2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

A Frequency Domain Iterative Feed-Forward Learning Scheme for High Performance Periodic Quadrocopter Maneuvers

Markus Hehn and Raffaello D'Andrea

Abstract— **Quadrocopters exhibit complex high-speed flight dynamics, and the accurate modeling of these dynamics has proven difficult. Due to the use of simplified models in the design of feedback control algorithms, the execution of high-performance flight maneuvers under pure feedback control typically leads to large tracking errors. This paper investigates an iterative learning scheme aimed at the non-causal compensation of repeatable trajectory tracking errors over the course of multiple executions of periodic maneuvers. The learning is carried out in the frequency domain and uses a simplified model of the closed-loop dynamics of quadrocopter and feedback controller. The resulting algorithm requires little computational power and memory, and its convergence is shown for the nominal model. This paper further introduces a time-scaling method that allows the initial learning to occur at reduced speeds, thus extending the applicability of the algorithm for high performance maneuvers. The presented algorithms are validated in experiments, with a quadrocopter flying a figure-eight maneuver at high speed.**

I. Introduction

Aerial robotics research has embraced quadrocopters as a popular platform, and a large number of commercial and open-source systems are available (e.g., [1]–[3]). Compared to other aerial robotics platforms, quadrocopters provide mechanical simplicity, robustness, and a large dynamic potential due to high thrust-to-weight ratios and fast rotational dynamics.

Like many other robotic systems, quadrocopter aerial platforms are often applied to tasks that involve the repeated execution of identical or similar motions. Such applications include for example inspection, monitoring, and filming [4]. The tasks typically require precise tracking of a reference trajectory, which is complicated by the inherently unstable, nonlinear dynamics of quadrocopters, as well as significant model uncertainties and disturbances caused primarily by a number of aerodynamic effects [5].

When a motion is executed repeatedly, causal feedback control laws may be augmented through the use of non-causal learning control. Using input-output data of past executions, the added learning control is designed to provide additional control inputs such that repeated disturbances are eliminated. A number of authors have demonstrated the application of iterative learning control (ILC) in combination with real-time feedback control (e.g., [6], [7]). The combination was shown to be capable of achieving high performance trajectory tracking, while maintaining good disturbance rejection for non-repetitive noise. For quadrocopters, significantly increased tracking performance has been demonstrated through the use of ILC [8], [9].

Besides ILC, a number of other learning strategies have been developed for high performance quadrocopter flight. A first group of learning schemes aim to account for model mismatches, for example through sliding mode control and reinforcement learning techniques [10], neural networks [11], and adaptive control [12].

A second approach is the use of parameterized motion primitives, where a set of parameters defines the motion. These parameters are then adapted in order to correct for tracking errors at specific points along the trajectory (referred to as 'key frames'). Demonstrations of this approach applied to quadrocopters include flying through narrow openings and perching [13], as well as multiple flips and time-optimal translations [14]. Compared to ILC, these approaches offer improved computational efficiency as a result of the low dimensionality of the maneuver parameterization.

This paper investigates the performance of a learning scheme that aims to compensate for trajectory tracking errors in the execution of periodic maneuvers in a fashion similar to ILC. However, it is comparable to parameterized motion primitives with respect to its use of a significantly reduced parameterization of the learning control input. Instead of relying on a specific parameterization of the maneuver, we decompose the tracking error into a truncated Fourier series. The order of the series provides a means to trade off the ability to compensate for local disturbances against computational complexity. The learning algorithm is closely related to repetitive control formulations [15], which have been suggested to be equivalent to ILC [16].

Learning is performed on the closed-loop system of a quadrocopter controlled by an input-output linearizing feedback controller, and the learning input is a set point correction for the feedback control. The nominally linear time-invariant closed-loop dynamics of the system provide a straightforward way to compute the necessary input corrections from measured tracking errors. The iterative learning algorithm thereby reduces to the inversion of the closed-loop dynamics.

The treatment of the iterative learning problem in the frequency domain offers an additional advantage in that it provides a straightforward means of transferring learnt corrections for a given maneuver geometry between different execution speeds. We extend the learning scheme in order to allow the learning to begin at low maneuver execution speeds, and transfer the learnt corrections to higher speeds.

The remainder of this paper is structured as follows: Sec-

The authors are with the Institute for Dynamic Systems and Control, ETH Zurich, Switzerland. {hehnm, rdandrea}@ethz.ch

978-1-4673-6357-0/13/$31.00 ©2013 IEEE

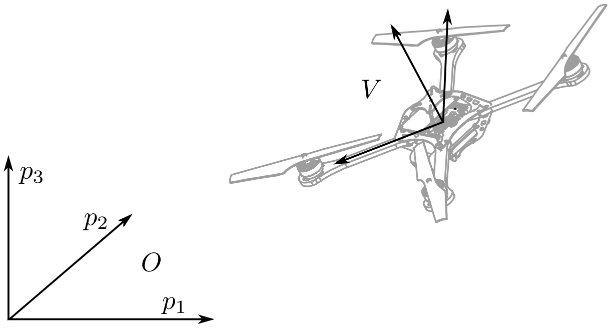

Fig. 1. The inertial coordinate system O and the vehicle coordinate system V, used to describe the dynamics of the quadrocopter.

tion II introduces the quadrocopter dynamics, the real-time feedback law, and the resulting closed-loop system dynamics. Section III presents the proposed iterative learning algorithm and its convergence. Section IV extends the algorithm by the transfer of learnt corrections between different execution speeds. Section V introduces the experimental setup and an example maneuver the algorithm was applied to. We show experimental results in Section VI. Advantages and limitations of the algorithm are discussed in Section VII. A conclusion, along with an outlook, is given in Section VIII.

