Autonomous Autorotation of an RC Helicopter

Pieter Abbeel[1], Adam Coates[1], Timothy Hunter[2], and Andrew Y. Ng[1]

[1] Computer Science Department, Stanford University, Stanford CA 94305
[2] Electrical Engineering Department, Stanford University, Stanford CA 94305

Summary. In case of engine failure, skilled pilots can save a helicopter from crashing by executing an emergency procedure known as autorotation. In autorotation, rather than relying on the engine to drive the main rotor, the pilot has to control the helicopter such that potential energy from altitude is transferred to rotor speed. In fact, maintaining a sufficiently high rotor speed is critical to retain sufficient control of the helicopter to land safely. In this paper, we present the first autonomous controller to successfully pilot a remotely controlled (RC) helicopter during an autorotation descent and landing.

1 Introduction

Autonomous helicopter flight represents a challenging control problem with high-dimensional, asymmetric, nonlinear dynamics. Helicopters are widely regarded to be significantly harder to control than fixed-wing aircraft. (See, e.g., [12, 18].) At the same time, helicopters provide unique capabilities, such as in place hover and low-speed flight, important for many applications.

Recently, there has been considerable progress in autonomous (RC) helicopter flight. Examples range from basic upright hovering and forward flight [4, 10, 15, 16, 17] to inverted hovering [14], and even to extreme aerobatic maneuvers [1, 6, 5].

All of this prior work pertains to helicopters operating with normal engine power. By contrast, in this paper we consider autonomous helicopter flight when the engine is not engaged. Indeed, even when the engine has failed, a skilled pilot can safely descend and land a helicopter through autorotation.

Whereas during powered flight, the rotor drag is overcome by the engine power, during autorotation the rotor drag is overcome by the airflow through the blades. Effectively the potential energy of the helicopter (corresponding to its altitude) is transferred to rotor speed. This rotor speed allows the pilot to control the helicopter throughout its descent, and then slow down the helicopter before touching down.

Autorotation landings are a challenging maneuver and improperly executing an autorotation maneuver often leads to severe damage or even complete loss of the helicopter: If the main rotor speed becomes too low, it becomes impossible to reliably control the helicopter, and the helicopter will typically crash. If the helicopter touches the ground with substantial horizontal velocity, it will tip over. If the helicopter touches the ground with too low a main rotor speed then the main rotor blades will flex sharply downward. This can result in one of the rotor blades striking the tailboom, destroying both that rotor blade and the tailboom.

Moreover, in contrast with regular landings—where the pilot could abort a landing attempt and try again later— autorotation landings only give you a single shot at the approach. One autorotation landing attempt with a poor controller suffices to destroy a helicopter. This makes autorotation landings logistically a particularly challenging research problem.

While engine failure is likely the better known reason to fly a helicopter in autorotation, autorotation is also crucial in case of tail-rotor failure. In case of tail-rotor failure, if one keeps the engine running, the torque from the engine causes the helicopter to rotate (fast) around its vertical axis, which makes it very hard (if not impossible) to fly the helicopter reliably. Switching off the engine removes the torque that causes this rotation. Hence, in case of tail rotor failure, the pilot can still maintain control of the helicopter by disengaging the engine and performing an autorotation descent and landing.

In this paper, we present the first controller to successfully pilot a (RC) helicopter during an autorotation descent and landing. We start by collecting flight data from our expert human pilot, which includes several autorotation descent and landing demonstrations. Next we learn a dynamics model from the flight data.[3] The dynamics model we propose in this paper builds upon the dynamics model proposed in [2]: we extend it for the autorotation setting. In particular, the model we present in this paper explicitly incorporates a model for the rotor speed dynamics, a crucial aspect of helicopter flight during autorotation.[4] Then, since it can be very difficult to specify helicopter maneuvers by hand, we use the expert demonstrations to define the autorotation task. (See also, e.g., [5], where demonstrations were used to enable a helicopter to fly high performance helicopter aerobatics.) Once we have the task specification and the dynamics model, we use differential dynamic programming (an extension of the linear quadratic regulator to nonlinear systems, see, e.g., [8, 3]) to find a feedback controller to perform the autonomous autorotations.

We extensively tested our autonomous autorotation controller on our helicopter. Concretely, we had our helicopter perform a large number (25) of autorotations, each of which resulted in a successful landing.

There is a significant body of work studying helicopter flight in autorotation (see, e.g., [18, 11, 9]). However, prior work has only considered the analysis of autorotation controllers and autorotation dynamics—often with the goal of pilot training. No

[3] Note on terminology: we do not intend to make a distinction between "system identification" and "learning a dynamics model." For the purposes of this paper these can be considered equivalent.

[4] In powered helicopter flight, the variation in rotor speed is relatively minor as the engine tries to keep the rotor speed close to constant.

prior work has autonomously descended and landed a helicopter through autorotation.

