Q-Learning for Navigation Control of an Autonomous Blimp

Yiwei Liu, Zengxi Pan, David Stirling, and Fazel Naghdy
University of Wollongong,Wollongong, NSW Australia
fyl359, zengxi, stirling, fazel@uow.edu.au

Abstract

In this research, an autonomous control system for blimp navigation was developed using reinforcement learning algorithm. The aim of this research is to provide a blimp the capability to approach a goal position autonomously in an environment, where the dynamical models of the blimp and the environment are unknown. Webots™ Robotics Simulator (WRS) was used to simulate and evaluate the control strategy obtained through a one-step Q-learning method. The simulation data generated via WRS were then processed and analysed within MATLAB. The simulation results showed that the control policy acquired from Q-learning is much more effective compared to the traditional control methods.

1 Introduction

In recent years, research and development on autonomous blimps has shown a significant growth. New applications have been found in areas such as freight carriers, advertising, atmospheric monitoring, surveillance, transportation, military and scientific research, etc. Furthermore, the military has shown a special interest in adopting autonomous blimps for reconnaissance and surveillance missions. The development of an intelligent navigation control strategy is the core research issue for these applications.

The blimp studied in this research was a test platform for the "2007 UAV Outback Challenge" organized by Australian Research Centre for Aerospace Automation (ARCAA). For this challenge, an intelligent control system needed to be developed to autonomously navigate the blimp to the target position. A reinforcement learning algorithm was proposed as the core for this blimp control system so as to acquire information from its environment and learn the best control policy through iterative learning. The developed control strategy neither requires a model of the blimp nor its environment, which presents a huge advantage over the traditional model-based control strategies in the same context where such models are difficult to acquire or are significantly time variant.

This paper is organized as follows. Section 2 presents the related work on blimp control, and Section 3 introduces the basic concept of the reinforcement learning. The Q-learning algorithm used in blimp control is described in Section 4, and the simulation results of the blimp continuous navigation tasks using Q-learning is presented and analysed in Section 5. Section 6 provides the conclusion and future work of this research.

2 Related Work

The control methods implemented on a many autonomous blimps have significantly improved in the last decade. At the end of last century, blimp control was mainly based on manual operations, such as direct pilot control or manual radio control. Brett and his research group used a radio-controlled blimp as a platform for microwave remote sensing in 2000 [Walkenhorst *et al*., 2000]. The semi-autonomous blimp was developed as a low cost alternative for a radar system, which was used for archaeological and geological studies typically gathering information from aircraft or satellites. At the same time, Brandreth regarded the remote controlled blimp as an ideal platform for remote sensing in the maritime environment [Brandreth, 2000].

Imaged-based control is an important technique which makes it possible to change the control patterns from manual or semi-manual control to autonomous control. The earliest work on imaged based blimp control was presented by Zhang and Ostrowski in 1999 [Hong Zhang and Ostrowski, 1999]. Recently, other successful applications of image-based control on blimps were reported [Silveria *et al*., 2002] [Fukao *et al*., 2003]. Azinheira also implemented a visual servo controller for hovering, or station control, of an outdoor robotic airship [Azinheira *et al*., 2002]. All the image-based blimp control systems mentioned above rely on the information processed by computers on the ground station. Visual devices in these control systems are able to work fast enough to collect the flight information of the blimp in real time.

With the development of more powerful and smaller sized microcomputers, it is now possible to handle the processing of significant volumes sensory data, such as visual interpretation for navigation and motion control onboard. In 2007, Rottmann and colleagues developed an onboard Linux operation system and device driver

interface on their autonomous blimp to apply intelligent control algorithms [Rottmann *et al*., 2007]. The total weight of this onboard controller was less than 200 grams. All the control tasks for this smart blimp were managed by the onboard microcontroller automatically without any human intervention.

The control methods utilised on autonomous blimps have also been significantly improved as a natural consequence of the advancement of relevant new technologies. In particular, more advanced control theories have been adopted to address the issues in navigation control of the blimp, which is a nonlinear and under-actuated dynamic system. The concept of periodic blimp control was mentioned in 2001 by [Hong Zhang and Ostrowski, 2001]. Model Predictive Control was also proposed and developed to achieve good performance for autonomous blimp control [Fukushima *et al.*, 2006] [Fukushima *et al*., 2006]. The Backstepping technique [Hygounenc and Soueres, 2002], [Beji *et al.*, 2002] is a new attempt to deal with autonomous control in the environment with low perturbations. Inverse Optimal Tracking Control was also implemented for this under actuated system in an attempt to provide the autonomous blimp a stability margin which guarantees robustness with respect to the input uncertainties [Fukao *et al*., 2005].

