Open access • Proceedings Article • DOI:10.1109/ROBOT.2010.5509452

A simple learning strategy for high-speed quadrocopter multi-flips — Source link

Lupashin, Schöllig, Sherback, D'Andrea

Published on: 01 Jan 2010 - International Conference on Robotics and Automation

Topics: Acceleration and Simple (abstract algebra)

A simple learning strategy for high-speed quadrocopter multi-flips

A Simple Learning Strategy for High-Speed Quadrocopter Multi-Flips

Sergei Lupashin, Angela Schöllig, Michael Sherback, Raffaello D'Andrea

***Abstract*— We describe a simple and intuitive policy gradient method for improving parameterized quadrocopter multi-flips by combining iterative experiments with information from a first-principles model. We start by formulating an N-flip maneuver as a five-step primitive with five adjustable parameters. Optimization using a simple 2D vertical plane model of the quadrocopter yields an initial set of parameters and a corrective matrix. The maneuver is then repeatedly performed with the vehicle. At each iteration the final state error at the end of the primitive is used to update the maneuver parameters via a gradient adjustment. The method is developed in simulation and demonstrated at the ETH Zurich Flying Machine Arena testbed on quadrotor helicopters performing and improving on flips, double flips and triple flips.**

I. INTRODUCTION

Our objective is to use a simple model of a quadrotor in order to be able to perform and improve upon single, double and triple flips. In particular we desire a formulation of a flip primitive such that it is able to return the quadrotor exactly to its initial state, plus a $2\pi N$ change in rotation about one of its principal axes. In addition, we seek an approach that avoids complex online computations and does not require or attempt to enforce an a priori known feasible trajectory.

Miniature quadrotor helicopters in both indoor and outdoor environments are a popular and challenging autonomous aerial research platform. Several established quadrotor research groups exist, focusing both on indoor and outdoor applications and utilizing both home-built and off-the-shelf vehicles, for example [1], [2], [3]. Most research has so far been focused on near-hover mode operation using simplified linear models, with a variety of extensions such as autonomous long-term operation [1] and various controller design methodologies such as in [4], [3]. More recently several groups began exploring aggressive maneuvers such as fast translation [5] and outdoor backflips [6].

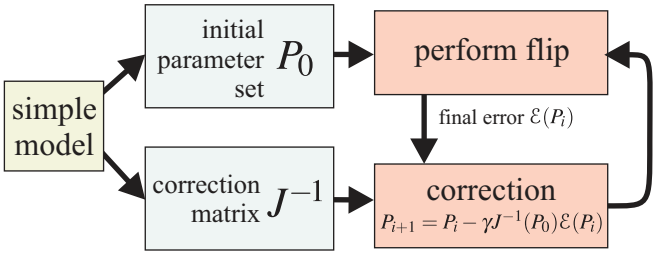

Fig. 1. Overview of the described approach

The authors are with the Institute for Dynamic Systems and Control (IDSC), ETH Zurich, Sonneggstr. 3, 8092 Zurich, Switzerland {sergeil,aschoellig,michaesh,rdandrea}@ethz.ch.

Associated video and relevant source code can be found online at http://www.idsc.ethz.ch/people/staff/lupashin-s

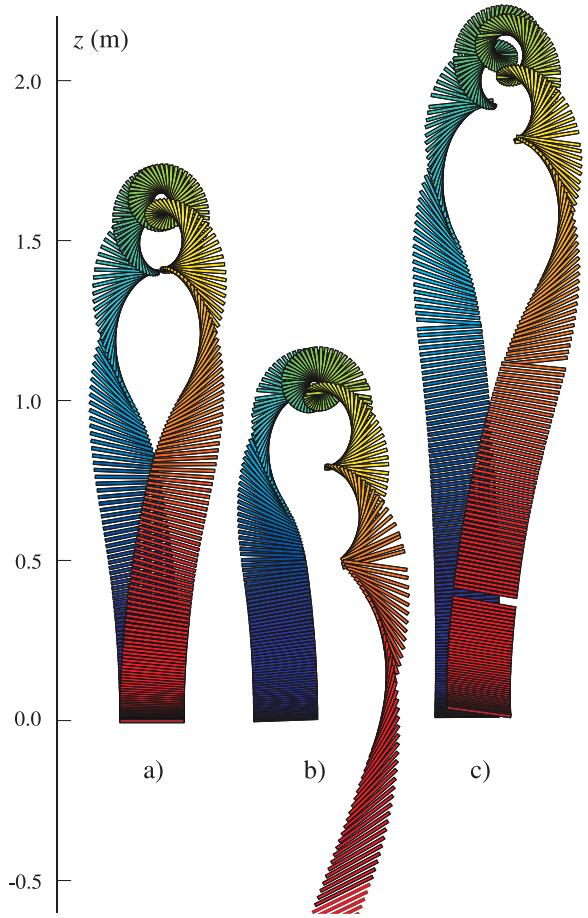

Fig. 2. Side view of quadrotor triple flips (5ms steps) with $\dot{\theta}_{max}$=1600°/s: **a)** simulated with a simple model and model-optimized parameter set P_0, **b)** on the real system with P_0, and **c)** with a corrected parameter set P_{69} after 69 learning iterations on the real system. Note that b) and c) are plots of actual experiments and so have some jumps where data is missing. Also note that b) is cut short as the actual z final state error is about -2m. The triple flip learning process is also shown in the accompanying video.