II. Closed-Loop Quadrocopter Dynamics

In this section, we introduce the first-principles model of the quadrocopter dynamics, and the input-output linearizing feedback controller used to control it. The combination of vehicle and feedback controller form the closed-loop dynamics that the iterative learning algorithm is applied to.

A. Vehicle Dynamics

The quadrocopter is modeled as a rigid body with six degrees of freedom: its position (p_1, p_2, p_3) in the inertial coordinate system O and its attitude, represented by the rotation matrix ${}^{O}_{V}R$ between the inertial coordinate system O and the body-fixed coordinate system V, as shown in Figure 1.

The control inputs are the rotational rates of the vehicle about the three body axes $(\omega_x, \omega_y, \omega_z)$ and the collective, mass-normalized thrust applied by the vehicle along its third body axis (a; in units of acceleration). The rotational rates are tracked by high-bandwidth controllers on board the vehicle using gyroscopic rate sensors. We therefore assume that control inputs are followed by the vehicle without dynamics or delay.

The translational dynamics are then given by

$$\begin{bmatrix} \ddot{p}_1 \\ \ddot{p}_2 \\ \ddot{p}_3 \end{bmatrix} = {}^{O}_{V}R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\mathrm{g} \end{bmatrix} \tag{1}$$

where g denotes gravitational acceleration, and the rotational kinematics are [17]

$${}^{O}_{V}\dot{R} = {}^{O}_{V}R \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} . \tag{2}$$

The mass-normalized thrust control input is subject to saturation due to the minimum and maximum speed of the propellers:

$$a_{\min} \leq a \leq a_{\max} \tag{3}$$

and the allowable body rate control inputs are bounded by the measurement range of the onboard gyroscopic sensors:

$$|\omega_i| \leq \omega_{\max} \text{ for } i = \{x, y, z\} . \tag{4}$$

B. Feedback Control

Within this paper, we assume that an existing feedback control law is used to stabilize the quadrocopter and track set points. The feedback control law is described in detail in [18], and is reproduced here for completeness. It consists of cascaded control loops for position and attitude along with the inversion of the rotational kinematics as follows:

1) Position Control: For all three degrees of freedom, a feedback control law determines the desired acceleration $\ddot{\hat{p}}_i$ from the position and velocity errors such that the loop is shaped to the dynamics of a second-order system with time constant τ_i and damping ratio ζ_i:

$$\ddot{\hat{p}}_i = \frac{1}{\tau_i^2}\left(\hat{p}_i - p_i\right) - 2\frac{\zeta_i}{\tau_i}\dot{p}_i \quad \text{for } i = \{1, 2, 3\} . \tag{5}$$

With R_{xy} denoting the x-th element of the y-th column of ${}^{O}_{V}R$, the thrust is computed to enforce the desired vertical acceleration according to (1):

$$a = \frac{1}{R_{33}}\left(\ddot{\hat{p}}_3 + \mathrm{g}\right) . \tag{6}$$

2) Reduced Attitude Control: The desired rotation matrix entries $\hat{R}_{13}$ and $\hat{R}_{23}$ for the given desired accelerations are computed from (1), (5) and (6) to be

$$\hat{R}_{13} = \frac{\ddot{\hat{p}}_1}{a} \text{ and } \hat{R}_{23} = \frac{\ddot{\hat{p}}_2}{a} . \tag{7}$$

The attitude control loop is shaped such that the rotation matrix entries R_{13} and R_{23} react in the manner of a first-order system with time constant τ_{rp} by computing the desired derivative of the rotation matrix elements:

$$\dot{R}_{i3} = \frac{1}{\tau_{rp}}\left(\hat{R}_{i3} - R_{i3}\right) \text{ for } i = \{1, 2\} . \tag{8}$$

Inverting the rotational kinematics (2), this is converted to the commanded rotational rates about the vehicle x- and y-axes:

$$\begin{bmatrix} \omega_x \\ \omega_y \end{bmatrix} = \frac{1}{R_{33}} \begin{bmatrix} R_{21} & -R_{11} \\ R_{22} & -R_{12} \end{bmatrix} \begin{bmatrix} \dot{R}_{13} \\ \dot{R}_{23} \end{bmatrix} . \tag{9}$$

The rotational rate about the third body axis, ω_z, can be determined separately as it does not influence the translational dynamics of the vehicle. We employ a proportional controller on the Euler angle describing the vehicle heading.

