ROBOT AIRSHIP FOR EDUCATION AND RESEARCH - MODELLING AND CONTROL

Jörg Kuhle
Hubert Roth
Christoph Klein

University of Siegen, Institute of Automatic Control Engineering,
Hölderlinstr. 3, 57068 Siegen, Germany
Phone: 0049-271-740-4439
joerg.kuhle@uni-siegen.de
hubert.roth@uni-siegen.de
chriskl@freenet.de

Abstract: At the University of Siegen a robot airship for cooperation with a heterogeneous multi-robot team is built to perform search and rescue as well as exploration missions. The airship control is based on an aerodynamic model derived from real airship models. The airship is able to fly autonomously over the operation field and is sharing sensor information with the ground based robots and a tele-operator.

Keywords: Mobile Robots, Remote Control, Aerospace control, Telecontrol

1. INTRODUCTION

This paper describes the actual development of a non-rigid airship for indoor use. The airship is part of a heterogeneous multi-robot team at the University of Siegen. The robot team consists of ground based robots in a car style and the airship, which explores the environment from air. The airship and the robots are used in a laboratory environment as a platform for students' education and research.

Giving autonomy to an airship robot is strongly based on position and orientation determination and control. Since the airship has no contact to the ground, all sensors must work in the air. Therefore the sensor design is borrowed from plane technology realized in low-cost versions. It uses pressure sensors for altitude, a 3D-compass and an indoor absolute navigation system called V-Scope like GPS or beacons. To design the position controller in 3D-space, the airship was modelled first, based on potential theory and friction afflicted cross flow approach. The model leads to some different controller designs, which all were tested in simulation and partly on the real system. To enable the controller tests, a flexible test platform was designed, using the capabilities of the existing onboard hardware.

The airship is also part of the robot team, so it must communicate with different robots and especially with the human operator. For the use in student laboratories but also for rescue mission the robot should be easily accessible from any mission control computer. Therefore the robot provides a web-interface to get easy and secure control of the airship from remote computers.

2. THE COBRA AIRSHIP

After cooperating in different project with the airship company "Zeppelin NT" at Friedrichshafen, Germany, the University of Siegen designed a model airship robot to operate as flying platform. Flying robots give a third dimension to normal ground based robots. The idea of creating such a robot is to enhance the capabilities of a heterogeneous multi-robot team. The airship robot was created to observe the environment and detect possible targets, which may be handed over to the mobile ground based robots. An airship is relatively stable and energy saving in operation and may observe the ground in a better way than helicopters or planes. Thus the airship is called COBRA for "Cooperative Observation Robot Airship". The airship has a size of around 3.5m length and 1m diameter and is used indoor. The indoor airship has some simplification in comparison to the large outdoor ones. There are no ballonets inside to compensate pressure differences and the thrust vector control is restricted. For outdoor use a larger model may be created later, and control mechanisms of real airships might be applicable (Roth and Adermann 1994/1995). The airship itself was not part of the robot design; it was bought from an advertisement airship company.

2.1 Position control

One of the first steps when creating a mobile robot is to implement position control and its sensor equipment. Certainly measuring the position and orientation based on low-cost sensors in the air is quite complicated and inaccurate compared to measuring on ground. COBRA got a three-dimensional gyroscope and low-gravity accelerometers to detect any movement. The sensors are mounted on a special circuit called BeeCon Bluewand at the back of the gondola, underneath the centre of volume and send the data via Bluetooth connection to mission control. At this robot control server the data are integrated to gather position, speed and angle of the airship. This information is send back to COBRA via serial radio link and considered into local control algorithms in the onboard microcontroller. Unfortunately these sensors give bad results since the airships movement and acceleration is very low, furthermore the sensors tends to drift after a while, so the robot must get some more, reliably data. The sensor has to be calibrated, accordingly.

