Applications in Robot Helicopter Acrobatics

Sergej Hardock
Department of Computer Science
Technical University Darmstadt
sergej.hardock@gmx.de

Abstract

Helicopters performing autonomously different tasks exist already for more than twenty years. New learning methodologies take each time the complexity of such tasks at a new higher level. However, the helicopters were always flying in general flight regimes, like simple directional flight, hovering, etc. Highly challenging manoeuvres were always prerogative of very experienced pilots, and have been for a long time unattainable for an autonomous helicopter. The main reason for that fact is the complexity of the helicopter dynamics, and therefore inability to cover all aerodynamic effects in a control model. In stationary flight regimes the influence of such unmodelled effects is minimal. However, during extreme manoeuvres they affect the flight significantly, which makes the simplified model more fallible. The situation has fundamentally changed in recent years with applying for controlling tasks reinforcement-learning methods. This approach enables to find an accurate dynamics model and a trajectory, which are further used to determine the optimal control policy.

1 Introduction

Autonomous helicopter flight is far away from to be a new task, but is still one of the most complicated in aircraft robotics. There are to date lots of solutions that enable a helicopter to perform certain tasks in an autonomous mode. The most significant examples were demonstrated in the International Aerial Robotics Competition (IARC)[1] in the time period from 1991 till now. Altogether were performed six different competitions, each time with a different and more complicated task (mission). During this 10 years many research groups from different universities in USA, Canada, Germany, India, etc. have been taken part in the project, and there were always at least one team, whose helicopter has done the specified mission perfectly. In a nutshell, the tasks which helicopters have needed to perform autonomously were:

- 1st mission: move small object from one place to another;
- 2nd mission: identify among others the toxic waste drum and take a sample from this drum;
- 3rd mission: rescue works in a disaster area (fire region); for example, among others it was required to identify dead and live persons;
- 4th mission: find specified in a task building, fly into one of its windows, take photos inside and return back;
- 5th mission: orientation inside unknown building without any feedback and control signals from outside;
- 6th mission (the current one): fly into a building without being noticed by security, find a certain room and a USB flash drive in this room, replace it with another and return back again avoiding security.

[1] http://iarc.angel-strike.com/

Despite of these really complex and challenging tasks, the helicopters in all such experiments were always flying in general stationary regimes. The latter ones include taking-off, hovering, directional flight, landing and some limited number of simple aerobatics (e.g., an axial roll). On the other hand, performing extreme manoeuvres, like split-S, double loops, chaos, rolls, flips, tic-toc, auto-rotation landing (see Figure 1), were possible only for experienced pilots, and for long time could not be successfully replicated in an autonomous mode. There are several reasons for this; each one corresponds to a certain problem in a flight control. At first, the helicopter dynamics are very complex, and even though, there are some unknown or mathematically unmodelled aerodynamic issues. Therefore, dynamics models used for controlling tasks are simplified to a certain degree. This is mostly not a problem for stationary flight regimes; however, the more complex they are, the more costly errors are produced by such model simplifications. Second, specifying a good trajectory for a manoeuvre is also quite a difficult task. Nevertheless, even a good dynamics model and a known intended trajectory solve the control problem only half. The reason for this is that the helicopter especially during extreme aerobatics is very unstable, and thus without additional control would soon significantly deviate from the planned trajectory.

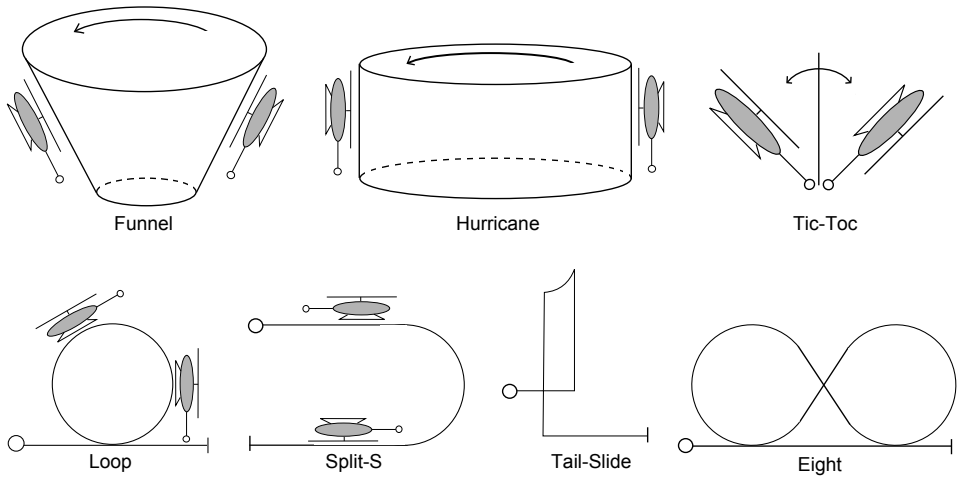

Figure 1: Extreme helicopter manoeuvres.

All these tree main complications were successfully solved at Stanford University by Ng et al. [1], [2], [3] . They first have achieved autonomous helicopter flight, which was consisting of more than dozen of most challenging manoeuvres. Aerobatics were performed in almost ideally form and often were even better than the pilot-driven demonstrations. The schema of their approach is shown in Figure 2. In a nutshell, the algorithm consists of the following steps:

1. build a simplified helicopter dynamics model containing unknown parameters;
2. collect some flight data from a real pilot-driven helicopter;
3. use flight data to find unknown parameters of the model;
4. collect data from several pilot-driven demonstrations, which have the same flight program that should be replicated in autonomous mode;
5. align data of all demonstrations in time;
6. based on time-aligned data refine the dynamics model;
7. use found intended trajectory and refined dynamics model to build a close-loop control;
8. if results are not sophisticated, repeat the algorithm from step 3, but use as test data, the data from autonomous flight.