In parallel there exists a rich history of successful autonomous acrobatic helicopters such as [7] and [8]. In both projects a reference aerobatic trajectory was followed by an autonomous helicopter. In the latter project an innovative approach was taken where an algorithm extracted the reference trajectory from human-operated demonstrations and attempted to improve on autonomous performances of the said maneuvers.

However, designing or extracting reference aerobatic trajectories is not a straightforward task. Various aerodynamic effects such as vortex-ring-state, translational lift and blade flapping, among others become significant if not dominant at rotor descent and translation speeds comparable to the induced wind speed [2], [9]. To compound this problem, most of these effects have been studied only in steady state

Preprint submitted to 2010 IEEE International Conference on Robotics and Automation. Received September 15, 2009.

(i.e. descent at a constant rate with constant angle of attack, etc), while for fast aggressive aerodynamic maneuvers we are concerned with transients. Furthermore, even after decades of dedicated research on modeling helicopter aerodynamics, some of the rotor phenomena encountered in aerobatic maneuvers simply do not have analytical models, most well-known of these being the vortex ring/turbulent wake rotor operating mode [9]. It's also not practical for a human pilot to fly an acrobatic maneuver that depends on millisecond-accuracy control input switches.

There is a strong argument for using simple models with minimal parameters that need to be identified. For example, while much research recently has been focused on extremely precise modeling of propeller effects in quadrotors [10], the identification of all parameters requires devoted, carefully-designed experiments with an extremely careful treatment of measurement errors, unwanted aerodynamic effects, etc. On the other hand, it has been demonstrated that a very straightforward approach where only the most essential parameters are learned yields good hover performance, for example by [11].

The outline of the method used to design and improve on the flips is shown in Fig. 1. A result of running the method on triple flips is shown in Fig. 2. In overview, the approach described in this paper consists of the following: first we formulate the flip primitive as a five step maneuver using five free parameters. Then we use a numerical solver combined with a simple 2D model and a rough initial guess to find a parameter set that causes the model to reach the desired final state. We approximate the effect of parameter perturbations about this point by numerically calculating a Jacobian matrix. The inverse Jacobian is used to adjust the parameters in an iterative fashion based on the final state error produced by running the primitive on the actual quadrotor. A step size parameter can be used to provide robustness to model errors and noise.

The rest of this paper is organized as follows: we introduce the simplified 2D model of the quadrotor and define the vehicle's control envelope in Section II. We formulate the flip maneuver and specify the free parameters in Section III. We describe the method for correcting parameters from one experiment to the next in Section IV. Finally, we describe the experimental setup and the vehicle in Section V, go over experimental results in Section VI, and conclude the paper in Section VII.

II. SIMPLIFIED 2D QUADROTOR MODEL

We consider a simple 2D model of a quadrotor constrained to a vertical plane (Fig. 3). Roll and yaw dynamics are stabilized separately and are ignored. The model is:

$$M\ddot{z} = (F_a + F_b + F_c + F_d)\cos\theta - Mg \quad (1)$$

$$M\ddot{x} = (F_a + F_b + F_c + F_d)\sin\theta \quad (2)$$

$$I_{yy}\ddot{\theta} = L(F_a - F_b)\,, \quad (3)$$

where M is the mass of the vehicle, L is the distance from the center of mass of the vehicle to the propeller, I_{yy} is the moment of inertia about the out-of-plane principal axis, and F_a and F_b are the thrust forces produced by the two in-plane rotors. F_c and F_d are the thrust forces produced by each of the other two rotors, which are used to stabilize roll and yaw and are nominally set to the average of F_a and F_b,

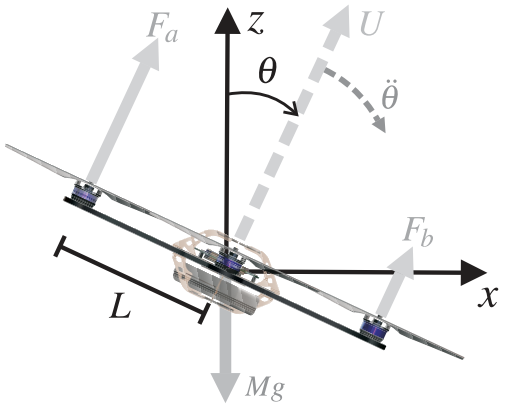

Fig. 3. Coordinate system and forces used in the 2D model in this paper.

$$F_c = F_d = \frac{F_a + F_b}{2}\,. \quad (4)$$

The combination of the propeller thrusts produces a collective acceleration U,

$$U = (F_a + F_b + F_c + F_d)/M = 2(F_a + F_b)/M\,. \quad (5)$$

Each propeller behaves approximately as a first-order system with significantly different up- and down- gains (i.e. we observe that the rotor slows down slower than speeding up). For each of the thrusts produced by rotors, $F_{[a,b,c,d]}$,

$$\dot{F} = \begin{cases} G_{up}(F_{des} - F) & \text{for } F_{des} \geq F \\ G_{down}(F_{des} - F) & \text{otherwise} \end{cases}\,, \quad (6)$$

where G_{down} is typically less than (slower) than G_{up}.