The problem of the BeeCon Bluewand sensor is the use of a relative measurement system, i.e. accelerometers and gyros. Therefore the airship is additionally equipped with a 3D-compass and inclinometer, combined with gyros. This sensor delivers absolute angles based on the magnetic field of the earth. The reliable resolution is about 1^o-2^o, which is far enough for the airship, since the position and direction of the floating ship should be just in an adequate area. The magnetic sensor is slightly delicate to magnetic influences, as metal pieces or fields generated by DC-motors. But the construction of the ship doesn’t change while running, and is hardly made from metal, so the compass can neglect this influence. Unfortunately the field produced by running motors of the blades has influence to the sensor. Also vibration and movement of the sensor give bad results, too. For this reason the 3D-compass is mounted as far as possible from the motors, but still fixed on the stable cabin and directly underneath the centre of volume, see fig. 1.

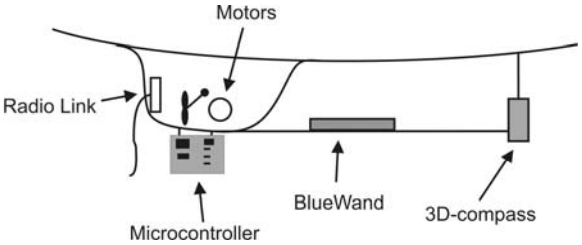

Fig. 1: The actuator, sensor and controller placement

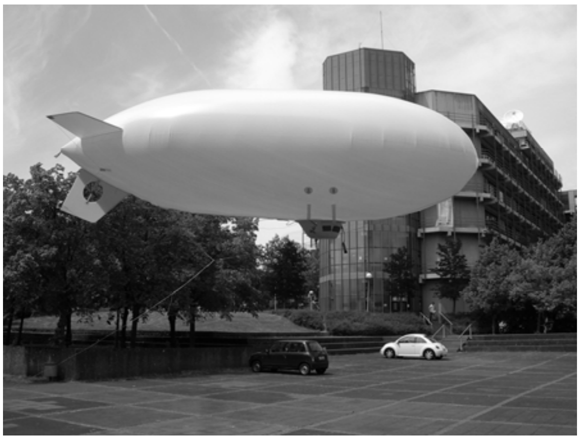

Fig. 2: The COBRA airship

The COBRA airship gets an absolute height measurement from a pressure sensor MPX4115A mounted on the ship. This information gives the robot its capability to control standstill for now. The resolution of the pressure sensor supports height differences of approx. 50cm.

When in use, the airship is flying for only some minutes, so no long term weather changes must be considered for the height control. But for later use a similar circuit is used as reverence at mission control. The airship should also be equipped with an ultrasonic or infrared sensor for relative height measurement for landing. The airship is shown in fig. 2 flying outside with no wind at the university site.

The airship is build up with low-cost model plane motors and has a handy size of 3.5m length, so the airship is only used indoor at present. The relatively large cross-section of the hull in comparison to the weak DC-motors with low force makes it difficult to manoeuvre reliable outside. Heavier equipment as needed outdoors is not applicable due to the payload of about 2.5kg overall. For indoor-use a GPS receiver for absolute position measurement may not be used, because the signal is blocked by walls. To really control the airship’s position an absolute position detection system is essential. An indoor absolute position system called V-Scope was used in tests. The V-Scope uses ultrasonic to detect small markers at the airship and gives all information to mission control. The airship is equipped with small markers, which can be detected by the V-Scope system, mounted on the ceiling. It uses time-of-flight of the sonar signals to detect the distance and infrared detectable colours on the markers to differ them. Unfortunately the range of the V-Scope system is limited to a light-cone of 2m radius and around 4m distance. So the airship can only use the system at start position or for landing operations. At present the use of another triangulation system, probably based on radio signals, is in research. In fact unreliable data may be the only information available in the field. The idea is to get an accelerometer based sensor for low acceleration that works fine for some travel and turns and can be updated by an absolute system time by time.