Note that the presented controller can be augmented by applying feed-forward velocities, accelerations, and control inputs for known input trajectories by extending Equations (5) and (8) with the corresponding feed-forward terms. A discussion of the effects and performance benefits thereof can be found in [19], where it is shown that feed-forward commands can improve tracking performance, though large systematic errors remain. A significant reason for these remaining errors are model mismatches: As the maneuvering speed rises, the equations of motion (1) represent an increasingly poor approximation of the flight dynamics due to aerodynamic effects (such as body drag and lift, as well as varying angles of attack of the propellers).

While many applications profit from additional feed-forward terms, we found them unnecessary in conjunction with the learning method presented herein. This is because they do not improve the repeatability of the flight performance, and the learning algorithm compensates for systematic tracking errors.

C. Approximate Closed-Loop System Dynamics

The feedback control design is based on cascaded control loops that are designed using a loop shaping approach. We assume time scale separation between the control loops (i.e., $\tau_{rp} \ll \tau_{xy}$), and approximate the closed-loop dynamics to depend only on the position control loops. The nominal dynamics of the closed-loop system from a position set point $\hat{p}$ to the vehicle position p can then be approximated by three decoupled linear time-invariant (LTI) second-order systems:

$$\ddot{p}_i \approx \frac{1}{\tau_i^2}\left(\hat{p}_i - p_i\right) - 2\frac{\zeta_i}{\tau_i}\dot{p}_i \text{ for } i = \{1, 2, 3\} \tag{10}$$

with time constant τ_i and damping ratio ζ_i. We will use these nominal closed-loop dynamics in the iteration-domain learning algorithm.

More accurate characterizations of the closed-loop dynamics could be used in the learning algorithm, e.g. by including the underlying control loops such as the attitude control (7)-(9). However, our experiments showed that the low-order model was sufficient to guide the iterative learning process. Higher-order LTI models would require linearization of the closed-loop dynamics about an operating point, and therefore may not accurately predict global system dynamics.

It was shown in [19] that the closed-loop dynamics of the quadrocopter and feedback controller, while exhibiting relatively large errors during fast motion, provide good repeatability. We will now exploit this property by introducing an iterative learning algorithm that compensates for systematic, repeated tracking errors over the course of multiple iterations.

III. Learning Algorithm

This section introduces the frequency domain iterative learning algorithm that is used to minimize repeated tracking errors, which may be caused for example by modeling inaccuracies, the closed-loop dynamics of the system, and repeated disturbances.

A. Learning Overview

The objective of the learning scheme is to find control inputs such that the quadrocopter precisely follows a desired, periodic trajectory, which is given by $p^*(t)$ for $t \in [0\, T]$, with T being the maneuver period.

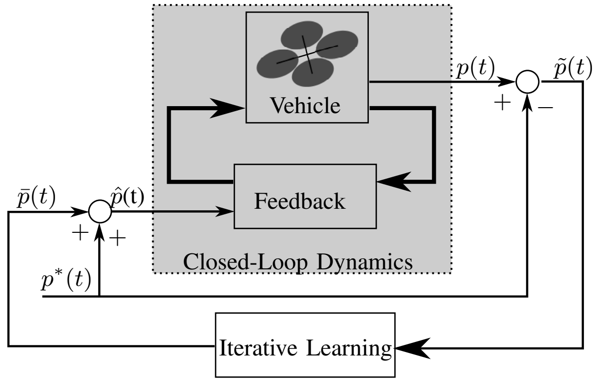

Fig. 2. The system structure used in the learning algorithm. The learning scheme has a serial architecture, meaning that it changes the set point used by a feedback controller. Due to the feedback linearizing architecture of the feedback controller, the iterative learning law need not consider the nonlinear dynamics of the vehicle: The closed-loop dynamics are nominally linear time-invariant.

We use the approach of adapting the set point $\hat{p}(t)$ of the controller, also called a serial architecture [7] or indirect learning-type control [16]. A set point correction $\bar{p}(t)$ is added to the desired trajectory set point $\hat{p}(t) = p^*(t) + \bar{p}(t)$, as shown in Figure 2. In comparison to modifying the control inputs directly, this offers the advantage that the dynamics from a change in the set point $\hat{p}$ to a change in the vehicle position p are those of the closed loop system, which are designed to be nominally LTI.

The core idea of the adaptation law is to measure the output error $\tilde{p}(t) = p(t) - p^*(t)$ of the closed-loop system over an iteration, and decompose it into a truncated Fourier series. Due to the additivity of the LTI nominal system dynamics [20], it is possible to compute a set point correction truncated Fourier series $\bar{p}(t)$ from $\tilde{p}(t)$ that compensates for repeated tracking errors. This correction, scaled by a step size in order to increase robustness with respect to non-repeatable disturbances and model inaccuracies, is then added to the feedback control set point for the next iteration.