The intent and main approaches of each step are discussed in more detail in the following sections. However, roughly the whole learning algorithm could be seen as a nested reinforcement learning (RL) approach [4]. At first, via a model-based RL the parameters of the model are determined, i.e., the dynamic model and the trajectory, which in terms of RL are seen as a transition function and

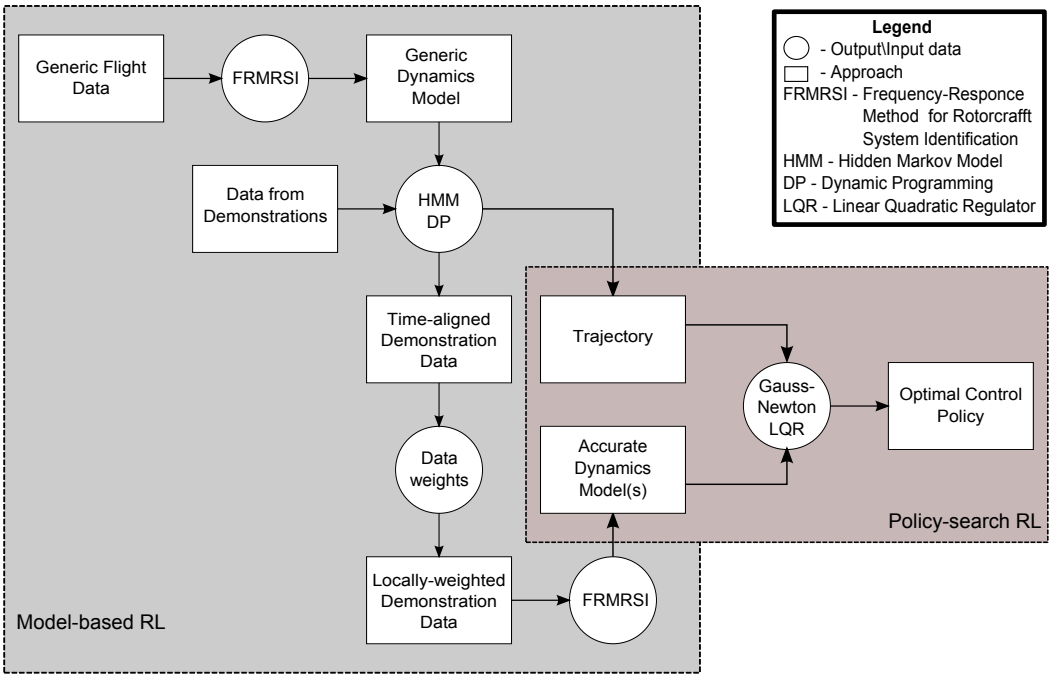

Figure 2: General schema of apprenticeship-learning approach.

a reward function, respectively. Further this model is used in policy-search RL in order to find the optimal control policy.

The rest of the paper is organised as follows. Helicopter basics and an approach of obtaining the dynamics model from the flight data, as well as possible inaccuracies of the latter are described in Section 2. An apprenticeship learning algorithm for finding intended trajectory from multiple pilot-driven demonstrations is explained in Section 3. The method of refining the dynamics model by its adjustment to the intended trajectory is shown in Section 4. In Section 5 is described the approach of finding in online regime an optimal control policy. Some details about experimental procedure are shown in Section 6. In Section 7 briefly discussed some other current research projects and works in sphere of autonomous helicopter flight; and Section 8 finalizes the paper with some conclusions.

2 Helicopter dynamics model

In fixed-wing aircraft, e.g., airplanes, the main elements of the aerodynamics are supplied by the separate physical devices. The thrust, which is responsible for the forward moving, is caused by the jet engine or propeller. Lift - the compensating force for the weight, which enables hovering of the plane or staying on a certain height, is supplied by the wings. Last not least the direction of the flight is mainly maintained by the vertical tail wing, as well as by horizontal wings. The control of the flight refers, respectively, to controlling the properties of these devices.

In contrast to the airplanes, the trust, lift and direction of the helicopter flight are supplied, and therefore controlled only by one physical device, the main rotor. This fact cause a lot of complex aerodynamics effects, relationships and influences between them, which in many cases are very difficult to describe mathematically. Thus, to model the helicopter dynamics is one of the most complicated tasks in the aircraft. On the other hand, the model, which covers all these aspects, would be too complex to work with, especially for the autonomous flight task, where the time constraints are playing significant role. Hence, it seems to be reasonable to apply the simplified dynamics model.

The model of the helicopter describes the relationships between the state and control elements. The helicopter has four control elements: three of them control the main rotor (collective, latitudinal and longitudinal pitches) and one - the tail rotor (tail-rotor pitch) (see Figure 3).

Primary purpose of the tail rotor is to compensate the torque of the main rotor (see Figure 4), which causes the helicopter to rotate in the direction opposite to rotor. When the trust from the tail rotor is equal to the torque, the helicopter body would not change its orientation during the flight. However, any deviation from this equation would result in turning the helicopter clockwise or counter clockwise depending on which force dominates. Hence, to control the position of the helicopter according to its vertical axe the trust of the tail rotor should be adjusted. This is one of the four control inputs (tail-rotor pitch) and done through changing the angle of attack of the blades of the tail rotor.

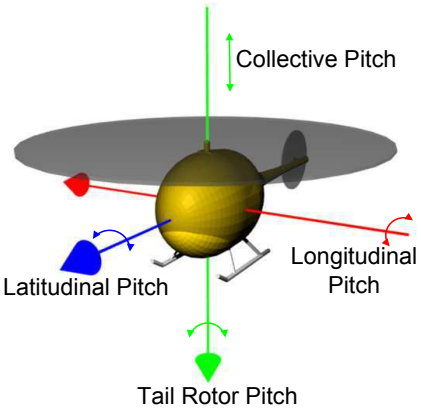

Figure 3: Helicopter control elements.

