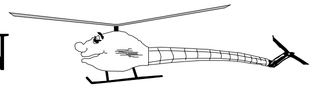
MARVIN 

Technische Universität Berlin's Never-Depressed Flying Robot for the IARC'98

Uwe Wolfgang Brandenburg Günter Hommel Marek Musial

http://pdv.cs.tu-berlin.de/leute/{brg,hommel,musial}
Technische Universität Berlin
Real-Time Systems Group

Abstract

MARVIN is an autonomously flying robot based on a model helicopter. It is designed for search and reconnaissance operations in an unknown environment with possible treats such as fires, water fountains, and smoke. The overall system consists of the helicopter with a microcontroller based on-board computer and a ground station computer. While flight control is done on-board, mission planning, human user interaction, and video image processing take place on ground. Sensors on board the helicopter include carrier phase DGPS, acceleration, compass, and gyro sensors, ultrasonic and infrared surface and threat detection, and a color CCD camera. MARVIN uses a hierarchy of fuzzy controllers for flight control. Flight path planning and collision avoidance are realized using a potential field approach with a dynamic object map.

INTRODUCTION

As the reader might suspect, MARVIN is an abbreviation. It stands for **M**ulti-purpose **A**erial **R**obot **V**ehicle with **I**ntelligent **N**avigation. It has to be strongly affirmed that any resemblance to Marvin in [1] is more or less incidental, because TU Berlin's flying robot does not contain any components, neither in hardware nor software, susceptible to depression.

The requirements to be fulfilled by MARVIN are defined by the mission of the 1998 International Aerial Robotics Competition Collegiate Event, as described in [2]. The mission task consists of finding and classifying hazards and victims in a simulated disaster area. There are black drums containing dangerous materials, with the contents distinguishable by symbols on the drums' surface, and there may be fires, water fountains, and smoke and fog threatening the operation of the robot. Victims may be dead or injured persons, the survivors recognizable by motion and sound. Neither their appearance nor the characteristics of the surrounding terrain are specified by the rules.

In contrast to TU Berlin's participation in 1995 [3, 4] with the blimp-based robot TubRob, for 1998 a model helicopter has been chosen because it has more power reserves to steer against winds, it can be set up more quickly, and it is more precise for the object acquisition tasks planned in the future competitions.

The remaining sections of this paper deal with an overview of the system, the air vehicle, the sensors on board the robot, computer and communication hardware, software structure, and the safety system. The paper closes with acknowledgements to all those who have made the creation of MARVIN possible.

