Proceedings of the 13th IASTED International Conference
ROBOTICS AND APPLICATIONS
August 29–31, 2007, Würzburg, Germany
ISBN Hardcopy: 978-0-88986-685-0 / CD: 978-0-88986-686-7

DEVELOPMENT OF AN INDOOR BLIMP ROBOT WITH INTERNET-BASED TELEOPERATION CAPABILITY

Yasuhiro Ohata Satoshi Ushijima Dragomir N. Nenchev
Department of Mechanical Systems Engineering
Musashi Institute of Technology
1-28-1 Tamazutsumi, Setagaya Ward, Tokyo, 158-8557, Japan

ABSTRACT

We are developing an indoor blimp-type robot to be used at sports and other events. The user gets access to the robot via a mobile phone or a desktop PC (the client), using Internet connectivity. The user then steers the robot to a desired location and takes pictures with the on-board camera.

This paper describes the system architecture, with hardware and software components. The so-called Predictive Motion Display (PMD) technique [1] is used to support the user during teleoperation. The operation mode is a shared control one: the user has full control only within a prespecified area. In this way, undesirable collisions of the blimp can be avoided. Partial autonomous feedback control is envisioned to stabilize yaw and altitude. The paper introduces also a dynamical simulator, which we found helpful in testing the PMD.

KEY WORDS

Indoor blimp robot, Internet-based teleoperation, Entertainment robot, Predictive motion display

1 Introduction

Recently, research on airships or lighter-than-air vehicles, called also blimps, is expanding rapidly. These vehicles can be divided into two main classes: for outdoor use and for indoor use. Main fields of application for outdoor airships include information gathering during rescue operations in disaster areas [2], land mine detection [3], observation of effects of global warming [4], [5], [6]. The main challenge for this type of airships is control in the presence of disturbances, such as air currents.

On the other hand, the main challenge for indoor-use blimps is their size and weight lifting capability. At present, miniaturized components are readily available on the market, so that feasible designs are emerging rapidly. Main areas of application of indoor blimps are observation and entertainment [7], [8]. It should be noted that, when compared to other indoor flying vehicles, such as small aircraft [9] or small helicopters [10], indoor blimps have several advantages; they are energy efficient, can easily hover for a relatively long period of time, and they are safe to operate – even if a blimp collides with a wall, no damages will occur. The main mode of control of indoor blimps is visual servoing [11], [12].

In this paper, we describe the design of an indoor blimp for entertainment use during various events. Users get access and teleoperate the blimp via the Internet, through their mobile device or through a local PC. The blimp is equipped with an on-board camera for taking pictures. Besides system design, we focus here on means of supporting inexperienced users, such as children, during teleoperation. This is an important point, because blimps are operated by propeller thrust, and hence, inexperienced users find it difficult to stop the blimp at a desired location. We implement a technique called "Predictive Motion Display" or PMD, for short [1].

The paper is organized as follows. In Section 2, we describe the system architecture. Section 3 introduces the PMD-based technique for teleoperation and the dynamical simulator. Sections 4 and 5 report experimental results from simulations and real experiments, respectively. Finally, Section 6 gives the conclusions.

2 System Architecture

2.1 System components

The system architecture comprises three main components: client site, server site and robot site (see Fig.1). The user operates the blimp from the client site either via a mobile phone or via a desktop/laptop PC. Appropriate graphical user interfaces (GUI's) using Java applets are under development. The client site sends/receives control/status information from/to the server site via the Internet.

The server cite processes the command information to be send via the parallel port to a MBH8 (Renesas) microcontroller. The microcontroller then prepares a serial signal which is sent through a small RF transmitter (R. F. Solutions Ltd) to the robot site. The server site includes also a base camera with vision processing board for visual feedback control, and a receiver/vision board for the on-board wireless camera.

