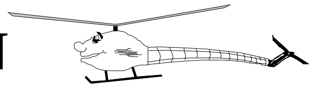
MARVIN 

Technische Universität Berlin's Flying Robot Competi
IARC'99

M. Musial G. Hommel U. W. Brandenburg
E. Berg M. Christmann C. Fleischer C. Reinic
V. Remuß S. Rönnecke A. Wege

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

Abstract

MARVIN is an autonomously flying robot based on a model helicopter. It is desi
participation in the International Aerial Robotics Competition (*IARC*) Millennial Even
2000. The competition task consists of a search operation in an unknown environm
possible threats such as fires, water fountains, and smoke. The overall system co
the helicopter with an on-board computer and a second computer serving as a grounc
While flight control is done on-board, mission planning, human user interaction, and di
age processing take place on ground. Sensors for autonomous operation include carri
DGPS equipment, acceleration, magnetic field, and rotation sensors, ultrasonic transdu
a flame sensor. Image acquisition is done through a digital photo camera. Flight path p
and collision avoidance are realized using a dynamic object map.

INTRODUCTION

MARVIN is an abbreviation for **M**ulti-purpose **A**erial **R**obot **V**ehicle with **I**ntellige
Besides reading as a frequent name for robots, this description expresses that MAR
tonomously flying robot that can fulfill certain missions purely on the basis of senso
any human interaction.

The requirements to be fulfilled by MARVIN are defined by the mission of the
tional Aerial Robotics Competition Collegiate Event, as described in [1]. The missio
of finding and classifying hazards and victims in a simulated disaster area. There ar
containing dangerous materials, with the contents distinguishable by symbols on th
face, and there may be fires, water fountains, and smoke threatening the operatio
Victims may be dead or injured persons, the survivors recognizable by motion and s

The historic background of MARVIN involves TU Berlin's participation in the 19
the blimp robot TubRob [2, 3] and the presentation of MARVIN's technical conceptio
competition [4, 5].

The remaining sections of this paper deal with an overview of the system, t
emergency and safety procedures, the navigation scheme, measure for threat avoi
image acquisition and processing approach. The paper closes with acknowledgeme
who have made the creation of MARVIN possible.

SYSTEM OVERVIEW

Figure 1 provides an overview of the MARVIN system.

Sensors

The sensors on board are:

- one infrared light barrier to measure the current rotor rpm,

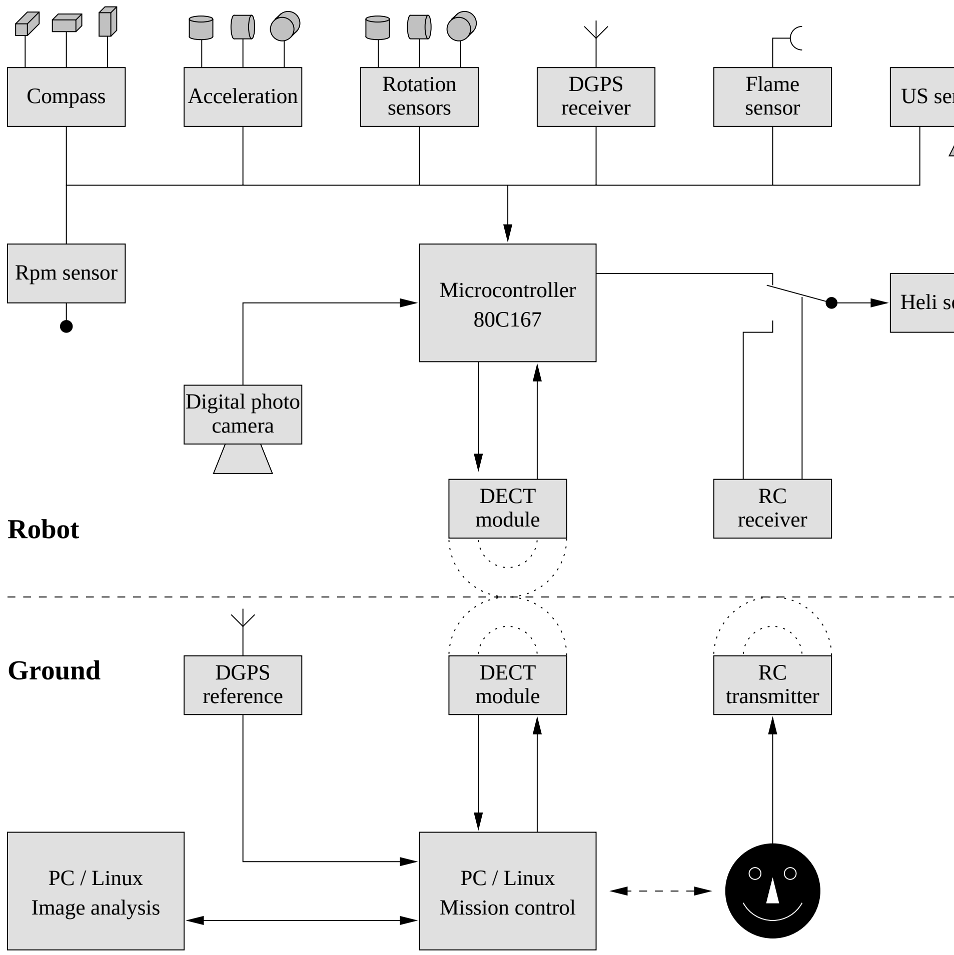


Figure 1: Overview of the MARVIN system structure

- three magnetic field sensors,
- three semiconductor acceleration sensors,
- three piezo-electric rotation sensors,
- one flame sensor based on ultraviolet light detection,
- two high-performance ultrasonic echo sensors, one of them looking down and t ing ahead,
- a digital photo camera looking down, and
- a NovAtel RT-2 carrier phase differential GPS receiver.

Computers

All devices on board are connected to the on-board computer, which is a si SAB80C167 microcontroller. The ground station consists of two PCs running Linux reference receiver. One of the PCs run the mission control software and acts as a us the overall system, while the other runs the computationally expensive computer visi

The on-board computer carries 256 KBytes RAM. The controller board has b designed for MARVIN by the TU Berlin team. The 80C167 controller has been chos

- its performance is sufficient for all on-board computations and
- its peripheral connectivity allows the processing of all analog and digital se control of the servos at minimum circuitry overhead and almost no additional

Communication

Communication between helicopter and ground is performed via two Siemens Gigas communication modules. These modules use the DECT (*digital enhanced cordl* technology stemming from cordless phones and operate at 1.8 Ghz. A usual RC 37 MHz serves for both backup manual control and the termination mechanism ac competition rules.

The communication software written for MARVIN provides a notion of a *shared* that is visible to all computers and processes in the system. The microcontroller and processes constitute nodes in the overall communication network, and whenever a *shared memory* is changed at any of these nodes, the *shared memory* contents are transparently throughout the whole network. The *shared memory* contains sensor d erence information, image data, flight path information and all other things that are local interest. While the network link to the on-board computer is a serial interface TCP/IP sockets. Thus, the number of PCs working together on the ground can be without changing a single line of software. MARVIN could even be controlled via th

AIR VEHICLE

The basis of the MARVIN air vehicle is a conventional model helicopter, the "Petro by SSM, Germany. It has a rotor diameter of 1.8 m and is equipped with a 22 cm^3 two engine producing about 2 kW. MARVIN's maximum takeoff weight is about 10.5 of 500 g of fuel, 6 kg unladen weight of the helicopter and about 4 kg of "specia This allows for an operation time of about 15 minutes. With respect to the 60 min for performance judging, 15 minutes flight time per attempt are considered sufficien neither realistic nor desirable to plan for less than – say – three attempts.

In order to alleviate both manually controlled flights and automatic flight contro tor control signal is fed through a conventional piezo-electric gyroscope, as used i controlled model helicopters today. This gyro module adds its controller output to th nating from the RC receiver or – in autonomous flight – from the microcontroller.

A wooden platform beneath the helicopter carries all the special equipment nec tonomous operation. Electric power is supplied by accumulators. Figure 2 show MARVIN on ground.

SAFETY PROCEDURES

This section describes the safety measures provided in the MARVIN system for case
operation. There are two stages of such procedures: Procedures for secure retrieval of
the flight termination procedure as required by the competition rules.

Secure Retrieval

The first component which might fail is the digital communication link. Its failure m

1. the GPS reference data are no longer transmitted to the on-board receiver, ca
   to loose precision quickly, and

2. flight path data are no longer provided from the “mission control” module

In this case, the flight control software on-board the robot switches automatically
flight mode. Failsafe flight mode means that flight control ignores GPS data complet
controlling the robot’s position and speed, it only tries to keep the robot’s current or
preventing the helicopter from turning over and possibly crashing. This maximizes th

- flight control can easily regain full command of the helicopter without huma
  when the GPS output is soon reestablished, or that
- the human “backup” pilot manages to stabilize the helicopter whenever this s
  to be necessary (see next paragraph).