In Section 2 we describe our modeling approach. In Section 3 we describe our control design and give details on how we use the demonstrations to define the control task. In Section 4 we describe our experimental results. Section 5 concludes the paper.

Videos of our autonomous autorotations are available at:

`http://heli.stanford.edu`.

2 Modeling the Dynamics

Helicopters are well-known to have complex dynamics. For instance, to completely capture the state of the "helicopter system" one would have to include the—typically extremely complex—state of the air around the helicopter into the dynamics model. (See, e.g., [12, 18].) However, various prior work has shown it is possible to build a sufficiently accurate model for control by treating the helicopter as a rigid-body, possibly including the blade-flapping dynamics and the main rotor speed. (See, e.g., [13, 7, 6, 2, 1, 5].)

We model the helicopter with a thirteen dimensional state consisting of position, orientation, velocity, angular rate and main rotor speed. The helicopter is controlled via a 4-dimensional action space: the cyclic pitch controls $i_{\mathrm{lon}}, i_{\mathrm{lat}}$, which cause the helicopter to pitch forward/backward or sideways; the tail rotor (rudder) control i_{rud}, which affects tail rotor thrust, and can be used to yaw (turn) the helicopter; the main rotor collective pitch control i_{col}, which changes the main rotor thrust by changing the pitch of the rotor blades.

Following [2], we first subtract out the effects of inertia and gravity, and then learn a model from data to predict accelerations in a coordinate frame attached to the helicopter. Doing so allows us to use a dynamics model with a relatively small number of parameters to be estimated from flight data. We integrate the accelerations over time to obtain position, velocity, orientation, angular rate and main rotor speed.

Concretely, our dynamics model uses the following parameterization to predict accelerations:

$$\begin{aligned}
\dot{u} &= v * r - w * q - g_u + C'_u * [u], \\
\dot{v} &= w * p - u * r - g_v + C'_v * [v], \\
\dot{w} &= u * q - v * p - g_w + C'_w * [1; \quad w; \quad i_{\mathrm{col}} * \Omega; \quad \sqrt{u^2 + v^2}], \\
\dot{p} &= C'_p * [1; \quad p; \quad i_{\mathrm{lat}} * \Omega], \\
\dot{q} &= C'_q * [1; \quad q; \quad i_{\mathrm{lon}} * \Omega], \\
\dot{r} &= C'_r * [1; \quad r; \quad i_{\mathrm{rud}} * \Omega], \\
\dot{\Omega} &= C'_\Omega * [1; \quad \Omega; \quad i_{col}; \quad w; \quad \sqrt{u^2 + v^2}; \quad (i_{\mathrm{lat}}^2 + i_{\mathrm{lon}}^2)].
\end{aligned}$$

The velocities (u, v, w) and angular rates (p, q, r) are expressed in the helicopter's reference frame. Here g_u, g_v, g_w refer to the components of gravity in the helicopter's reference frame; Ω is the main-rotor speed. Similar to [2], we evaluate the model by considering its simulation accuracy over several seconds. In Section 4 we document this accuracy in detail for the rotor speed, which is particularly important for autorotation.

We estimate the parameter vectors $C.$ from flight data using least squares. Note that only the last equation in the dynamics model is specific to autorotation. Hence, thanks to the non-linear parameterization, we can use powered flight data to estimate the parameters appearing in the first six equations. This is an interesting practical property of the proposed model: during autorotation it is dangerous to apply large control inputs as is often done when collecting data with the purpose of learning a dynamics model—large control inputs would slow down the rotor speed and make the helicopter hard (or even impossible) to control. The nonlinear parameterization allows one to still use data with large control inputs to learn the model. In our experience this improves the accuracy of the learned model, especially so when the flight data is noisy—as is often the case for (small-scale) helicopters where vibration tends to pollute the sensor measurements, and the resulting state estimates.

The ground is well known to affect helicopter dynamics whenever the helicopter is within two rotorspans of the ground. In our experiments, we found it diffult to accurately model the influence of the ground effect on the helicopter dynamics.[5] However, the net effect relevant for control during an autorotation landing was sufficiently well captured by adding a vertical offset relative to the vertical position predicted in the absence of ground effect. This vertical offset was easily estimated from flight data and taken into account accordingly.

3 Control

Once we have an accurate dynamics model, two key challenges remain: (i) Formulating the control problem in a format amenable to control design algorithms, and (ii) Solving the resulting control problem.

For the case of optimal control (or reinforcement learning), we need to specify a cost function that corresponds to the task at hand. For complex tasks, such as helicopter flight, it is often very challenging to hand-specify the task in the form of a cost function. Concretely, for many helicopter maneuvers a fairly natural description of the task requires one to specify a trajectory (a sequence of states and control inputs to be visited over time). However, helicopter dynamics are very complex and this makes it difficult (if not impossible) to hand-specify a trajectory that (even approximately) obeys the dynamics. Similarly to [5], we leverage our expert pilot's demonstrations to find a good target trajectory.

