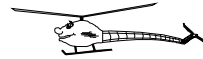


MARVIN



Ein autonom fliegender Erkundungsroboter

Uwe Wolfgang Brandenburg, Günter Hommel, Marek Musial

<http://pdv.cs.tu-berlin.de/leute/{brg,hommel,musial}>
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 gefunden und klassifiziert werden sollen. Das Gesamtsystem besteht aus dem Helikopter mit Bordrechner und einer Bodenstation. Die Flugregelung verwendet ein Netz von Fuzzy-Reglern. Die Flugroutenplanung mit Kollisionsvermeidung erfolgt mit Hilfe eines Potentialfeldansatzes.

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).

Die Anforderungen an MARVIN ergeben sich aus der Wettbewerbsaufgabe für die 1998er *International Aerial Robotics Competition* (IARC) [1], die am 14. August 1998 in Richland, Washington, stattfand. Die Roboter sollten innerhalb einer Stunde in einem mit Hindernissen versehenen und den Teilnehmern vorher unbekanntem „Katastrophengebiet“ Tonnen mit imaginärem Gefahrgut sowie Überlebende und Opfer der Katastrophe suchen und lokalisieren. Die schwarzen Tonnen sind mit weißen Symbolen versehen, die ihren Inhalt kennzeichnen – es gibt solche mit „radioaktiven“, mit „giftigen“ und mit „biologisch gefährlichen“ Inhaltstoffen sowie Tonnen ohne Symbole, die für die Wettbewerbsaufgabe nicht relevant sind. „Überlebende“ unterscheiden sich von „Opfern“ dadurch, daß die sie darstellenden Puppen Bewegungen und Geräusche machen. Das Katastrophenszenario wird abgerundet durch Gasfeuer, Wasserfontänen und Rauch, die z. T. sporadisch auftreten.

Nach der Teilnahme des Fachgebiets Prozeßdatenverarbeitung der TU Berlin mit einem Heliumballon 1995 [2, 3] – bei anderer Aufgabenstellung – wird diesmal ein Modellhubschrauber als Roboter benutzt [4]. Leider konnte das System für den diesjährigen Wettbewerbstermin nicht rechtzeitig fertiggestellt werden.

Der Rest dieses Papiers gliedert sich folgendermaßen: Es folgt ein Überblick über den Aufbau des MARVIN-Systems. Daran schließen sich Beschreibungen einzelner Komponenten an, nämlich des Helikopters, der Sensorik, des Computer-Systems und wichtiger Teile der Software. Abschnitt 7 schließt das Papier mit einer Zusammenfassung und Danksagung.

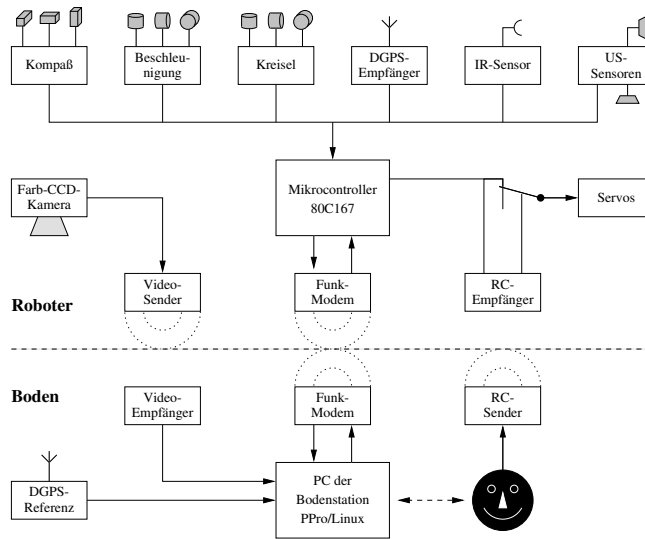


Abbildung1. Struktur des MARVIN-Systems.