2.2 Operator control

To achieve the goal of a rescue or exploration mission, the airship must be also controllable by a human operator. At present the default mode is to display the airships position data and give a target position to the airship. Additionally the control can be done by a joystick, the same equipment as used for the mobile ground robots. The manual control overrides the autonomous control of the airship. In this mode COBRA will stabilize the position after release of the joystick control. As there is no obstacle avoidance or autonomous guiding system for the airship at present, the operator gets full control of the airship while taking over. The idea for this mode is to manoeuvre the ship quickly into its starting or target position. The Microsoft sidewinder joystick gives many ways to implement 3 axes control needed for the airship. Even if the control of a 3D flight might be tricky using a joystick, all robots should be able to use the same software and interfaces for human interaction. This allows an easy integration of new robots and ensures the usability for the operator. As the ground robot use a Java application for mapping the movement of the robots, the airship should also use the same or similar system. Thus the existing software will be enhanced for third dimension. To ensure the easy implementation of a uniform platform, all robots use the same onboard microcontroller C167 and a radio link based on RS-232 serial link. They also use similar communication protocols for sensor, joystick and control data (Kuhle, Roth and Ruangpayoongsak, 2004).

The network configuration of the system under normal operation is shown in fig. 3. The operator will work at the client site and launch the application of the web-server. The airship control server will send data from camera and sensors to the client. The communication to the airship's controller is done by serial radio link, which can be also replaced by a standard serial cable. Since the operator should be able to use a joystick as well as communicate to the airships control server, a Java applet would not be appropriate; security restrictions of applets will limit the functionality. A Java application will provide the functionality for the test-phase. But when using the airship control via internet from any place, the system should be as easy to use as an applet, should follow security aspects and give access to local hardware. These requirements are provided by Sun's Java Web Start technology. The user front-end is implemented as Java application, packed and signed by an authorised institution to ensure the reliability of the software. The software defines its requirements in a *.jnlp file, describing hardware needs, network communication, etc. The user is prompted to grant the access, getting full information about author, communication, signature and local access. The Web Start plug-in will download a local copy of the software, to speed-up the use in future, but will also guarantee the check for updates of the software, so no old version is running at the client. Furthermore the client will now directly communicate to the control application running on the robot server. The communication is not restricted to the web server. This configuration is shown in fig. 4.

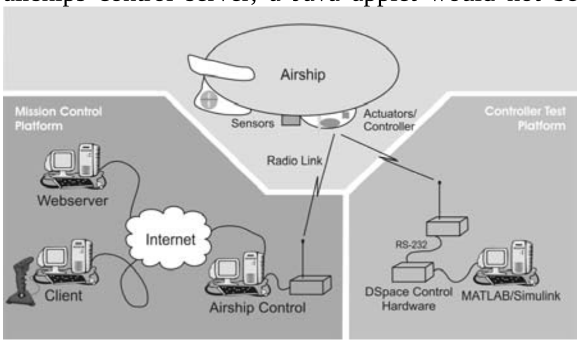

Fig. 3: COBRA network configuration

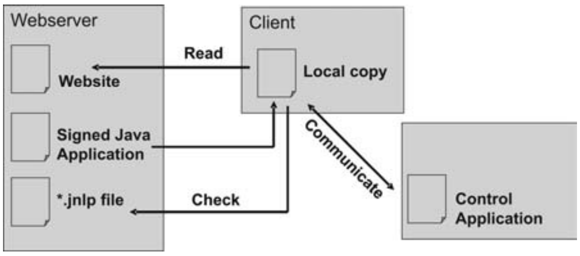

Fig. 4: The Web Start configuration

3. CONTROLLER DESIGN

The robot airship controller is based on a mathematical model of the aerodynamic kinematical and dynamical behaviour.

3.1 Modelling

To design the controllers of the COBRA airship first a mathematical model of the whole system is needed. The first part of the controller is the altitude control, based on the absolute pressure measurement by the pressure sensor MPX4115A. The model is based on force balance equations for vertical movement (Puangmali 2003):