B. Iteration-Domain Feedback Law

We rewrite the measured output error $\tilde{p}(t)$ as a combination of a truncated Fourier series $\mathrm{e}(t)$ of order N, and a remaining truncation error $\mathrm{v}(t)$:

$$\tilde{p}(t) = \mathrm{e}(t) + \mathrm{v}(t) \tag{11}$$

with

$$\mathrm{e}(t) = a_0 + \sum_{k=1}^{N} a_k \cos\left(k\Omega_0 t\right) + \sum_{k=1}^{N} b_k \sin\left(k\Omega_0 t\right) \tag{12}$$

where the fundamental frequency is $\Omega_0 = 2\pi/T$.

The objective of the iterative learning scheme is to eliminate the effects of errors described by the Fourier series $\mathrm{e}(t)$. The additional error $\mathrm{v}(t)$ captures the truncation effects of $\mathrm{e}(t)$, and could be captured by $\mathrm{e}(t)$ through an increase of the series order N.

Let $\tilde{P}(\omega)$ denote the Fourier series representation of $\tilde{p}(t)$, composed of the Fourier series representations of $\mathrm{e}(t)$

and $v(t)$:

$$\tilde{P}(\omega) = E(\omega) + V(\omega) \tag{13}$$

with

$$E(0) = a_0 \tag{14}$$

$$E(k\Omega_0) = a_k - jb_k \text{ for } k = 1 \dots N. \tag{15}$$

The signals considered herein are real, and we therefore limit the analysis to $\omega \geq 0$ without loss of generality [20]. We further note that, by the assumption that $e(t)$ is a truncated Fourier series and $v(t)$ its truncation error, it follows that

$$E(\omega) = 0 \text{ for } \omega > N\Omega_0 \tag{16}$$

$$V(\omega) = 0 \text{ for } \omega \leq N\Omega_0 \ . \tag{17}$$

Let $G(\omega)$ be the transfer function of the nominal closed-loop feedback system (10) from the position set point to the vehicle position:

$$G(\omega) = \frac{P(\omega)}{\hat{P}(\omega)} = \frac{1}{1 - \tau^2\omega^2 + 2j\zeta\tau\omega} \ . \tag{18}$$

Assume that we have executed the trajectory for iteration i, and measured the tracking error $\tilde{p}^i$. The iteration feedback law is then given by the correction Fourier series

$$\bar{P}^{i+1}(k\Omega_0) = \bar{P}^i(k\Omega_0) - \gamma^i G^{-1}(k\Omega_0)\, E^i(k\Omega_0) \tag{19}$$

for $k = 0 \dots N$, with the step size γ^i being a tuning parameter. The time signal $\bar{p}^i(t)$ is then constructed from $\bar{P}^i(\omega)$ and applied to the system in the next iteration.

C. Convergence

We now consider the convergence of the frequency domain tracking error $\tilde{P}(\omega)$. Using the additivity of the closed-loop LTI dynamics, the iteration update leads to an error progression of

$$\tilde{P}^{i+1} = \tilde{P}^i + G(\bar{P}^{i+1} - \bar{P}^i) \tag{20}$$

$$= \tilde{P}^i - G\left(\gamma^i G^{-1} E^i\right) \tag{21}$$

$$E^{i+1} + V^{i+1} = E^i + V^i - \gamma^i E^i \tag{22}$$

and therefore an evolution of the tracking errors E and V according to

$$E^{i+1} = \left(1 - \gamma^i\right) E^i \tag{23}$$

$$V^{i+1} = V^i \ . \tag{24}$$

It can be seen that E converges to zero for constant step sizes $0 < \ \gamma = \gamma^i \leq 1$, while the higher-frequency error V (which is not captured by the truncated Fourier series) remains constant.

Note that high values of the step size γ provide faster convergence, while being more susceptible to disturbances caused by non-repeatable effects. The step size can therefore be used to trade off the convergence rate of E and noise rejection. This may be necessary because $\tilde{P}$ could include non-repetitive process and measurement noise, and the Fourier series decomposition is therefore only an estimate of the true repeatable error E. In our experiments, we choose decaying values of γ^i, combining fast initial convergence and noise-resistance in the long run.

IV. Time Scaling for High Performance Maneuvers

When attempting to learn high performance maneuvers that approach the performance limits given by the system dynamics (1)-(4), it may not be possible to execute iterations and measure the corresponding tracking errors. For example, the tracking errors may grow large enough to cause the vehicle to collide with its environment, invalidating the error measurement. Furthermore, tracking errors may be so large that the approximate dynamics (10) no longer accurately predict the behaviour of the closed-loop control system, leading to instabilities in the learning algorithm.

In order to allow the algorithm to be applied to such maneuvers, we extend it by introducing a speed scaling factor λ, giving control over the execution speed of the maneuver. Consider two different maneuver execution speeds λ_1 and λ_2. We define the scaled maneuver durations

$$T_1 = \frac{T}{\lambda_1} \text{ and } T_2 = \frac{T}{\lambda_2} \tag{25}$$

and the corresponding nominal trajectories

$$p_1^*(t_1) = p^*(\lambda_1 t) \ \text{ for } t_1 \in [0\, T_1] \tag{26}$$

$$p_2^*(t_2) = p^*(\lambda_2 t) \ \text{ for } t_2 \in [0\, T_2] \ . \tag{27}$$

Now assume that $\lambda_1 < \lambda_2$, and that the iterative learning law (19) has converged to a correction signal $\bar{P}_1(\omega)$ that provides good tracking performance at the execution speed λ_1. The objective is then to use $\bar{P}_1(\omega)$ in order to construct an initial guess of $\bar{P}_2(\omega)$ such that the initial tracking errors at the execution speed λ_2 are sufficiently small for the learning control to converge.