SYSTEM OVERVIEW

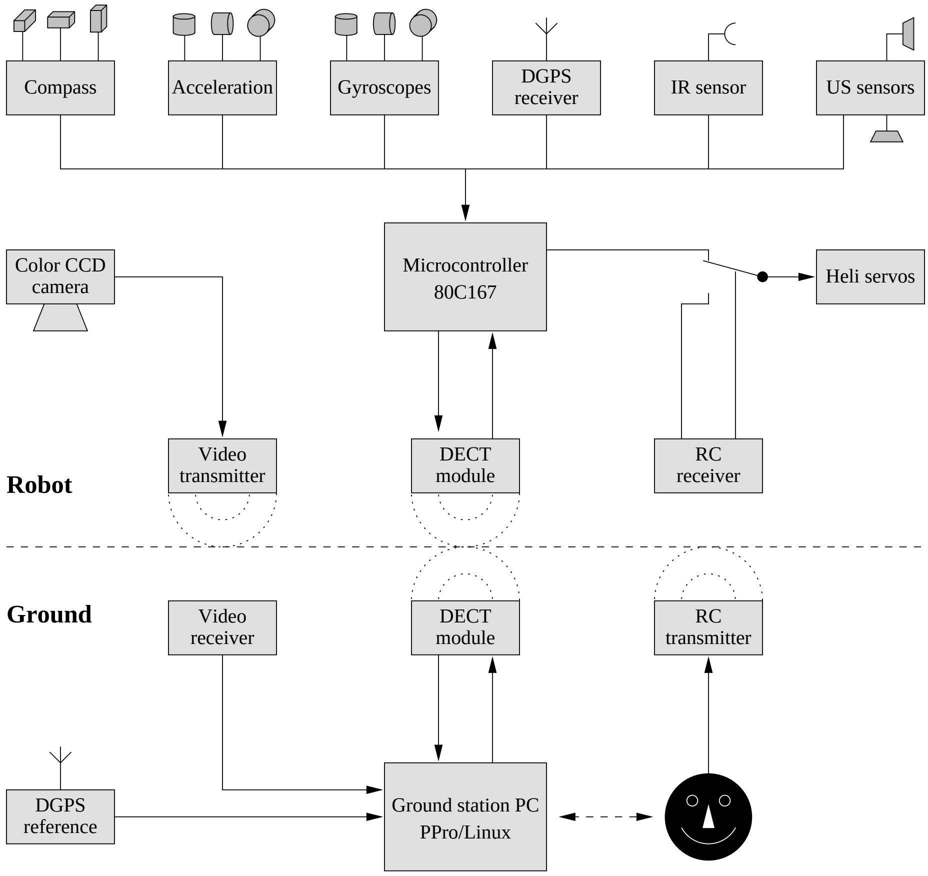


Figure 1: Overview of the system structure for MARVIN

Figure 1 provides an overview of the MARVIN system. The sensors on board are three magnetic field sensors, three acceleration sensors, three gyroscope rotation sensors, one infrared photo sensor, two high-performance ultrasonic echo sensors, a color CCD camera looking down, and a carrier phase differential GPS receiver. The sensors (except the camera) and the actuators – the servos in the helicopter – are connected to the board computer, which is a Siemens SAB80C167 microcontroller. The ground station consists of a PC running Linux with a frame-grabber card and the DGPS reference receiver. The communication channels between helicopter and ground station are a digital bidirectional data link set up by modified components for digital cordless telephony, the video image transmission and an RC transmitter for both backup manual control and the termination mechanism.

HELICOPTER

The helicopter is a conventional model helicopter, the “Petrol Trainer” 180 by SSM, Germany. It has a rotor diameter of 1.8 m and is equipped with a 22 cm^3 two-stroke petrol engine producing about 2 kW. The usable payload reaches about 5 kg including fuel. For an operation time of 30 minutes, this payload has to be divided into 1 kg of fuel and 4 kg of that equipment that turns the helicopter into MARVIN.

The helicopter controls are driven by six servos:

- Four servos for combined collective and cyclical pitch adjustment of the main rotor – front, rear, left, and right.
- Tail rotor pitch servo.
- Engine throttle servo.

Four servos for main rotor pitch are redundant, because the pitch plane is already determined by three points. This means that the servo signals have to be calculated suitably to prevent the servos from pushing against each other. On the other hand, the benefit from redundancy is that pitch control might still work when one of the servos fails – without blocking, anyway.

In order to alleviate both manually controlled flights and automatic flight control, the helicopter's tail rotor servo is connected to a conventional piezo-electric gyroscope sensor with integrated p-controller, as used in most radio-controlled model helicopters today. This gyro module adds its controller output to the input originating from the RC receiver or – in autonomous flight – from the microcontroller.

SENSORS

This section deals with the sensors on board the robot and the evaluation of their outputs.

Position and Attitude

Table 1 describes the position and attitude sensors and lists the physical data delivered by each of them, possibly after some straightforward calculations on the sensor output. The upper index ^{H} denotes parameters in the helicopter's reference system, whereas ^{W} denotes the world (= arena) reference system.