$m\ddot{y} = 2F_p \sin\theta - F_{dy} + F_l - F_g$, with m as mass of the airship, F_p is thrust vector produced from a single motor, F_{dy} is vertical drag force, F_l is lifting force, F_g is gravitational force and θ is the thrust vector angle. The resulting system model, with u as input voltage to the motor is

$$\ddot{y} = \begin{cases} 2a_f \cdot u^{b_f} \cdot \frac{1}{m} \cdot \sin\theta - \frac{1}{2} \cdot \frac{C_d}{m} \rho_a \dot{y}^2 A_y + D_g, u > 0 \\ -2a_b \cdot |u|^{b_b} \cdot \frac{1}{m} \cdot \sin\theta - \frac{1}{2} \cdot \frac{C_d}{m} \rho_a \dot{y}^2 A_y + D_g, u < 0 \end{cases} \quad (1)$$

$D_g = \frac{F_l - F_g}{m}$ is called gravitational disturbance, A_y is the cross-sectional area of the airship body parallel with the horizontal plane, C_d is the drag coefficient or air resistance and ρ_a the density of air. All constants were obtained from experiments, partly flight experiments or motor tests, like $a_f = 0.0181, a_b = 0.0087$. For the vertical take of and landing operations the angle is $\theta = 90°$ and the equations can be simplified. Because of passive stability around roll axis and the centre of gravity is very deep the model works fine for vertical movement with fuzzy controller and even a combination between vertical and horizontal movement Thus a more enhanced model was designed, based on aerodynamics kinematics and dynamics aspects. Since the robot airship is used under special conditions, some restrictions could be made from the beginning. The system is considered with constant mass and volume. No effects as wind, superheat or pressure changes in and outside the bag due to the indoor environment are considered. Furthermore the fins and body of the airship are calculated separately, where the relation between is neglected (Klein 2004). For modelling the airship, the differential equations are based on a coordinate system with its origin at the centre of volume of the ship, not the centre of gravity, see fig. 5. This leads to the following basic equations:

$$\begin{aligned} m \cdot \left[\dot{\underline{V}} + \dot{\underline{\Omega}} \times \underline{r}_{CV \to CM} + \underline{\Omega} \times \left(\underline{V} + \underline{\Omega} \times \underline{r}_{CV \to CM}\right)\right] &= \sum \underline{F}_{ext} \\ \underline{I} \cdot \dot{\underline{\Omega}} + \underline{\Omega} \times \left(\underline{I} \cdot \underline{\Omega}\right) + m \cdot \underline{r}_{CV \to CM} \times \left(\dot{\underline{V}} + \underline{\Omega} \times \underline{V}\right) &= \sum \underline{Q}_{ext} \end{aligned} \quad (2)$$

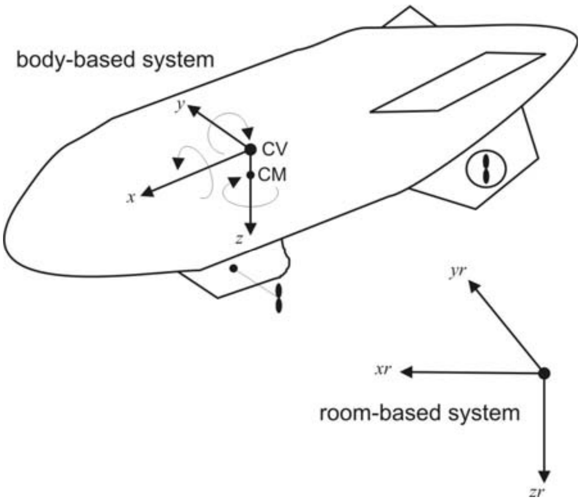

Fig. 5: The airship's coordinate system

The vectors V include the translational velocity, Ω the rotational velocity in the room-based fixed coordinate system and F, Q represent acting forces and torques. Since the airship's cross-section is symmetrically regarding the XY-plane, the inertia matrix $\underline{I}$ is symmetrically, too. The vector $\underline{r}_{CV \to CM}$ is pointing from the centre of volume to the centre of gravity.

For the motors the experiment results from the altitude control could be taken. For the buoyancy force the centre was considered to be coincident with the centre of volume, so no torque will be produced, since the airship is well balanced. The fins of the airship are only passive; they don't act as elevators or ailerons. Therefore the modelling for the fins is simplified. No drag force is considered and wind is neglected.