An obvious choice for the transfer between two maneuver speeds is to keep the learnt input corrections and simply re-map them to the corresponding frequencies (i.e., $\bar{P}_2(k\lambda_2\Omega_0) = \bar{P}_1(k\lambda_1\Omega_0)$). However, the varying sensitivity of the closed-loop transfer function at the two different frequencies could potentially lead to large errors. Instead, we assume that, without the application of the iterative learning correction $\bar{p}(t)$, the tracking error for the two different maneuver execution speeds λ_1, λ_2 are

$$\tilde{p}\left(\lambda_1 t_1\right) = \tilde{p}\left(\lambda_2 t_2\right) \tag{28}$$

implying that the tracking error will be scaled in time identically to the maneuver, and not change apart from this. Note that, for significant changes of λ, the assumption of similar tracking errors is likely to be false. For incremental changes of λ, however, the similarity of tracking errors can be expected.

It follows that the correction input Fourier series for the new maneuver execution speed λ_2 can be computed from the old maneuver execution speed λ_1 to be

$$\bar{P}_2(k\lambda_2\Omega_0) = G^{-1}(k\lambda_2\Omega_0)\, G(k\lambda_1\Omega_0)\, \bar{P}_1(k\lambda_1\Omega_0) \tag{29}$$

for $k = 0 \dots N$.

V. Experiments

Experiments were carried out in the Flying Machine Arena, an aerial vehicle research platform at ETH Zurich [21]. The quadrocopter vehicles used are modified Ascending Technologies 'Hummingbird' vehicles [1], equipped with custom electronics to allow greater control of the low-level control algorithms [14]. A motion capture system provides position and attitude information, which is filtered by a Luenberger observer. The filtered full state information is used by the feedback controller, and the filtered position information is used in the learning algorithm.

A. Nominal Maneuver Design

We will now design a high performance periodic maneuver, to which we will apply the presented learning algorithm. The maneuver consists of a figure-eight motion executed in the horizontal plane around two obstacle points, as shown by the dotted black line in Figure 3.

Assume, without loss of generality, that the first obstacle point lies at the origin of the inertial coordinate system O, and the second obstacle point is located at a distance L in the p_1-direction. The maneuver is composed of two half-circles about the obstacle points, and two splines connecting the half-circles. The maneuver is executed as fast as possible, with the speed being limited by the control input saturations (3)-(4).

1) Semi Circles: The radius R_s of the semi circles is user-defined, and the circular trajectory covers an angle of $180°$. The time required for the half circle is determined by computing the control inputs along the circular trajectory [22], and finding the fastest time for which the control input constraints (3)-(4) are satisfied.

2) Connecting Splines: Polynomial trajectories are used to connect the two half-circles. In order to achieve continuity in the control inputs, four boundary constraints arise at each end of the trajectory [23]: with the radius and rate of the semi circles fixed, the position, velocity, acceleration, and jerk at the beginning and end of the half circle are fully defined. In order to satisfy the four boundary constraints on both ends of the spline, we construct a seventh-order polynomial. The remaining degree of freedom in the spline design is the duration of maneuver. In order to achieve high speed, we iterate over the duration until the fastest maneuver that satisfies the control input bounds (3)-(4) is found, using the algorithm from [24] to compute the inputs.

An example of the figure-eight motion can be seen in Figure 3. In this specific example, the parameters were chosen to be $L = 4\,\mathrm{m}$, $R_s = 0.75\,\mathrm{m}$, $a_{\max} = 1.8\mathrm{g} = 17.65\,\mathrm{m\,s^{-2}}$, and $\omega_{\max} = 500°\,\mathrm{s}^{-1}$. The resulting maneuver duration is $T = 3.3\,\mathrm{s}$, with an average speed of $4\,\mathrm{m\,s^{-1}}$ and a maximum speed of $6\,\mathrm{m\,s^{-1}}$. This example will be used as the reference trajectory to be learnt in Section VI.