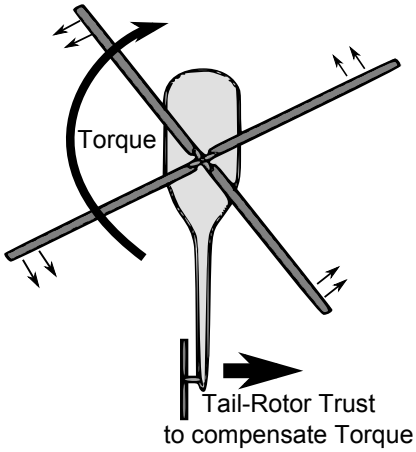

Figure 4: Torque and Tail-Rotor Trust.

Vertical rising, descent, hovering and directional flight are caused and controlled through the main rotor. The main rotor consists typically of two or more blades, which rotate with the roughly constant speed and produce the lift (cf. wings of airplane). The lift is affected by the angle of attack of each blade (see Figure 5). With increasing the angle of the blade, increased also the downward air pressure and further the lift, which results in bigger upward force for the helicopter.

If all blades have the same pitch angle over the whole rotation cycle, the air pressure would be evenly distributed, and therefore the helicopter could move vertically up or down, or hover. Such change of the blade's angle called the collective pitch control and is the second control input of the helicopter.

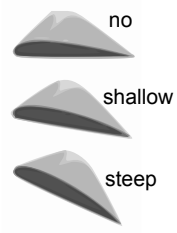

Figure 5: Angle of attack.

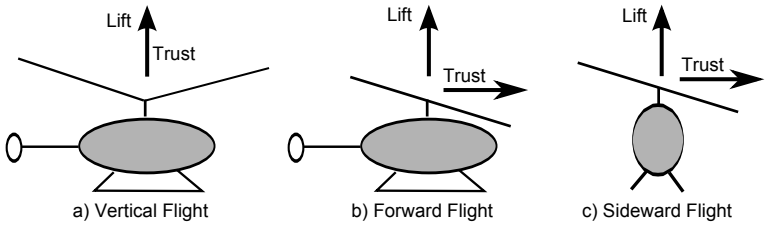

Figure 6: Rotation disk during directional flight.

Through the swash plate it is possible for blades to have different pitch angle at different points of rotation cycle. Thus, the air pressure under the blades changes during the rotation. Since the blades can also move up- and downward, uneven pressure causes so-called blade flapping. The achieved in that way tilted rotation disk (see Figure 6) changes the direction of the thrust, further enabling the flight of the helicopter forward, backward or sideways. Two control inputs that responsible for forward-backward and right-left flight are called longitudinal and latitudinal cyclic pitch controls, respectively.

The state of the helicopter at each point of time consists of its position, orientation, liner and angular velocities. The simplified dynamics model could be defined as following [1]:

$$
\begin{aligned}
\dot{u} &= vr - wq + Ax_u + g_x + w_u, \\
\dot{v} &= wp - ur + Ay_v + g_y + D_0 + w_v, \\
\dot{w} &= uq - vp + Az_w + g_z + C_4 u_4 + D_4 + w_w, \\
\dot{p} &= qr(I_{yy} - I_{zz})/I_{xx} + B_x p + C_1 u_1 + D_1 + w_p, \\
\dot{q} &= pr(I_{zz} - I_{xx})/I_{yy} + B_y q + C_2 u_2 + D_2 + w_q, \\
\dot{r} &= pq(I_{xx} - I_{yy})/I_{zz} + B_z r + C_3 u_3 + D_3 + w_r.
\end{aligned} \tag{1}
$$

Here: (u, v, w) – linear velocity; $(\dot{u}, \dot{v}, \dot{w})$ – linear acceleration; (p, q, r) – angular velocity; $(\dot{p}, \dot{q}, \dot{r})$ – angular acceleration; (g_x, g_y, g_z) – gravity; u_1, u_2, u_3, u_4 - latitudinal, longitudinal, tail-rotor and collective pitch controls, respectively; w_u, w_v, w_w, w_p, w_q, w_r – represent the noise for respective accelerations (result of simplified model); A_x, A_y, A_z, B_x, B_y, B_z, $C_1 - C_4$, $D_0 - D_4$ – unknown coefficients.

For simplicity the velocities in this model are determined in so-called body frame, which is fixed with the helicopter and placed in its centre of gravity. Thus, with changing the orientation of helicopter, the velocities would also change even without additional forces and moments applied to the helicopter. To reflect such changes the terms $(vr - vq)$, $(wp - ur)$, $(uq - vp)$, $qr(I_{yy} - I_{zz})/I_{xx}$, $pr(I_{zz} - I_{xx})/I_{yy}$, $pq(I_{xx} - I_{yy})/I_{zz}$ are added to the corresponding accelerations.

One possible way to determine unknown parameters of the model from the test data is via linear regression. However, the typical approach in the rotational aircraft is to use the frequency-domain system identification method [5]. Originally proposed in 1987 as the dissertation of Mark B. Tischler, this method is used nowadays by U.S. Army and NASA as the standard for identification model structures and unknown model characteristics, and therefore for analysing and predicting the behaviour of vehicles. In a nutshell, there are three main steps according to this approach.

