

MARVIN



Der autonom fliegende Erkundungsroboter der TU Berlin und sein Erfolg beim Wettbewerb IARC'99

Marek Musial, Uwe Wolfgang Brandenburg, Günter Hommel

<http://pdv.cs.tu-berlin.de/leute/{musial,brg,hommel}>
TU Berlin, Prozeßdatenverarbeitung und Robotik

Zusammenfassung MARVIN ist ein fliegender Beobachtungs- und Erkundungsroboter auf Basis eines Modellhubschraubers für einen Flugroboter-Wettbewerb, bei dem in einem unbekanntem Gelände Objekte und Personen lokalisiert werden sollen. Das Gesamtsystem besteht aus dem Helikopter mit Bordrechner und einer Bodenstation aus mehreren PCs. Die Bordsensorik umfaßt ein GPS, Magnetfeld-, Rotations- und Beschleunigungssensoren, Ultraschall, einen Flammensensor und einen digitalen Fotoapparat. Die Kommunikation stellt allen Rechnern im System den gesamten Systemzustand transparent zur Verfügung. Die Missionssteuerung wird am Boden durchgeführt, während der Bordrechner selbständig einen Kurs abfliegen kann. MARVIN flog als erster Teilnehmer im dreijährigen Wettbewerb autonom.

1 Einführung

MARVIN ist eine Abkürzung für **M**ulti-purpose **A**erial **R**obot **V**ehicle with **I**ntelligent **N**avigation (intelligent navigierender Vielzweck-Luftroboter). Er ist in mehreren Projekt-Lehrveranstaltungen von Studierenden der TU Berlin entworfen und gebaut worden für den internationalen Flugroboter-Hochschul-Wettbewerb IARC (*International Aerial Robotics Competition*) [2] in Richland, Washington. In den Jahren 1998 und 1999 haben Qualifikationswettkämpfe stattgefunden für die Endausscheidung im Jahr 2000, das *Millennial Event* [1].

Die Roboter sollen innerhalb einer Stunde in einem simulierten Katastrophengebiet schwarze Tonnen suchen und lokalisieren, die durch Symbole für gefährliche Inhaltsstoffe gekennzeichnet sind – es gibt solche mit „radioaktivem“, mit „giftigem“ und mit „biologisch gefährlichem“ Inhalt sowie Tonnen ohne Symbole, die für die Wettbewerbsaufgabe nicht relevant sind. Außerdem sollen Personen erkannt und lokalisiert werden, dargestellt durch Puppen. Es gibt sowohl „Überlebende“, nämlich Puppen, die sich mechanisch bewegen, und „Opfer“, die keine Bewegungen (mehr) machen. Die Katastrophensituation ist weiter gekennzeichnet durch eine unspezifizierte Geländestruktur, z. T. sehr hohe Hindernisse und Gefahrenquellen wie Gasfeuer, Wasserfontänen und Rauchentwicklung. Die Zielobjekte sind teilweise eingegraben oder verdeckt.

Das Fachgebiet Prozeßdatenverarbeitung der TU Berlin hat 1995 bei einem Vorläufer-Wettbewerb mit einem Heliumballon den zweiten Platz belegt [3] und

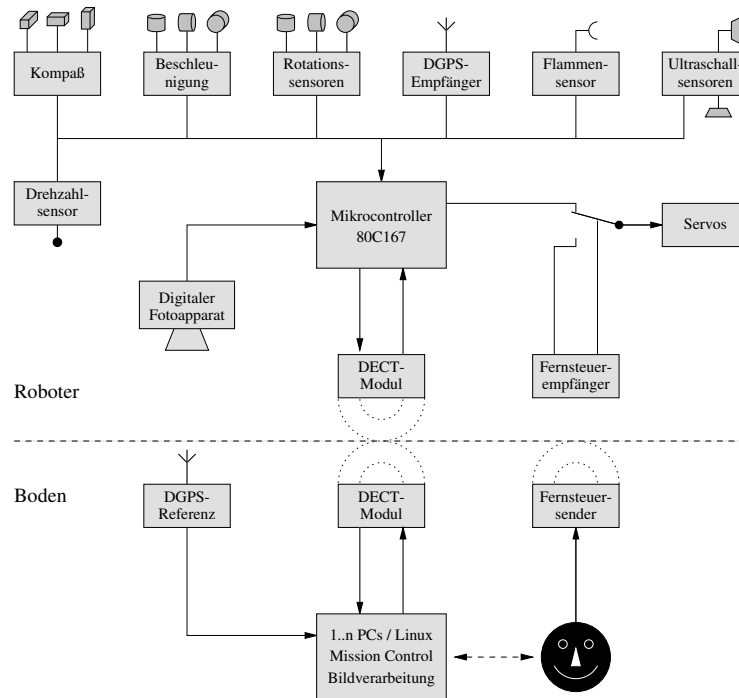


Abbildung1. Struktur des MARVIN-Systems.