If the flight control software obviously fails to stabilize the helicopter, the servo in
can at any time be switched away from the microcontroller and back to the RC
an otherwise free channel of the usual remote control unit. This switching facili
in figure 1. When the microcontroller is reset due to a failure in its power supply
accumulator supplying the microcontroller is exhausted, control of the helicopter is
to the human pilot even without his intervention. This is guaranteed because the co

are relays that fall back mechanically into “human” mode when power becomes ins RC receiver and the servos are independently powered, which means that MARVI fully operational conventional RC helicopter whenever one of the above cases occurs

Termination Procedure

A second switch on the RC transmitter – the autorotation switch – serves as in-flight Its use renders the robot ballistic – as far as possible for a helicopter with a rotor at order to do emergency termination, the backup pilot just needs to throw two switche switch for regaining manual control (with his left forefinger) and the autorotation sw right forefinger). Additionally, this double-switch approach provides some protecti unintentional use of the cutoff feature. This termination mechanism fulfills the requir in the rules because

- the RC transmitter is not used during autonomous operation and
- the RC receiver on board, which is not used during autonomous operation eith pletely separate power supply.

NAVIGATION

MARVIN’s navigation scheme consists of two parts:

1. Flight route planning, also referred to as *mission control*, which is performed b ground.

2. Flight control, i. e. the *autopilot*, which runs on the on-board computer.

Flight route planning or *mission control* is the high level control task within the overall sion control transmits an intended course line to the robot, and the autopilot on-board of following this course line. The following sections deal with these two parts respec

Flight Route Planning

The objectives to be fulfilled by the mission control software are the success of the s
on the one hand and the generation of collision free paths that can be safely followe
on the other.

Mission control operates on the basis of a dynamic object map covering the com
This map contains entries for

- potential or recognized target objects and
- obstacles and threats.

Note that certain obstacles, such as fires, constitute both a target and an obstacle. Bri
can be entered into the initial object list through a graphical user interface prior to th

When the mission begins, the robot ascends to about 50 m above ground in order
ber of overview images. The coordinates where these images are taken are calculat
control according to the extents of the competition arena, which, of course, have to b
operator. The overview images are processed by the vision software running on th
images are searched for potential objects, which are inserted into the object map t
measure of recognition probability.

After this phase, mission control sorts the object map entries according to the rec
abilities, their relevance with respect to the scoring scheme, and their distance fr
position of the helicopter. In the order obtained in this way, mission control guide
the object positions for further clarification of the "potential" entries. At these posi
images are taken, resulting in either the deletion of the respective object entry fr
the classification of the object as "certainly recognized". In the latter case, the obj
displayed on the screen of the mission control PC together with the object type. Dur
clarification phase, the object map is updated every time a new object is found, possi
and the prioritization of the objects is recalculated whenever necessary.

Figure 3 depicts the X-Window user interface of the mission control software wi
object positions displayed.

Flight Control

The autopilot module first has to provide flight stabilization, since a helicopter is by its air vehicle. Second, it has to follow the linear flight path segments determined by mi

A prescribed course as being issued by mission control consists of:

- Two support points, *from* and *two*, defining a straight line in three-dimensional nates.
- A designed scalar speed *v* at which the course line shall be followed.
- A flight phase specifier *ph* to distinguish between passive mode, on ground with rpm, start, normal flight, and landing.

The autopilot uses a hierarchy of controllers with bounded piecewise linear tran Most of the controllers get one state variable x and its deviation $\dot{x}$ as inputs and comp y as follows:

$$\begin{aligned} y_i &:= y_i + f_3(x) + f_4(\dot{x}) \\ y &:= y_i + f_1(x) + f_2(\dot{x}) \end{aligned}$$

That is, y_i serves as a memory for controller outputs being integrated over time. are computed according to the transfer functions f_3 and f_4, whereas f_1 and f_2 affe y directly. Thus there are P (proportional), I (integral) and D (differential) comp controllers, but arranged in a somewhat unorthodox manner.