Because of the relatively big error of the inexpensive gyroscopes, the integrated angle values might drift by up to $5°\text{min}^{-1}$. Therefore, the ϕ-angles have to be recalibrated regularly based on the compass and acceleration data: First, the compass only supplies the two angles β_z^H, β_y^H because it measures the direction of the magnetic field vector, but a single vector does not tell anything about rotations around itself. Second, The acceleration measurement consists of the gravity field vector of the earth and the dynamic acceleration $\vec{a}^W$ obtained by the robot's movement, rotated according to the robot's actual orientation:

$$\vec{a}_g^W = (0, 0, -1\text{g})^\text{T} - \vec{a}^W \qquad \vec{a}^H = \text{rotate}_{\vec{\phi}^W}(\vec{a}_g^W) \tag{1}$$

In (1), $\vec{a}^W$ can be obtained from the GPS data. The two angles γ_z^H, γ_y^H describe the yaw and pitch of the $\vec{a}_g^W$ vector in the helicopter system, again not telling about the rotation around this vector. In general, i.e. when $\vec{a}_g^W$ is neither zero nor parallel to the magnetic field, $\vec{\phi}^W$ can be calculated from β_z^H, β_y^H and γ_z^H, γ_y^H. This result is used for the recalibration of the gyroscopes every few seconds, depending on data availability and quality.

TABLE 1: POSITION AND ATTITUDE SENSORS AND THEIR MEASUREMENTS

Sensor(s)	Description	Parameters
Carrier phase differential GPS receiver	RT-2 by NovAtel, Canada (offered on loan by the manufacturer to all IARC teams).	*Position* p_x^W, p_y^W, p_z^W and *speed* v_x^W, v_y^W, v_z^W in world coordinates.
Electronic compass	Three orthogonally mounted fluxgate magnetic field sensors with PWM digital output.	Yaw and pitch angles β_z^H, β_y^H of the earth's magnetic field vector within the helicopter system.
Acceleration sensors	Three orthogonally mounted semiconductor acceleration sensors with analog output.	Total virtual acceleration a_x^H, a_y^H, a_z^H in the helicopter system.
Gyroscopes	Three orthogonally mounted model helicopter tail rotor controllers with piezo-electric rotation sensors. Output is PWM digital.	Helicopter roll rotation frequencies $\omega_z^H, \omega_y^H, \omega_x^H$ and, by numerical integration, corresponding angles $\phi_z^H, \phi_y^H, \phi_x^H$.
Ground detector	Echo-sounder using an ultrasonic transducer by Polaroid	Altitude over ground h^W.

Threat Detection

For in-flight threat detection, the following sensors are used to prevent MARVIN from crashing into obstacles or being burnt down by large fires:

- A forward-looking ultrasonic echo-sounder, like the one for ground detection. Its operating range is 10 to 15 meters, depending on the structure and surface of the obstacles.
- An infrared sensor for the recognition of the heat of fires that could become threatening to MARVIN at its current position.

Computer Vision

A downward-looking wide-angle fix-focus color CCD camera with TV resolution is used for image acquisition. The recognition of the different objects is performed in a sequence of steps, as usual in computer vision. It profits from the measurement of the altitude over ground, which allows to estimate the object size within the image for drums and human bodies. In the following, the recognition algorithms for the most important objects are briefly described:

Drums
1. Color segmentation by the pixels' distance from "black" in color space.
2. Eliminate noise and unwanted segments and gaps by erosion and dilatation.
3. Kill segments of unsuitable size.
4. Calculate rotation-invariant features, e.g. central moments, and classify the feature vector to yield a "drum probability".

Drum Labels
1. Color segmentation of "white" pixels in the region of a recognized drum.
2. Look for accumulations of segments from step 1.
3. Calculate a feature vector of rotation- and scaling-invariant features for the classification of the potential labels.

Human bodies
1. Color segmentation – according to a high-sophisticated size-related rotation-invariant sample pixel mask – of pixels that are "unusual" in their 1–2 m environment.
2. Evaluation of a human-body-specific rotation-sensitive filter matrix at image positions preselected by step 1, for a number of rotation angles.

Body movement
1. Calculation of central moments for segments according to step 1 of body recognition in a sequence of subsequent images.
2. FFT of the time/moment pairs from 1 to detect periodic changes.

Fires
1. Find spots exceeding a certain brightness threshold.
2. Filter out those which are only detected from a certain position (which are direct reflections of sunlight).