- First, the pilot-driven flight data should be collected, verified and filtered. The flight data is usually a set of state elements and control inputs received with some frequency from vehicle's sensors. Some state elements are calculated, i.e., estimated from others, since not all state elements could be effectively measured. Further, using Kalman filter the log data is verified for compatibility; and errors caused by sensors and scale factors are eliminated.
- On the second step, the flight data is transformed from time to frequency domain. Here at the beginning, using techniques of multi-variable spectral analysis, multi-input/multi-output (MIMO) frequency-responses are obtained. However, the real point of interest is a set of frequency-responses for all possible input-output pairs, i.e., single-input/single-output (SISO) frequency-responses. The latter ones are obtained by applying the Chirp-Z transform and techniques of spectral averaging using composite windows.
- Having the frequency-responses for each pair, the next step is to obtain and verify the transfer-function models for each pair. The verification is done by applying the determined transfer-function to the flight data, which was not used in previous identification steps. By comparing the received time history with the real one, the accuracy of transfer-function model is checked according to the system requirements, and if needed another transfer-function is selected.
- At the end, frequency-responses and appropriate transfer-functions are used to determine the unknown model characteristics, e.g., the unknown coefficients A, B, C, D of the helicopter dynamics model.

2.1 Model simplifications and errors

As already mentioned, the applied dynamics model is simplified in order to make easier further computations. Main simplifications are considered by the model explained in the following.

Each blade depending on the certain rotor system has different number of degrees of freedom (DOF). The most widely used today are articulated and semi-rigid rotor types. The former provides the blade with three DOF: up/down (flapping), right/left in the horizontal plane and around its longitudinal axis (pitch angle, i.e., angle of attack (see Figure 5)). In the latter version the movements of blade

in the horizontal plane are eliminated. The dynamics model, however, explicitly covers only one movement of the blade – the pitch angle, since in most cases it is the only one, which is directly controlled. The flapping effects are not considered by the model.

On the other hand, the current model assumes that the helicopter is the rigid body; however, especially for blades, it is not the case. The airflow could significantly influence the blade deformation, which further results in unmodelled aerodynamics effects.

Among other significant simplifications are not considered effects of:

- airflow around the helicopter;
- additional lift during the forward flight;
- inter-influence of control inputs;
- deviations in keeping the constant rotor speed;
- deviations of helicopter weight during the flight (e.g., due to the volume of fuel);
- non-linear correlations between drag forces and velocities.

Naturally, that these unmodelled issues cause inaccuracies and errors. The weightiness of such errors for the model is the bigger, the more complicated flight regimes the helicopter should perform. Thus, for some basic regimes, like hovering or directional flight, the differences between the predicted and actual helicopter states are quite small. In contrast, when the helicopter follows complex trajectories performing aerobatic manoeuvres (see Figure 1), lots of hidden and unmodelled aerodynamics effects influence the flight more extremely. This would further result in serious controlling errors, i.e., the real trajectory followed by the helicopter would significantly differ from the predicted/modelled one. The possible solution to this problem is explained in Section 4.

3 Trajectory learning

In an autonomous flight, the helicopter should follow some specific trajectory. However, the term trajectory here stands not only for the set of the desired geographical positions of the helicopter. Instead it defines also at each point of time the helicopter state (e.g., orientation, velocities) and control elements (e.g., cyclic and collective pitch controls). Therefore, specifying the valid trajectories by hand is very challenging and complex task, especially for aerobatic manoeuvres. First, it is quite difficult to predict the trajectory states at all, having only the image of how the manoeuvre should look like. Second, states and control inputs should be conformed to the dynamics model of the helicopter, which makes the specification task more complicate and sometimes even impossible. The solution to these problems could be in learning the trajectory from the real pilot-driven demonstrations.

Even for experienced pilots, it is quite difficult to perform complex aerobatic manoeuvres. Thus, due to errors some flight elements would be better in one demonstration, others in another. Performing and analysing dozens of pilot attempts in order to have the best trajectory is very expensive task in terms of time, work and money. One can think that calculating the average trajectory from the given set of demonstrations could be a good idea; however, it is not the case. There is no sense in simple averaging positions, velocities, etc. at each point of time, because each demonstration has its own time aligning. This means, that helicopter states and control inputs are, in most cases, unrelated at the same time in different flights .

Ng et al. [1] have proposed so-called apprenticeship learning algorithm for obtaining the optimal trajectory from demonstrations. The basic idea of that approach is to consider the pilot-driven demonstrations as the observations of some desired optimal trajectory, which is hidden. The task of finding the hidden trajectory is then solved using the hidden Markov models (HMM). However, before applying the maximum likelihood search, some important transformations should be applied. First, the available data and the one, which is looked for, should be appropriately modelled.

The trajectories are represented through the sequence of elements, where each element comprises the helicopter state and control inputs at a certain point of time:

$$\begin{aligned} \textit{intended trajectory:} \quad & z = \{z_t | t = 0, ..., T-1\}, \\ \textit{trajectories from demonstrations:} \quad & y = \{y_j^k | j = 0, \ldots, N_k - 1; k = 0, \ldots, M-1\}, \end{aligned}$$

where T, N_k – the lengths of trajectories; M – number of pilot-driven demonstrations. The length of the hidden trajectory T is unknown, and therefore should be selected experimentally. Herewith, as demonstrations are observations, the length of the searched trajectory preferred to be more or equal than the length of the longest demonstration. One possible variant could be choosing this length equal to the double arithmetical mean of the observed trajectories' lengths. Experiments have shown that such length effectively covers all time aligned differences, and therefore results in more accurate result.