The robot site contains the RF receiver, an MBH8 microcontroller for signal processing, a small motor controller for the three propeller motors, the wireless camera, and two batteries. In future, we plan to add a gyro sensor for onboard yaw angle feedback control, and a camera pointing mechanism with two small RC servomotors.

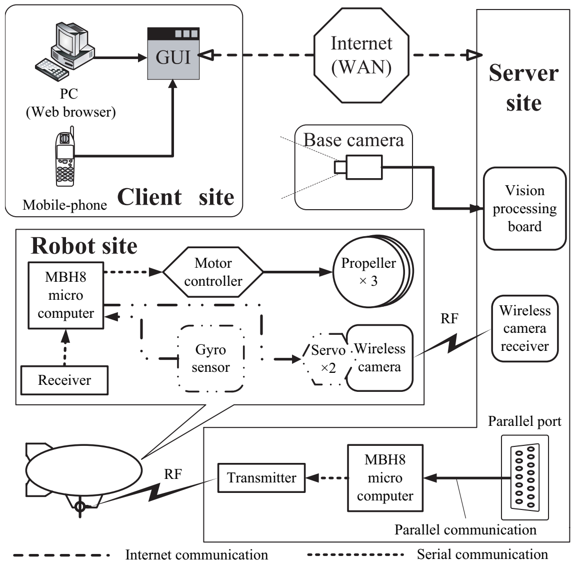

Figure 1. Blimp robot system.

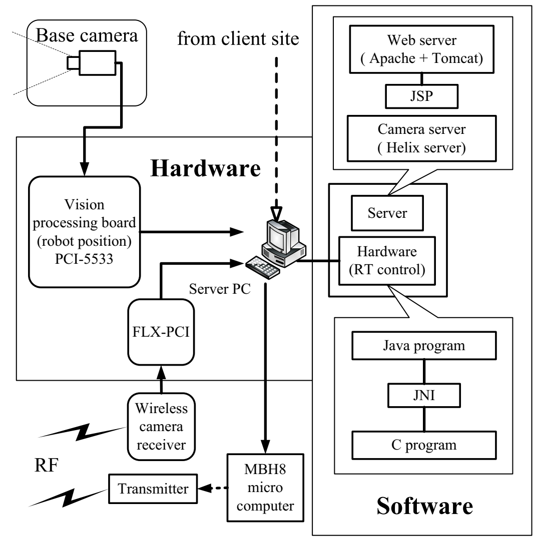

Figure 2. Server site details.

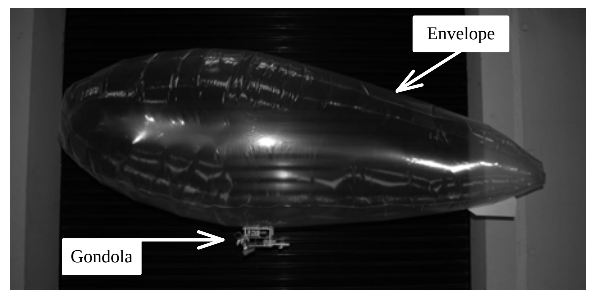

Figure 3. Blimp robot.

2.2 Server site details

First, we focus on the software components (see Fig.2). The operating system is Linux, the web server is Apache, in combination with a Tomcat server to enable the use of Java Server Pages (JSP) technology[13]. We use JSP in order to avoid loading Java applets at the client site.

Further on, the image from the onboard wireless camera is received by the server site, then streamed to the client site via the Helix streaming server (RealNetworks). The base camera is used to track a marker on the blimp robot. The vision processing board calculates the area and the center of mass of the marker, and based on this information, the position/orientation and the velocity of the blimp robot are estimated. These data are then used for feedback control.

Finally, we note that information processing at the server site is done using both C-programming and Java programming. Java programs communicate with C programs via the JNI interface [14].