2 Systemaufbau

Abbildung 1 stellt den Aufbau des MARVIN-Systems dar. Die Sensorik an Bord des Helikopters umfaßt einen 3D-Kompaß, einen 3D-Beschleunigungsmesser zur Messung von dynamischer Beschleunigung und der Vertikalrichtung, drei preiswerte Piezo-Kreisel-Sensoren, einen Infrarot-Sensor zur Erkennung sehr naher Feuer, zwei Ultraschallsysteme zur Abtastung des Bodenprofils und zur Hinderniserkennung, eine Farbkamera und einen auf bis zu 2 cm genauen DGPS-Empfänger mit Trägersignal-Phasenvergleich. Diese GPS-Geräte vom Typ RT-2 werden von NovAtel, Kanada, an alle interessierten Teilnehmer kostenlos verliehen. Als Bordcomputer dient ein SAB80C167-Mikrocontroller von Siemens. Die Bodenstation bildet ein Linux-PC mit Videodigitalisierungskarte. Die digitale Datenkommunikation zwischen der Bodenstation und MARVIN erfolgt über Spread-Spectrum-Funkmodems. Außerdem gibt es eine analoge Videostrecke von MARVIN zur Bodenstation und eine normale Modellbau-Fernsteuerung für Notfälle und Testzwecke.

3 Helikopter

Als Helikopter wird ein handelsüblicher Modellhubschrauber vom Typ „Trainer 180“ der deutschen Firma SSM verwendet. Er hat einen Rotordurchmesser von 1,8 m und ist mit einem 22 cm³ Zweitaktmotor ausgestattet, der mit Benzin (statt Methanol) arbeitet und in etwa 2 kW Leistung abgibt. Das Helikopter-Leergewicht beträgt 6,5 kg, die verfügbare Nutzlast liegt bei etwa 5 kg. Die Helikoptermechanik wird von sechs Servos angesteuert.

4 Sensorik

Dieser Abschnitt behandelt die Sensoren an Bord und die Weiterverarbeitung ihrer Meßdaten.

4.1 Position und Lage

In diesem Abschnitt zur Positions- und Lagesensorik kennzeichnet ein hochgestellter Index H Angaben relativ zum Bezugssystem des Helikopters, während W für das Weltkoordinatensystem steht.

Wegen der relativ großen Meßfehler der preiswerten Piezokreisel kann die Orientierung des Helikopters nicht dauerhaft durch Integration der Rotationsgeschwindigkeiten ermittelt werden. Daher muß die Orientierung regelmäßig auf Basis der Kompaß- und Beschleunigungsdaten berechnet werden. Nun liefert der Kompaß effektiv nur zwei Winkelparameter, weil jede Rotation *um* den Feldstärkevektor zu keinerlei Änderung der Kompaßmeßwerte führt. Das gleiche gilt für den Vektor \mathbf{a}_g^W der Scheinbeschleunigung. Dieser setzt sich zudem zusammen aus dem Gravitationsfeldvektor der Erde und der dynamischen Beschleunigung \mathbf{a}^W infolge der Bewegung des Roboters:

$$\mathbf{a}_g^W = (0, 0, -1g)^T - \mathbf{a}^W \quad (1)$$

In (1) kann \mathbf{a}^W durch Differenzieren der GPS-Daten zumindest näherungsweise ermittelt werden. Die Richtung des magnetischen Feldstärkevektors der Erde \mathbf{B}^W im Welt-Bezugssystem ist außerdem bekannt. Nun kann im Weltkoordinatensystem eine vollständige Orientierung in Form eines Orthonormalsystems als 3×3 -Matrix \mathbf{K}^W eindeutig ermittelt werden: Die x -Achse dieses Systems wird durch die Richtung von \mathbf{B}^W bestimmt, die $x - y$ -Ebene durch die $\mathbf{B}^W - \mathbf{a}_g^W$ -Ebene. Entsprechend erhält man ein Orthonormalsystem \mathbf{K}^H in Hubschrauberkoordinaten aus den Meßwerten \mathbf{B}^H und \mathbf{a}^H . Daraus läßt sich die Koordinatentransformation \mathbf{H}^W der Helikopterorientierung berechnen, denn:

$$\mathbf{K}^W = \mathbf{H}^W \cdot \mathbf{K}^H \quad \mathbf{H}^W = \mathbf{K}^W \cdot (\mathbf{K}^H)^T \quad (2)$$

Die Berechnung nach (2) ist allerdings nur dann möglich, wenn \mathbf{a}_g^W weder sehr klein wird noch parallel zum Magnetfeldvektor \mathbf{B}^W steht, denn in diesen Fällen lassen sich die Systeme \mathbf{K}^W und \mathbf{K}^H nicht bestimmen. Zur Überbrückung solcher Situationen wird \mathbf{H}^W durch Integration der Kreiselmeßwerte aktualisiert.