Most of this related research is based on the analysis of the dynamic models of the blimp. Sergio has provided a thorough analysis of the dynamic modeling of a blimp, and has made a comprehensive description of the physical principles of general airship operation [Gomes and Ramos, 1998]. Ko, has alternatively used Gaussian processes and reinforcement learning to help find the dynamic model of an autonomous blimp in a single formulation [Ko *et al*., 2007].

The blimp navigation control methodologies employed in the previous work assumed time-invariant environment models, which are neither true or not applicable to an actual autonomous airship. Acquisition of behavioural skills of an expert human operator and their codification in an intelligent autonomous system is an important but rather challenging task. A systematic method to realize this process will greatly simplify the development, commissioning and maintenance of autonomous blimp systems.

3 Reinforcement Learning

Generally over prolonged periods the presence of feedback, in either either positive or negative forms, can ultimately help people obtain better solutions when they are dealing with an unknown environment. This experience and exportation ability is one kind of learning process, which provides critical judgments that bias appropriate decisions based on rewards or punishments mostly from personal experience. Interaction with the surrounding environment inturn provides the basic reinforcement for learning experiences which often leads to intuitions in a human. In a typical reinforcement learning process, the agent uses rewards or punishments from the environment to accommodate with an unknown circumstance and produce adaptive actions to it.

The advantage of reinforcement learning is that it can be used to solve problems that occur in a complex environment which an agent has little information and knowledge about. Reinforcement learning will enable an agent to achieve good performance after an adequate training period, where most if not all available feedback from a range of learning process trials is utilised. Of course, this learning progress will only develop and adapt knowledge, such as control solutions, for a particular environment. The basic scenario in reinforcement learning is to provide an appropriate classification of rewards or punishments according to the result of each iteration episode.

However, reinforcement learning can take significant time to complete all possible trials during the learning process. For this reason, the efficiency of learning is mostly influenced by the aspects of each learning iteration. More or less, there is no guarantee that the best solution, or skills, can be found after training for a long time. However, compared to classical control, reinforcement learning can provide a quicker response to the changes of the environment, because the current optimal control policy can be acquired by this algorithm after each episode.

4 Q-learning algorithm in blimp control

One of the most important breakthroughs in reinforcement learning was the development of an off-policy TD control algorithm known as Q-learning [Watkins and Dayan, 1992]. The difference between on-policy TD algorithm and off-policy TD methods is in the learned action-value function. In particular, for an On-policy method, we must estimate $Q^{\pi}(s,a)$ for the current behaviour policy π and for all states S and actions A. Such as in the State-Action-Reward-State-Action (SARSA) learning methods which is an on-policy TD algorithm, we continually estimate Q^{π} for the behaviour policy π, and at the same time change π toward a greediness condition with respect to the successor state. For any action-value function Q, the corresponding greedy policy is the one that deterministically chooses an action with maximal Q-value, which can be noted as that in Equation 1.

$$\pi(s) = \arg\max_{a} Q(s,a) \tag{1}$$

The simplest form of off-policy TD learning algorithm is the one-step Q-learning, defined by Equation 2.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}) - Q(s_t, a_t)] \quad (2)$$

In this notation, it can be seen that the learned action-value function, Q, directly approximates Q^* (the optimal action-value function), which is independent of the policy being followed. Learning in this way, the policy has an effect because it determines which state-action pairs are visited and updated. This dramatically simplifies the analysis of the algorithm and enables early convergence. The requirement for correct convergence is that all pairs continue to be updated. This is a minimal requirement in the sense that any method guaranteed to find optimal behaviour in the general case must require it.