1998 in Richland zumindest das technische Konzept von MARVIN präsentiert [5] – der Roboter selbst war zu diesem Zeitpunkt noch nicht weit genug entwickelt. 1999 reiste das TU-Team mit MARVIN zum Wettbewerb [4].

Im folgenden wird zunächst ein Überblick über die Systemarchitektur gegeben. Die nächsten Abschnitte greifen einige wichtige Aspekte in MARVINs Design heraus, nämlich die Kommunikation, die Flug- und Lageregelung und die Strategie der Missionssteuerung. Es schließt sich ein Erfahrungsbericht aus der Entwurfsphase und vom Wettbewerb an. Das Papier endet mit einem Ausblick auf das *Millennial Event* im nächsten Jahr.

2 Systemarchitektur

Abbildung 1 stellt den Aufbau des MARVIN-Systems dar.

Als Fluggerät dient ein handelsüblicher Modellhelikopter der Firma SSM. Er hat einen Rotordurchmesser von 1,8 m und verfügt über einen Zweitakt-Benzinmotor mit ca. 2 kW Leistung aus 22 cm³ Hubraum. Bei etwa 10 kg Startgewicht kann MARVIN rund 15 min in der Luft bleiben.

Die Lagesensorik an Bord des Helikopters umfaßt ein differentielles GPS mit Auswertung der Trägersignal-Phasenlage (NovAtel RT-2). Dieses System wird vom Hersteller an alle interessierten IARC-Teams kostenlos verliehen und er-

reicht eine relative Genauigkeit von 2 cm. Dazu gesellt sich eine preiswerte im Projekt entwickelte Orientierungssensorik, bestehend aus drei Magnetfeld-, drei Rotations- und drei Beschleunigungssensoren, mit einem Materialwert von insgesamt unter 400 DM. Um die erhebliche Drift der preiswerten Rotationssensoren („Kreisel“) in der Nick- und Rollachse zu kompensieren, wird der gemittelte gemessene Beschleunigungsvektor verwendet, der ja die Richtung der Erdbeschleunigung angibt. Die Mittelung ist erforderlich, damit der Einfluß der kinematischen Beschleunigungen verschwindet. Für die Gierachse dient der Kompaß als Driftkompensation.

Um Hindernisse, Gefahrenquellen und Unebenheiten im Gelände zu detektieren, verfügt MARVIN über zwei Ultraschall-Sonars mit Polaroid-Transducern, eines nach vorne und eines nach unten. Außerdem ist ein Flammensensor montiert, der auf UV-Strahlung zwischen 185 und 260 nm reagiert – dieser Spektralbereich liegt unterhalb des Sonnenlichts und tritt fast nur beim Verbrennen fossiler Brennstoffe wie Gas, Holz und Benzin zutage. Der Flammensensor registriert bereits eine Streichholzflamme in 5 m Entfernung.

Als einziger Bordrechner dient ein SAB80C167-Mikrocontroller von Siemens. Die Rechnerplatine wurde im Projekt entworfen und gefertigt, weil verfügbare Boards den Anforderungen an Platzbedarf und Anschlußmöglichkeiten nicht gerecht wurden. Mit dem Rechner sind alle Sensoren und Aktoren direkt verbunden, und die Lage- und Flugregelung läuft auf ihm. Damit kann MARVIN selbständig einen Kurs abfliegen, ohne daß die Bodenstation in die Regelung eingreift. Die Bodenstation, bestehend aus einem Netz von PCs unter Linux, erledigt die Bildauswertung und die Vorgabe von Zielpunkten im Missionsablauf. Außerdem muß sie die Daten des GPS-Referenzempfängers zum Roboter weiterleiten.