Each demonstration's state y_j^k is an observation of some state $z_{\tau_j^k}$ in the intended trajectory:

$$y_j^k = z_{\tau_j^k} + w_j, \tag{2}$$

where w_j – Gaussian noise, which model possible pilot errors during each demonstration. τ_j^k is the time aligned index, that matches observed and desired (unknown) states. These indices

$$\tau = \{\tau_j^k | j = 0, \ldots, N_k - 1; k = 0, \ldots, M - 1\}$$

are also unknown and should be determined in order to find the intended trajectory. Additionally, the intended trajectory should satisfy the dynamics model of the helicopter (see Section 2). To meet this condition, the relation between current and next trajectory states is defined through the helicopter dynamics:

$$z_{\tau+1} = f(z_\tau) + w_\tau, \tag{3}$$

where z_τ, $z_{\tau+1}$ – current and next trajectory states; $f()$ – helicopter dynamics model; w_τ - Gaussian noise, which covers errors and inaccuracies in the dynamics model.

Graphically such data model is represented in Figure 7. So far the time indexes are unknown, unknown is also the correlation between observed y_j^k and hidden states z_t. Thus, for example, the state y_1^k of the k-th pilot-driven trajectory could be an observation of states z_1, z_2, z_3 of the intended trajectory; the number of possible dependencies increases with the time. Solving the system under such circumstances would result in enormous high complexity with requiring a lot of time and resources costs. One possible workaround to the model complexity is to use data estimation in

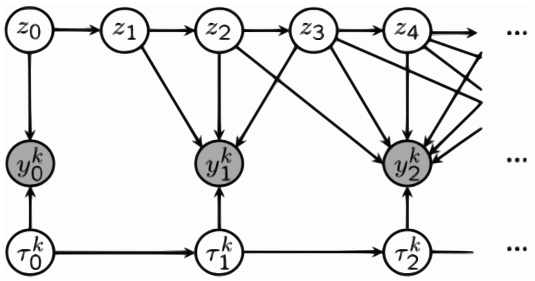

Figure 7: Data model for the trajectory learning [1].

combination with the dynamic programming. According to this approach, on initial step the time indexes are estimated based on some expert preferences, thus the model is restricted to one-to-one relations between observed and hidden states. Such a model is a typical example of a hidden Markov model, and therefore standard expectation-maximization approaches (e.g., Baum-Welch) could be applied to find/learn unknown system parameters (e.g., covariance matrices of the Gaussian noise) and the intended trajectory. The received trajectory is the one that maximizes overall likelihood of the all observations under defined time indexes. However, as there is no guarantee that current guess of time indexes is optimal, the dynamic programming is used to estimate new values for t and alternate the search algorithm. The process stops, when there is a trajectory, for which the total likelihood of all observations is bigger than the acceptable accuracy threshold.

As already mentioned, the specification of the trajectory by hand results in significant costs and inaccuracies. However, the existed knowledge about the trajectory could significantly increase the efficiency and performance of the search algorithm. Among typical examples of such prior knowledge are: the helicopter during a certain manoeuvre should not change its orientation or position; or

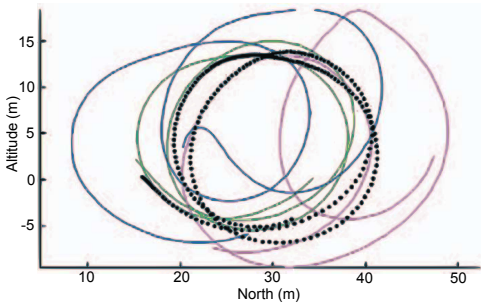

Figure 8: The example of observed and intended trajectories for the "double loop" manoeuvre [1].

at some points it should have zero vertical part of the linear velocity, etc. This knowledge is integrated in the system as a separate observation, which defines at certain points of time prior-known features of the helicopter state or control inputs. Included in the maximization likelihood search, this data would positive influence on the accuracy of the intended trajectory.

The example of one intended trajectory, which was identified using the described trajectory learning approach, is shown in Figure 8. Here for simplicity are presented only 2D positional parts of four trajectories – three observed (lines) and one intended (dots). The modelled manoeuvre called "double loop", and as it can be seen, the found trajectory describes almost the ideal loops.

4 Refining the helicopter dynamics model

The applied so far global dynamics model of the helicopter is many places simplified in order to avoid high complexity, and therefore computation costs. Directly unmodelled aerodynamic effects are implicitly covered by the model parameters, which are obtained from the flight data using procedure described in Section 2. However, for extreme aerobatic manoeuvres, the influence of these implicitly modelled aspects is much more significant than in stationary regimes. Consequently, the learned from general flight data parameters would inaccurately correspond to the real helicopter behaviour, which further results in control errors.

Interesting to note, that such errors depend on a specific manoeuvre, and especially on time during the manoeuvre. This means, that by repeating same aerobatics the model would produce almost the same errors on certain aligned (not absolute!) time points. The fact could be explained as follows: the errors are caused by unusual aerodynamics effects; the latter ones are, naturally, common for certain helicopter state and control inputs, thus similar effects produce similar errors.

The model inaccuracies give the motive for the refining the helicopter dynamics; and the character of these inaccuracies (i.e., stability of errors) provide the effective approach for such refining. The basic idea of improving the dynamics model is to provide instead of one general model, which should cover all possible aerodynamics effects, the set of dynamics model $F = \{f_t | t = 0, ..., T-1\}$. Each of such models would be specific for a concrete time point during the manoeuvre. This approach concentrates on estimating the model based on relevant flight data, and therefore unrelated effects would be eliminated.

The test data for the model refining is prepared on the previous step, i.e., as an intermediate result during the process of finding the intended trajectory, all observed demonstrations have been time-aligned. Such aligning enables to weight the flight data for a specific dynamics model based on time index. The weights of data points could be, for example, defined by:

$$W(t') = \exp\left(-(t–t')^2/\sigma^2\right), \tag{4}$$

where $W(t')$ is the weight of the data point at aligned time t' for the dynamics model f_t calculated for time t; σ is the configurable parameter to adjust the data relevance. Thus, the bigger distance between data point and time, for which model is estimated, the smaller is the weight, i.e., relevance and influence of this data. Obtained in such way models would with high accuracy cover both explicit and implicit modelled aerodynamic effects specific for a concrete point of time in the intended trajectory. One might think that since there are more accurate dynamics models, it would be reason-

able to recalculate the intended trajectory. However, the experiments [1] have shown that already obtained level of precision is fully acceptable.