Each of the quadrotors accepts a collective acceleration command U_{des} and three desired body angle rates (in the 2D case, we consider just $\dot{\theta}_{des}$ and set the others to 0). The desired thrusts relevant to the flip are then specified by

$$F_{des,a} = MU_{des}/4 + I_{yy}\text{PI}(\dot{\theta}_{des} - \dot{\theta})/2L \quad (7)$$

$$F_{des,b} = MU_{des}/4 - I_{yy}\text{PI}(\dot{\theta}_{des} - \dot{\theta})/2L\,, \quad (8)$$

where PI is a proportional-integral controller given by

$$\text{PI}(\dot{\theta}_{des} - \dot{\theta}) = P_{\dot{\theta}}(\dot{\theta}_{des} - \dot{\theta}) + I_{\dot{\theta}}\int_0^t (\dot{\theta}_{des} - \dot{\theta})dt\,. \quad (9)$$

In total, the model is fully specified by 10 parameters, summarized in Table I. All parameters are either measured directly or taken from the on-board controller (designed and tuned separately), with the exception of $\dot{\theta}_{max}$, which is a design parameter and lets the user specify how quickly the flip should be performed. With the exception of M and L, all measured parameters were quick, data-based approximations but required no further measurements for the algorithm to converge.

Preprint submitted to 2010 IEEE International Conference on Robotics and Automation. Received September 15, 2009.

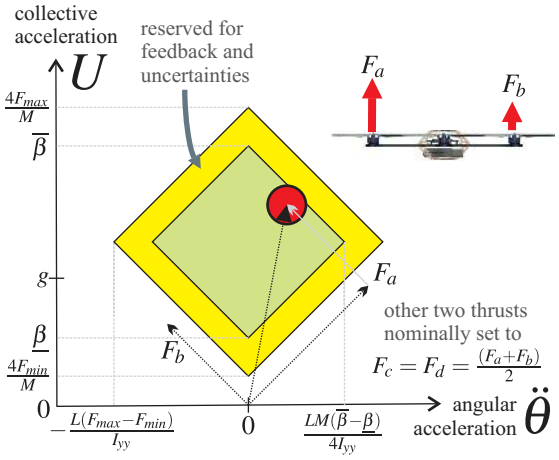

Fig. 4. Control envelope for a quadrotor constrained in a 2D plane.

III. PARAMETERIZED MULTI-FLIP PRIMITIVE

The quadrocopter should perform a flip such that in the end the vehicle's rotation is offset by a multiple of 2π with all the other states unchanged. We ignore the out-of-plane dynamics. The initial and final state conditions for the multi-flip maneuver can then be stated as:

$$x_0 = x_f = 0 \quad (10)$$

$$z_0 = z_f = 0 \quad (11)$$

$$\dot{x}_0 = \dot{x}_f = \dot{z}_0 = \dot{z}_f = 0 \quad (12)$$

$$\theta_f = \theta_0 + 2\pi N = 0 \quad (13)$$

where N is 1 for a single flip, 2 for a double flip, etc.

We do not seek a time-optimal flip, but we do use basic concepts from optimal control to guide how we construct the trajectory. If the system were linear, a time optimal control strategy for the quadrotor would consist of control actions that lie on the edge of the control envelope [12]. In addition, experience shows that for many systems bang-bang control strategies provide results that are very close to the optimal, with greatly reduced complexity [13], [14]. We restrict our attention to such control actions. We use a reduced control envelope, denoted as a range of accelerations ($\underline{\beta}$,$\overline{\beta}$), to account for modeling uncertainties and to reserve some control authority for the on-board feedback controllers. The desired propeller forces must then be consistent with a slightly reduced range of accelerations. A convenient way to parametrize this is

$$F_{min} \leq \frac{M\underline{\beta}}{4} \leq F_{[a,b]} \leq \frac{M\overline{\beta}}{4} \leq F_{max}\,, \quad (14)$$

TABLE I

2D MODEL PARAMETERS

	Source	Value
M	measured	0.468 kg
L	measured	0.17 m
I_{yy}	measured	0.0023 kg m^2
G_{up}	measured	50 s^{-1}
G_{down}	measured	25 s^{-1}
$\dot{\theta}_{max}$	design parameter	1000-1800 °/s
F_{min}	measured	0.08 N per prop
F_{max}	measured	2.8 N per prop
$P_{\dot{\theta}}$	onboard controller	240 rad/s
$I_{\dot{\theta}}$	onboard controller	3600 rad/s^2

so that if no control margin is reserved, $\overline{\beta}$ corresponds to the vehicle's acceleration at full thrust in the absence of gravity. The feasible 2D control envelope of the vehicle can then be depicted as Fig. 4.

Since the quadrotor accepts a collective thrust command and desired rotation rates, we express the control action as $U_{des}, \dot{\theta}_{des}$ where U_{des} is a desired collective acceleration and $\dot{\theta}_{des}$ is a desired angular acceleration. We integrate the desired angular acceleration over the maneuver to produce the desired angular rates at each time instant. This allows us to respect the dynamic limitations of the vehicle while allowing local feedback on board the vehicle to compensate for disturbances and for modeling errors as described in Section V.