2.3 Onboard hardware components

The current blimp robot design parameters are:

- dimensions: 1700 600 600 mm
- total weight (envelope included): 300 g
- number of propellers: three
- maximum speed: 2 m/s

All hardware components are located at the gondola (see Fig. 4). Two of the propellers are used for forward/backward thrust in the x direction and for rotation (yaw angle θ). The third propeller is used for altitude control. It is clear that this blimp robot is a nonholonomic vehicle.

The hardware components are summarized in Table 1. The total weight is 178.1 g, which means that there is some space to include additional parts such as the gyro sensor and the camera pointing mechanism. With the batteries selected, we can operate the robot for up to 25 minutes.

3 PMD-Based Teleoperation

3.1 Predictive Motion Display (PMD)

It is well-known that a blimp robot is difficult to control, mainly because of the actuation by thrusts from the propellers. Note that the operator may not be able to stop the blimp at the desired location, by thrust command input. Hence, the operator needs to be supported by the control system. We will adopt here a technique called “predictive motion display” or PMD for short, which was developed in relation to a free-flying space robot [1]. The idea is quite simple: the operator focuses on the predicted posi-

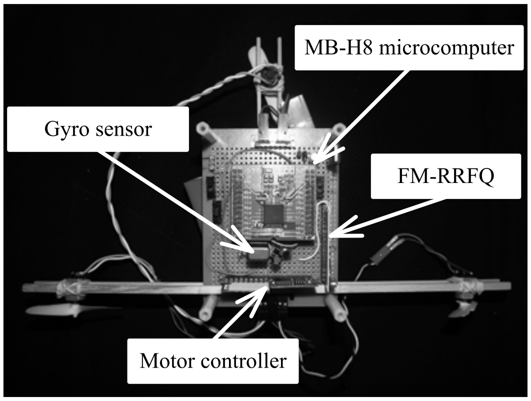

Figure 4. Onboard components.

Table 1. Hardware components

Parts	Model number	Weight [g]
RF receiver	FM-RRFQ	3 g
Micro computer	MB-H8(H8/3694F)	7 g
Motor controller	PololuSMC02B	2×1 g
Control battery	PQ-300XP-2S	23 g
Motor battery	Li-355SP 1S	23 g
Structure	Balsa, Perfboard	78 g
Propeller	made-up article	3×0.5 g
Propellent motor	TN-1333	$2\times$ 3 g
Pitch motor	MABUCHIFFN20PN	5 g
Servo motor	BA-TS-4.3	2×4.3 g
Fin	Polystyrene paper	1 g
Wireless Camera	RC-12	20 g
Total weight		178.1 g

tion/orientation of the blimp, instead on the current one. In this way, we do not need to cope directly with position control via propeller thrust.

Let the predicted position/orientation be denoted as $\boldsymbol{p}_{pre}$, and the current one – as $\boldsymbol{p}_{cur}$. Then, we use the following equation:

$$\boldsymbol{p}_{pre} = \boldsymbol{p}_{cur} + \boldsymbol{v}_{cur} t_{pre}, \tag{1}$$

where $\boldsymbol{v}_{cur}$ is the current velocity and t_{pre} is the time span of prediction. The operator observes a planar map of the environment together with the predicted and the current position/orientation (see Fig. 5).

3.2 Dynamical simulator

In order to confirm the validity of the proposed approach, we designed a dynamical simulator, using Open Dynamics Engine (ODE) [15]. The software model is composed of five bodies, as follows:

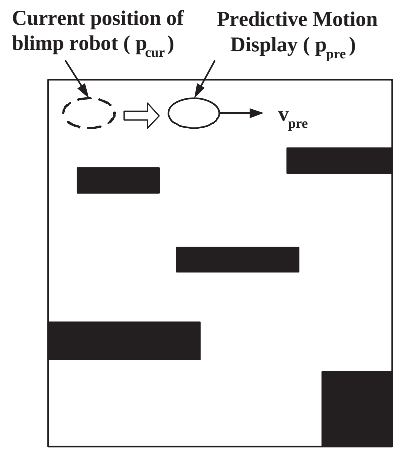

Figure 5. PMD map.