A fundamental requirement for autonomous blimps is to achieve the goal of autonomous navigation. The height can be controlled separately form the navigation control, but it can be affected by the payload, physical structure and characteristics of a blimp envelope, as well as the thrusters (propellers) locations and the angles of rudders on the body. The problems of height control for autonomous blimps has been widely studied in different implementations [Azinheira *et al.*, 2002], [Kampke and Elfes, 2003], [Rottmann *et al.*,2007]. So here, the research focus is on the blimp autonomous navigation. The pitch and roll of the blimp have little effect on the turning motions of navigation tasks because blimp angular accelerations in X and Y (horizontal) axes are not able to provide sufficient thrust for yaw tuning during flight. The tail motor of the autonomous blimp accounts for the majority of the yaw turning moments, which in-turn directly impacts on the heading and thus the navigation of the blimp. Under these conditions, the control problem can be significantly simplified in converting it from 6 to 1 degree of freedom.

In order to program for Q-learning, we need an equation that increases the Q-value when a reward is positive, decreases the value when it is negative, and holds the value at equilibrium when the Q-values are optimal. The equation utilised for this follows:

$$Q(a,i) \leftarrow Q(a,i) + \beta(R(i) + Q(a_1, j) - Q(a, j)) \quad (3)$$

- Q - a table of Q-values
- a - previous action
- i - previous state
- j - the new state that resulted from the previous action
- a_1 - the action that will produce the maximum Q-value
- β - the learning rate (between 0 and 1)
- R - the reward function

Off-policy learning techniques are especially suitable for systems that include existing controllers, other behaviours or existing knowledge, in addition to the learning system. This advancement was developed from the classical reinforcement learning situation in which the learning system learns from scratch, interacting purely with its environment.

To evaluate the reinforcement learning algorithm, a small model sized blimp was chosen to implement the autonomous navigation control system. The physical dimension of the body envelope of this small blimp is 1.4 meters long and 0.75 meters in diameter. The gondola of the blimp is located under the middle of the main body envelope. At both sides of this gondola two main propellers are mounted as the main propulsion force. These two propellers are driven by 2 DC motors which are suitable for the limited indoor flight tests, and the angles of mounting position of these two propellers is fixed along a common shaft between them, which can be rotated to control the final thrust from both by a main servo in the gondola. The servo combined with main DC propellers is able to turn the propulsion force around the horizontal axes. The basic body structure of this blimp is shown in Figure 1.

Figure 1: The blimp undergoing development tests.

5 Simulation of Autonomous Blimp

Webots is a three dimensional mobile robot simulator. It was originally developed as a research tool for investigating various control algorithms in mobile robotics. It contains a rapid prototyping tool allowing the user to create 3D virtual worlds with physics properties, such as mass repartition, joints, friction coefficients, etc. The user can add simple inert objects or active objects called mobile robots. Moreover, they can be equipped with any number of sensor and actuator devices, such as distance sensors, motor wheels, cameras, servos, touch sensors, grippers, emitters, receivers, etc. Webots contains a large number of robot models and controller program examples that help the users get started. A controller is an executable binary file which is used to control a robot described in a world file.

Figure 2 shows the physical structure of the blimp

model in the virtual environment. The reference coordinate axes of the blimp body are indicated as red and blue lines in the figure. Here the X axis of the coordinate system is aligned with the blimp's heading direction and the Y axis is pointing to the side (Port and Starboard) of the blimp body. Some of the blimp flight simulation data, such as target difference, angular speed of turning, angular rotation acceleration, are referenced to this coordinate system.

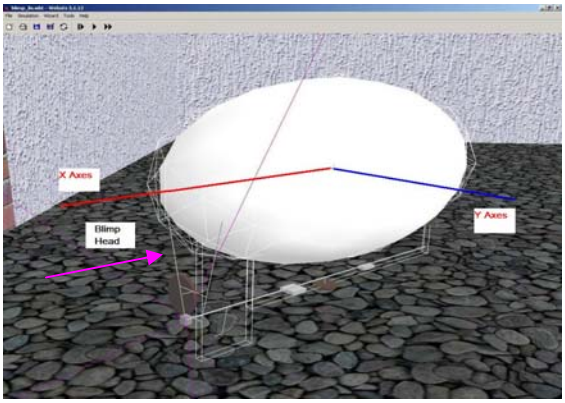

Figure 2: The body coordinate of the blimp in the virtual world.