4.2 Bildauswertung

Eine senkrecht nach unten gerichtete CCD-Farbkamera mit Weitwinkelobjektiv dient zur Aufnahme der Videobilder vom Missionsgebiet. Dank der Messung der Flughöhe über Grund ist die zu erwartende Größe der Tonnen und menschlichen Körper im Bild in etwa bekannt, was deren Erkennung erheblich erleichtert. Die Algorithmen zur Bildauswertung und ihre Optimierung sind momentan noch in der Entwicklung, weil der Fähigkeit zum autonomen Flug höhere Priorität einzuräumen war. Am Beispiel der Tonnenerkennung soll eine mögliche Vorgehensweise erläutert werden:

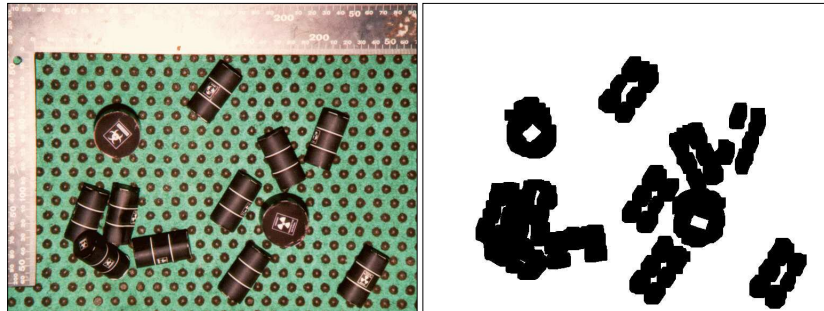


Abbildung 2. Beispielbild (*links*) und Ergebnis der Tonnensegmentierung (*rechts*).

1. Da die Tonnen schwarz sind, wird das Videobild anhand einer Helligkeitsschwelle segmentiert. Ergebnis ist ein Binärbild, das nur die dunklen Pixel umfaßt.
2. Rauschen, Störungen und Glanzlichter werden durch Erosion und Dilatation beseitigt [5]. Die verwendete Zahl der Erosions- und Dilatationsschritte wird in Abhängigkeit von der Flughöhe gewählt. Abbildung 2 zeigt ein Beispiel für das Ergebnis nach diesem Schritt.
3. Die resultierenden Segmente werden anhand ihrer Ausdehnung und der Zahl der zum Segment gehörenden Pixel innerhalb des kleinsten umgebenden Achtecks mit 45° -Winkeln bewertet.

5 Computer-System

Dieser Abschnitt beschreibt die Komponenten von MARVINs Computer-System. Diese sind der Rechner der Bodenstation und der Bordcomputer.

5.1 Bodenstation

Der Rechner der Bodenstation ist ein PC auf Basis eines Pentium-Pro-Prozessors mit 200 MHz Takt und 64 MB Arbeitsspeicher, auf dem Linux läuft. Er ist über eine serielle Schnittstelle mit dem Boden-Referenzempfänger des GPS verbunden. Dieser Rechner dient auch als Benutzerschnittstelle des MARVIN-Systems. Gegenüber Ansätzen, das Computer-System vollständig an Bord des Roboters unterzubringen, hat diese Lösung praktische Vorteile in der Entwicklungsphase und im Hinblick auf die verfügbare Rechenleistung für die Bildauswertung.

5.2 Bordcomputer

Herzstück des MARVIN-Roboters ist ein 16-bit-Mikrocontroller vom Typ Siemens SAB80C167CR-16F als Bordcomputer [7]. Der 167 wurde gewählt, weil er über genug Rechenleistung für alle an Bord zu erledigenden Aufgaben verfügt

und über ausreichend Peripherieanschlüsse, um sämtliche Sensoren und Aktoren direkt anzuschließen. Auf eine Trennung zwischen CPU und zusätzlichen Prozessoren zur Peripherieanbindung konnte so verzichtet werden. Hier einige wichtige technische Daten des Controllers: 20 MHz interner Takt, 16-bit-Speicher-Interface, 32 Capture-/Compare-Einheiten, 9 Zeitgeber, 10-bit-A/D-Umsetzer mit 16 Eingängen, *Peripheral Event Controller* (PEC) mit Speicherdirektzugriff, 128 KByte Flash-ROM, CAN-Bus-Schnittstelle.