All objects reported by the vision algorithms get a probability attribute denoting the certainty of recognition.

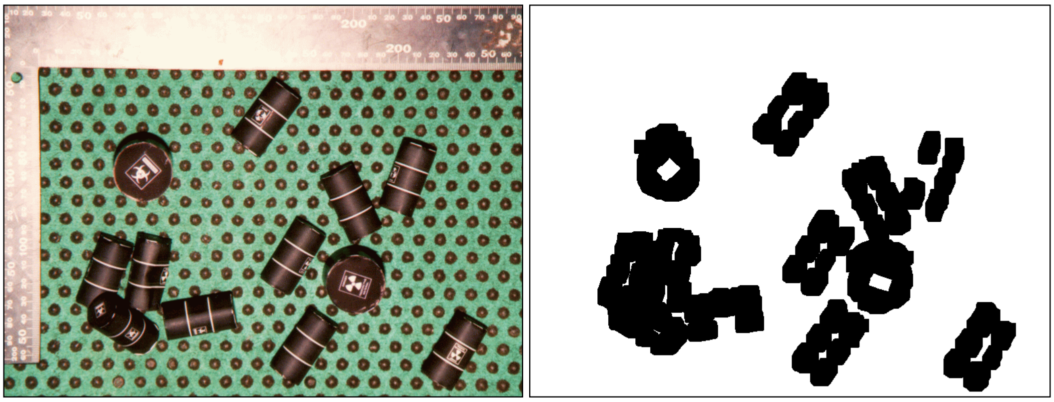


Figure 2: Sample input (*left*) and result (*right*) of drum segmentation

Figure 2 shows a test image and the drums segments found by an early version of the recognition algorithm.

COMPUTER SYSTEM

This section describes the components of MARVIN's computer system. These are the ground station computer, the on-board computer, and the communication links between.

Ground Station

The ground station computer is a PC with a PentiumPro CPU (200MHz) with 64 MB RAM, running Linux and X-Windows. It hosts the frame-grabber board and is connected to the DGPS reference receiver via an RS232 interface. This computer serves as the user interface to the MARVIN system: It displays the mission progress, reports the recognized targets and allows to give the "start" and "stop" commands. Throughout system development, the ground PC provided valuable help in the real-time monitoring of arbitrary parameters. The principal advantage of performing the image processing on ground is that the computational power available for this complex task could be easily scaled by using more than one computer. By now, the PC can process an average of one image per three seconds, depending on the different phases in object recognition (see *Computer Vision*).

On-Board Computer

The heart of MARVIN is a Siemens SAB80C167CR-16F 16bit microcontroller, which constitutes the on-board computer. The 167 has been chosen because it is powerful enough to run all the on-board tasks *and* provides an inexhaustible number of peripheral interface lines. Thus, it is not necessary to use different kinds of processors for calculation and hardware interfacing, as it is always the case when PC-boards are used.

These are – in short – some outstanding features of the controller chip: 20 MHz internal clock rate, 16 bit memory interface, 32 capture/compare-units with dedicated interface lines for PWM signal generation or measuring at no CPU-cycle cost, 9 programmable timers, 16 A/D-converter lines (multiplexed), a peripheral event controller (PEC) for external event handling without actually calling an interrupt handler, very short response times thanks to interruptible instructions, programmable chip-select lines, 128 KByte internal flash memory, and a CAN bus interface.

The μC-board in MARVIN has been developed by the project team, for no commercially available board allows to make use of *all* important features provided by the 167. The MARVIN board carries 512 KByte SRAM with 0-wait-state access and can be *stacked* with special peripheral boards. It can also be stacked with further 167 boards for applications that require this, automatically providing a CAN bus through the stack.

Communication

The main communication link is the bidirectional digital data transmission between the PC and the on-board computer. It is realized using a pair of modified DECT station modules. DECT (*Digital European Cordless Telephone*) is the standard for current cordless telephones sold in Europe and, possibly, for local loop setup by new communication network providers who cannot use the telephone wire. Up to now, no DECT components are commercially available for data transmission, since voice communication is the original application of DECT. Siemens Corp., Germany, helped the MARVIN team to modify some of their DECT devices to overcome this gap.

