CS229 Final project
Control of an autonomous helicopter in autorotation

Timothy Hunter

Autonomous helicopters have been studied for years because they represent an interesting class of unmanned aerial vehicles (UAVs). Recent works (abb [1]) have successfully shown that unmanned helicopters can accomplish the most difficult acrobatic figures and can be controlled in difficult attitudes by learning from a human model. In the wake of these studies, the present work is interested in handling an unusual configuration such as landing with the engine disabled (i.e. in autorotation). The ultimate goal that we fixed to ourselves is to build a controller that would handle the approach and the landing at a given point without the engine at any time. In the course of this class project, we built upon the previous works made by the Stanford AI/Robotics Lab. By customizing the remote-controlled helicopter used for #Pieter, we were able to perform successfull autorotations without the final landing, and we laid out the ground for future works on the landing itself.

A. Introduction

Even if modern UAVs have become very reliable over the time, their inherent mecanical complexity make them prone to failures while in flight. The engine failure is the most common mecanical failure for helicopters (see for example Padfield [3]) and pilots are trained to deal with this situation. In this case, the blades, instead of pulling the helicopter up in the air, are used as a break to slow down the fall of the machine. This steady state is called autorotation. A helicopter UAV is usually considered as expandable and its irremediable loss is the usual outcome in case of such a failure. However, now that UAVs are loaded with expensive sensors and fly over densely populated areas, it seems worth trying to recover from such delicate configurations. Since a human pilot can safely land a remote-controlled helicopter without its engine and since to our knowledge this had never been attempted before for small-scale helicopters, it seemed interesting to try to give this task to the computer as well.

The helicopter dynamics in normal conditions is now well-understood and well-simulated in nearly all the range of maneuvers. However, the understanding of the dynamics during autorotation requires a fine model of the air circulation around the blades and is much more complicated. Instead of attempting to build such a model from theoretical consideration, we built a simple model from flight data. This model is a linear model of non-linear features, and thus can be used through standard techniques to derive a controller from it.

A crucial factor is the speed of the blades, counted in number of rounds per minutes (rpm). Due to the air drag, the rpm number tends to go down. If the blades turn too slowly, the helicopter cannot sustain itself and follows a free fall path. Thus the capture and the identification of an optimal autorotation blade speed was an important part of our work.

We followed the following pattern in our work: we first recorded a number of autorotations made by our pilot. Using this data, we built a model for the steady-state part of the flight in autorotation and identified what we considered as critical parameters for the controller. Using the model and the parameters, a controller was automatically derived, tested in simulation and then in a full experiment.

B. Modelization

We use the usual representation of a helicopter as a 6 degrees of freedom solid object controlled by 4 inputs (Padfield [2] is a reference text on the helicopter dynamics) We also add as a state the rpm of the blades. The influence of the wind is neglected. Thus our model has 13 states and 4 inputs.

A helicopter usually a non-linear behaviour due to the relations between the angles and the velocities. Following the approach seen in abb [1], we build a model that predicts the physical acceleration in the body coordinates, and which is usually a simple function of the other parameters in the body frame. According to physics, if we know the acceleration, we can then predict the state using the equation

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix}_t^{(b)} = R\left(\begin{pmatrix} \dot{\phi} \\ \dot{\vartheta} \\ \dot{\omega} \end{pmatrix}_{t-1}^{(b)}\right) * \left(\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix}_{t-1}^{(b)} + \Delta t \begin{pmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{pmatrix}_{t-1}^{(b)}\right)$$

where R is the rotation matrix that maps the body coordinate frame at time t to the body coordinate frame at time $t+1$ (and depends only on the angular velocity at time t). We have a similar equation for the angular velocities. Thus we have a non-linear model (even though the accelerations are predicted using linear equations), which means we cannot use standard frequency analysis methods and instead rely on quadratic optimization.