5 Autonomous flight control

The data obtained on previous stages consists of 1) desired helicopter trajectory, and 2) accurate helicopter dynamics models for each time step of the trajectory. The last, however, definitely not the least step of the autonomous helicopter flight is to perform an appropriate control. The result of such control is a sequence of actions, also called control policy, applied by a helicopter during the autonomous flight. Found intended trajectory already contains the control inputs, which correspond to the optimal helicopter states. Therefore, one might think, that directly performing these actions would result in following by helicopter the desired trajectory. The main reason, which interferes with this idea, is that the helicopter is very unstable in an open-loop control, i.e. by directly performing the control without feedback channel for corrections. The perturbations, and consequently trajectory deviations are mainly caused by unmodelled aerodynamics effects (see Section 2.1), and by some external factors (e.g., wind). The typical solution for such unstable systems is to use a close-loop control via feedbacks. In terms of autonomous helicopter flight, that means the optimal control policy should be adjusted at runtime, i.e., during the flight, based on actual helicopter states. Thus, the "optimal" trajectory of states and controls given by the trajectory-learning method (see Section 3) could be seen as just a "good" trajectory to optimize around. In a nutshell, the policy search algorithm, described below, at each time step analyses together current helicopter state, previous taken action, desired trajectory and helicopter dynamics model to find the next optimal action, which would minimize trajectory deviations.

For linear models such trajectory following task usually solved via reinforcement learning by using linear quadratic regulator (LQR), a special case of Markov decision process (MDP). The general finite-horizon MDP is defined using following 5-tuple:

$$\{S, A, \{Psa\}, T, R\},$$

where S - set of possible states; A - set of possible actions; $\{Psa\}$ – state transition probabilities (e.g., the possibilities of arriving at state s' after performing some action a at state s; T – time horizon, i.e., which time period is analysed; R – reward function, which describes expected rewards of performing actions and transforming the system to another state. LQR concretizes the above model by the following assumptions:

1. state and action spaces are multi-dimensional:

   $$S \in R^n; A \in R^d;$$

2. state transitions are represented via linear ("L" from LQR) time-varying model (e.g., obtained from helicopter dynamics model via linear approximation):

   $$\{Psa\} : s_{t+1} = B_t s_t + C_t a_t + w_t, \tag{5}$$

   where $B_t \in R^{n \times n}$; $C_t \in R^{n \times d}$; w_t – Gaussian noise;

3. reward function is quadratic ("Q" from LQR) and negative ($R \leq 0$):

   $$R_t(s_t, a_t) = -s_t^T D_t s_t - a_t^T F_t a_t, \tag{6}$$

   where D_t, F_t – positive semi-defined matrices.

Matrices B_t, C_t, D_t, F_t change over time and assumed to be known in advance. In order to solve the task of trajectory following the states and actions in the reward functions are defined as deviations from desired values, i.e., $(s_t - s_t^*)$ and $(a_t - a_t^*)$; therefore the bigger errors, the more system is penalized by the negative reward value. The value function V_t represents the total expected system reward at each time step t as a sum of current reward R_t from performing action a_t at state s_t and all other expected rewards for remaining time:

$$V_t(s) = E[R_t(s_t, a_t) + R_{t+1}(s_{t+1}, a_{t+1}) + ... + R_T(s_T, a_T)]. \tag{7}$$

The policy π is defined as a mapping from states to actions, $\pi : S \rightarrow A$. The task is to find such optimal policy π^*, which would result at each step in maximal total expected system reward, i.e., $V_t^*(s)$:

$$\pi^* = \{\pi_t^*(s)|t = 0, ..., T; \pi_t^*(s) = \arg\max_a E[R_t(s_t, a_t) + ... + R_T(s_T, a_T)]\}. \tag{8}$$

$V_t^*(s)$ and $\pi_t^*(s)$ could be recursively represented as follows:

$$\begin{aligned} V_t^*(s) &= \max_a [R_t(s_t, a_t) + \sum Psa_t(s')V_{t+1}^*(s')], \\ \pi_t^*(s) &= \arg\max_a [R_t(s_t, a_t) + \sum Psa_t(s')V_{t+1}^*(s')], \end{aligned} \tag{9}$$

where the term $Psa_t(s')V_{t+1}^*(s')$ is the expected total reward from new state s' weighted by the probability, that the system turns to this state; thus summing these terms over all possible states gives the total system reward. Standard approach to find the sequence $V_t^*(s)$, and therefore the optimal policy $\pi_t^*(s)$ is to use dynamic programming. In the last time step $t = T$, since the system has no more state transitions, the term $\sum Psa_T(s_T)V_{T+1}^*(s') = 0$, and equations 9 have the form:

$$\begin{aligned} V_T^*(s) &= \max_a [R_T(s_T, a_T)] = \max_a [-s_T^T D_T s_T - a_T^T F_T a_T], \\ \pi_T^*(s) &= \arg\max_a [R_T(s_T, a_T)], \end{aligned} \tag{10}$$

which could be easily solved (e.g., $a_T = 0 \Rightarrow V_T^*(s_T) = -s_T^T Q_T s_T$; $\pi_T^*(s_T) = 0$). Then using dynamic programming $V_T^*(s)$ is recursively propagated backwards (using discrete time Riccati equation) to find all $V_t^*(s)$ and $\pi_t^*(s)$:

$$\begin{aligned} V_T^*(s) \rightarrow V_{T-1}^*(s) \quad &\rightarrow V_{T-2}^*(s) \rightarrow V_{T-3}^*(s) \quad \rightarrow ... \rightarrow V_0^*(s), \\ \pi_T^*(s) \rightarrow \pi_{T-1}^*(s) \quad &\rightarrow \pi_{T-2}^*(s) \rightarrow \pi_{T-3}^*(s) \quad \rightarrow ... \rightarrow \pi_0^*(s). \end{aligned} \tag{11}$$

In order to apply LQR controller for non-linear system, which in reality helicopter dynamics is, the above algorithm should be appropriately modified. The modified approach, also called Gauss-Newton LQR, is an iterating process, which continues until the optimal control policy not obtained. Each iteration consists of two steps.