Concretely, an autorotation maneuver is naturally split into three phases.

[5] Close to the ground, one cannot safely exert the large control inputs standardly used to collect flight data for system identification.

1. **Autorotation glide.** The helicopter descends at a reasonable velocity while maintaining a sufficiently high main rotor speed, which is critical for the helicopter to be able to successfully perform the flare.
2. **Autorotation flare.** Once the helicopter is at a certain altitude above the ground, it transitions from the glide phase into the flare phase. The flare slows down the helicopter and (ideally) brings it to zero-velocity about 50cm above the ground.
3. **Autorotation landing.** Once the helicopter has completed the flare, it lands by using the remaining rotor speed to maintain a level orientation and slowly descend until contacting the ground.

We recorded several autorotations from our expert pilot and split each of the recorded trajectories into these three phases.

The glide is a steady state (rather than a trajectory) and we picked as our target glide state a typical state (in particular velocity and rotor speed) from the glides our expert performed.

The landing is simply characterized by a level orientation, a rotor speed, and a very low downward velocity. Concretely, our target for the landing is for the helicopter to maintain zero velocity and level orientation. Since the helicopter's engine is disabled, the main rotor speed will gradually decrease during this in-place hover, and the helicopter will slowly descend and land.

The flare is very challenging to specify—it does require a state trajectory. A natural candidate for the flare trajectory would be the best expert demonstration, or even an idealized version automatically estimated from the expert's many suboptimal demonstrations (as proposed for helicopter aerobatics in [5]). We use an idealized version of the best expert demonstration for our flare target trajectory. (See Section 4 for details.)

Once we have the target state or trajectory for each phase in the autorotation, we use differential dynamic programming (DDP), an extension of the linear quadratic regulator (LQR) formalism for non-linear systems. We penalize quadratically for deviations from the target state or trajectory. See, e.g., [3], for more details on linear quadratic methods, [8] for more details on DDP, [1] for more details on DDP in the context of autonomous helicopter flight.

4 Experimental Results

4.1 Experimental Setup

Figure 1 shows our helicopter platform: an XCell Tempest (length 54", height 19") powered by a 0.91-size, two-stroke engine. We instrumented our XCell Tempest with a Microstrain 3DM-GX1 inertial unit, which measures three-axis acceleration, angular rate and magnetic field. A ground-based four-camera system measures the helicopter's position.[6] A (extended) Kalman filter uses both of these sets of measurements to track the helicopter's position, velocity, orientation and angular rate. In

[6] Using ground-based vision is just one of many possible ways of solving the localization problem. Of course, the presented control approach is independent of the particular sens-

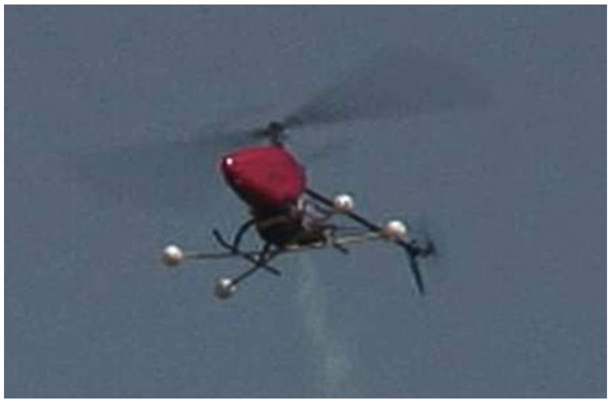

Fig. 1. Instrumented XCell Tempest during autonomous autorotation.

addition, our Tempest includes a custom tachometer, which uses a magnet attached to the main rotor shaft and a Hall effect sensor to monitor the rotational speed of the main rotor. Our Tempest also includes a sonar unit, which measures distance from the ground. XBee Pro 2.4GHz wireless radios relay sensor information from the helicopter to the ground-based flight computer.

4.2 Modeling and Simulation Results

First we had our (human) pilot perform autorotations and sweeps on each of the four control inputs through their normal operating range. In particular, we collected 10 minutes of autorotation demonstrations and 10 minutes of (powered) frequency sweeps for each of the control inputs. During the powered frequency sweeps, the governor regulated the main rotor speed around 1700rpm. During autorotation the control sweeps are small and gentle to avoid expending the rotational energy of the rotor blades. Then we learned a model from the flight data as described in Section 2.[7]

Model accuracy for position and orientation for the family of models we use has been validated in earlier work (see, e.g., [2]). Here we focus on the novel modeling aspect: the rotor speed model. We simulated the rotor speed over time. The rotor speed's evolution over time depends on the velocity and control inputs, which we provide to our simulator for this evaluation. Figure 2(a) shows both the simulated rotor speed and the actual rotor speed for a typical autorotation descent. Our rotor speed dynamics model accurately captures the true rotor speed dynamics throughout.

ing setup. One could conceivably localize the helicopter using, e.g., GPS, or any other localization system that might be available.