Eine besonders gelungene Entwurfsentscheidung besteht darin, daß MARVIN die Bilder zur Objektlokalisierung mit Hilfe eines digitalen Fotoapparats aufnimmt und nicht, wie bei allen anderen Teams, mit einer Videokamera. Der Fotoapparat wiegt einschließlich Akkus nur 250 g und liefert am Helikopter hervorragende Bilder mit automatischem Weißabgleich und automatischer Belichtung. Diese Bilder mit einer Auflösung von 640 mal 480 Pixeln im JPEG-Format werden durch den Bordrechner über eine serielle Schnittstelle digital ausgelesen und digital über die ohnehin benötigte Datenfunkstrecke zur Bodenstation übertragen. Damit entfällt eine zweite, analoge Sendestrecke für Videobilder, die erfahrungsgemäß sehr störanfällig ist und bei hinreichender Qualität erhebliches Gewicht und zusätzlichen Stromverbrauch mit sich bringt. Auf diese Weise erreicht zwar nur alle 9 s ein Bild die Bodenstation, was aufgrund der ständig erforderlichen Neupositionierung des Roboters beim Suchen aber kaum eine Einschränkung darstellt.

3 Kommunikation

MARVIN ist über zwei Funkverbindungen mit dem Boden verbunden, von denen aber nur eine im autonomen Betrieb verwendet wird:

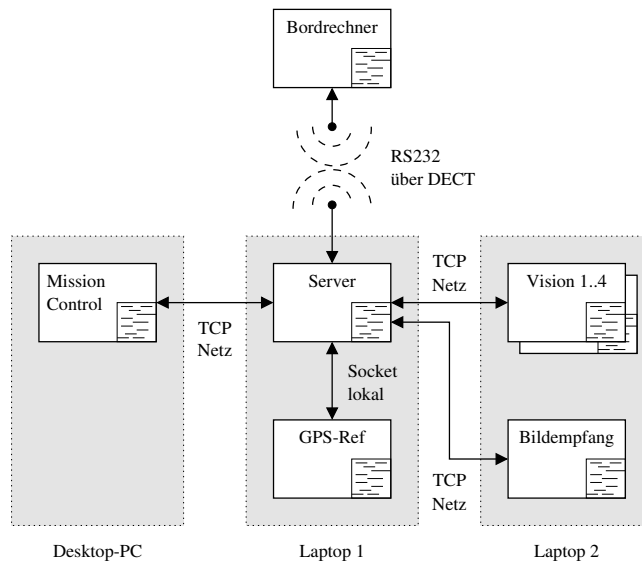


Abbildung2. Verteilung der Knotenprozesse im Wettbewerb 1999.

- Eine handelsübliche Funkfernsteuerung dient zur Steuerung des Helikopters bei manuellen Testflügen und als Notfallsystem zur sicheren manuellen Landung. Mittels eines Schalters auf der Fernsteuerung kann der menschliche Pilot jederzeit die Kontrolle über MARVIN übernehmen. Beim autonomen Flug wird die Fernsteuerung aber nicht verwendet.
- Ein Paar Datenübertragungsmodule vom Typ Siemens Gigaset M101 Data stellt eine drahtlose bidirektionale Übertragungsstrecke zwischen einer RS232-Schnittstelle eines PCs am Boden und einer RS232-Schnittstelle des Bordrechners zur Verfügung. Die Übertragungsrate in jeder Richtung liegt bei 115 kbit/s. Diese Module arbeiten mit Hilfe der von schnurlosen Telefonen bekannten DECT-Technologie und ersetzen ein Nullmodemkabel. Über diese Funkstrecke wird die gesamte Kommunikation zwischen Boden und Roboter abgewickelt. Die Reichweite liegt im Freien bei mindestens 300 m.

3.1 Shared-Memory-Ansatz der Kommunikation

Die Kommunikations-Software präsentiert den einzelnen Modulen eine Datenstruktur, die systemweit bekannt ist und den vollständigen Systemzustand mit allen interessierenden Daten repräsentiert. Diese Datenstruktur steht als *virtueller gemeinsamer Speicher (Shared Memory)* systemweit zur Verfügung.

Jeder sogenannte *Knoten* im System verfügt über eine lokale Kopie des Shared Memory. Als *Knoten* gilt jeweils ein Prozeßkontext, repräsentiert durch

- einen Linux-Prozeß auf einem Rechner der Bodenstation oder

- die Bordrechner-Software, die in einer auf 20 Hz synchronisierten Hauptschleife als ein einziger Prozeßkontext läuft.