We performed the following data fitting:

$$\begin{aligned} \dot{u}_t - (g_u)_t &= C_u \cdot (u_t\ 1)^T \\ \dot{v}_t - (g_v)_t &= C_v \cdot (v_t\ 1)^T \\ \dot{w}_t - (g_w)_t &= C_w \cdot (w_t\ i_{\text{col}}\ 1)^T \end{aligned}$$

and

$$\begin{aligned}\dot{p}_t &= C_p \cdot (p_t \; i_{\text{lat}} \; 1)^T \\ \dot{q}_t &= C_q \cdot (q_t \; i_{\text{lon}} \; 1)^T \\ \dot{r}_t &= C_r \cdot (r_t \; i_{\text{cyc}} \; 1)^T\end{aligned}$$

We express the acceleration in the body coordinates at time t, from which it is easy to find the acceleration $(\ddot{x}\; \ddot{y}\; \ddot{z})^T$ in the reference earth frame by a rotation and compute the velocity at time $t+1$. Note that we substract the effect of the gravitation in our model. Indeed, gravitation has such a strong (and known) influence that it can be left out in the model fitting and integrated back in the simulation.

Our simulation showed us that the blade speed had only a small influence on the other parameters. The blade speed (expressed in rpm) follows the following model:

$$\dot{v_{\text{b}}} = C_{\text{b}} \cdot \left(v_{\text{b}} \; \sqrt{u^2 + v^2} \; w \; i_{\text{col}} \sqrt{i_{\text{lat}}^2 + i_{\text{lon}}^2} \right)^T$$

C. Data collection

The helicopter used in this experiment is equipped with a MaxStream IMU (instrumentation measurement unit) that gives the angular accelerations and the attitude of the helicopter in the body frame at a rate of 15Hz. The position was measured either with a GPS or with a setup of 2 cameras on the ground that tracked the helicopter. The GPS was eventually disabled because its weight would penalize the performance of the helicopter and furthermore the signal would be unreliable close to the ground (due to obstructions by ground objects).

The blade speed is captured by a magnetic sensor positioned on the main gear of the helicopter. In order to relay this data to the ground, we programmed a microprocessor that digitizes the information on-board. This processor transmits the rpms at a rate of about 25Hz though a radio serial link. The blade speed we get from this setup is accurate up the millisecond (for a practical working limit of 20 milliseconds) and does not need to be filtered.

The camera setup used for estimating the position is unfortunately not robut enough close to the ground. The helicopter is identified by centroid separation from the background. However, because of the way the cameras adapt to the luminosity, this setup works well when pointing either on the sky or on the ground, but not when both backgrounds are present. As a first workaround, we mounted an ultrasound sonar that was shown to provide accurate measurements of the distance to the ground for distances of up to 5 meters (at a sampling rate of 10Hz). This sonar shares the microprocessor unit used for the blade counter to digitize and relay the information packets.

D. Controller

The controller was automatically generated using Linear Quadratic Regulatic, thanks to the fact that our estimates of the accelerations are expressed as linear combinations (of non-linear features). Note that were are evaluating a steady-state model in which we do not care about the position (we only want to regulate the blade speed, the velocity and the attitude in a first time). By looking at the flight records, we identified an constant value for the collective pitch:

$$\begin{aligned} i_{\text{lon}} &= 0 \\ i_{\text{lat}} &= 0 \\ i_{\text{cyc}} &= 0 \\ i_{\text{col}} &= 0.11 \end{aligned}$$

(We maintain a positive collective pitch throughout the descent) The target model was the following

$$\begin{aligned} \dot{z}_{\text{target}} &= -5 \text{ m/s} \\ \dot{x}_{\text{target}} &= 10 \text{ m/s} \\ v_{\text{blade}} &= 1200 \text{ rpm} \\ (\phi, \theta, \psi) &= (0, 0, 0) \end{aligned}$$

i.e., we want to have a constant velocity, blade speed and a level angle during descent. The blade speed and the velocities were identified by by looking at the data of previous flights.