[7] The parameters we found for our helicopter were: $C_u = -0.05$; $C_v = -0.06$; $C_w = [-0.47; -1.42; -0.01; -0.15]$; $C_p = [-1.46; -5.74; 0.02]$; $C_q = [-0.23; -5.32; -0.01]$; $C_r = [0.52; -5.43; 0.02]$; $C_\Omega = [106.85; -0.23; -68.53; 22.79; 2.11; -6.10]$.

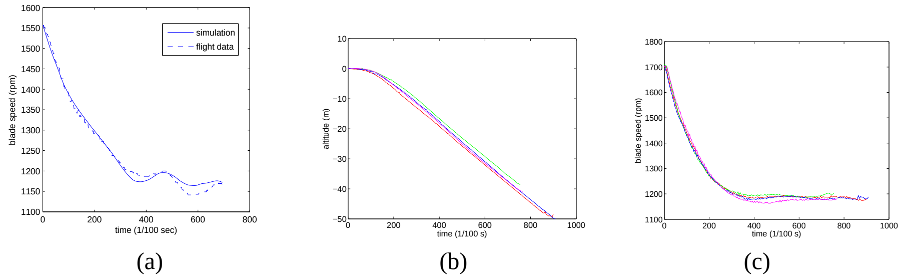

Fig. 2. (a) Main rotor speed simulation. (b) Altitude during four autonomous autorotation descents. (c) Main rotor speed during four autonomous autorotation descents. (See text for details.)

An accurate rotor speed model is crucial for model-based control design. In particular, a controller which would bring the rotor speed too low, would make it hard (if not impossible) to recover and the helicopter would crash into the ground.

4.3 Autonomous Flight Results

In our autonomous flight experiments, the helicopter starts in autonomous hover. We then enable a forward flight controller for 2 seconds. This ensures the helicopter has some forward speed. Then we disable the engine, we enable the autorotation controller and the helicopter begins its (unpowered) autorotation maneuver.[8]

During the first phase of the autorotation, the glide, our controller tries to maintain a state similar to the state (crudely) maintained by our pilot during his demonstrations. In particular, we set a target rotor speed of 1150rpm, a forward velocity of 8m/s, a downward velocity of 5m/s, and a level orientation. Similar to our pilot's demonstration, once the helicopter is 9 meters above the ground, we switch to the second phase.

During the second phase, the flare, our controller tries to follow an idealized version of our pilot's flare demonstrations. In particular, we chose our pilot's best demonstration, and slowed it down to ensure zero horizontal velocity at the end of the flare.[9] Throughout the maneuver, we penalize for deviation from the target trajec-

[8] The engine is not actually turned off. Instead, the throttle is reduced to idle, causing the clutch attached to the main rotor to disengage. In this state the main rotor spins freely and is no longer driven by the engine.

[9] Indeed, in principle it might have seemed a natural choice to just use the slowest demonstrated flare. However, there is a big discrepancy between sensing capabilities of our autonomous helicopter and our expert pilot. In particular, our expert pilot has better accuracy in sensing the distance of the helicopter from the ground. On the other hand, our autonomous helicopter has better rotor speed sensing accuracy. As a consequence, the naturally safest autorotation trajectories are different for our expert pilot and our autonomous helicopter. Our expert pilot prefers the helicopter to have high velocity, and can then time his controls just right relative to the ground to transfer (forward) velocity into rotor speed