Weil jede Shared-Memory-Kopie zu einem einzigen Prozeßkontext gehört, kann dieser Speicher jederzeit ohne Synchronisationsmaßnahmen gelesen und beschrieben werden. Die Kommunikations-Software stellt einen Dienst `shm_sync` zur Verfügung, der in jedem Knoten regelmäßig aufgerufen wird und die globale Konsistenz der Shared-Memory-Datenstruktur transparent mit Hilfe eines Konsistenzprotokolls erhält.

Der Shared Memory ist in zusammenhängende logische Einheiten, sogenannte *Slots*, unterteilt. Ein Slot könnte z. B. die aktuelle Lagebeschreibung des Roboters als homogene 4x4-Matrix enthalten. Mittels der Dienste `shm_put` und `shm_get` wird der Kommunikations-Software mitgeteilt, wann ein Slot lokal verändert wurde, bzw. abgefragt, ob ein Slot in einem entfernten Knoten eine Änderung erfahren hat. Veränderte Slots werden grundsätzlich nur im ganzen aktualisiert, weil sie – wie bei der Lage-Matrix – voneinander abhängige Daten enthalten können.

Die einzelnen Knoten sind durch Punkt-zu-Punkt-Verbindungen miteinander verbunden. Für jede Übertragungsrichtung ist pro Slot eine maximale Kommunikationsbandbreite zugewiesen, so daß die insgesamt verfügbare Bandbreite niemals überschritten wird. Häufige Aktualisierungen einzelner Slots können also nicht zum Aushungern anderer Slots führen, für sämtliche Slots werden Echtzeitbedingungen eingehalten.

Durch dieses Konzept stehen in jedem Knoten immer möglichst aktuelle Zustandsdaten zur Verfügung. Wenn ein Slot häufiger aktualisiert als übertragen wird, werden ältere Daten überschrieben, weil sie mit Vorliegen der neueren Daten für die Zustandsbeschreibung ohnehin bedeutungslos sind. Slot-Daten werden nirgends gepuffert, sondern erst beim Vorliegen einer Bandbreitenzuteilung für den jeweiligen Slot direkt aus dem Shared Memory kopiert. Dies spart Pufferspeicher und verhindert weitgehend, daß Bandbreite für veraltete Daten verschwendet wird.

3.2 Physikalischer Aufbau des Systems

Die Kommunikations-Software ist in C geschrieben und fast vollständig plattformunabhängig. Nur die Treiberfunktionen für die Kommunikationsverbindungen unterscheiden sich zwischen Linux und dem Bordrechner. Unter Linux gibt es Treiberfunktionen für RS232 einerseits und Internet-Stream-Sockets andererseits, während an Bord nur RS232 unterstützt wird.

Durch die Verwendung der Sockets kann die Software der Bodenstation beliebig auf ein Netz von Rechnern verteilt werden. Theoretisch kann sogar *jeder* Rechner im Internet einen Knotenprozeß übernehmen, ohne daß eine einzige Zeile der Software geändert werden muß. Abbildung 2 zeigt die Verteilung der verschiedenen Knotenprozesse, wie sie im 1999er Wettbewerb vorlag. Der Server-Prozeß, der mit dem DECT-Modul am Boden verbunden ist, erfüllt drei wichtige Aufgaben:

```

kvt <3>
Gyro_Sensor      Gyro_Winkel      Gyro_MGeschw
545 x             524285 x         0 x
506 y             524287 y         0 y
475 z             3723 z           0 z
Beschl_Sensor    Beschl_Beschl    Beschl_Geschw    Beschl_Position
497 x             0 x              250 x            37565 x
488 y             0 y              22 y             811 y
514 z             0 z              41 z             3207 z
Magnet_Sensor    Magnet_Normwert  Magnet_Winkel    Magnet_Winkel2
37094 x          75 x             976 x            3740 xy
38690 y          27 y             1006 y           4048 zx
0 z              1020 z           57 z             17 yz
Lage_Position    Lage_Orientierung Gyro_Null        Beschl_Null
0 x              -2 x             545 x            497 x
0 y              0 y              506 y            488 y
0 z              3740 z           475 z            514 z
Ultraschall     Kamera          BC_idle          36 %
2054 dx          38 t/o          18 seq0          16 seq1          17 seq2
412 dz          11 err          2048 len0        2048 len1        2048 len2
GPS-VLHA-Log    GPS-PVAA-Log    Arenakoordinaten VLHA-Infos
0 hsp           0 a_x           0 p_x            0 lat
0 vsp           0 a_y           0 p_y            0 age
0 nrd           0 a_z           0 p_z            0 vel
GPS-Direktdaten Vel-GPS-Position
0 brt           0 hoh           -1 ptp           0 s_x
0 lng           0 sat           -1 rtk           0 s_y
0 sol           0 s_a           0 sec           0 s_z
Servos          Vel-GPS-Geschwindigkeit
0 ha1           0 ha2           0 gas            0 s_x
0 ha3           0 ha4           0 hec            0 s_y
0 ha3           0 ha4           0 hec            0 s_z
Feuer
0 uvr
0 ir1           0 nu1           0 ga1
0 ir2           0 nu2           0 ga2
0 ir3           0 nu3           0 ga3
Verbindung mit CAMERA aufgebaut.