For the remainder of this paper all collective and rotative accelerations are understood as the desired values. In the description of the flip below, we make the assumption that the quadrotor always reaches rotation rate of $\dot{\theta}_{max}$. This can be assured by sufficiently lowering $\dot{\theta}_{max}$ depending on the physical characteristics of the quadrotor.

We perform the flip in five steps (Fig. 5):

1) **Acceleration** Accelerate up at near-maximum collective acceleration while rotating slightly away.
2) **Start Rotate** Use maximum differential thrust to achieve $\dot{\theta}_{max}$.
3) **Coast** Hold $\dot{\theta}_{max}$ (use a low collective thrust command to prevent accelerating into ground).

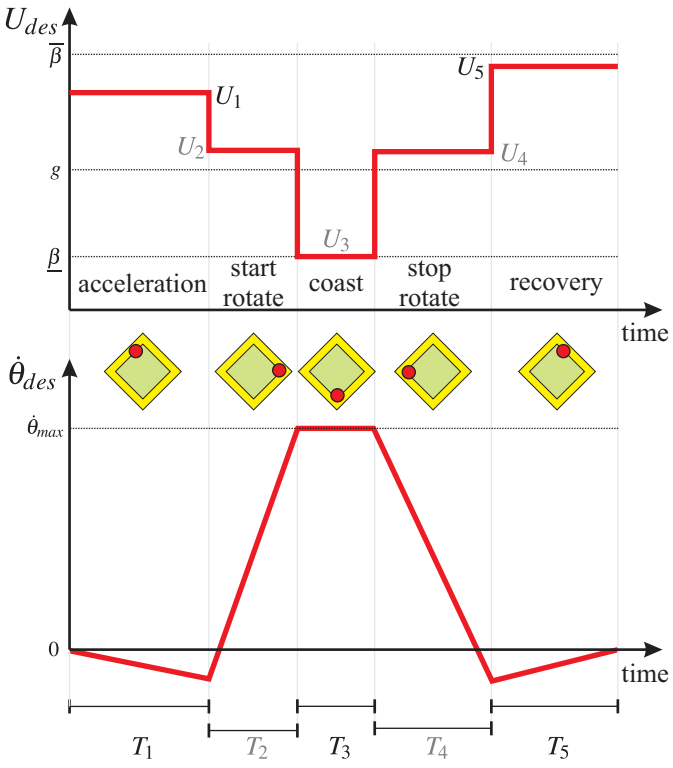

Fig. 5. The collective thrust and commanded angular rate profile of the multi-flip maneuver, with control actions depicted with respect to the reduced control envelope at each stage of the primitive. Note that the grayed out variables along with the rest of the profile are fully determined by the five selected parameters.

Preprint submitted to 2010 IEEE International Conference on Robotics and Automation. Received September 15, 2009.

4) **Stop Rotate** Maximum differential thrust to reach $\dot{\theta}$ slightly less than 0.
5) **Recovery** Accelerate up with near-maximum collective thrust with a slight rotational acceleration to stop vertical descent and any remaining horizontal movement.

Each step of the primitive is fully described by 3 values: a duration T_n, a constant collective acceleration U_n, and a constant rotational acceleration $\ddot{\theta}_n$. Given that we always want to be issuing commands on the edge of the reduced control envelope, U_n and $\ddot{\theta}_n$ fully determine each other.

We select the following parameters:

1) U_1 - collective acceleration during step 1.
2) T_1 - duration of step 1.
3) T_3 - duration of step 3 (coasting at $\dot{\theta} = \dot{\theta}_{max}$).
4) U_5 - collective acceleration during step 5.
5) T_5 - duration of step 5.

and define a vector $P_i = [U_1, T_1, T_3, U_5, T_5]$ as a collection of these parameters at iteration i.

For conciseness, we define a normalized mass distribution variable $\alpha = 2I_{yy}/ML^2$. The other steps are then fully described given these parameters and start/end and coast conditions:

$$\ddot{\theta}_1 = -(\overline{\beta} - U_1)/\alpha L \quad (15)$$
$$\ddot{\theta}_2 = -\ddot{\theta}_4 = (\overline{\beta} - \underline{\beta})/2\alpha L \quad (16)$$
$$U_2 = U_4 = (\overline{\beta} + \underline{\beta})/2 \quad (17)$$
$$T_2 = (\dot{\theta}_{max} - \ddot{\theta}_1 T_1)/\ddot{\theta}_2 \quad (18)$$
$$\ddot{\theta}_3 = 0 \quad (19)$$
$$U_3 = \underline{\beta} \quad (20)$$
$$T_4 = -(\dot{\theta}_{max} + \ddot{\theta}_5 T_5)/\ddot{\theta}_4 \quad (21)$$
$$\ddot{\theta}_5 = (\overline{\beta} - U_5)/\alpha L \quad (22)$$

The multi-flip maneuver is parameterized with five variables. There are also exactly five final error states to minimize when attempting to improve the flip. The problem of optimizing the flips is thus fully constrained.