The model of the body itself is based on the potential-theory approach to determine the aerodynamic forces and moment as a result of frictionless air-flow as well as the friction afflicted cross flow to get friction forces and torques. The airships body is considered as a rotational ellipsoid.

$$\underline{\tau}_{AD(R_{rf})} = \begin{bmatrix} X_{\dot{u}}\dot{u} + Z_{\dot{w}}wq - Y_{\dot{v}}rv \\ Y_{\dot{v}}\dot{v} + X_{\dot{u}}ur - Z_{\dot{w}}wp \\ Z_{\dot{w}}\dot{w} + Y_{\dot{v}}vp - X_{\dot{u}}qu \\ 0 \\ M_{\dot{q}}\dot{q} - N_{\dot{r}}pr + M_{uw}uw \\ N_{\dot{r}}\dot{r} + M_{\dot{q}}qp + N_{uv}uv \end{bmatrix} \quad (3)$$

M_{uw} and N_{uv} are destabilising torques, u, v, w is velocity in x, y, z direction. p, q, r is angular velocity in roll, pitch, yaw angles.

When considering the air friction, the airship gets a stabilising moment by unsymmetrical stream, which effects opposite to the destabilizing torque.

$$\underline{\tau}_{AD(R_{rb})} = \begin{bmatrix} -\frac{1}{2}\rho_{air} \cdot c_{w_{rot}} \cdot u|u| \cdot A_x \\ -\frac{1}{2}\rho_{air} \cdot \left(C_1 \cdot v|v| + C_3 \cdot r|r|\right) \\ -\frac{1}{2}\rho_{air} \cdot \left(C_1 \cdot w|w| - C_3 \cdot q|q|\right) \\ 0 \\ -\frac{1}{2}\rho_{air} \cdot \left(C_4 \cdot q|q| - C_2 \cdot w|w|\right) \\ -\frac{1}{2}\rho_{air} \cdot \left(C_4 \cdot r|r| + C_2 \cdot v|v|\right) \end{bmatrix} \quad (4)$$

Using the vector $\underline{X} = [u, v, w, p, q, r]^T$ including the translational and rotational velocities of the body-based system, the model computes to

$\underline{M}\cdot\dot{\underline{X}} = \underline{\tau}_m + \underline{\tau}_P + \underline{\tau}_{AS} + \underline{\tau}_{AD(F)} + \underline{\tau}_{AD(R_{rb})}$. The parts of the vector $\underline{\tau}_{AD(R_{rf})}$ from the frictionless approach are bundled into the mass-matrix $\underline{M}$. $\underline{\tau}_m$ contains centrifugal and coriolis-forces and torques, $\underline{\tau}_{AS}$ aerostatic forces and torques, $\underline{\tau}_P$ is the propulsion vector and $\underline{\tau}_{AD(F)}$ consists of the torques and forces of the fins.

3.2 Controller design

Since the system is not completely controllable, the first attempt is a switching controller, based on the linearized model of the airship. Because of the coupling of all equations and important missing parts in the linearized model it is not possible to design a linear controller for the system at once. A mission control strategy is necessary, thus a switching controller first turns the airship into the correct direction, by PD controller. Afterwards the airship moves forward to the desired target position. Finally the airship moves up to the specific height. This easy approach lacks in speed and a general stability analysis, since the coupling of the system will cause side-effects on the single control steps. Thus the use of decoupling controllers is sensible. To simplify the controller design the following assumptions are made: roll and pitch angles and velocities are zero (the model airship is almost still in these directions), acceleration is neglected, buoyancy equals gravitation (the airship is balanced) and the forces and torques of the fins are neglected. The controller is actually designed for movement in the XY-plane, i.e. horizontal movement. Therefore the state-vector is $\underline{x} = [u, v, r, X, Y, \varphi]^T$, now containing the position, orientation and the derivations. In this two-dimensional situation the equations simplify to $\dot{X} = u\cdot\cos\varphi - v\cdot\sin\varphi$, $\dot{Y} = u\cdot\sin\varphi + v\cdot\cos\varphi$ and $\dot{y} = r$. The control vector calculates to

$$\underline{R}(\underline{x}) = \begin{bmatrix} \frac{1}{2}m_y\cos\varphi & \frac{1}{2}m_x\sin\varphi \\ m_y\sin\varphi & -m_y\cos\varphi \end{bmatrix} \cdot\left(\underline{c}^*(\underline{x}) + \begin{bmatrix} q_{10}X + q_{11}(u\cdot\cos\varphi - v\cdot\sin\varphi) \\ q_{20}Y + q_{21}(u\cdot\sin\varphi + v\cdot\cos\varphi)\end{bmatrix}\right) \tag{5}$$