```

Abbildung 3. Bildschirmanzeige des Server-Prozesses.

1. Er verteilt die Daten innerhalb der Bodenstation, weil alle Kommunikationsstrecken ja Punkt-zu-Punkt-Verbindungen sind.
2. Er zeigt alle wichtigen Zustandsdaten in einem eigenen Fenster an (siehe Abbildung 3) und erlaubt somit die verzögerungsfreie Überwachung aller Systemparameter.
3. Er erzeugt eine Log-Datei aller interessierenden Daten aus dem Shared Memory, mit deren Hilfe z. B. Plots von Flugbahndaten erzeugt werden können, noch bevor der Helikopter wieder gelandet ist.

4 Flugregelung

Die Lage- und Flugregelung erfolgt vollständig auf dem Bordrechner. Nur gelegentlich gibt Mission Control vom Boden aus über den Shared Memory einen neuen Zielpunkt und eine Sollgeschwindigkeit auf dem Weg dorthin vor. MARVIN erreicht diesen Punkt dann selbständig, wobei er allerdings auf die Referenzdaten vom Boden-GPS angewiesen ist, um seine Position mit voller Genauigkeit zu messen.

Die Hierarchie der Flugregelung ist in Abbildung 4 dargestellt. Eine erste Reglerstufe in der Abbildung links berechnet aus dem seitlichen Abstand von der Kurslinie und dem Fehler der Geschwindigkeit längs dem Kurs eine Soll-Schräglage der Rotorebene quer zum Kurs und eine Soll-Schräglage längs zum Kurs, mit deren Hilfe die Steuerkräfte zur Ausregelung der beiden Fehler aufgebracht werden sollen. Gemäß der aktuellen Orientierung des Helikopters in der

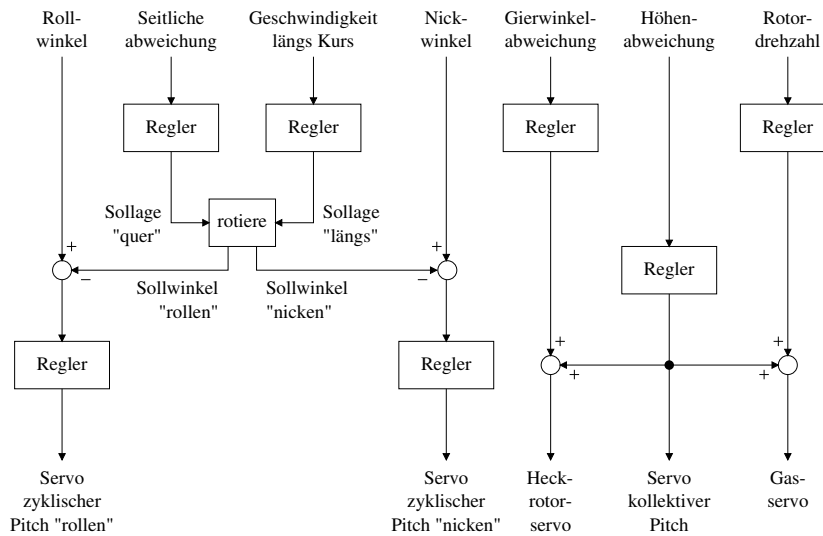


Abbildung 4. Hierarchie der Elementarregler für MARVINs Flugregelung.