- First, having current policy (from intended trajectory or from previous iteration) we compute the linear approximation of the dynamics model (e.g. linear approximating of non-linear function using its derivative at some point).
- Thus the typical LQR problem is obtained, by solving which at the second step, we receive an optimal policy for the current model.

To refine the linear model approximation, and therefore to obtain a better "optimal" policy, the process is repeated using the policy from previous iteration (i.e., $current\_policy_i = optimal\_policy_{i-1}$ for approximation.

As the policy search is done online, there are quite limited time windows for the search algorithms to be performed. Because of this, in practice the time horizon of the LQR is selected to nearest future (e.g., 2 sec.) and number of search iterations is restricted (e.g., 3 iterations).

The quality of autonomous flight strongly depends on accuracy of the dynamics model, which is applied during the policy search. Thus if the observed helicopter flight is not satisfactory, the learning algorithm turns back to the defining the dynamics model (see Section 2). Model refining is done based on current flight data, obtained from autonomous flight; therefore recent aerodynamic effects could be modelled. The new policy search is further done using improved helicopter dynamics.

6 Experimental Procedures

The experimental results provided in [1] prove the efficiency of the described above reinforcement learning approach for autonomous helicopter flight. The tests were performed on the remote control helicopter XCell Tempest (see Figure 9). The helicopter has the following characteristics: weight ca. 5 kg, length 135 cm and the height 51 cm. Additionally, the XCell was equipped with sensors,

Figure 9: RC helicopter X-Cell Tempest.

like: camera, GPS receiver and sonar for measuring the position; accelerometers; tachometer, rate gyros and magnetometers for measuring the trust and speed of both rotors; and the radio transmitter which sends data from these sensors to the base. The base is a stationary PC, which receives the current state of the helicopter, runs the policy search algorithm and transmits encoding actions back to the helicopter. The price for the helicopter itself is about 1.000 Euro[2], and with the additional equipment it could even reach the point of 5.000 Euro (e.g. price for orientation sensor vary from 1.500 to 3.000 Euro[3]).

A typical experimental autonomous flight consisted of sequentially (without pauses) performed high-challenging manoeuvres (e.g., 12-15 manoeuvres), and has taken in a usual about two minutes. However, to perform such a flight, at first, it was needed about half an hour in order to collect the pilot-driving demonstrations of the same manoeuvres in the same sequence. The number of demonstrations was usually 10, but only five of them with the better performance were further used for learning. Second, the offline part of the reinforcement learning algorithm (intended trajectory and refining the dynamics model) was taking another half an hour. Therefore, about an hour was required altogether to be able performing with a high accuracy (sometimes even better than the pilot) a two-minute autonomous flight. The deviations in performing challenging manoeuvres, such as: split-S, tic-toc, flips, rolls, loops, chaos, etc. (see Figure 1), were only about a meter or less from the intended "optimal" trajectory.

A separate attention should be paid to, probably, the most difficult helicopter aerobatics, so-called auto-rotation landing. This manoeuvre usually can be performed only by very skilled pilots, who have already more than two thousands hours of flight time as experience. It is actually more an emergency manoeuvre, which enable save and successful landing of the helicopter in case of engine failure. The main idea is to rotate the blades of the main rotor (which is not more driven by engine) using the potential energy of the helicopter (i.e., $U = mgh$). By falling down, the air goes through the blades and forces them to rotate. When the speed of the rotor is enough, the pilot uses this accumulated energy (already kinematic) for producing the lift, which decreases the vertical down-oriented speed up to zero near the ground, and therefore enables safe landing. The general dynamics model of the helicopter described in Section 2 is actually not suited for the auto-rotation manoeuvre, since it assumes a constant main rotor speed, which is here not more the case. To overcome this problem, the general model should be enlarged with the following equation:

$$\dot{\Omega} = D_5 + C_5\Omega + E_5 u_4 + F_5\sqrt{u^2 + v^2} + G_5(u_1^2 + u_2^2) + w_\Omega, \tag{12}$$

where Ω is a speed of a main rotor; D_5, C_5, E_5, F_5, G_5 are unknown parameters, which identified using flight data. Apart from this modification the learning algorithm remains unchanged. For performing this manoeuvre the engine of the helicopter was being turned off. The control policy considering the current height tries at first, in so-called glide phase, to maximize the rotor speed during the descending. Then about 10 meters above the ground it switches into the flare phase, where takes place the slowdown of the helicopter. And at last, having zero (ideally) vertical velocity, the helicopter using remaining rotor speed corrects its orientation and performs a safe landing. As it is shown in [3], all experiments of auto-rotation flight were successful and never produce any damage of the machine. Therefore, this extreme challenging manoeuvre, which has never been done in autonomous control mode before, proves once more the efficiency of the proposed learning strategy.