The second communication link is the analog transmission of the video signal from the CCD camera. It uses a transmitter/receiver pair by VTQ, Germany, which transmits an FBAS signal with TV quality.

The third links is the PWM digital signal of the conventional remote control transmitter, which serves two purposes:

1. It allows to fly MARVIN manually for test purposes or in an emergency. Manual control is switched on or off via the RC system.
2. It serves as technically independent trigger for the required emergency termination mechanism.

Table 2 summarizes the main characteristics of the three radio links.

TABLE 2: TECHNICAL DATA OF THE RADIO COMMUNICATION LINKS TO MARVIN

	DECT link	**Video link**	**RC link**
Direction	full-duplex	down	up
Bandwidth	384 kbit/s	TV	$\approx$ 630 pulses/s
Range (outside)	300 m	500 m	2 km
Operation frequency	1.8 GHz	2.6 GHz	40 MHz
Signal power	200 mW	25 mW	(unspecified)

MISSION SOFTWARE

This section explains the structure of the MARVIN software and reveals some details about the most important parts of the software – the on-board "operating system", Mission Control, and the helicopter flight controller.

Software Modules

Figure 3 gives an overview of the software modules and their distribution among the two computers. A "module" stands for one or several processes on the Linux PC or a *task* on the 167.

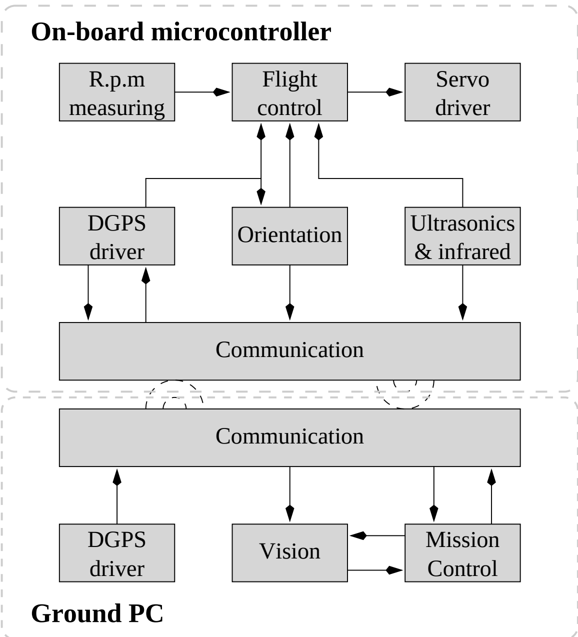


Figure 3: Software modules on the ground and in the air

Microcontroller "Operating System"

No real operating system is used on the controller board. Instead, the controller task modules are simply called from a main loop which is guaranteed to be executed at a rate of 20 Hz. Busy waiting is strictly forbidden for the controller tasks. Every task is allotted a maximum time slot per call from the main loop. It would have to return prematurely to proceed in the next cycle if some work cannot be finished within a single call. This tasking scheme simplifies the cooperation between tasks because there is no concurrency involved.

Inter-task communication is realized using a *shared-memory* approach: Every task, both on-board and on ground, accesses a "global" state description in this shared memory. The communication software takes care to synchronize the shared memory copies on the controller and the PC transparently via the DECT link. This provides a very comfortable data-level communication interface to the task programmers. In every case, tasks operate on the latest available values respectively of all parameters.