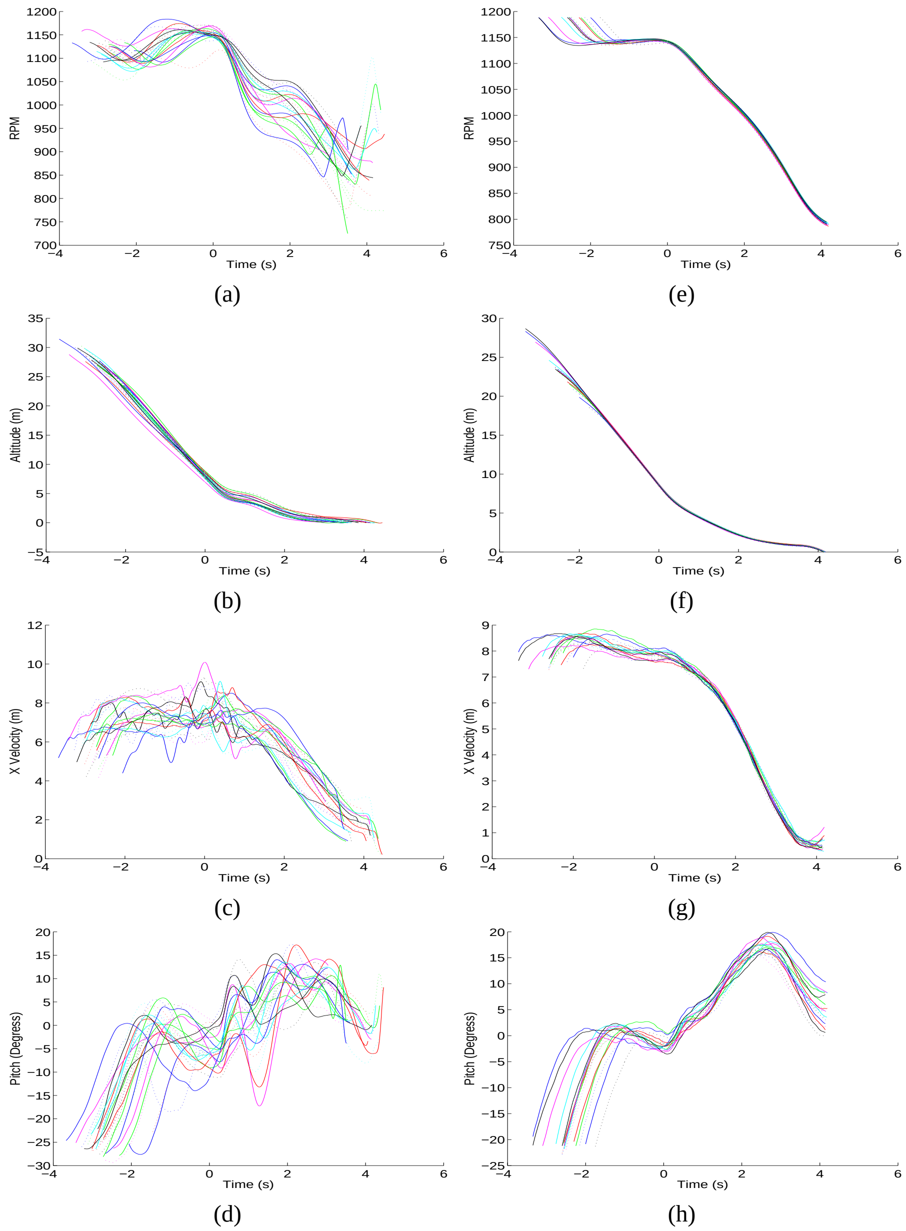

Fig. 3. (a) Main rotor speed during autonomous autorotation flights. (b) Altitude of the helicopter during autonomous autorotation flights. (c) Forward velocity of the helicopter during autonomous autorotation flights. (d) Pitch angle for the helicopter during autonomous autorotation flights. (e) Main rotor speed in (closed-loop) simulation. (f) Altitude of the helicopter in (closed-loop) simulation. (g) Forward velocity of the helicopter in (closed-loop) simulation. (h) Pitch angle for the helicopter in (closed-loop) simulation. (See text for details.)

tory's velocity, angular rate, altitude, orientation and rotor speed. Once the helicopter is 0.5 meters above the ground, we switch to the third phase.

During the third phase, the landing, our controller tries to hover the helicopter in place. The helicopter will slowly lose rotor speed while doing so, and touch down.

We performed the maneuver twenty-five times to experimentally validate our controller's performance. Each of the autorotation landings successfully touched the helicopter down gently, never causing any damage. Figure 3 (a-d) shows our autonomous flight results. In particular, it shows the main rotor speed, the altitude, the forward velocity and the pitch angle of our helicopter throughout each of the autonomous autorotations we performed. Since the glide phase can take an arbitrary amount of time (depending on how long it takes to reach the altitude that triggers the flare phase), the flights are time-aligned by setting time to be zero at the start of the flare phase. The plots start at the time we switch to power-off mode. The plots show that our autorotation controller successfully holds the main rotor speed around 1150rpm during the glide. It consistently manages to descend at a reasonable velocity, and bring its velocity close to zero during the flare.

Figure 3 (e-h) shows our simulator's predictions for our autorotation descents. Our simulator's predictions fairly closely match the flight results.

Figures 4, 5 and 6 show mosaics of three of our autorotation maneuvers. To make the mosaics, we subsampled videos of three of our autonomous autorotation flights at 4Hz. Then we overlaid the images, ensuring the background is correctly aligned. Finally,for every frame we put the patch containing the helicopter in the first layer.

We posted videos of our autonomous autorotations at the url provided in the introduction.