5.1 Blimp navigation tasks

An initial simulation task that was chosen involved an indoor environment with slight disturbances of the air flow in order to test the learning and control performance of the autonomous navigation task from the original position to the target position. The goal of blimp navigation in this simulation is to search and turn to the direction of a moving target. The control strategy for this navigation task is broken down to two steps. The first step is to steer the heading of the blimp towards the goal direction by rotating the blimp at the original location. The second step is to approach the target position in a prismatic motion.

During the simulation of blimp navigation, the independent control of position and orientation was implemented to test the performance of reinforcement learning. Four target positions were set up initially to evaluate the efficiency of the Q-learning algorithm in blimp navigation control. The heading of the blimp is required to turn to these 4 different target locations separately via Q-learning. In each trial, the blimp needs to discover the most appropriate actions for rotation and control its heading in order to face each new target position accurately without excessive oscillations. Figure 3 shows the target positions and the initial setting of the blimp simulation environment. Targets A, B, C, and D are typical positions in I, II, III, and IV quadrants referencing to the blimp body coordinate frame.

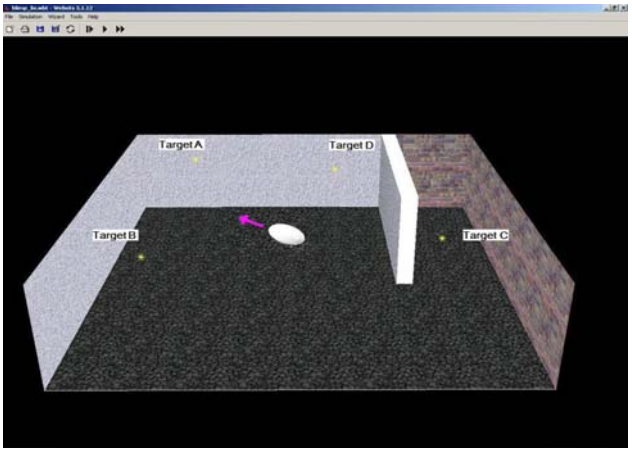

Figure 3: Initial set up of a four quadrant target trials.

5.2 Simulation results of continuous Q-learning

In the navigation control tasks, a blimp would often need to automatically seek and reach various new goal positions - such as in following a predefined flight plan. In order to simulate this procedure, different target positions were designed as a sequence of goal directions for blimp to achieve in one simulation test. In this blimp navigation test, the control strategy of the blimp turning is that of rotating the blimp heading to the first target position, and subsequently turning toward the second target orientation after achieving a stable orientation, for a specified number of iterations, of the former target. By repeating the above control procedures for the blimp a number of times, it can complete an exhaustive range of turning tasks in various directions at the same point in virtual world. This can assist the autonomous blimp to amass enough learning from various turning experiences to achieve correct goal orientations with a robust performance. Further, by combining linear forward (straight) motion, in certain iterations at stable states, the blimp is able to handle autonomous navigation tasks efficiently.

Figure 4 presents the blimp turning results of orientation and angular difference in the simulation of autonomous navigation under the (long-time) continuous-targets task. The sequence of target positions in the planar body coordinate frame are (16, -16), (-16, 16), and (16, -16) again.

The angular difference shown in Figure 4 provides details of turning motions. Three obvious short ranges with the value of zero can be readily observed along the red line in plot of angular difference. The average duration of these three periods of zero is approximately 200 iterations, which represent stable stages in the blimp turning that was achieved after each new target position was given. When the blimp rotates towards each new goal direction, it will maintain this orientation for some 200 iterations, before the next target is issued. The orientation plot of Figure 4 clearly reflects the same events. The red

line in the orientation plot represents first target potion (16, -16), and the blue line identifies the second target position (-16, 16). It can be seen that, after the initial learning phase, the blimp turns correctly to the first goal direction ($3\pi/4$), and subsequently moves to face the second goal ($-\pi/4$) after a suitable relearning process.