- Body 1 : envelope
- Body 2 : gondola
- Body 3 : right propeller
- Body 4 : left propeller
- Body 5 : propeller shaft

The model is depicted in Fig. 6. Note that we did not include the altitude propeller, because altitude will be controlled autonomously, and hence is not related to the PMD.

To determine the thrust force, we made a dynamical model, as shown in Fig. 7. The equation of motion of the blimp is [16], [17]:

$$\boldsymbol{M}\dot{\boldsymbol{x}} + \boldsymbol{C}(\boldsymbol{x}) + \boldsymbol{G} = \boldsymbol{D}(\boldsymbol{x}) + \boldsymbol{W}, \tag{2}$$

where $\boldsymbol{M}$ denotes the inertia matrix, $\boldsymbol{x} = \begin{bmatrix}\boldsymbol{v}^T & \boldsymbol{\omega}^T\end{bmatrix}^T$ is the spatial velocity vector, $\boldsymbol{C}(\boldsymbol{x})$ stands for nonlinear (Coriolis and centrifugal) force, $\boldsymbol{D}(\boldsymbol{x})$ denotes air drag, $\boldsymbol{G}$ is the gravity term and $\boldsymbol{W} = \begin{bmatrix}\boldsymbol{F}^T & \boldsymbol{T}^T\end{bmatrix}^T$ is the force/moment (wrench) vector acting at the center of mass G.

Henceforth, we will ignore the gravity term, as it is unrelated to the (planar) PMD. Also, we will ignore the air drag term under the assumption that the blimp moves with low speed. The last equation can be expanded as:

$$\begin{bmatrix} m\dot{\boldsymbol{v}} \\ \boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} \end{bmatrix} = \boldsymbol{W} = \hat{\boldsymbol{A}}\boldsymbol{F}_c, \tag{3}$$

where

$$\boldsymbol{A} = \begin{bmatrix} -\boldsymbol{u}_1 & -\boldsymbol{u}_2 \\ -\boldsymbol{r}_1 \times \boldsymbol{u}_1 & -\boldsymbol{r}_2 \times \boldsymbol{u}_2 \end{bmatrix}$$

and $\hat{\boldsymbol{A}}$ is $\boldsymbol{A}$ mapped into inertial coordinates.

The two components of $\boldsymbol{F}_c = \begin{bmatrix} f_1 & f_2 \end{bmatrix}^T$ denote thrust forces for the two propellers. m stands for the total mass, $\boldsymbol{I}$ is the inertia tensor. Vectors $\boldsymbol{u}_i$, $i = 1, 2$ are

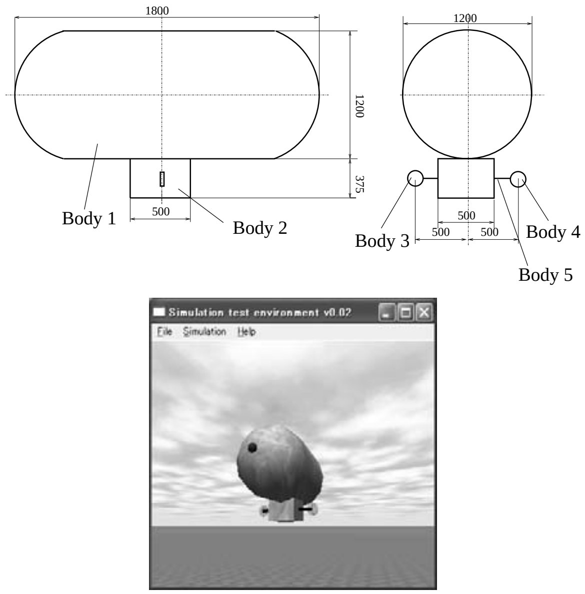

Figure 6. ODE model of the blimp.