Gierachse rotiert, ergeben sich ein Soll-Rollwinkel und ein Soll-Nickwinkel, die nun auf Helikopterachsen bezogen sind. Subtraktion dieser Sollwerte von den Ist-Roll- und Nickwinkeln liefert die Roll- und Nickwinkelfehler, die als Eingabe der Regler für die zyklische Rotorblattverstellung für die entsprechenden Achsen dienen. Rechts in der Abbildung sieht man einstufige Regelungen für die Blattanstellung des Heckrotors aus dem Gierwinkelfehler, die kollektive Blattanstellung des Hauptrotors aus der Höhenabweichung und die Stellung des Gas-Servos aus der Rotordrehzahl. Letztere soll während des Fluges möglichst konstant gehalten werden, bei MARVIN auf 1200 Umdrehungen/min. Die Reglerausgabe für die kollektive Hauptrotor-Anstellung wird zusätzlich in die Heckrotor- und Gasstellung eingemischt, weil Gegendrehmoment und nötige Antriebsleistung mit stärkerer Blattanstellung zunehmen und diese Fehler nur mit erheblicher Verzögerung durch die Einzelregler ausgeglichen würden.

Jeder der Elementarregler aus Abbildung 4 erhält einen Fehler x und dessen Ableitung \dot{x} als Eingaben und erzeugt eine Ausgabe y nach der Vorschrift:

$$I(t) := I(t-1) + f_{I,P}(x(t)) + f_{I,D}(\dot{x}(t)) \quad (1)$$

$$y(t) := I(t) + f_{P,P}(x(t)) + f_{P,D}(\dot{x}(t)) \quad (2)$$

$I(t)$ bildet also eine integrierte Regler-Ausgabe, was einige Vorteile gegenüber dem klassischen Fehlerintegral bietet: Der Wert von $I(t)$ hat eine offensichtliche Semantik in bezug auf die gesamte Reglerausgabe, so daß eine Begrenzung dieses Wertes auf einen sinnvollen Bereich leichtfällt. Außerdem erlaubt $I(t)$ als Zustandsvariable, die aktuelle Ausgabe des Gesamtreglers auf einen Zielwert zu setzen und so einen vorgegebenen Arbeitspunkt zu übernehmen. Von dieser Möglichkeit macht MARVINs Flugregler intensiv Gebrauch: Wann immer der Flugregler MARVINs Steuerung vom menschlichen Piloten übernimmt oder eine

neue Flugphase einleitet, wird die aktuelle Stellung aller Servos rückwärts durch den Flugregler „durchgerechnet“. Alle $I(t)$ -Werte werden dabei so eingestellt, daß die Reglerausgabe im ersten Regelzyklus *genau* der aktuellen Servostellung entspricht. Dadurch ist sichergestellt, daß es niemals durch falsch eingestellte Arbeitspunkte der Regler plötzlich zu kritischen Fluglagen kommen kann.

Die Transferfunktionen $f_{x,y}$ sind stückweise lineare Kennlinien, definiert durch einige Stützpunkte. Außerhalb der äußeren Stützpunkte sind die Funktionen jeweils konstant. Die Stützpunkte der Reglerfunktionen werden in einer Textdatei festgelegt und mit Hilfe eines kleinen Programms von der Bodenstation aus per Shared Memory zum Roboter übertragen. Damit läßt sich die Reglercharakteristik sogar im Flug – allerdings nicht bei laufendem Regler – ohne Eingriff in die Bordrechner-Software ändern. Mit Hilfe eines sehr einfachen Simulators kann eine geänderte Reglerparameter-Datei vor der Übertragung zum Roboter schnell auf grobe Fehler geprüft werden.

5 Missionsstrategie

Ein mit *Mission Control* bezeichnetes Software-Modul der Bodenstation implementiert die Missionsstrategie und dient als Benutzerschnittstelle zum MARVIN-System.

Beim Start der Mission wird eine initiale Liste von Zielpunkten auf dem Gelände vorgegeben. Diese initialen Zielpunkte befinden sich beispielsweise in 25 m Höhe und sind Orte für Übersichtsaufnahmen. Auf diesen Übersichtsaufnahmen können die Symbole auf den Tonnen nicht erkannt werden, weil sie zu gering aufgelöst sind. Daher ermittelt ein spezielles Bildverarbeitungsmodul auf den Übersichtsbildern Objekte, die aufgrund ihrer Farbe, Größe und Kompaktheit Zielobjekte sein *könnten*. Für jedes potentielle Zielobjekt wird ein neuer Zielpunkt in die Zielliste eingetragen, diesmal in 5 m Höhe über dem potentiellen Objekt. Aus dieser geringen Flughöhe macht MARVIN Detailaufnahmen, die zur sicheren Erkennung der Objektart – einschließlich der Tonnenkennzeichnung – dienen.