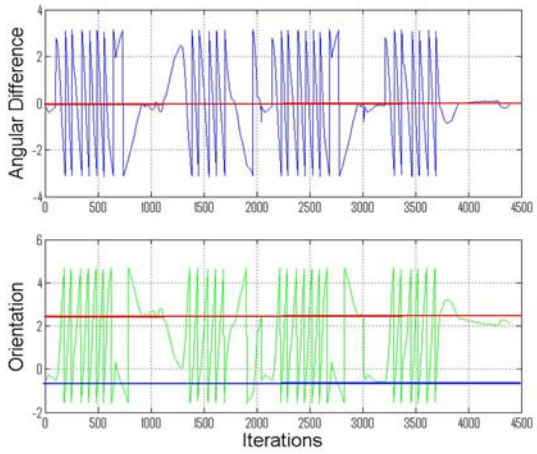

Figure 4: Angular difference and Orientation in the learning of continuous turning.

In Figure 5, the sequence of actions are displayed that have enabled the required learning for the blimp to realise the various goals (stable periods) of continuous turning tasks. Three stages of small actions (trimming control) around state number 3 can be readily identified in Figure 5, which inturn match the noted performances in orientation and angular difference already mentioned.

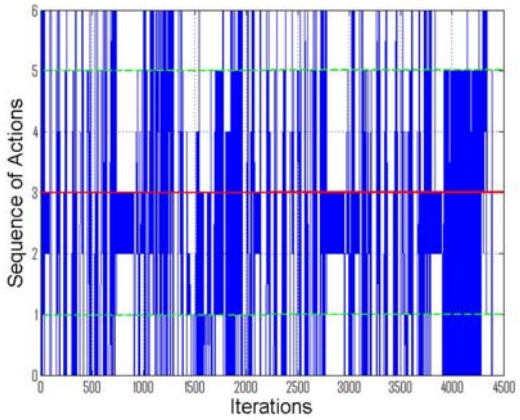

Figure 5: Sequence of the actions in the continuous turning.

Another aspect of the Q-learning in this continuous turning task can be obtained from the sequence of blimp states. As analysed in a similar previous manner, the stable stages of this learning process have been labelled in Figure 6. Here, the three stable regions correspond to the acquisition of the three target orientations previously discussed. The reference for stable states is represented as number 20 (marked in red).

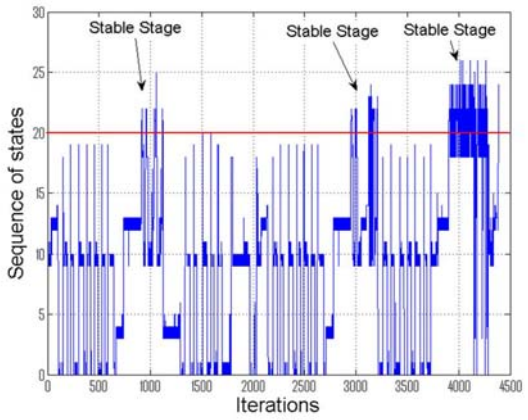

Figure 6: Sequence of the states for the continuous turning task.

5.3 Learning performance: Q–value tables

The main results to show the learning process in this simulation experiment is that of the Q-value tables. The values in the Q-table can be used to evaluate the learning efficiency of Q-learning exploration. These tables recorded the Q-values of all state-action pairs, and were updated after each Q-learning step. With the blimp simulation running, an exploration process within the Q-learning algorithm is enabled, which is managed through the Q-value updating procedure for the Q-table. All of these Q-value tables are analysed with MATLAB based on the data records from Webots.

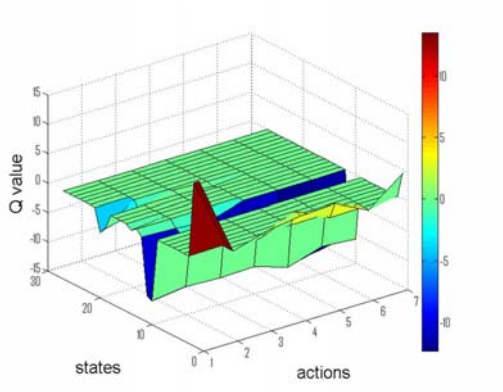

Figure 7: Q-value surface plot of the restricted Q-learning.

Figure 7 corresponds to the Q-value table produced after a small number of iterations. From this figure, it can be seen that most of the Q-values in this table are 0 (green coloured), which represents that none, or a very limited amount of value updating has occurred. That is to say the degree of learning at this stage is not sufficient, and that the exploration of the Q-learning has been, thus far, too limited. This is a very important characteristic in determining whether or not there is sufficient learning in the control system for more complicated flights, or indeed, more demanding environmental challenges during the blimp navigation tasks.