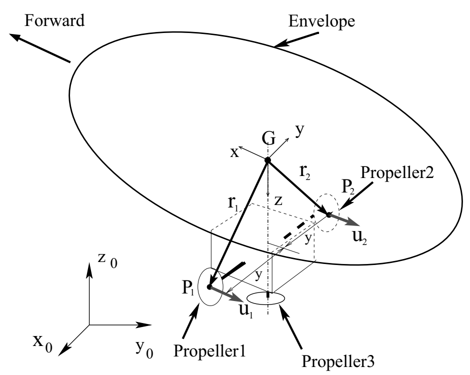

Figure 7. Dynamical model.

unit vectors in the direction of the thrust force. In body coordinates, these vectors have null y and z components. Vectors $\boldsymbol{r}_i$, $i = 1, 2$ are position vectors for the propellers, with respect to the body frame (cf. Fig. 7).

Further on, we can simplify the equation of motion, since we will consider just forward/backward propulsion and yaw rotation, that is, components related to speeds v_x and ω_z, respectively. Hence, rewrite equation (3) as:

$$\boldsymbol{A}\boldsymbol{F}_c = \tilde{\boldsymbol{W}} \equiv \begin{bmatrix} m\dot{v}_x \\ I_{zz}\dot{\omega}_z + (\boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega})_z \end{bmatrix}, \tag{4}$$

where

$$\boldsymbol{A} = \begin{bmatrix} u_{1x} & u_{2x} \\ (\boldsymbol{r}_1 \times \boldsymbol{u}_1)_z & (\boldsymbol{r}_2 \times \boldsymbol{u}_2)_z \end{bmatrix} \in \Re^{2\times 2}. \tag{5}$$

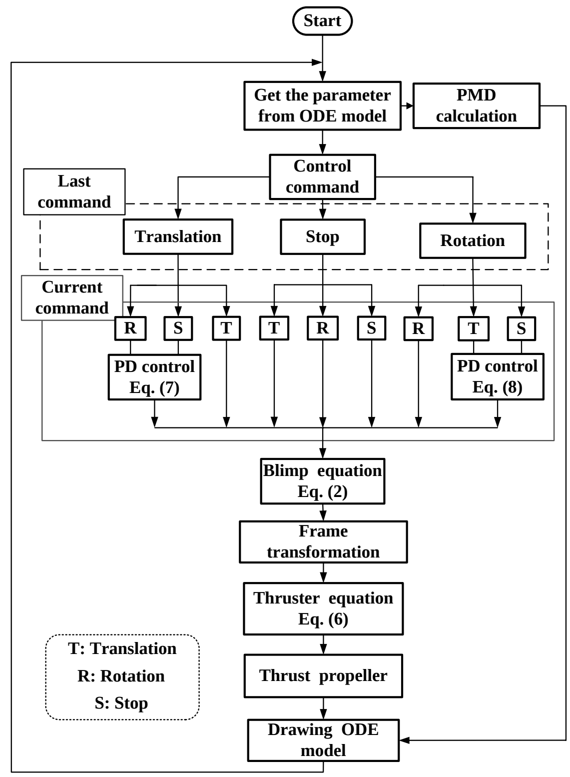

Figure 8. ODE flow chart.

From the last equation, we calculate finally the propeller thrusts as

$$\boldsymbol{F}_c = \boldsymbol{A}^{-1}\tilde{\boldsymbol{W}}. \tag{6}$$

3.3 PD controller

To ensure the blimp model's tracking of the predicted motion from the PMD, we need a feedback controller. Hence, we calculate the input force along f_x and the yaw torque t_θ from the following PD feedback control equations:

$$f_x(t) = K_{px}e_x(t) + K_{dx}\frac{de_x(t)}{dt} \tag{7}$$

$$t_\theta(t) = K_{p\theta}e_\theta(t) + K_{d\theta}\frac{de_\theta(t)}{dt}, \tag{8}$$

where K_{px} and $K_{p\theta}$ are the proportional feedback gains, and K_{dx} and $K_{d\theta}$ are derivative ones. $e_x(t)$C$e_\theta(t)$ denote errors with respect to translation and rotation, respectively. The force and the yaw torque are transformed to propeller thrust force via (6).