After a decoupled controller was designed by pole setting to get $q_{10}\dots q_{21}$, the complete controller design was implemented and tested in Matlab/Simulink, see also the XY-plot of a model test flight in fig. 6.

This concept is just in the realization phase. Until now different control approaches are just simulated, the altitude and angular control is already implemented. The height is measured by the pressure sensor MPX4115A and gives resolutions up to 0.5m, the direction is measured by the 3D-compass and inclinometer 3DM, which gives resolutions up to 1 degree.

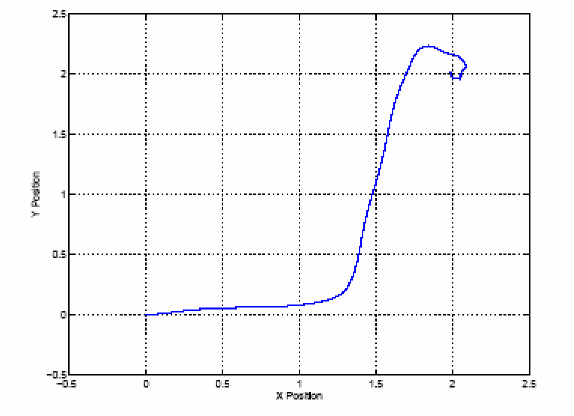

Fig. 6: XY-Plot of the flight control

3.3 Controller Test platform

When operating as an autonomous robot vehicle the airships controller must be implemented inside the onboard controller. The usual communication to the user is done by a Java application. Nevertheless for designing the controller, a test-platform for rapid design and implementation of different approaches is necessary. But the test-platform should use the implemented components onboard. To communicate with mission control the COBRA already needs a radio link from the C167 microcontroller to a ground computer. This link is a RS232-compatible serial communication, so a cable can be used as well as the radio module later. The airship system is relatively slow and the sensor as well as control data don't cause problem due to the transmission rate. With the radio link the system can be used on the test-platform as well as in the multi-robot environment, a big advantage regarding to a default analogue connection to the dSPACE hardware. As test platform now a PC running MATLAB/Simulink under Windows 2000 working on a dSPACE DS1104 PCI-Card is used.

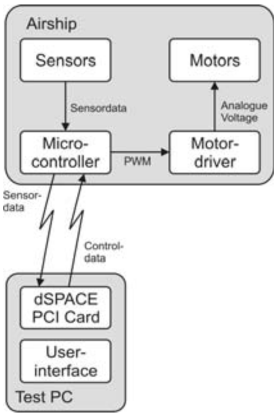

Fig. 7: The test-platform, consisting of the airship and a PC with dSPACE DAQ hardware

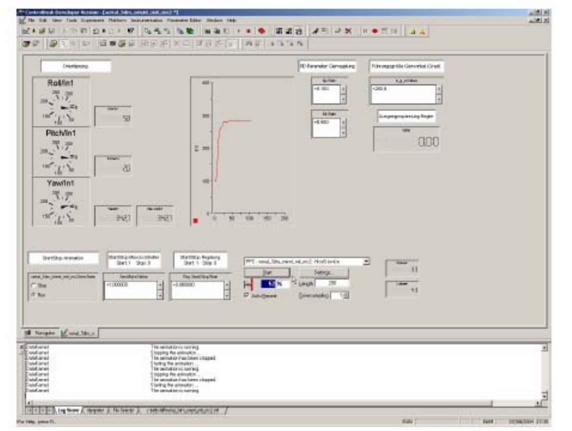

Fig. 8: Test-platform control panel (dSPACE)

All control algorithms can be implemented in MATLAB/Simulink, the dSPACE card sends the control data via serial radio link to the microcontroller. The loop is closed by the measured data sent back from the controller to the PC via radio link, too, see fig. 7. Actually the airship is equipped with the altitude sensor as well as a 3D-compass and inclinometer for absolute angle measurement. Position detection was formerly done with accelerometers, but seems to be too inaccurate for good performance of the controller. At present the use of indoor absolute navigation systems is discussed.