Mission Control sucht sich als nächstes anzufliegendes Ziel immer dasjenige mit dem geringsten Abstand von der aktuellen Roboterposition aus der Zielliste heraus. Die Mission endet, wenn die Zielliste leer oder eine vorgegebene Flugzeit abgelaufen ist, für die Akkus und Benzin sicher reichen. MARVIN kehrt dann zur Startposition zurück und landet dort autonom.

Zusätzlich zur Zielliste führt Mission Control eine Karte mit Hindernissen, die entweder initial vorgegeben oder während des Fluges mit den Ultraschall- und Flammensensoren detektiert worden sein können. Da der Flammensensor nur Standlinien von Flammen liefert, werden mit diesen Triangulationen ausgeführt und eine Flamme erst für einen hinreichend gut bestimmten Schnittpunkt mindestens zweier Standlinien eingetragen. Flammen auf dem aktuellen Flugkurs werden mit Hilfe von Zwischen-Zielpunkten umflogen, während MARVIN feste Hindernisse durch schrittweise Erhöhung der Sollflughöhe überfliegt.

6 Erfahrungen und Ergebnisse

Mit der Entwicklung von MARVIN wurde am Fachgebiet Prozeßdatenverarbeitung im Wintersemester 1997/98 begonnen. Erst im Frühjahr 1999 waren alle erforderlichen Komponenten beschafft und integriert, um an autonome Flugversuche zu denken.

Bei der Abstimmung und Erprobung der Lage- und Flugregelung wurde von Anfang an ein pragmatisch-mutiger Ansatz gewählt: Der Reglerentwurf erfolgte nicht anhand eines geschlossenen Modells, weil solche Modelle für die Flugdynamik von Helikoptern sehr kompliziert sind, zahlreiche schwer bestimmbar Parameter enthalten und eine Vielzahl von Effekten modellieren, die quantitativ hinter den „wichtigen“ Steuerkräften fast nicht zutage treten. Stattdessen wurde der Regler anhand der Steuereingaben beim Helikopter und ihrer grundlegendsten Wirkungen konstruiert und mit Hilfe eines stark vereinfachenden Simulators grob getestet. Nachdem das Verhalten in dieser Simulation zufriedenstellend war, erfolgten Übernahmeveruche im Flug, der erste am 23. Mai. Dabei startet ein Modellhelikopter-Pilot den Roboter, und der Flugregler übernimmt auf Funkbefehl in der Standschwebe nicht etwa nur einzelne, sondern stets *sämtliche* Steuerfunktionen auf einmal. Dank der in Abschnitt 4 beschriebenen Übernahme des Arbeitspunktes sind abrupte Steuerfehler dabei ausgeschlossen, und die maximale Stellgeschwindigkeit von der übernommenen Servostellung weg ist begrenzt. Der Pilot kann MARVIN jederzeit wieder übernehmen, was am Anfang nach jeweils wenigen Sekunden erforderlich wurde. Mit der Zeit wurden noch einige Fehler beseitigt und die Transferfunktionen optimiert, bis die Regelung zufriedenstellend arbeitete. Schäden am Helikopter im Zusammenhang mit dieser Art der Reglerentwicklung sind nicht entstanden.

Kritischer sind der autonome Start und die autonome Landung, weil hier für den Piloten im Fehlerfall kaum Reaktionszeit bleibt, bevor der Rotor den Boden zu berühren droht. Die Startstrategie besteht daher darin, möglichst schnell eine „sichere“ Höhe zu erreichen. Zur Überraschung der Autoren erwies sich der Start als unkritisch, weil der Bordrechner den Helikopter sehr gerade nach oben steuern kann, noch bevor die Regelung an sich beginnt. Aber selbst, als das initiale Hochziehen des Helikopters durch einen Programmierfehler unterblieb und MARVIN nur langsam „losschwebte“, kam es nicht zu einer kritischen Situation.