Die in MARVIN benutzte Controller-Platine wurde im Rahmen des Projekts entwickelt, weil keine kommerziell verfügbare Lösung einen komfortablen Zugriff auf alle benötigten Merkmale und Anschlußmöglichkeiten des 167 erlaubte. Die Platine trägt 0,5 MByte statisches RAM mit Zugriff ohne Wartezyklen und erlaubt das Aufstecken von Peripherieplatinen über Steckkontakte.

6 Software

Dieser Abschnitt geht auf die drei wichtigsten Module der MARVIN-Software näher ein: auf das „Betriebssystem“ des Mikrocontrollers, das *Mission-Control*-Modul auf dem Boden-PC und die Flugregelung an Bord.

6.1 „Betriebssystem“ des Bordrechners

Auf der Mikrocontroller-Platine wird kein Betriebssystem im eigentlichen Sinne verwendet. Statt dessen werden die Task-Module auf dem Controller als Unterprogramme aus einer Hauptschleife heraus aufgerufen. Durch obere Schranken für die Ausführungszeit jedes Task-Durchlaufs sowie eine Warteoperation am Ende der Hauptschleife läuft diese mit einem garantierten Takt von 20 Hz. Dieses Konzept hat im Rahmen der vorliegenden Aufgabe mehrere Vorteile gegenüber einem „echten“ Betriebssystem:

1. Die Einhaltung von Echtzeitanforderungen kann durch den deterministischen Programmablauf sehr einfach gewährleistet werden.
2. Weil das geschilderte Task-Konzept nichtpräemptiv ist, sind kaum Synchronisationsmaßnahmen erforderlich.
3. Durch den festen Takt können die meisten mit Zeitmessung verbundenen Aufgaben von 50 ms zwischen zwei Task-Aufrufen ausgehen.

Die systemweite Kommunikation wird mit dem abstrakten Konzept eines virtuellen gemeinsamen Speichers (*shared memory*) realisiert: Es existiert eine globale Zustandsbeschreibung in Form einer Datenstruktur, die sowohl für die Prozesse am Boden als auch für die Tasks an Bord zugreifbar ist. Die Kommunikations-Software sorgt im Hintergrund für die Synchronisation der beiden Kopien der Zustandsbeschreibung über die digitale Datenfunkverbindung. Wenn ein Modul auf ein Datum zugreift, erhält es automatisch den jeweils aktuellsten verfügbaren Wert. Dies erlaubt eine sehr komfortable Nutzung der Kommunikation auf einer sehr abstrakten Ebene.

6.2 Mission Control

Das *Mission-Control*-Modul ist für die Zielauswahl und Routenplanung zuständig. Während die Zielauswahl im Hinblick auf eine maximale Punktzahl gemäß den Wettbewerbsregeln [1] ausgerichtet ist, stellt die Kollisionsvermeidung den wesentlichen Aspekt der Routenplanung dar – Hindernisse und Gefahren müssen in einem für MARVIN sicheren Abstand umflogen werden.

Mission Control verwaltet eine Liste von Objekten, in die alle bekannten Ziele und Gefahrenquellen eingetragen werden. Ein Listeneintrag besteht aus

- dem Objekttyp,
- dem bisherigen Zuverlässigkeitsgrad der Erkennung (zwischen 0 und 1),
- der Objektposition in der Wettbewerbsarena,
- der Objektgröße (Breite und Höhe) und
- dem Gefährdungsgrad, das vom Objekt ausgeht.

Solche Einträge resultieren einerseits aus Ergebnissen der Bildauswertung, welche immer mit einer Erkennungswahrscheinlichkeit versehen sind. In der Wettbewerbspraxis können bestimmte Hindernisse, z. B. Laternenmasten, auch vorher bekanntgemacht werden, diese ergeben initiale Einträge in der Objektliste mit Zuverlässigkeit 1.