B. Implementation of the Learning Algorithm

The nominal maneuver design introduced in the previous section is periodic, meaning that one execution immediately follows the other. The learning algorithm was implemented such that adaptation of the control inputs occurs without interrupting the repeated execution of the nominal maneuver. Each iteration of the learning algorithm consists of three steps:

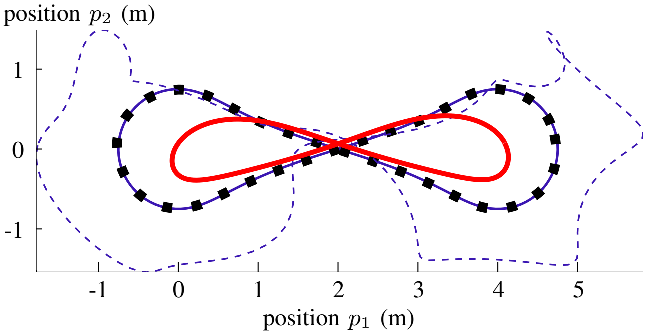

Fig. 3. Two-dimensional view of the learnt maneuver for $L = 4\,\mathrm{m}$, $R_s = 0.75\,\mathrm{m}$. The nominal maneuver design is shown by the black dotted line. The set point trajectory $\hat{p}(t)$ after convergence of the algorithm is shown in dashed blue. It can be seen that tracking performance is improved considerably from the initial trial (thick solid red) to execution after convergence (thin solid blue). Final execution errors are always below $5\,\mathrm{cm}$. The period of the maneuver is $T = 4.7\,\mathrm{s}$.

1. Measure the tracking error $\tilde{p}$ for at least one period T. By averaging the error over multiple iterations, improved rejection of non-repetitive disturbances can be achieved.
2. Apply the iteration-domain feedback law (19).
3. Execute at least one period of the motion before measuring the tracking error in order to permit the system to converge to a repeatable trajectory.

The learning step size was chosen to be

$$\gamma^i = \begin{cases} 1 \text{ for } i = \{1, 2\} \\ \frac{3}{i} \text{ otherwise} \end{cases} \tag{30}$$

with the objective of providing a good compromise between fast learning and good rejection of non-repeatable errors.

VI. Results

This section presents experimental results that demonstrate the performance of the introduced learning scheme. We begin with a discussion of the learning performance at a fixed maneuver execution speed, followed by the extension to increasing maneuver speeds.

A. Learning at Fixed Maneuver Execution Speed

In order to demonstrate the effectiveness of the presented learning algorithm at a fixed maneuver speed, the maneuver is learnt at a constant speed of $\lambda = 0.7$, i.e. 70% of the nominal maneuver speed. At this speed, the maneuver could be safely executed with no initial set point correction (i.e. $\bar{p}^1(t) = 0$), and the learning algorithm converged.

Figure 3 shows the trajectories of the vehicle and the set points in the horizontal plane, both at the start of the learning process and after convergence. Figure 4 shows the evolution of the error coefficients over 22 iterations of the learning algorithm. It can be seen that the error coefficient magnitude quickly reduces from values in excess of $100\,\mathrm{cm}$

to values below $2\,\mathrm{cm}$. The peak tracking error was reduced from approximately $200\,\mathrm{cm}$ to $5\,\mathrm{cm}$.

For fixed maneuver execution speeds λ significantly larger than 0.7, the learning process failed to converge when initialized with no initial set point correction. Our experiments showed that this behaviour was largely independent of the learning step size γ. We believe this to be caused by the inaccurate characterization of the closed-loop dynamics for very large errors as a LTI system because nonlinear effects such as actuator saturation become significant.

B. Transferring Learnt Data Between Execution Speeds

In a second experiment, the same maneuver was executed at increasing speeds. The execution speed was initialized at $\lambda = 0.7$, and was increased by 0.05 every eight iterations until the nominal speed $\lambda = 1$ was reached. The learning rate was again chosen according to Equation (30), but the iteration count index i was reset to $i = 1$ each time λ was changed. At each change of λ, the update law (29) was applied in order to produce updated correction inputs. Figure 5 depicts the norms of the error Fourier coefficients, showing good initial convergence to values on the order of $1\,\mathrm{cm}$, followed by increases to around $10\,\mathrm{cm}$ every time the execution speed was changed.

Note that the maneuver could only be learnt at $\lambda = 1$ by starting learning at lower values of λ and then transferring the learnt trajectory to higher values. As discussed for the fixed-speed experiments, an immediate execution at $\lambda = 1$ resulted in divergence of the error during learning.

VII. Advantages and Limitations

The experimental results in the previous section demonstrated the ability of the learning algorithm to significantly improve the tracking performance for periodic maneuvers. A key enabler for this is the underlying feedback control law that makes maneuver executions highly repeatable.

It can also be seen that a simplified model of the closed-loop dynamics, though only capturing a relatively rough approximation of the true behaviour, suffices to guide the learning process. Similar results have been demonstrated for other learning algorithms, e.g. reinforcement learning [25].

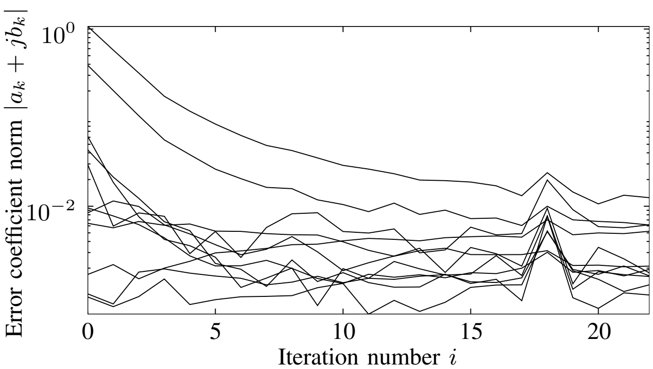

Fig. 4. Logarithmic plot of the evolution of error coefficient norms during the learning of the example maneuver at a fixed speed of $\lambda = 0.7$. Each line represents the norm of the three-dimensional error coefficients for one frequency. The lines at iteration zero are from top to bottom: $k = \{1, 2, 4, 5, 0, 6, 3, 7, 10, 8, 9\}$.