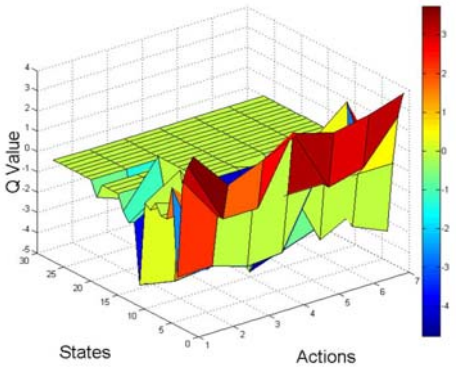

Figure 8: Q-value surface plot of the extended Q-learning.

Further results illustrated in Figure 8 provide another view of the Q-learning table. Here, a more extended number of Q-values have been updated during the leaning process. The maximum value of the Q-value in this figure reaches to approximately 3.7 rad., and the minimum value is around -4.8 rad. Both of these are in response to various different state-action pairs. The large difference between these two values indicates that the majority of the possible flight states have been visited by the Q-value updating mechanism. It can be seen from Figure 8 that the colour of most state-action pairs is modified from green (zero or few updates) to a range of other colours. Thus, this Q-learning process is more exclusive than the previous example shown in Figurer 7.

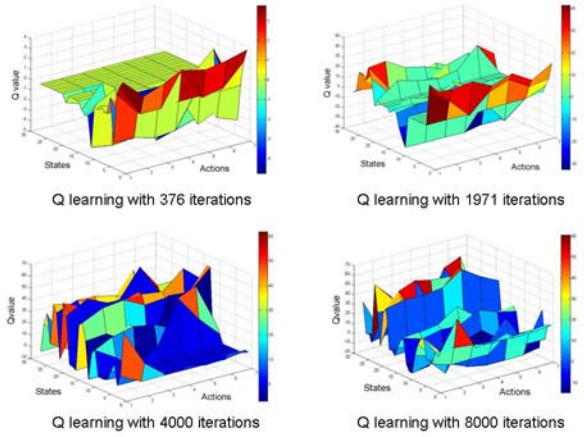

Figure 9: Results of Q-value tables with different iterations.

In each of the blimp simulation tests, the initial values of all variables in the Q-value table were set to 0. As a result, the blimp agent nullified or removed the experience from previous learning. Through progressive updating of the Q-value tables, the blimp controller learns from the unknown environment through various rewards or punishments. However, the initial values of Q-table were not all zeros; the exploration of an optimal control policy in Q-learning would be disturbed, or delayed, by comparing these original values to the current environment experience. This observation can also be identified by the simulation results of the various Q-value tables.

The greater the number of Q-learning iterations being experienced, the more extensive the exploration of the Q-value tables becomes. The various views presented in Figure 9 demonstrate this. These results explain the relationship between the accuracy and the number of iterations (cost) of the learning.

6 Conclusion and Future Work

This paper investigated the issues of developing a Q-learning algorithm for autonomous blimp navigation control in an unknown environment. Through the interaction with the environment during the flight, the Q-learning algorithm is able to acquire the optimal control policy for the blimp. The autonomous blimp observes the effects of its actions and based on this observation learns to select the proper actions to reach to the target positions in the navigation task. This learning strategy is able to be applied in a wide range of problems, in which neither human intervention nor expert supervised knowledge are required, and therefore has a clear future potential in many areas, such as in economic, social and industrial applications.

Some popular methods for improving Q-learning performance have not been applied in this work yet. Domain knowledge may be used to convert the parameters of the input states into measurements which are easier to learn from. The state-action space can be enlarged by further considering continuous action variables, which inturn would add further issues to the development of the Q-value function. This would be of value to explore in future work as a possible means to improve the rate of Q-learning.

References

[Walkenhorst et al., 2000] Walkenhorst, B.T. and Miner, G.F. and Arnold, D.V. A low cost, radio controlled blimp as a platform for remote sensing. In Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International, pages 2308-23092, vol. 5, 2000.