Zur Planung der Missionsaktivitäten werden die Zielobjekte in der Objektliste regelmäßig anhand ihrer Wichtigkeit im Hinblick auf den Punkteerwerb, die Sicherheit ihrer bisherigen Erkennung und den Abstand von der aktuellen Roboterposition bewertet. Das Objekt mit dem höchsten Wert dieser Bewertungsfunktion wird zum aktuellen Zielobjekt. Die Planung des Flugweges dorthin erfolgt mit Hilfe eines dreidimensionalen Potentialfeldansatzes: Objekte in der Objektliste mit Gefahrenpotential werden durch jeweils einen Zylinder modelliert, dessen Breite und Höhe den Objektmessungen und dessen abstoßendes Potential an seiner Oberfläche dem Gefährdungsgrad des Objektes entspricht. Dem Boden ist ein ebenfalls abstoßendes Potential zugeordnet. Die Beobachtungsposition über dem Zielobjekt erhält ein punktförmiges, die bevorzugte Flughöhe – aktuell 10 m über Grund – ein flächenförmiges anziehendes Potential. Die Stützpunkte der Soll-Flugbahn ergeben sich als Folge von 50 cm weiten Schritten, deren Richtung durch den Gradienten des Potentialfeldes definiert wird. Im Falle eines lokalen Minimums wird das Sollflughöhe-Potential vorübergehend „abgeschaltet“ – der Roboter weicht dann nach oben aus. Sobald während des Zielflugs ein neues Ziel- oder Hindernisobjekt von der Bildauswertung registriert wird, erfolgt eine sofortige Neuplanung im Hinblick auf das aktuelle Ziel und die Kollisionsvermeidung.

Am Beginn des Flugs werden so lange Beobachtungspunkte in großer Höhe vorgegeben, bis die von dort gemachten Übersichtsaufnahmen Anhaltspunkte für mögliche Zielobjekte liefern, die anschließend – mit Hilfe der beschriebenen Strategie – gezielt angefliegen werden, um den Zuverlässigkeitsgrad der Erkennung zu erhöhen.

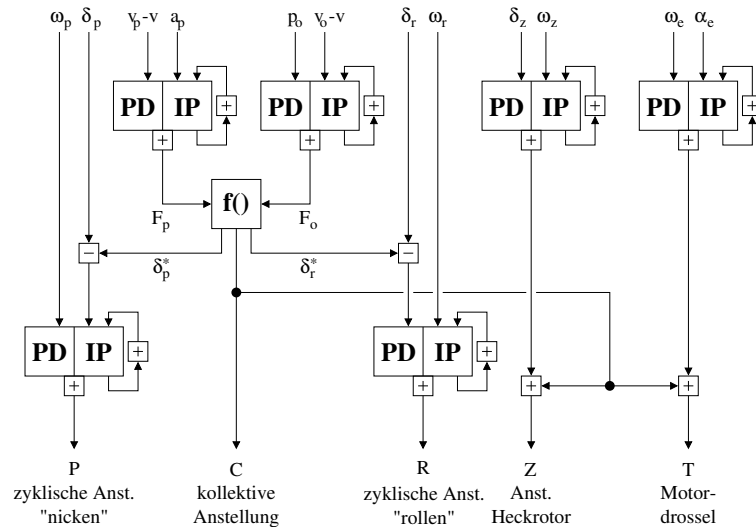


Abbildung3. Hierarchie der Fuzzyregler in der Flugregelung.

6.3 Flugregelung

Das Flugreglermodul erhält die Sollkursvorgabe sowie alle gemessenen Positions- und Lagedaten als Eingaben und berechnet daraus die Signale zur Ansteuerung der Servos. Der Sollkurs ist stets eine Gerade und wird durch zwei benachbarte Stützpunkte (siehe Abschnitt 6.2) und eine Sollgeschwindigkeit beschrieben. In der folgenden Ein-/Ausgaben-Tabelle sind für die Meßwerte hinter „/“ die jeweiligen Ableitungen angegeben:

Sollkurs	Meßwert-Eingaben	Servo-Ausgaben
A 1. Stützpunkt	p_o/v_o Abstand vom Kurs	C kollektive Anstellung
B 2. Stützpunkt	v_p/a_p Geschw. entlang Kurs	P zykl. Anst. nicken
v Geschwindigkeit	δ_z/ω_z Gierwinkel zu $\overline{A_{xy}B_{xy}}$	R zykl. Anst. rollen
	δ_p/ω_p Nickwinkel	Z Heckrotoranstellung
	δ_r/ω_r Rollwinkel	T Motordrossel
	ω_e/α_e Motordrehzahl	

Der Regelalgorithmus selbst besteht aus einer Hierarchie von Fuzzy-Reglern [6]. Jeder Elementarregler erhält eine skalare Regelgröße und deren Ableitung zur Eingabe und bestimmt einen skalaren Ausgabewert. Beide Eingabevariablen werden zu linguistischen Variablen mit je fünf Termen fuzzifiziert. Die Auswertung der 5×5 -Regelmatrix erfolgt mit der Minimum-Operation als Fuzzy-Konjunktion. Da als Regelantworten Einzelwerte (*singletons*) verwendet werden, erfolgt die Zusammenfassung der Regelantworten als mit den Erfüllungsgraden der Regeln gewichtete Summe.