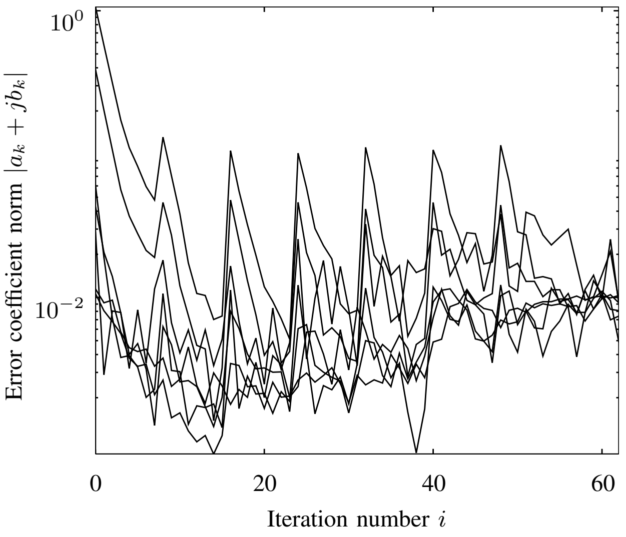

Fig. 5. Logarithmic plot of the evolution of error coefficient norms during learning with changes in execution speed. Each line represents the norm of the three-dimensional error coefficients for one frequency. The maneuver execution speed λ was increased by 0.05 every 8 iterations, starting at 0.7 (resulting in a maneuver period of $T = 4.7\,\mathrm{s}$) and ending at 1.0 ($T = 3.3\,\mathrm{s}$).

The analysis of the tracking error signal as a truncated Fourier series allows the use of the order of the series to define a trade-off between 1) the ability to compensate for highly localized tracking errors, and 2) computational complexity and memory requirements. The limitation to relatively low Fourier series orders also provides a convenient way to suppress high-frequency jitter in the learnt compensation, an effect that can be frequently observed in iterative learning control approaches [9]. Furthermore, inaccuracies of the dynamic model at high frequencies can be circumvented by limiting the learning to lower frequencies.

Due to the simplified dynamic model, the serial architecture, and the frequency domain approach in the learning algorithm, it is not trivial to extend the learning law to incorporate additional learning constraints such as input constraints or penalties. Large tracking errors can lead to behaviour that is not captured by the simplified model, causing the learning algorithm to fail.

The frequency domain approach to iterative learning also provides a convenient way to transfer learnt correction inputs between different execution speeds of the same maneuver. This allows initial learning to occur at reduced speeds, thus providing a safe way to learn high-speed maneuvers where a poor initial guess of the correction input can lead to a crash or to non-convergence.

VIII. Conclusion and Outlook

We have investigated a frequency-domain iterative learning scheme for periodic quadrocopter flight. The iteration-domain feedback law leverages the nominally linear time-invariant closed-loop dynamics of the quadrocopter feedback system in order to determine correction values from observed errors in a straightforward manner. The approach allows the learning of high performance maneuvers by executing them at a reduced speed initially, and then transferring learnt corrections to higher speeds.

Experimental results demonstrated the application of the learning scheme, and highlighted the advantages of learning at reduced speeds. An example figure-eight maneuver could only be learnt at full speed when using learnt parameters from lower speeds to initialize the learning process.

The transfer of learnt corrections between different execution speeds involves the estimation of how tracking errors depend on this speed. Due to the varied nature of possible error sources in quadrocopter flight (e.g., external disturbances such as wind or ground effects, sensor or actuator miscalibrations, or unmodeled system dynamics), this estimation is not a trivial task. The learning scheme presented herein is based on the assumption that the tracking error is essentially independent of execution speed.

It is obvious from experimental results – as well as from more accurate models of quadrocopter flight [26] – that the tracking error depends on the execution speed. It would therefore be useful to develop a means of predicting the change in tracking error when altering execution speed, for example by using knowledge of the dynamical model, or by extrapolating changes in tracking errors from previous speed changes.

The frequency domain approach used herein also lends itself to accounting for model uncertainty, which can often be directly converted to uncertainty in the transfer function (see, for example, [27]). This uncertainty could potentially be used to design a robust iterative learning scheme. For example, the iterative feedback correction gain could be chosen separately for varying frequencies. This would allow the fast compensation of errors at frequencies where the model is well known, and more conservative adaptation at frequencies with high uncertainty. The comparison of such a frequency-domain robust iterative learning control approach to other robust learning control formulations (see e.g. [7] for an overview) could provide further insight into the applicability of the algorithm under uncertainty.