We also performed a completely separate set of flight tests focusing on the glide phase to verify our controller's capability of prolongedly maintaining a sufficiently high main rotor speed, while descending relatively slowly. Figure 2 (b) and (c) show the main rotor speed and altitude throughout several long glides. Our controller successfully maintains a sufficiently high main rotor speed throughout the glides: From a nominal (power on) rotor speed of roughly 1700 RPM, the main rotor is slowed to a steady-state rate around our target for this case of 1200RPM—usually within just 30RPM.

5 Conclusion

Autorotation is a maneuver that allows one to safely bring down a helicopter in case of engine failure and in case of tail-rotor failure. We first collect flight data from an expert pilot. We use the flight data to (i) build an accurate dynamics model of a helicopter in autorotation, and (ii) help us define the autorotation task. Then we

when pitching back during the flare. By contrast, our autonomous helicopter can more accurately maintain rotor speed during the descent. Hence it does not need as much forward velocity to ensure sufficient rotor speed in the flare and landing phase. As a consequence, the safer approach for our autonomous helicopter is to execute a slowed-down version of our expert's autorotation.

use tools from optimal control (in particular, differential dynamic programming) to design a controller for autonomous autorotation. Our experiments present the first successful autonomous autorotations of (RC) helicopters.

Acknowledgments

We thank Garett Oku for piloting and building our helicopter. This work was supported in part by the DARPA Learning Locomotion program under contract number FA8650-05-C-7261. Adam Coates is supported by a Stanford Graduate Fellowship.

References

1. P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *NIPS 19*, 2007.
2. P. Abbeel, V. Ganapathi, and A. Y. Ng. Learning vehicular dynamics with application to modeling helicopters. In *NIPS 18*, 2006.
3. B. Anderson and J. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall, 1989.
4. J. Bagnell and J. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *International Conference on Robotics and Automation*. IEEE, 2001.
5. A. Coates, P. Abbeel, and Andrew Y. Ng. Learning for control from multiple demonstrations. In *Proceedings of the International Conference on Machine Learning*, 2008.
6. V. Gavrilets, I. Martinos, B. Mettler, and E. Feron. Control logic for automated aerobatic flight of miniature helicopter. In *AIAA Guidance, Navigation and Control Conference*, 2002.
7. V. Gavrilets, I. Martinos, B. Mettler, and E. Feron. Flight test and simulation results for an autonomous aerobatic helicopter. In *AIAA/IEEE Digital Avionics Systems Conference*, 2002.
8. D. H. Jacobson and D. Q. Mayne. *Differential Dynamic Programming*. Elsevier, 1970.
9. W. Johnson. *Helicopter Optimal Descent and Landing After Power Loss*. NASA TM-73244, 1977.
10. M. La Civita, G. Papageorgiou, W. C. Messner, and T. Kanade. Design and flight testing of a high-bandwidth $\mathcal{H}_\infty$ loop shaping controller for a robotic helicopter. *Journal of Guidance, Control, and Dynamics*, 29(2):485–494, March-April 2006.
11. A. Lee. *Optimal Landing of a Helicopter in Autorotation*. PhD thesis, Stanford University, July 1985.
12. J. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge University Press, 2000.
13. B. Mettler, M. Tischler, and T. Kanade. System identification of small-size unmanned helicopter dynamics. In *American Helicopter Society, 55th Forum*, 1999.
14. Andrew Y. Ng, Adam Coates, Mark Diel, Varun Ganapathi, Jamie Schulte, Ben Tse, Eric Berger, and Eric Liang. Autonomous inverted helicopter flight via reinforcement learning. In *ISER*, 2004.
15. Andrew Y. Ng, H. Jin Kim, Michael Jordan, and Shankar Sastry. Autnonomous helicopter flight via reinforcement learning. In *NIPS 16*, 2004.
16. Jonathan M. Roberts, Peter I. Corke, and Gregg Buskey. Low-cost flight control system for a small autonomous helicopter. In *IEEE Int'l Conf. on Robotics and Automation*, 2003.
17. S. Saripalli, J. Montgomery, and G. Sukhatme. Visually-guided landing of an unmanned aerial vehicle, 2003.
18. J. Seddon. *Basic Helicopter Aerodynamics*. AIAA Education Series. America Institute of Aeronautics and Astronautics, 1990.