The MATLAB/Simulink model is connected to the dSPACE Controldesk front-end to visualize the incoming data and control data, shown in fig. 8. At present the Controldesk software displays roll-, pitch- and yaw-angle as well as the altitude of the airship. Even when the installed 3D-compass is mounted as far as possible from the motors, there is some influence of the magnetic field to the sensor. Thus a discrete low-pass filter was implemented to eliminate disturbances.

5. CONCLUSION

The model based controller design gives a good result for the airship robot control; especially the simulation opens wide possibilities to test and simulate different approaches. Nevertheless the transfer to the real system results in good behaviour, at present in altitude and direction control. The step response of a simulated and real system is shown in fig. 9.

The quality of control not only depends on the model but a lot on sensors, so a good position sensor is the next step for the autonomous airship. The test platform, using existing onboard hardware without changes gives the possibility to design controllers, modelled in MATLAB/Simulink, without touching the running configuration.

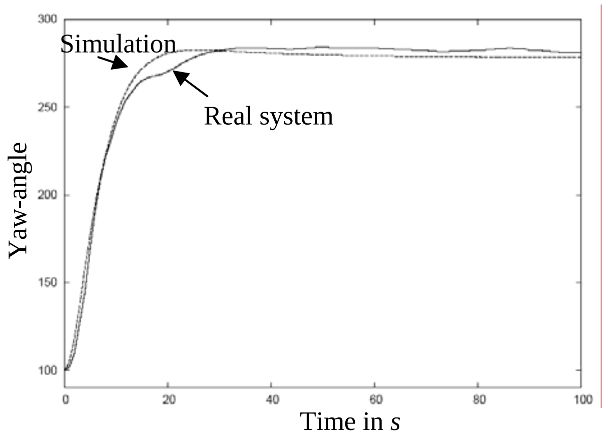

Fig. 9: Simulation and real system comparison of yaw-angle control

The airship can be used in the mutli-robot environment with operator control as well as in the test-platform. After the controller design has finished, means in particular the implementation of an absolute navigation system, the optimal controller code will be implemented on the microcontroller.

6. REFERENCES

Klein, Ch. (2004). *Modellierung und Lageregelung eines Luftschiffes (Diploma Thesis),* Siegen, Germany, August 2004

Kuhle, J.; Roth, H.; Ruangpayoongsak, N. (2004). *Mobile robots and airship in a multi-robot team,* The 1st IFAC Symposium on Telematics Applications in Automation and Robotics, Helsinki, Finland, June 2004

Puangmali, P. (2003). *Mechatronic Design of an Altitude Control Module for Unmanned Airship Automatic Flight Control (Master Thesis),* Siegen, Germany, November 2003

Roth, H.; Adermann, H.-J. (1994). *Flight Control Concept of the Zeppelin NT Airship*, Proc. “ICSE ‘94”, 10th International Conference on Systems Engineering, Coventry, UK, September 1994

Roth, H.; Adermann, H.-J. (1995). *Airship Control using Adaptive Fuzzy Logic Strategies*, Proceedings of the 6th International Fuzzy Systems Association World Congress, Sao Paulo, Brasilia, p. 683-686, 1995

Roth, H.; Schwarte, R.; Ruangpayoongsak, N.; Kuhle, J.; Albrecht, M.; Grothof, M.; Heß, H. (2003). *3D Vision Based on PMD-Technology for Mobile Robots,* Aerosense - Technologies and Systems for Defense & Security 2003, SPIE Conference, Orlando, Florida, April 2003