Acknowledgments

This work is supported by and builds upon prior contributions by numerous collaborators in the Flying Machine Arena project. A list of past and present participants of the project is available at `http://bit.ly/RdO6g1`

This research was funded in part by the Swiss National Science Foundation (SNSF).

References

[1] 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 *International Conference on Robotics and Automation*, 2007.

[2] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Frandorfer, and M. Pollefeys, "PIXHAWK: A Micro Aerial Vehicle Design for Autonomous Flight Using Onboard Computer Vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.

[3] A. Kushleyev, D. Mellinger, and V. Kumar, "Towards A Swarm of Agile Micro Quadrotors," in *Robotics: Science and Systems*, 2012.

[4] S. Bouabdallah, P. Murrieri, and R. Siegwart, "Design and Control of an Indoor Micro Quadrotor," in *International Conference on Robotics and Automation*, 2004.

[5] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," *Control Engineering Practice*, vol. 19, no. 9, pp. 1023–1036, Sept. 2011.

[6] I. Chin, M. Cho, S. J. Qin, and K. S. Lee, "A Two-Stage Algorithm for Combined Iterative Learning Control With Real-Time Feedback; a State Space Formulation," in *IFAC World Congress*, 2005.

[7] D. A. Bristow, M. Tharayil, and A. G. Alleyne, "A Survey of Iterative Learning Control," *Control Systems Magazine*, vol. 26, no. 3, pp. 96–114, 2006.

[8] O. Purwin and R. D'Andrea, "Performing and Extending Aggressive Maneuvers Using Iterative Learning Control," *Robotics and Autonomous Systems*, vol. 59, no. 1, pp. 1–11, 2011.

[9] A. P. Schoellig, F. L. Mueller, and R. D'Andrea, "Optimization-Based Iterative Learning for Precise Quadrocopter Trajectory Tracking," *Autonomous Robots*, vol. 33, no. 1-2, pp. 103–127, 2012.

[10] S. L. Waslander, G. M. Hoffmann, J. S. Jang, and C. J. Tomlin, "Multi-Agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning," in *International Conference on Intelligent Robots and Systems*, 2005.

[11] T. Dierks and S. Jagannathan, "Output Feedback Control of a Quadrotor UAV Using Neural Networks." *IEEE transactions on neural networks*, vol. 21, no. 1, pp. 50–66, 2010.

[12] I. Palunko and R. Fierro, "Adaptive Control of a Quadrotor with Dynamic Changes in the Center of Gravity," in *IFAC World Congress*, 2011.

[13] D. Mellinger, N. Michael, and V. Kumar, "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors," in *International Symposium on Experimental Robotics*, 2010.

[14] S. Lupashin and R. D'Andrea, "Adaptive Fast Open-Loop Maneuvers for Quadrocopters," *Autonomous Robots*, vol. 33, no. 1-2, pp. 89–102, 2012.

[15] L. Cuiyan, Z. Dongchun, and Z. Xianyi, "A survey of repetitive control," in *International Conference on Intelligent Robots and Systems*, 2004.

[16] Y. Wang, F. Gao, and F. J. Doyle, "Survey on iterative learning control, repetitive control, and run-to-run control," *Journal of Process Control*, vol. 19, no. 10, pp. 1589–1600, Dec. 2009.

[17] P. C. Hughes, *Spacecraft Attitude Dynamics*. John Wiley & Sons, 1986.

[18] A. Schoellig, C. Wiltsche, and R. D'Andrea, "Feed-Forward Parameter Identification for Precise Periodic Quadrocopter Motions," in *American Control Conference*, 2012.

[19] F. L. Mueller, A. P. Schoellig, and R. D'Andrea, "Iterative learning of feed-forward corrections for high-performance tracking," in *International Conference on Intelligent Robots and Systems*, 2012.

[20] H. P. Hsu, *Schaum's Outline of Signals and Systems*. McGraw-Hill, 1995.

[21] S. Lupashin, A. P. Schoellig, M. Hehn, and R. D'Andrea, "The Flying Machine Arena as of 2010," in *International Conference on Robotics and Automation*, 2011.

[22] M. Hehn and R. D'Andrea, "A Flying Inverted Pendulum," in *International Conference on Robotics and Automation*, 2011.

[23] M. Hehn and R. D'Andrea, "Quadrocopter Trajectory Generation and Control," in *IFAC World Congress*, 2011.

[24] A. Schoellig, M. Hehn, S. Lupashin, and R. D'Andrea, "Feasiblity of Motion Primitives for Choreographed Quadrocopter Flight," in *American Control Conference*, 2011.

[25] J. Z. Kolter and A. Y. Ng, "Policy Search via the Signed Derivative," in *Robotics: Science and Systems*, 2009.

[26] P. Pounds, R. Mahony, and P. Corke, "Modelling and Control of a Quad-Rotor Robot," in *Australasian Conference on Robotics and Automation*, 2006.

[27] J. Doyle, "Analysis of feedback systems with structured uncertainties," *IEE Proceedings, Part D*, vol. 129, no. 6, pp. 242–250, 1982.