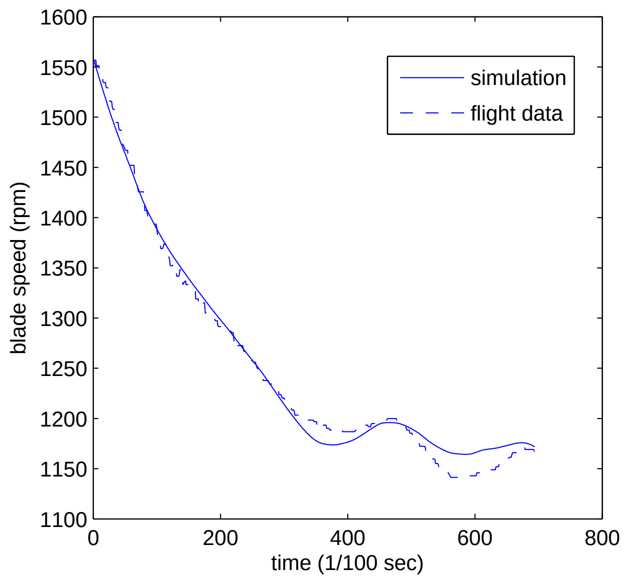
Note that we are building a model for the steady state. Since of course the helicopter is not at the beginning in this steady state, the penalities introduced in the LQR model had to be adjusted with respect to this fact. In particular, the blade speed target had to be severely penalized. Indeed, this parameter is already an order of magnitude above the other parameters and the helicopter would try to gain some forward speed at all cost in order to build a good blade speed. This induced in the first second of autorotation a dangerous free fall-like behaviour. Note also we are not specifying a position target right now. This is a point to consider in further research.

E. Results

The following parameters were used to modelize our UAV helicopter:

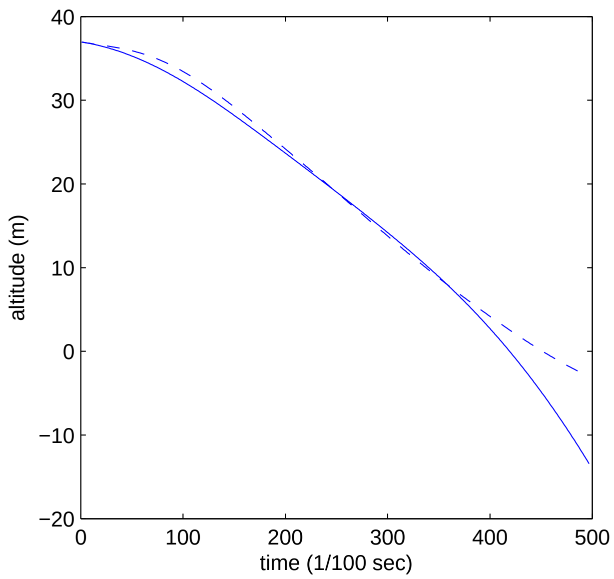
$$\begin{aligned} C_p &= (-0.4904 \;\; -6.1590 \;\; 25.2748) \\ C_q &= (-0.2463 \;\; -2.0976 \;\; -16.5976) \\ C_r &= (0.5215 \;\; -5.4296 \;\; -16.5976) \\ C_u &= (-0.0378) \\ C_v &= (-0.6127 \;\; -0.1643) \\ C_w &= (3.4978 \;\; -0.7368 \;\; -31.3544) \\ C_{\text{b}} &= (106.85 \;\; -0.22 \;\; -68.53 \;\; 22.79 \;\; 2.11 \;\; -6.09) \end{aligned}$$

This model yielded good results both in simulation and in practice. The rpm model was very accurate as shows graph (1) wich represents the simulated value of the rpms (single value simulation) against the actual one during one flight. As one can see, the values match closely:

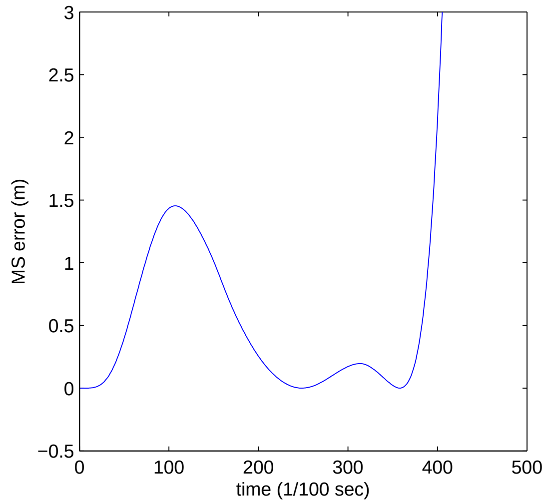


The first thing to notice is the range covered by the rpm throughout the flight, wich further justifies our choice to model this feature. However, it was not found to influence our model for the accelerations in a significant way. It may be interesting to further refine the analysis of data to find a direct influence of the rpms on the acceleration (as it exists in theory).

We validated our model with a full open-loop simulator that uses the physical model described by equation (1). The simulator yielded good results in the steady part of the autorotation, especially for the critical parameters such as the altitude:



The dotted plot is the actual flight data whereas the plain plot is the simulated (open-loop) altitude. Also shown is the mean-square error usually obtained for this simulation:



Note how the the two curves drift apart at the end of the flight. Not only does the pilot add some power and reengage the engine, but also the ground effect becomes increasingly important. At low altitudes, the air flow is blocked by the ground, and this effect becomes a major contribution to the sustentation of the aircraft. Indeed, our pilot could sustain the helicopter in hovering flight for a few seconds and with the engine cut off by using the ground effect.

The best test we can submit our controller to is the control of the real helicopter. The UAV was programmed to begin in stationary flight at altitudes ranging from 20 meters to 40 meters (in order to stay inside the vision system). Then the autopilot would disengage the engine from the blade gears and reach a new steady state in autorotation. When the helicopter would go out of the vision system, the pilot would take back the commands and reengage the engine. This experiment was repeated successfully 6 times with slight variations each time on the initial attitude and the target in order to have a practical idea of the robustness of the controller. After the controller penalty parameters had been properly adjusted, the helicopter would fly successfully (see the videos).

F. Conclusion and future plans

In the course of this project, we were able to accurately predict the state of the helicopter during its descent in autorotation by building a model from flight data. This model showed good accuracy for the steady-state part of the autorotation flight and enabled us to derive a working controller from it. This controller was successfully tested on the real UAV helicopter and demonstrated great accuracy and stability once the control targets and penalties had been refined.

The practical results we obtained show that controlling a helicopter in steady-state autoroation is not very different from the techniques employed for normal flight. Our future work will concentrate on the landing itself wich presents different challenges. Since we can precisely guide the helicopter before te landing stage, an easy solution would be to always guide it to the same configuration and apply a defined sequence of controls from that point. However, it will be interesting to see how the helicopter can be controlled up until the landing from different states. We have already begun to lay the ground for this work with the sonar implementation.

G. Acknowledgements

My first thanks go to Professor Ng for the idea of the project and his kind words of encouragements at some critical moments. I also want to express all my gratitude to Pieter Abbeel for his mentorship thoughout this project, and his profound insight of dynamics and control. Some technical details would have stopped short this project without the expertise of Morgan Quigley and the coding skills of Adam Coates. Finally, I was able to work on excellent data thanks to Garett, our pilot, who recovered the helicopter more than once from perilous situations.

[1] Modeling vehicular dynamics with application to modeling helicopters. *NIPS*, 2005.
[2] G.D. Padfield. *Helicopter flight dynamics.* 2007.
[3] R.R. Padfield. *Learning to fly helicopters.* 1992.