[2] http://www.heliproz.com/products.asp?dept=41

[3] http://www.microstrain.com/inertial/sensors

7 Related Research Work

Autonomous unmanned aerial vehicles (e.g., helicopter, quadrotor, etc.) are not new, but are still a subject for many research groups in science, industry and military. The tasks, which could be executed by such aircraft robots, have changed from simple to very challenging and complex (e.g., rescue works, indoor flying, etc.); and sometimes even with results that outperform ones from experienced pilots. Some of current research projects and results in this sphere are described below.

Sensing, Unmanned, Autonomous Aerial Vehicles (SUAAVE)[4] is a project carried out by the University of Oxford. Its main focus is to create the swarm of autonomous helicopters, which would collaborate with each other in order to perform the specified task (e.g., environment exploration). The "team" of autonomous vehicles has significant advantages over single robot: higher level of robustness, performance, complexity of performed tasks, etc. On the other hand, synchronization between agents causes additional implementation problems.

The Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control (STARMAC)[5] is another example of research project (Stanford University, USA), which also specializes on autonomous cooperative behavior of multiple quadrotor helicopters (see Figure 10). Quadrotors comparing to standard helicopters are more stable during the flight, have better power-size ratio and are significantly cheaper. Applicability of autonomous quadrotors for different civil tasks is also research interest of the SkeyeCopter[6] project at Chemnitz University of Technology (Germany).

Figure 10: Quadrotor (STARMAC project).

A. Saxena et al. in [6], [7] described the main principles of indoor autonomous helicopter flight using a single onboard camera as the only one available sensor. They have combined different image processing techniques to build in a runtime the 3D map of environment, which allows safe flight in unknown, narrow indoor spaces. Similar results, however using multi-camera visual feedback have been also achieved by H. Oh et al. [8].

There are lots of other research groups, which concentrate either on outdoor or indoor autonomous helicopter flight. The differences in tasks and environments between these two types result in solving quite different design (e.g., size, power of robot) and control problems (e.g., level of flight accuracy; navigation, communication principles, etc.). However, in almost all such applications the helicopters are flying in general, stationary flight regimes. Extreme helicopter aerobatics performed in autonomous mode are, in opposite, less popular field among research community. One of the main reasons for this is, probably, the character of achieved results – more scientific than practical. Nevertheless, this statement is only explicit true, in reality solving such complex control problems at the edge of machine's capabilities, takes not only aircraft robotics, but the robotics in all at a new research level. Therefore, the work of A. Ng et al. [1], [2], [3], the main ideas of which were in a nutshell presented in this paper, has significant meaning for autonomous systems.

[4] http://www.cs.ox.ac.uk/projects/SUAAVE/

[5] http://hybrid.eecs.berkeley.edu/starmac/

[6] http://www.tu-chemnitz.de/etit/proaut/forschung/quadrocopter.html.en

8 Conclusions

The nested reinforcement learning approach enables the helicopter to perform autonomously very challenging aerobatic manoeuvres. The higher level model-based RL overcomes errors from simplified dynamics model and extracts the intended trajectory from multiple pilot-driven demonstrations. Then policy-search RL methods use this model in combination with online helicopter feedback to find the optimal control policy. All performed according to this control strategy manoeuvres have a high accuracy and in some cases are even better than the pilot-driven examples. Moreover, the major part of these aerobatics, e.g., tic-toc, chaos, auto-rotation landing, etc., has been never before done in an autonomous mode.

References

[1] Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *International Journal of Robotics Research*, 29(13):1608–1639, 2010.

[2] Andrew Y. Ng, H. Jin Kim, Michael I. Jordan, and Shankar Sastry. Autonomous helicopter flight via reinforcement learning. *Advances in Neural Information Processing Systems (NIPS)*, 16:363–372, 2004.

[3] Pieter Abbeel, Adam Coates, Timothy Hunter, and Andrew Y. Ng. Autonomous autorotation of an rc helicopter. In *Experimental Robotics, The Eleventh International Symposium, ISER 2008, July 13-16, 2008, Athens, Greece*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 385–394. Springer, 2008.

[4] Michael L. Littman. Model-based reinforcement learning. WWW Page, 2009. http://mlg.eng.cam.ac.uk/mlss09/mlss_slides/Littman_1.pdf [Accessed Jan. 23, 2012].

[5] Mark B. Tischler. *System identification methods for aircraft flight control development and validation*. National Aeronautics and Space Administration (NASA), Ames Research Center : US Army Aviation and Troop Command, 1995.

[6] Sai Prashanth Soundararaj, Arvind K. Sujeeth, and Ashutosh Saxena. Autonomous indoor helicopter flight using a single onboard camera. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5307–5314. IEEE Press, 2009.

[7] Cooper Bills, Joyce Chen, and Ashutosh Saxena. Autonomous mav flight in indoor environments using single image perspective cues. In *ICRA'11*, pages 5776–5783, 2011.

[8] Hyondong Oh, Dae-Yeon Won, Sung-Sik Huh, David Hyunchul Shim, Min-Jea Tahk, and Antonios Tsourdos. Indoor uav control using multi-camera visual feedback. *Intelligent and Robotic Systems*, 61:57–84, 2011.

[9] Andrew Y. Ng. Reinforcement Learning and Linear Quadratic Regulator. Video Lecture no. 18, 2008. http://www.virtualprofessors.com/machine-learning-stanford-cs-229-andrew-ng [Accessed Jan. 24, 2012].

[10] J. Andrew Bagnell and Jeff C. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *International Conference on Robotics and Automation*, pages 1615–1620. IEEE Press, 2001.

[11] Mathias Ramskov Garbus, Gustav Hogh, Rasmus Nielsen, Soren Lynge Pedersen, Jeppe Moller Holm, and Jan Vestergaard Knudsen. Trajectory tracking control of autonomous helicopter for terrain following. Master Thesis, AALBORG University, 2007. http://projekter.aau.dk/projekter/files/9578164/report.pdf [Accessed Jan. 24, 2012].