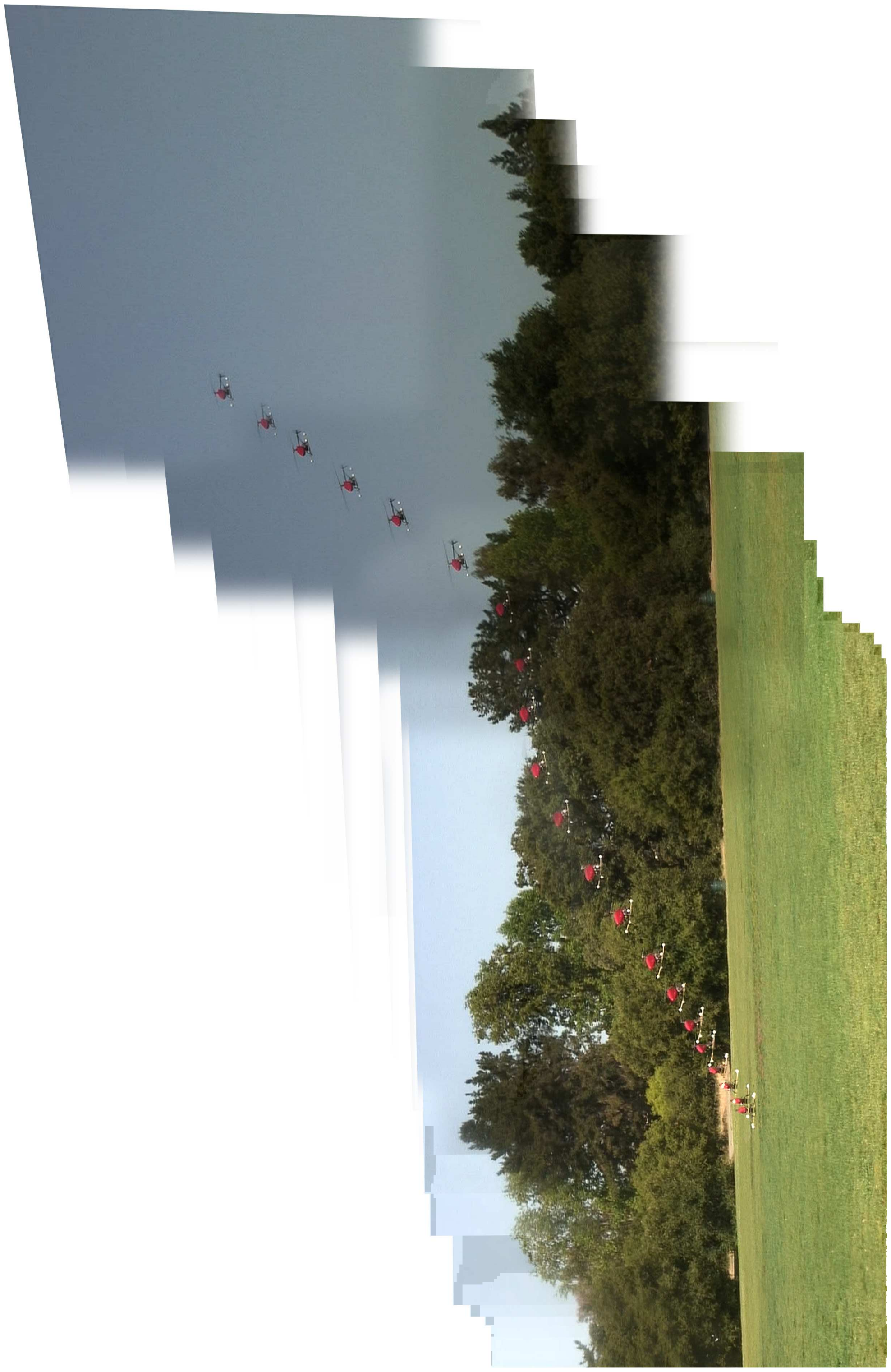

Fig. 4. Mosaic of one of our autonomous autorotation flights as viewed from to the left of the helicopter. (Sampled at 4Hz.)