A. Initial Rough Parameter Guess

It is useful to have a rough guess of the parameter values for initializing the numerical optimization scheme. To this end we can drastically simplify the multi-flip primitive and compute rough guesses for the five parameters. We assume that the maneuver is perfectly symmetric and make several simplifications:

- $U_1 = U_5 = 0.9\overline{\beta}$ We assume that we'll need most of the available acceleration, minus a small margin so that we do not violate the reduced control envelope during gradient calculation and during the initial few iterations.
- We assume that the vehicle is roughly level when entering step 2 and roughly level when exiting step 4. Since steps 2 and 4 are mostly a ramp from 0 to $\dot{\theta}_{max}$, and so have fixed known duration, we can calculate

$$T_3 = \frac{2\pi N - \dot{\theta}_{max}^2/2\ddot{\theta}_2}{\dot{\theta}_{max}}, \quad (23)$$

where $\ddot{\theta}_2$ is defined above.

- Steps 2,3 and 4 are roughly ballistic from a vertical acceleration perspective, so we can compute a guess for the change in $\dot{z}$ accumulated during those steps. This gives us a requirement for vertical velocity at the end of step 1, which should be roughly equal to the negative of the vertical velocity to be cancelled by step 5. Therefore,

$$T_1 = T_5 = \frac{g(T_2 + T_3 + T_4)}{2U_1}. \quad (24)$$

IV. PARAMETER IMPROVEMENT SCHEME

While the true model of the vehicle performing flips is not known, we use the fact that a simple model provides the correct overall direction for corrective action. The main idea behind this approach is similar to the algorithm described in [15], although we retain the full corrective matrix and not just the signs.

We acquire a model-optimal parameter set P_0 by following a procedure outlined in Fig. 6. First we run a numerical optimization on the parameters using the simple models described in Section II, minimizing a weighted 2-norm of the final error state. We generated a starting set of parameters by the approach described above in Section III-A.

The optimization of the parameter set using the simple model results in an initial parameter set P_0. If the solver succeeded then this parameter set allows the vehicle to perform the required maneuver in simulation, returning exactly to the starting state with a $2\pi N$ pitch offset.

We define $\mathcal{F}(P_i)$ to be a column vector of the final error from running the flip primitive with parameter set P_i using the simple model and $\mathcal{E}(P_i)$ as the final error from running the same on the actual vehicle in the Flying Machine Arena testbed.

We calculate a numerical approximation of the Jacobian matrix $J(P_0)$ reflecting the sensitivity of the final error states to the parameters about the model's optimal parameter set P_0. Since the final error $\mathcal{F}(P_0) = \mathbf{0}$,

$$\mathcal{F}(P_0 + \tilde{P}) \approx J(P_0)\tilde{P}, \quad (25)$$

where, as noted above, $\mathcal{F}$ is the output of running the 2D quadrotor model. This expresses a linear approximation of the effects of a parameter perturbation $\tilde{P}$.

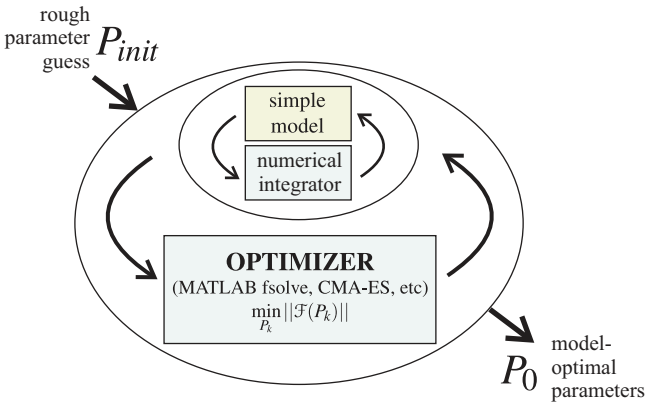

Fig. 6. Outline of the method for finding the initial parameter set P_0.

For problems where the size of the final state equals the number of parameters and where the Jacobian is invertible, the corrective matrix from final error to parameter space is simply the inverse of the Jacobian. To improve the maneuvers in the real world we use the inverse Jacobian matrix at each iteration combined with a step size γ,

$$P_{i+1} = P_i - \gamma J^{-1}(P_0)\mathcal{E}(P_i)\,, \tag{26}$$

where $\mathcal{E}(P_i)$ is the final error vector from running an experiment using the parameter set P_i and γ is a step size between 0 and 1. The step size γ can be used to trade off convergence rate for noise rejection.

V. EXPERIMENTAL SETUP

We tested our approach on the ETH Zurich Flying Machine Arena on our customized quadrocopters. The system is highly modular in both design and implementation, so we describe the quadrotors and the off-board hardware separately.

A. The Modified X3D Quadrocopter

The quadrotor vehicles used for the following experiments are highly modified Ascending Technologies X-3D quadrotors. We replaced the onboard sensors and central electronics completely while keeping the original propulsion system, the motor controllers, and frame. The design and physical properties of the standard X3D quadrotor are described in detail in [16].