This arrangement has been chosen because it allows to force the current control certain value by just calculating the required value of y_i. This enables the microcont take over control of the helicopter: The microcontroller permanently measures the se long as a human pilot is steering via remote control. When the pilot software is inst over control, the controller outputs are initially forced to the very same values the hu used. While this is not required for autonomous takeoff, it constitutes an condition for in-flight takeover during the development phase.

Figure 4 depicts the architecture of the flight controller. Each unlabeled rectan one of the elementary controllers as described above. Control of the cyclic rotor p performed in two stages: MARVIN's speed and position relative to the course lin calculate designed orientation angles, because the helicopter has to be tilted in orde certain direction. These designed orientation angles are rotated into the helicopter's cu angle and then compared with the current pitch and roll angles, yielding the steering cyclic rotor pitch setting.

THREAT AND COLLISION AVOIDANCE

The flight paths generated by mission control are concatenations of linear segments. avoidance, segments leading through or touching obstacle entries in the map are byp points in the neighborhood of the respective obstacle.

New obstacles are not only detected by the vision system (which would not work tudes anyway), but also using the ultrasonics sensor looking ahead and the flame sens on of these sensors detects an obstacle in the flight path, the robot is stopped and recognized edge of the obstacle is put into the map, thus forming a complete obsta the object has been approached from three or more sides.

Additionally, mission control uses the measurements of the downward-looking ul sor to add ground profile information to the arena map. This information is used to designed altitude during the clarification phase. This is necessary to prevent ground the environment is not flat.

VISION

In order to recognize the target objects in the mission, a computer vision system has consists of a part for image acquisition and a number of algorithms for image analys

Image Acquisition

In an earlier approach, an analog video camera has been used for image acquisition. crocontroller on board is not capable of image analysis, this approach required an add video data link and a frame-grabber card in the ground station PC. It caused strong and would have required substantial computational power to eliminate disturbances from the image data before one could have just started to think of object recognitio this first approach has been abandoned.

MARVIN carries a usual digital photo camera made by Sanyo. Image data are the camera's serial interface by the on-board computer and transmitted to the grou the digital data link, which has to be present anyway for GPS reference data, stat and human "intervention". This approach eliminates a second data link with its interference, power consumption, weight, and chance of failure.

The camera weighs only 200 g and transmits a JPEG-compressed image of 6 every 10 seconds. These images are of brilliant quality and free of disturbances o long as the helicopter is not moving too fast. The vision algorithms have been possible compression artefacts do not matter. Despite the low image rate in nor motion detection is still possible because the camera is able to record short "video up to 5 frames per second. The camera is even equipped with a microphone, which record short "comments" to images. In the IARC mission, this facility is used for th the sounds of survivors.

Image Analysis

For image analysis, there are a number of independent computer vision modules r or more PCs on the ground. These modules use different strategies to look for re in the camera images and pass their findings to the mission control software. M merges together the coordinates of a single object recognized by more than one v and arbitrates in the case of contradictory results. Every object recognition is att

recognition probability, so that mission control can use these hints both for result a
for target prioritization.

At the time of writing, most effort relates to the recognition of drum labels.
proaches used are basically different, the robustness of label recognition is remarka
following list outlines some of the vision modules implemented so far:

- The first algorithm for label detection computes correlations between pre-define
and the image at all possible positions and orientations. Since this would requ
amounts of computation time, there are a couple of optimizations. First, or
ages are used, generated from the camera images applying a measure of colo
locally adaptive threshold. Second, only sparse pixel masks are used to redu
of comparisons needed. Third, the symmetries of the labels are taken advanta
computation of a correlation is abandoned prematurely when there are to few
els. Figure 5 shows a part of a real aerial picture with two recognized labe
label.
- The second approach to label detection also starts with the binarization of th
segments within the binary image are collected by edge tracing and checked if
ably sized and rectangular. The remaining segments are used to cut out the p
from the original image, which are finally validated through cross-correlation
labels.
- The third label finding algorithm starts with image segmentation similar to th
Then, a number of central moments, being invariant to rotation and scaling,
for each segment, forming a very compact feature vector. The feature vectors
using a distance-based classifier in order to detect segments that are either com
parts thereof.
- One algorithm to find human bodies makes use of the fact that human skin ha
acteristic color – well, truly out one of a number of very characteristic color
suitably colored pixels are determined on behalf of a distance classifier in co
validated according to their size and arrangement. Of course, this algorithm do
every case, depending on the position of the body.