4 ODE Simulation Experiments

The ODE model parameters are given below:

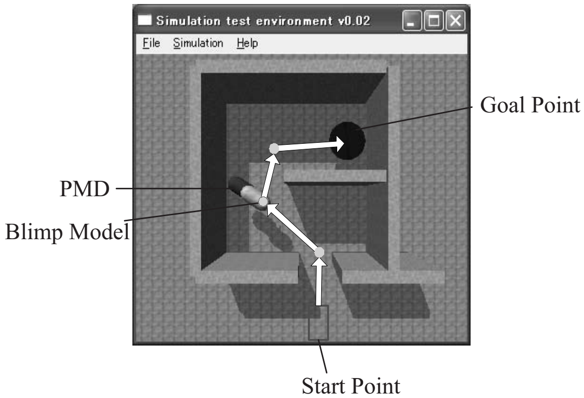

Figure 9. PMD simulation.

Figure 10. Blimp experiments.

Envelope volume	1800 1200 1200 mm^3
Gondola volume	500 500 375 mm^3
Blimp weight	300 g

The initial speeds v_x and $\dot{\theta}$ are set to zero. The position of the two propellers are: $\boldsymbol{r}_1 = \begin{bmatrix} 0 & 0.5 & 0.376 \end{bmatrix}$ m and $\boldsymbol{r}_2 = \begin{bmatrix} 0 & -0.5 & 0.376 \end{bmatrix}$ m, respectively. Hence, the matrix:

$$\boldsymbol{A} = \begin{bmatrix} -1 & -1 \\ 0.5 & -0.5 \end{bmatrix}. \tag{9}$$

The PMD time constant t_{pre} was set to 1 s [1]. The current position/orientation as well as the respective speeds are obtained from the ODE model via the "dBodyGet" set of commands [15]. The desired position/orientation as well as the respective speeds are obtained from the PMD equation. The operator uses keyboard input for the three commands: translation, rotation and stop. The flowchart of the ODE simulator is shown in Fig. 8. The graphical display is shown in Fig. 9. The graphs of the PMD model and current variables are shown in Fig. 11. It is seen that the ODE dynamic model tracks successfully the PMD model, and the user is able to stop at the desired position.

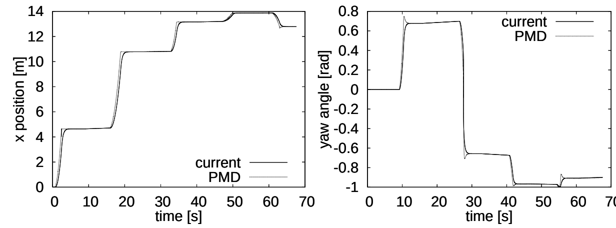

Figure 11. Current and PMD model variables.

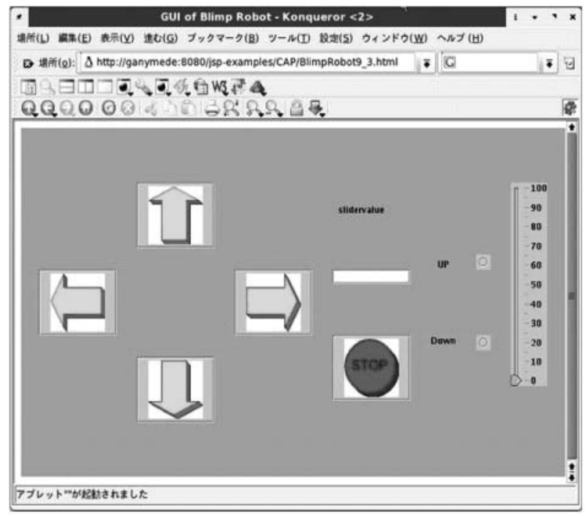

Figure 12. Java applet GUI outlay.