The standard firmware on the motor controllers was upgraded to RPM-control firmware from the standard torque-control version. The motor controllers accept 200-step commands at 1 kHz. We experimentally derived a function from command to nominal hover-condition thrust. The RPM control allows us to largely ignore effects of battery voltage and internal resistance, including transients, except for extremely high commands where the achievable RPM is limited by the current voltage.

In order to have better control over the onboard algorithms and to have access to better quality and higher range rate gyro data we replaced the central electronics with our own design. An overview of the onboard controller is shown in Fig. 7. We used the following angular rate sensors: a dual-axis IDG650 2000 $^\circ/\mathrm{s}$ rate gyro for pitch and roll and a single-axis ADXR300 with an extended 1200 $^\circ/\mathrm{s}$ range for sensing yaw rate.

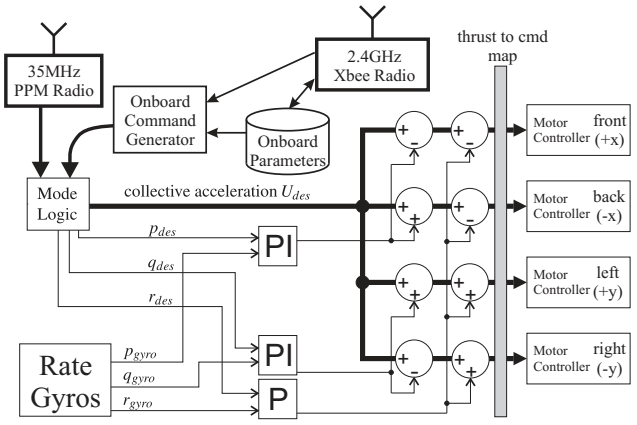

Fig. 7. Logical layout of the onboard controller. The variables p, q, and r refer to roll, pitch, and yaw body angle rates, respectively. In the case of the flip maneuver as described above, $q_{des} = \dot{\theta}_{des}$, while $p_{des} = r_{des} = 0$.

The onboard control loop samples the rate gyros and computes new motor commands at 1KHz. The attitude rate control loops are decoupled from one another. A PI controller produces a differential thrust command based on the current pitch rate and the current desired pitch rate command. The roll rate is controlled similarly. Yaw rate is controlled via a proportional controller without an integral gain. Center of mass location trim values allow for precise balancing of the quadrotor. The outputs of the controller are combined as shown in Fig. 7 and constrained between maximum and minimum command values before being sent to the motor controllers.

Each vehicle is equipped with two radio systems: a one-way 35Mhz analog hobbyist PPM receiver and a bidirectional 2.4GHz XBee transceiver for non-time-critical communication such as data feedback or onboard parameter read/writes.

Usually commands are received by the vehicle via the 35MHz radio at approximately 50Hz. During open-loop maneuvers, commands are instead generated on-the-fly at 1kHz via a function that uses the current onboard maneuver parameters. In the case of the flip, the open-loop command profile corresponds exactly to that shown in Fig. 5, sampled at 1KHz.

The approach of generating commands directly onboard the vehicles allows us to update the desired angle rates and collective thrust commands with virtually no communication delay with the maximum 1-ms time interval resolution. While this approach assures good maneuver repeatability, it does add some difficulty to detecting when exactly the vehicle begins and ends the open loop maneuver and switches to normal control. We have found that a good understanding of communication delays is vital to measuring the final state error accurately.

B. The ETH Flying Machine Arena

The ETH Flying Machine Arena (FMA) is a $10\times10\times10\mathrm{m}$ space built for research involving small flying vehicles. The overall organization of the system is similar to [1]. The space is equipped with a motion capture system for localization and a set of protective nets to reduce the occurrence of catastrophic crashes. We use a Vicon motion capture system with 8 cameras to achieve redundant retroreflective marker localization at 200Hz with millimeter accuracy. Each quadrotor carries a unique arrangement of three such markers allowing the Vicon system to measure each vehicle's full position and attitude at each frame.

The conceptual organization of the components in the FMA is shown in Fig. 8. A flexible, reflective data serialization scheme allows for convenient online visualization of all data sent over the network and also for recording, playback, and export of near-arbitrary data series. All data is sent back and forth using the multicast UDP scheme; any specific hardware interfacing is handled by dedicated bridges,

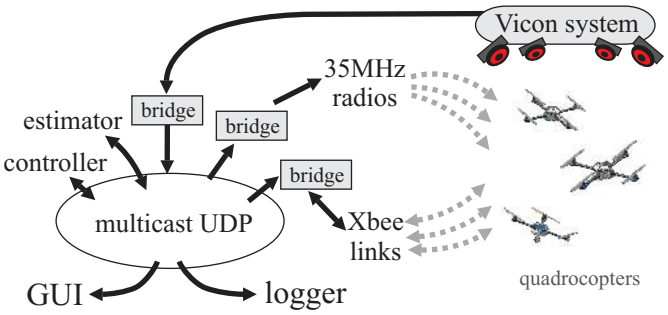

Fig. 8. Overview of the ETH Flying Machine Arena testbed.

allowing the core processes to be completely separate from hardware interfacing issues. A convenient side-effect of this setup is that components are binary-identical when running in the real system or in simulation. In addition, played back data is automatically accepted by components as actual, real-time data, allowing for convenient debugging functionality.