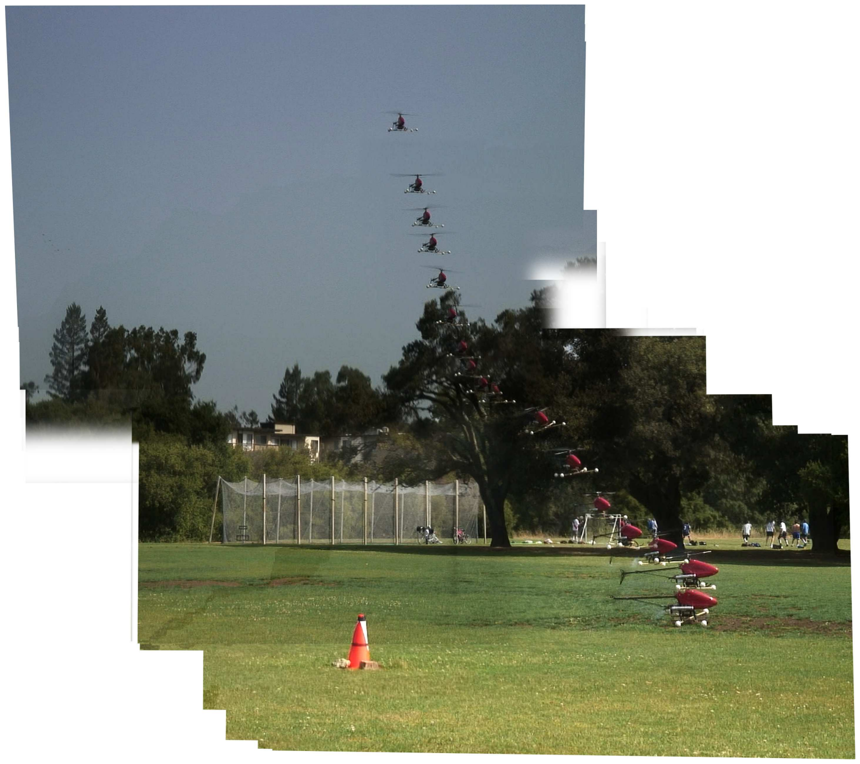

Fig. 5. Mosaic of one of our autonomous autorotation flights as viewed from in front of the helicopter. (Sampled at 4Hz.)

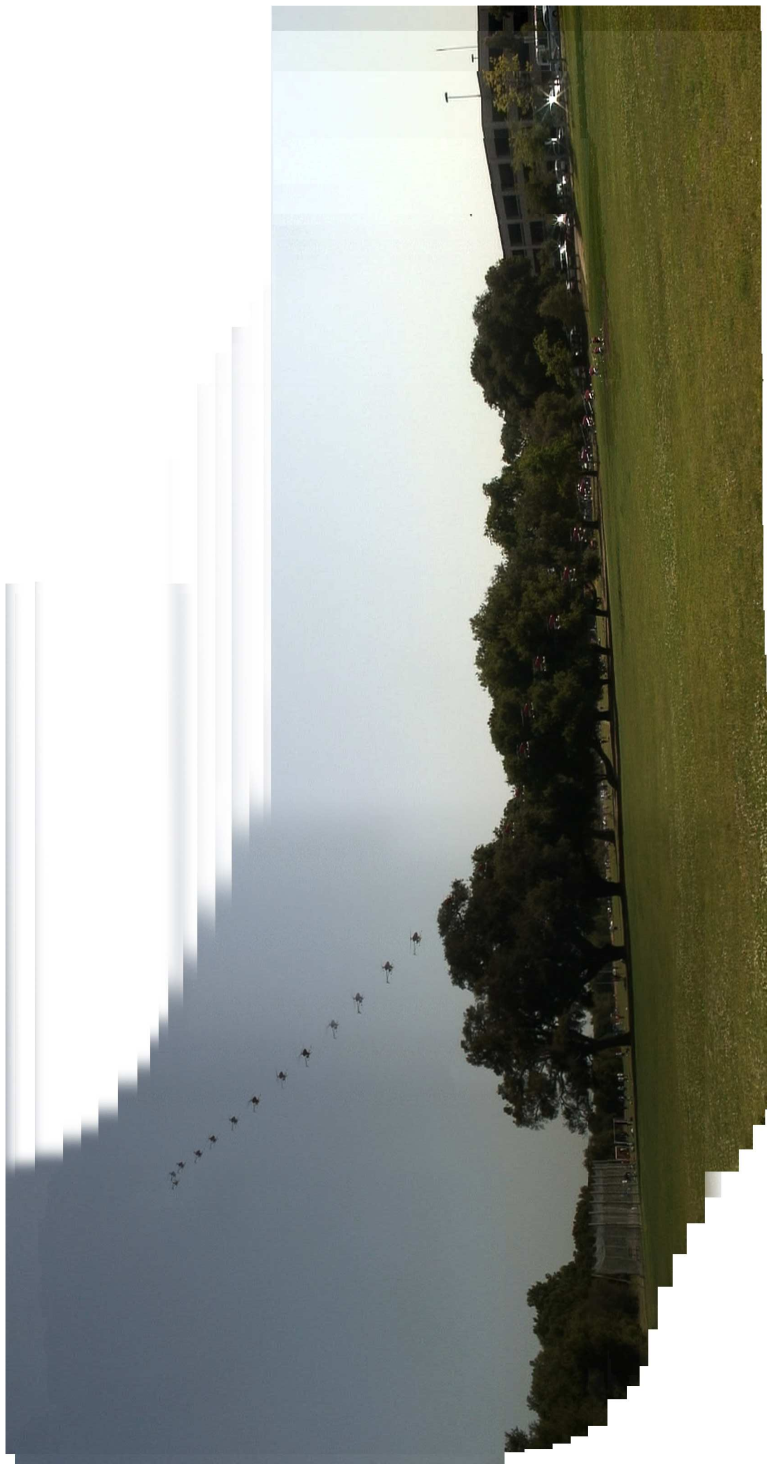

Fig. 6. Mosaic of one of our autonomous autorotation flights as viewed from to the right of the helicopter. (Sampled at 4Hz.)