[Brandreth, 2000] Brandreth, E.J.,Jr. Airships: an ideal platform for human or remote sensing in the marine environment. In OCEANS 2000 MTS/IEEE Conference and Exhibition, pages 1883-1885, vol. 3, 2000.

[Hong Zhang and Ostrowski, 1999}] Hong Zhang and Ostrowski, J.P. Visual servoing with dynamics: control of an unmanned blimp. In Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference, pages 618-623, vol. 1, 1999.

[Silveria et al., 2002] Silveria, G.F. and Carvalho, J.R.H.

and Rivers, P. and Azinheria, J.R. and Bueno, S.S. and Madrid, M.K. Optimal visual servoed guidance of outdoor autonomous robotic airships. In American Control Conference, 2002. Proceedings of the 2002, pages 779-784, vol. 1, 2002.

[Fukao et al., 2003] T. Fukao and K. Fujitani and T. Kanade. Image-based tracking control of a blimp. In Proc. of the IEEE Conf. on Decision and Control, 2003.

[Azinheira et al., 2002] Azinheira, J.R and Rives, P. and Carvalho, J.R.H. and Silveira, G.F. and de Paiva, E.C. and Bueno, S.S. Visual servo control for the hovering of all outdoor robotic airship. In Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference, pages 2787-2792, vol. 3, 2002.

[Rottmann et al., 2007] Rottmann, A. and Plagemann, C. and Hilgers, P. and Burgard, W. Autonomous blimp control using model-free reinforcement learning in a continuous state and action space. In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference, pages 1895-1900, Oct.-Nov., 2007.

[Hong Zhang and Ostrowski, 2001] Hong Zhang and Ostrowski, J.P. Periodic control for a blimp-like dynamical robot. In Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference, pages 3396-3401, vol. 1, Oct. 2001.

[Fukushima et al., 2006] Fukushima, H. and Kon, K. and Matsuno, F. and Hada, Y. and Kawabata, K. and Asama, H. Constrained Model Predictive Control: Applications to Multi-Vehicle Formation and an Autonomous Blimp. In SICE-ICASE, 2006. International Joint Conference, pages 4515-4520, Oct. 2006.

[Fukushima et al., 2006] Fukushima, H. and Saito, R. and Matsuno, F. and Hada, Y. and Kawabata, K. and Asama, H. Model Predictive Control of an Autonomous Blimp with Input and Output Constraints. In Control Applications, 2006. CCA '06. IEEE International Conference, pages 2184-2189, Oct. 2006.

[Hygounenc and Soueres, 2002] Hygounenc, E. and Soueres, P. Automatic airship control involving backstepping techniques. In Systems, Man and Cybernetics, 2002 IEEE International Conference, pages 6, vol. 6, Oct. 2002.

[Beji et al., 2002] Beji, L. and Abichou, A. and Bestaoui, Y. Stabilization of a nonlinear underactuated autonomous airship-a combined averaging and backstepping approach. In Robot Motion and Control, 2002. RoMoCo '02. Proceedings of the Third International Workshop, pages 223-229, 9-11 Nov. 2002.

[Fukao et al., 2005] Fukao, T. and Kanzawa, T. and Osuka, K. Inverse optimal tracking control of an aerial blimp robot. In Robot Motion and Control, 2005. RoMoCo '05. Proceedings of the Fifth International Workshop, pages 193-198, 23-25 Jun. 2005.

[Gomes and Ramos, 1998] Gomes, S.B.V. and Ramos, J.G., Jr. Airship dynamic modeling for autonomous operation. In Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference, pages 3462-3467, vol. 4, 16-20 May. 1998.

[Ko et al., 2007] Ko, J. and Klein, D.J. and Fox, D. and Haehnel, D. Gaussian Process and Reinforcement Learning for Identification and Control of an Autonomous Blimp. In Robotics and Automation, 2007 IEEE International conference, pages 742-747, 10-14 Apr. 2007.

[Watkins and Dayan, 1992] Watkins, C.J.C.H. A. and Dayan, P. Technical note: Q-learning. In Machine leaning, 8(3/4): 279-292, 1992.

[Kampke and Elfes, 2003] Kampke, T. and Elfes, A. Optimal aerobot trajectory planning for wind-based opportunistic flight control. In Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference, pages 67-74, vol. 1, 27-31 Oct. 2003.