- For the evaluation of overview images – which do not allow the recognition of an algorithm has been implemented that searches for compact segments of a d color and suitable size. The size of interest is computed from the altitude at t has been taken. Potential drums have to be dark and uncolored, while human show any color from a greater distance because of clothing.

ACKNOWLEDGEMENTS

MARVIN would never have been possible without lots of valuable support. We w Oliver Klenke, Stefan Pohle and Roland Böhrenz, who developed parts of the vision cannot join our Hammer expedition; Matthias Jeserich for helping us as a human p welcoming us on their helicopter "airport" for testing; NovAtel Inc. for the loan of th ment; Daimler Benz Aerospace AG for the helicopter; SSM Technik Peter Schröppel some kilos of spare parts; Rotor Modellsport Center, Berlin, for even more thereof; Fa for a (no longer used) frame-grabber; Novedia Media Solutions for financial support for a PC, 20 microcontrollers, financial support, and their development of the marv M101 Data DECT module; Tasking BV for the rebate on our C167 development s Berlin for providing their board production facilities; and everyone else who has in work or is still doing so to keep MARVIN running – aeh, flying.

REFERENCES

[1] THE MILLENNIAL EVENT: Rules for the 1999 International Aerial Robotic qualifier. http://avdil.gtri.gatech.edu/AUVS/CurrentIARC/1999CollegiateRules.

[2] U. W. Brandenburg, M. Finke, D. Hanisch, M. Musial, and R. Stenzel. TUBI tonomously flying robot. In *Proc. AUVS Symposium*, pages 627–636, Washingto 1995.

[3] U. W. Brandenburg, M. Finke, and M. Musial. Aufbau und Steuerung des flieger
TUBROB. In *Tagungsband 11. Fachgespräch Autonome Mobile Systeme*, pa
Karlsruhe, 1995. Springer-Verlag. In German.

[4] U. W. Brandenburg, M. Musial, and G. Hommel. MARVIN – Technische Unive
never-depressed flying robot for the IARC'98. In *Proc. AUVS Symposium*, Was
USA, 1998.

[5] Marek Musial, Uwe Wolfgang Brandenburg, and Günter Hommel. Marvin -
fliegender erkundungsroboter. In *Autonome Mobile Systeme 1998, 14. Fachge*
226–233, Karlsruhe, 1998. Springer-Verlag.