5 Blimp Experiments

We made experiments with the blimp during the University Open Door days (see Photo 10).

At that time, the PMD + feedback controller was not yet implemented. The user teleoperated the blimp directly, via the Internet. For this purpose, a simple GUI Java applet was created (see Fig. 12). The GUI has four buttons for forward/backward and left turn/right turn command inputs, a stop button, and a slider for speed adjustment. The experiments confirmed that an operator support is necessary, indeed.

6 Conclusion

We have designed a blimp robot for indoor use, which can be teleoperated via the Internet, by using a mobile phone or a PC. It was confirmed that such a robot is difficult to control directly. To alleviate the problem, we proposed means for operator support, called the Predictive Motion Display. Satisfactory performance has been confirmed with the help of a dynamical simulator, described also in the paper. In future, we plan to implement the PMD with the real robot.

References

[1] Y. Tsumaki and M. Yokohama, Predictive motion display for acceleration based teleoperation, Int. Conf. on Robotics and Automation, ICRA 2006, pp. 2927–2932

[2] Y. Hada et al., Rescue Communicators for Global Victim Search and Local Rescue PlanningCIEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2006, pp. 3510–3513

[3] K. Fujita, K. Onaka and N. Goto, A New Approach to Airborne Mine Detection Scheme Using a Magnetic Gradiometer, Memoirs of the Faculty of Engineering, Kyushu University 66(3), pp. 197–215

[4] K. Hosoi and M. SugimotoCAn Autonomous Security System Using Multi-Flying RobotsC The 18th Annual Conference of the Japanese Society for Artifical Intelligence, 2004, 2CI–04 (in Japanese)

[5] H. Kawano, Three–Dimensional Obstacle Avoidance of Autonomous Blimp Flying in Unknown Disturbance, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS 2006, pp. 123–130

[6] S. B. V. Gomes and J. Jr. G. Ramos, Airship Dynamic Modeling for Autonomous Operation, IEEE International Conference on Robotics and Automation 1998, pp. 3462–3467

[7] M. Onda et al., Aerobatic Airship Robot with Advanced Propulsors, Robotics and Automation, TExCRA ’04. First IEEE Technical Exhibition Based Conference, 2004, pp. 91–92

[8] H. Kawamura et al., Development of an Entertainment Indoor Blimp Robot Based on Hovering ControlCJapan Society for Fuzzy Theory and Intelligent Informatics Journal, vol. 7 2005Cpp. 203–211 (in Japanese)

[9] J. Zufferey et al., A 10-gram Microflyer for Vision-Based Indoor Navigation, IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS2006, pp. 3267–3272

[10] W. Wang, et al., Autonomous Control for Micro-Flying Robot and Small Wireless Helicopter X.R.B Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference, 2006, pp. 2906–2911

[11] H. Zhang and J. P. Ostrowski, Visual Servoing with Dynamics: Control of an Unmanned Blimp, IEEE Int. Conf. on Robotics and Automation, 1999, pp. 618–623

[12] T. Fukao, K. Fujitani and T. Kanade, Image-Based Tracking Control of a Blimp, In Proc. 42nd IEEE Conference on Decision and Control, pp. 5414–5419

[13] Java.sun.com,http://java.sun.com/developer/technicalArticles/javaserverpages/

[14] R. GordonCEssential JNI: Java Native Interface (Essential Java)CPrentice Hall Ptr.C1998

[15] http://ode.org

[16] A. AzumaCAeronautical Engineering (2)CSyoukabouC1989, (in Japanese)

[17] S. V. D. Zwaan, A. Bernardino and J. Santos-Victor, Vision based Station Keeping and Docking for an Aerial Blimp, IEEE/RSJ Int. Conf. Intelligent Robots and Systems, IROS 2000, pp. 614–619