Similarly to [1], the localization data is used by a set of processes that run estimation and control algorithms on a set of typical desktop PCs. The resulting commands are sent via hobbyist PPM channels to the quadrotors with a 50 Hz update rate as described above. Under usual operation the vehicle's translational degrees of freedom are controlled by linear PID controllers designed for near-hover operation. Yaw is held at a constant angle via a proportional controller.

To execute an iteration of the flip, a managing process first uploads a set of parameters and then signals the vehicle to begin executing the maneuver. The vehicle then executes the primitive on its own, ignoring hover controller commands for the duration of the flip. Once the primitive is over, or if certain safety constraints are broken, the vehicle resumes normal operation and reports the end of the primitive to the managing process. The final error is then recovered from filtered Vicon data, parameters adjusted as described in Section IV, and the process is repeated.

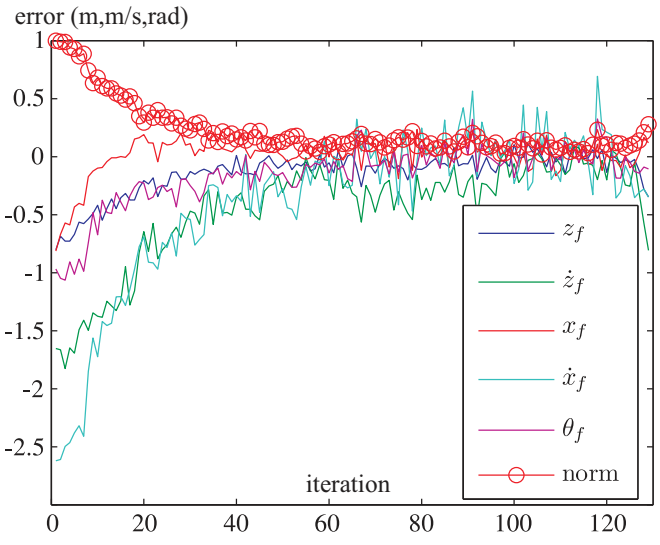

Fig. 9. Evolution of final state errors over a 129-iteration run of a 1600 °/s double flip. Step size γ was set to 0.1.

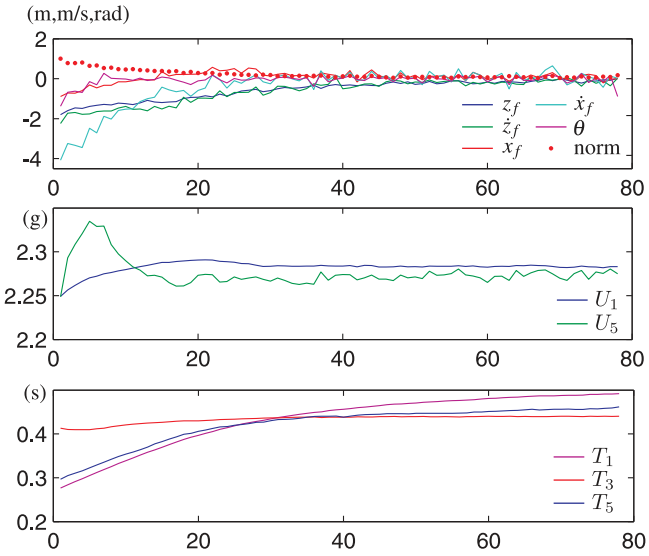

Fig. 10. Final state error and parameter evolution over 78 iterations for a triple flip. Step size γ=0.1. Note that the parameter values are relative to and normalized by their initial values.

VI. EXPERIMENTS

A. *Double flips* ($N = 2$)

Fig. 9 shows the evolution of final state errors for a 129-iteration 1600 °/s double-flip experiment. The maneuver converges within the first 40 to 50 iterations. Note the increase in error at the end due to the battery running low.

B. *Triple flips* ($N = 3$)

The trajectory for a triple flip maneuver according to the simple model and on the actual system can be seen in Fig. 2. Fig. 10 depicts the evolution of the final state errors and maneuver parameters over a 78-iteration experiment. Note that the initial error is quite large (-4 m/s and -2 m/s lateral and vertical velocities, respectively). Since this maneuver is longer in duration than the single and double flips, we experienced significantly worse repeatability problems than the shorter maneuvers. A small step size was especially important for the first few iterations as the thrust parameters typically initially hover right near their upper limit. The parameters continue a slow evolution throughout the experiment, compensating for the slight changes in the transient voltage response of the battery throughout the flight.

C. 1300 °/s *double flips with* 1600 °/s *triple-flip* J *and* P_0

One of the side effects of the described approach is that a Jacobian generated for triple flips will also work to improve single or double flips. For example, if the vehicle performs two flips on the first iteration (due to model errors or a $\dot{\theta}_{max}$ mismatch), it will converge to a double flip (Fig. 11). We found that the Jacobians generated for different numbers of flips are surprisingly similar, once again supporting the signed gradient intuition [15]. However, the initial error is much larger than with a properly-generated P_0.

Preprint submitted to 2010 IEEE International Conference on Robotics and Automation. Received September 15, 2009.