Beim Landemanöver muß die Höhe dagegen langsam abgebaut werden. Nach Unterschreiten einer Mindesthöhe reduziert die Flugregelung dann kontinuierlich die Hauptrotor-Blattanstellung auf einen Neutralwert. Spätestens dann ist der Roboter gelandet. Dieses Manöver wurde vorsichtshalber erstmals während der Wettbewerbsvorführung praktisch durchgeführt und erwies sich gleichsam als überraschend unproblematisch.

Im Wettbewerb 1999 konnte MARVIN als einziger Roboter unter 11 ange-reisten Teams autonom fliegen und fand eine Tonne, deren Radioaktiv-Symbol er auf 8 cm genau lokalisierte. 1998 hatte es keine autonomen Flüge gegeben, und 1999 konnte gerade ein einziges anderes Team, vom Rose-Hulman Institute of Technology, einzelne Regelungskomponenten im Flug demonstrieren, ohne vollständige Übergabe und ohne autonomes Starten und Landen. Daher erhielt

das Team der TU Berlin die ersten und einzigen Punkte für die Flugvorführungen seit 1998 und führt dadurch die Punktwertung der Qualifikation für das *Millennial Event* an.

7 Schluß und Ausblick

In diesem Papier wurde das Konzept des autonomen Flugroboters MARVIN beschrieben und die Entwicklung hin zum Erfolg im 1999er Qualifikationswettbewerb. Nach diesem Erfolg ist natürlich geplant, auch an der Endausscheidung um 30.000 US-Dollar Preisgelder im Juni 2000 teilzunehmen. Nötige Verbesserungen am MARVIN-System betreffen vor allem die Bildauswertung und ihr Zusammenspiel mit Mission Control. Das Erkunden des ca. 150 m im Quadrat messenden Wettbewerbsareals muß schneller und zuverlässiger erfolgen. Allerdings ist MARVIN schon jetzt insofern zu einem Endstand entwickelt, daß zahlreiche Detailverbesserungen den Austausch weiterer Komponenten nach sich zögen. Im Vergleich zu den übrigen Teams muß man mit Bedauern feststellen, daß diese fast nur aufholen können.

Wir würdigen die aufopferungsvolle Arbeit zahlreicher Studierender an MARVIN, ohne die der Erfolg des Projekts unmöglich gewesen wäre, besonders – aber nicht nur – das USA-Team mit Eike Berg, Christian Fleischer, Christian Reinicke, Volker Remuß und Andreas Wege. Besonderer Dank gebührt unserem Piloten Matthias Jeserich für seine sichere sichernde Hand und dem NLV Saarmund für die Testmöglichkeiten. Zahlreiche Partner haben wesentliche Beiträge zur Realisierung geleistet, darunter Daimler Benz Aerospace AG (Modellhelikopter), NovAtel Inc. (GPS-Leihgabe), Novedia Media Solutions (finanzielle Unterstützung), Siemens AG (PC, Mikrocontroller, DECT-Module), SSM-Technik Peter Schröppel sowie Rotor-Modellsport-Center Berlin (Ersatzteile für die USA-Reise) und TFH Berlin (Platinenherstellung).

Literatur

1. International Aerial Robotics Competition: The Robotics Competition of the Millennium. <http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/IARC2000Intro.html>.
2. THE MILLENNIAL EVENT: Rules for the 1999 International Aerial Robotics Competition qualifier. <http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/1999CollegiateRules.html>.
3. U. W. Brandenburg, M. Finke, M. Musial. Aufbau und Steuerung des fliegenden Roboters TUBROB. In *Tagungsband 11. Fachgespräch Autonome Mobile Systeme*, Seiten 100–109, Karlsruhe, 1995. Springer-Verlag.
4. M. Musial, G. Hommel, U. W. Brandenburg, E. Berg, M. Christmann, C. Fleischer, C. Reinicke V. Remuß, S. Rönnecke, A. Wege. MARVIN – Technische Universität Berlin's flying robot competing at the IARC'99. In *Proc. AUVS Symposium*, Washington D.C., USA, 1999.
5. Marek Musial, Uwe Wolfgang Brandenburg, Günter Hommel. Marvin - Ein autonom fliegender Erkundungsroboter. In *Autonome Mobile Systeme 1998, 14. Fachgespräch*, Seiten 226–233, Karlsruhe, 1998. Springer-Verlag.