Mission Control

The *Mission Control* module is responsible of planning and carrying out the search mission with respect to (**a**) maximum safety for MARVIN and (**b**) the achievement of a maximum number of points according to [2] in the shortest possible time. This important task has been divided into four subtasks, each of which is solved by a process of its own:

1. The *vision listener* receives information from the vision module about recognized objects and maintains an object map of the mission area. Entries in this map may be uncertain (see section *Computer Vision*).
2. The *mission sequencer* sorts the list of objects according to their importance for the mission and their respective distance from MARVIN's current position. The topmost entry represents the target to be flown to next for further identification or classification.
3. The *flight supervisor* plans the flight path to the current target position. This is realized using a 3D potential field approach known from collision avoidance in robotics: Recognized and briefed obstacles in the current map are assigned a repelling potential, the target and the recommended flight level get an attracting one. From MARVIN's current position, the flight supervisor calculates, several times a second, the flight direction and altitude that maximize the potential. When new obstacles are recognized during the flight, dynamic re-planning takes place.
4. The *user interface* visualizes the object list and the map and allows user intervention for testing purposes.

In case of problems, each of these processes can be restarted at run-time without aborting the mission.

The initial strategy for finding first possible objects is to take a number of "overview" pictures from an altitude of 30 to 50 meters yielding a relatively big number of quite uncertain object entries. These can then be processed by the mission sequencer as described above.

Flight Stabilization and Control

The flight controller module gets the intended course data as a single input and a number of position and attitude parameters as periodic inputs, generating a set of servo adjustment signals. The course is a straight line through a support point in the plane. For each controller input, the derivative is shown behind "/":

Course description		**Controller inputs**		**Controller outputs**	
c_x	support point x	p_z/v_z	z-position (altitude)	C	collective pitch
c_y	support point y	p_o/v_o	distance from course	P	cyclic pitch along course
ϕ_z	flight direction	v_p/a_p	speed along course	O	cyclic pitch around course
v	flight speed	δ_z/ω_z	yaw angle, from ϕ_z	Z	tail rotor
h	flight altitude	δ_p/ω_p	pitch angle, in course dir.	T	engine throttle
		δ_o/ω_o	roll angle, around course		
		ω_e/α_e	engine r.p.m.		

The controller module consists of a hierarchy of fuzzy controllers. Each controller gets one parameter and its derivative as inputs, yielding one scalar output. Each input is fuzzified according to five linguistic variables with completely overlapping triangular membership functions summing up to unity. The 5×5 output rules are evaluated by min-max-inference using minimum as fuzzy conjunction operator and maximum for disjunction. Defuzzification is done by the center-of-gravity-method using seven singletons per output linguistic variable[1].

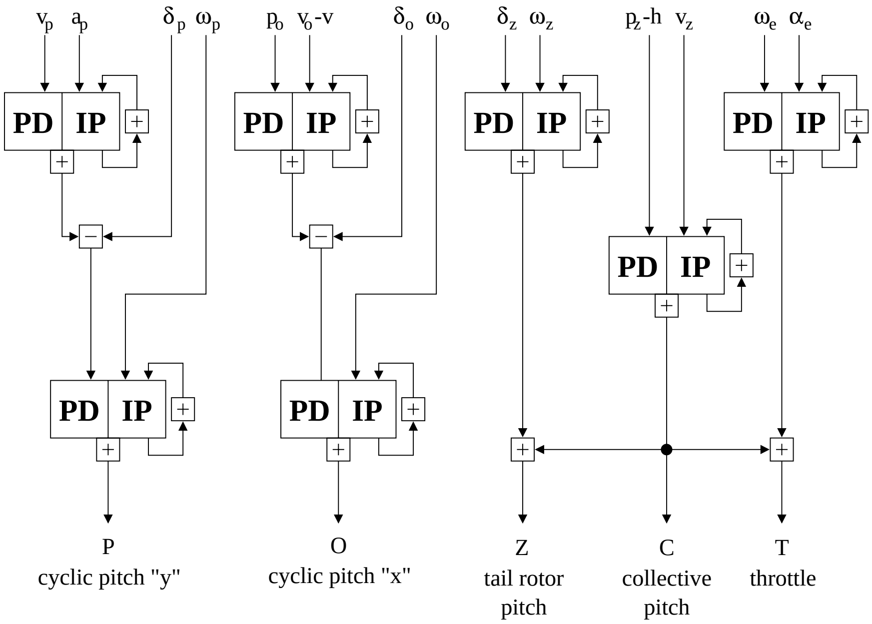


Figure 4: Hierarchy of fuzzy controllers for flight control

Figure 4 shows the structure of the flight controller. Double squares denote pairs of the explained fuzzy controllers getting the same inputs, one of them (IP) integrating its responses, the other (PD) outputting them directly, thus realizing a fuzzy PID-controller. Note that most controller in- and outputs refer to a coordinate system defined by the course direction.