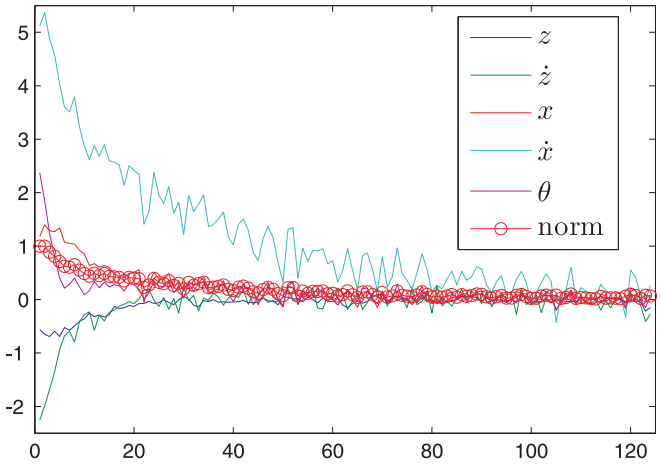

Fig. 11. Final state error and parameter evolution over 124 iterations for a vehicle with $\dot{\theta}_{max} = 1300°/\text{s}$ using J and P_0 calculated for $\dot{\theta}_{max} = 1600°/\text{s}$. Step size $\gamma = 0.07$. Note that θ_f is normalized to $(-\pi, \pi)$.

VII. CONCLUSION

We have demonstrated a simple and intuitive method for iteratively improving quadrocopter flips. Our method requires only the final state error to be measured and is simple and lightweight to implement. The model used in the method uses straightforward, measurable parameters and does not require extensive parameter identification experiments. During iterative parameter adjustment, user control over system convergence speed is provided by a step size parameter, an essential feature for successful implementation on real systems. The method is simple to extend to other aerobatic maneuvers and is well suited as a bootstrapping mechanism for generating feasible trajectories for more involved learning/adaptation algorithms. A video of the algorithm in action and relevant source code are available online at `www.idsc.ethz.ch/people/staff/lupashin-s`.

VIII. ACKNOWLEDGMENTS

We thank Felix Althaus, Guillaume Ducard and Matt Donovan for their contributions to the onboard controller, the hover controller and to the building of the FMA testbed, respectively. We thank Jonathan How for his advice during the design and setup of the FMA space. We thank Jan Stumpf and the rest of Ascending Technologies team for providing us with RPM- control motor controller firmware. Finally, we thank Haomiao Huang and Russ Tedrake for the fruitful discussions about autonomous aerobatics.

REFERENCES

[1] J. How, B. Bethke, A. Frank, D. Dale, and J. Vian, "Real-time indoor autonomous vehicle test environment," *Control Systems Magazine, IEEE*, vol. 28, no. 2, pp. 51–64, April 2008.

[2] G. M. Hoffmann, H. Huang, S. L. Wasl, and E. C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: theory and experiment," in *In Proc. of the AIAA Guidance, Navigation, and Control Conference*, 2007.

[3] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," in *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 3, Sept.-2 Oct. 2004, pp. 2451–2456 vol.3.

[4] S. Waslander, G. Hoffmann, J. S. Jang, and C. Tomlin, "Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, Aug. 2005, pp. 3712–3717.

[5] O. Purwin and R. D'Andrea, "Performing aggressive maneuvers using iterative learning control," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 1731–1736.

[6] J. H. Gillula, H. Huang, M. P. Vitus, and C. J. Tomlin, "Design and analysis of hybrid systems, with applications to robotic aerial vehicles," in *Robotics Research, 2009 International Symposium of*, Aug 2009.

[7] M. B. Gerig, "Modeling, guidance, and control of aerobatic maneuvers of an autonomous helicopter," Ph.D. dissertation, ETH Zurich, 2008, no. 17805.

[8] A. Coates, P. Abbeel, and A. Y. Ng, "Learning for control from multiple demonstrations," in *ICML '08: Proceedings of the 25th international conference on Machine learning*. New York, NY, USA: ACM, 2008, pp. 144–151.

[9] J. G. Leishman, *Principles of Helicopter Aerodynamics*, 2nd ed. Cambridge University Press, 2006.

[10] P.-J. Bristeau, P. Martin, E. Salaün, and N. Petit, "The role of propeller aerodynamics in the model of a quadrotor UAV," in *European Control Conference 2009*, 2009.

[11] M. Achtelika, A. Bachrachb, R. Heb, S. Prenticeb, and N. Royb, "Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments," in *Robotics: Science and Systems Conference*, June 2008.

[12] R. Stengel, *Stochastic optimal control: theory and application*. John Wiley & Sons, 1986.

[13] T. K. Nagy, R. D'Andrea, and P. Ganguly, "Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle," *Robotics and Autonomous Systems*, vol. 46, pp. 47–64, 2004.

[14] O. Purwin and R. D'Andrea, "Trajectory generation and control for four wheeled omnidirectional vehicles," *Robotics and Autonomous Systems*, vol. 54, pp. 13–22, 2006.

[15] J. Z. Kolter and A. Y. Ng, "Policy search via the signed derivative," in *Robotics: Science and Systems*, 2009.

[16] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz," in *Robotics and Automation, 2007 IEEE International Conference on*, April 2007, pp. 361–366.