Abbildung 3 zeigt den Aufbau des Flugreglers. Die Quadratpaare mit der Beschriftung „PD/IP“ stellen jeweils ein Paar der beschriebenen Elementarregler mit denselben Eingaben dar. Einer der beiden Regler (IP) integriert seine Ausgaben pro Regeltakt, der andere (PD) wirkt direkt auf die Stellgröße. Formal

für Regelgröße x und Stellgröße y :

$$I(t) := I(t-1) + f_{IP}(x(t), \dot{x}(t)) \quad y(t) := I(t) + f_{PD}(x(t), \dot{x}(t)) \quad (3)$$

Dies ergibt eine besondere Art eines Fuzzy-PID-Reglers. Die Integration der Reglerantwort statt des Fehlers der Regelgröße hat mehr technische Aussagekraft. Der Initialwert $I(0)$ kann z. B. dazu dienen, einen stetigen Verlauf der Stellgröße beim Übergang in eine neue Flugphase zu erzwingen.

In der Abbildung erkennt man, daß die Flugregelung zunächst Sollkräfte parallel und senkrecht zur Kursgeraden berechnet (F_p, F_o). Diese werden ins Helikopterkoordinatensystem transformiert und ergeben dort Sollwerte für die kollektive Hauptrotoranstellung (C) sowie für die Roll- und Nickwinkel (δ_r^*, δ_p^*), welche ja für horizontale Kraftwirkungen verantwortlich sind.

7 Schluß

In diesem Papier wurde das Konzept des autonomen Flugroboters MARVIN beschrieben. Die meisten vorgestellten Komponenten und Module sind bereits einzeln oder in Simulationen getestet worden. Die Systemintegration und der Systemtest – also letztendlich die Erprobung im autonomen Flugbetrieb – stehen zum aktuellen Zeitpunkt, zu dem dieses Papier geschrieben ist, noch aus.

MARVIN ist das Resultat einer mehrsemestrigen Projektveranstaltung mit zahlreichen Studierenden. Allen Beteiligten gilt unsere Anerkennung für ihren unermüdlchen Einsatz, der weit über den üblichen Lehrbetrieb hinausging. Besonderer Dank gebührt allen, die durch technische und finanzielle Unterstützung ein so aufwendiges Projekt erst möglich gemacht haben: Daimler Benz Aerospace AG (Modellhelikopter), Patrick Magnus (Helikoptertechnik und Probeflüge), Fast Multimedia (Framegrabber-Karte), NovAtel Inc. (GPS-Leihgabe), Novedia Media Solutions (finanzielle Unterstützung), Siemens AG (PC, Mikrocontroller), Tasking BV (preiswerte 167er-Entwicklungsumgebung) und TFH Berlin (Platinenherstellung).

Literatur

1. THE MILLENNIAL EVENT: Rules for the 1998 International Aerial Robotics Competition qualifier.
<http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/1998CollegiateRules.html>.
2. U. W. Brandenburg, M. Finke, D. Hanisch, M. Musial, and R. Stenzel. TUBROB – An autonomously flying robot. In *Proc. AUVS Symposium*, Seiten 627–636, Washington D.C., USA, 1995.
3. U. W. Brandenburg, M. Finke, and M. Musial. Aufbau und Steuerung des fliegenden Roboters TUBROB. In *Tagungsband 11. Fachgespräch Autonome Mobile Systeme*, Seiten 100–109, Karlsruhe, 1995. Springer-Verlag.
4. U. W. Brandenburg, M. Musial, and G. Hommel. MARVIN – Technische Universität Berlin's never-depressed flying robot for the IARC'98. Erscheint in *Proc. AUVS Symposium*, Washington D.C., USA, 1995.
5. Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
6. Jörg Kahlert and Hubert Frank. *Fuzzy-Logik und Fuzzy-Control*. Vieweg, Braunschweig, Wiesbaden, 1994.
7. Siemens AG, München. *C167 Derivatives*, Edition 03.96, 1996.