SECURITY SYSTEM

The safety measures in the MARVIN system are organized in four steps:

[1]This means they are two-dimensional response field controllers with 25 support points and a piecewise linear response field in between. “Fuzzy controller” is just the more common term for such items.

1. If the DECT communication link fails, the GPS positioning data loose their precision quickly and MARVIN does no longer obtain a secure flight path from Mission Control. Therefore, the flight controller enters a safety mode, trying to stabilize the helicopter and to keep the current position, when no valid data packets have been received for 2 s.
2. When the flight controller detects a serious problem – for instance a roll angle of $45°$ – an optic and acoustic warning is issued to the ground staff. Manual control is *never* selected automatically because this makes no sense unless the human pilot is really ready to take over.
3. Manual control can be switched at any time via the RC transmitter. This is done independently of the on-board computer and its power supply. There is a backup accumulator for the RC receiver and servos which is not used unless the main accumulator is empty.
4. The engine of the helicopter is adjusted such that it stops when the throttle servo is set to the minimum position via the RC transmitter. This constitutes the flight termination mechanism as required by the rules: It works on separate power supply and uses an independent radio link.

ACKNOWLEDGEMENTS

MARVIN would never have been possible without lots of valuable support. We wish to thank: Daimler Benz Aeorospace AG (helicopter), Patrick Magnus (helicopter optimization), Fast Multimedia (frame-grabber), NovAtel (GPS loan), Novedia Media Solutions (financial support), Siemens AG (DECT link, PC, microcontrollers), Tasking BV (rebate on C167 development software), and TFH Berlin (board production).

MARVIN is the result of a student project. People who have been working nights and days are: Thomas Baldauf, Martin Bender, Axel Berres, Roland Böhrenz, Michael Christmann, Jörg Conradt, Jan Ebert, Sven Fischer, Jens Glagau, Matthias Hendler, Oliver Hesse, Peter Heyer, Lars Kitschke, Oliver Klenke, Michael Knoke, Kay Kossowan, Enrico Krull, Martin Kuhn, Markus von Lehmann, Marco Liem, Günther Martinez Dreyer, Christian Merten, Frederik Meysel, Marina Müller, Roland Pauly, Jörn Plöger, Stefan Pohle, Marcus Pritsch, Frank Rehberger, Christian Reinicke, Volker Remuß, Stefan Rönnecke, Ulli Romahn, Simon Schaarschmidt, Michael Schmidt, Carsten Schmoll, Bernd Schöller, Bodo Sievers, Torsten Steinert, Markus Steiof, Malte Stien, Marc Stroh, Necmetti Türkkal, Andreas Wege, Thomas Zedler, Qi Zhu.

REFERENCES

[1] Douglas Adams. *The Hitchhiker's Guide to the Galaxy*. POCKET BOOKS, New York, 1981.

[2] THE MILLENNIAL EVENT: Rules for the 1998 International Aerial Robotics Competition qualifier. http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/1998CollegiateRules.html.

[3] U. W. Brandenburg, M. Finke, D. Hanisch, M. Musial, and R. Stenzel. TUBROB – An autonomously flying robot. In *Proc. AUVS Symposium*, pages 627–636, Washington D.C., USA, 1995.

[4] U. W. Brandenburg, M. Finke, and M. Musial. Aufbau und Steuerung des fliegenden Roboters TUBROB. In *Tagungsband 11. Fachgespräch Autonome Mobile Systeme*, pages 100–109, Karlsruhe, 1995. Springer-Verlag. In German.