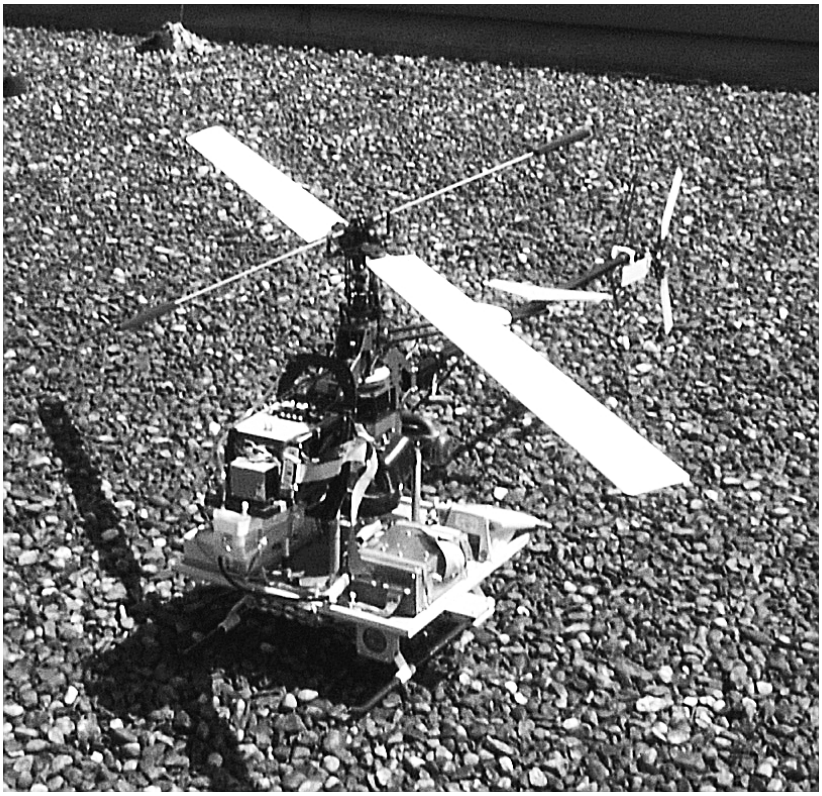


Figure 2: A view of MARVIN

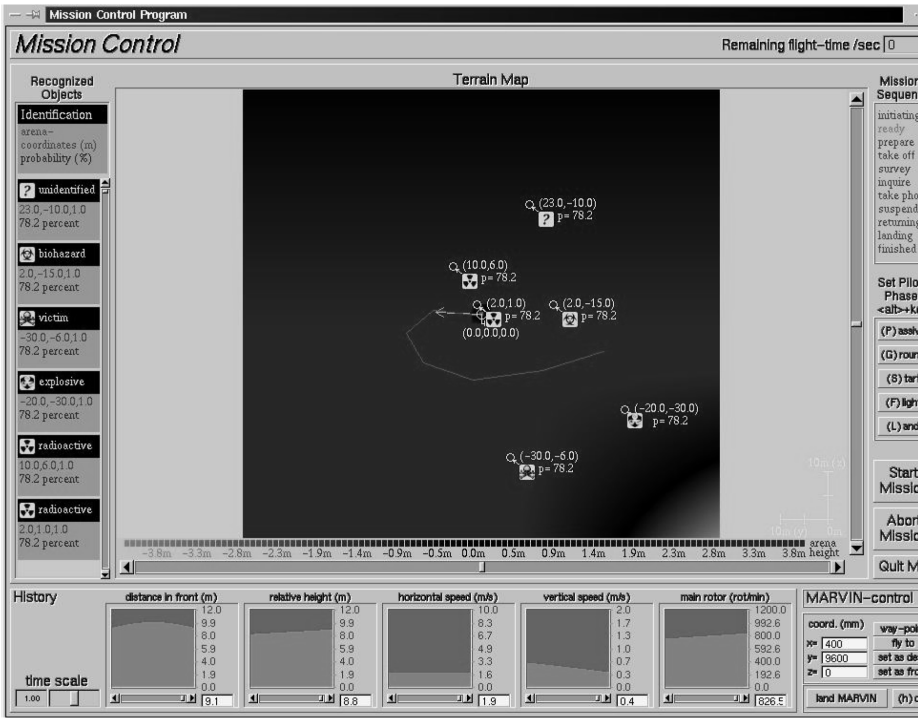


Figure 3: Screen-shot of the mission control software

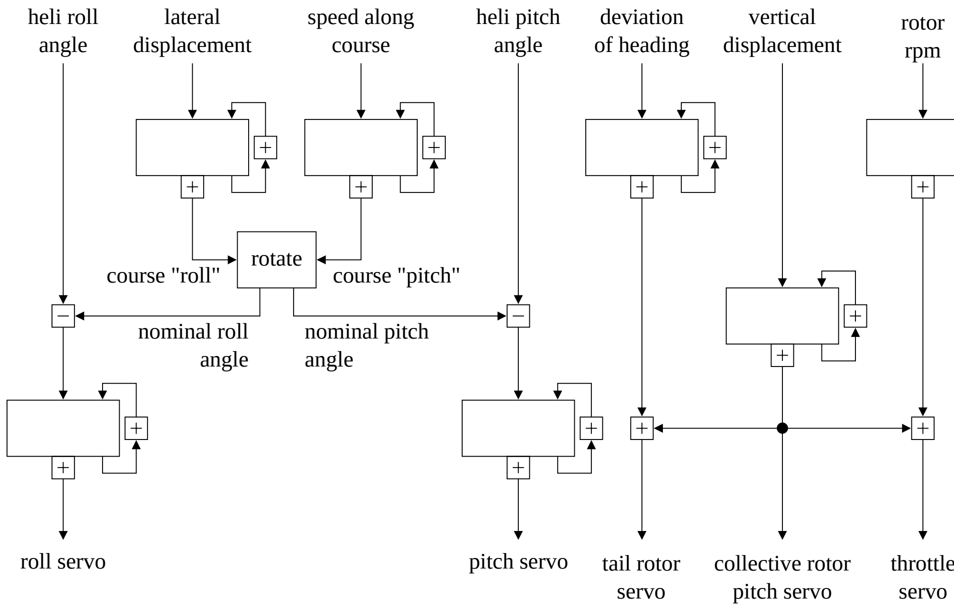


Figure 4: Architecture of the MARVIN flight controller

Figure 5: Recognized labels in an aerial picture