Mechatronics 24 (2014) 954–965

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

A frequency domain iterative learning algorithm for high-performance, periodic quadrocopter maneuvers

Markus Hehn *, Raffaello D'Andrea

Institute for Dynamic Systems and Control, ETH Zurich, Switzerland

ARTICLE INFO

Article history:
Received 2 September 2013
Revised 13 September 2014
Accepted 28 September 2014
Available online 23 October 2014

Keywords:
Learning control
Quadrocopter control
Frequency domain learning
Repetitive control
Unmanned aerial vehicle control

ABSTRACT

Quadrocopters offer an attractive platform for aerial robotic applications due to, amongst others, their hovering capability and large dynamic potential. Their high-speed flight dynamics are complex, however, and the modeling thereof has proven difficult. Control algorithms typically rely on simplified models, with feedback corrections compensating for unmodeled effects. This can lead to significant tracking errors during high-performance flight, and repeated execution typically leads to a large part of the tracking errors being repeated. This paper introduces an iterative learning scheme that non-causally compensates repeatable trajectory tracking errors during the repeated execution of periodic flight maneuvers. An underlying feedback control loop is leveraged by using its set point as a learning input, increasing repeatability and simplifying the dynamics considered in the learning algorithm. The learning is carried out in the frequency domain, and is based on a Fourier series decomposition of the input and output signals. The resulting algorithm requires little computational power and memory, and its convergence properties under process and measurement noise are shown. Furthermore, a time scaling method allows the transfer of learnt maneuvers to different execution speeds through a prediction of the disturbance change. This allows the initial learning to occur at reduced speeds, and thereby extends the applicability of the algorithm for high-performance maneuvers. The presented methods are validated in experiments, with a quadrocopter flying a figure-eight maneuver at high speed. The experimental results highlight the effectiveness of the approach, with the tracking errors after learning being similar in magnitude to the repeatability of the system.

1. Introduction

Aerial robots serve as platforms for robotic applications that provide numerous benefits, including the ability to move freely in three-dimensional space, and the significantly increased ability to overcome obstacles due to not being limited to motion on the ground. For relatively small platforms that require hovering capabilities, multi-rotor vehicles such as quadrocopters are often the vehicle of choice [1]. Compared to other such platforms, quadrocopters profit from high mechanical robustness due to a minimal number of moving parts [2], safety due to comparatively small rotor size, and high thrust-to-weight ratios allowing high-performance maneuvers as well as the transport of large payloads.

While the use of quadrocopters as robotic platforms was largely confined to research institutions in the past, a growing number of industrial applications are now in the process of being developed and deployed. Examples include aerial imaging for photogrammetry, motion picture production, and journalism [3], environmental monitoring and inspection tasks of hard-to-reach objects such as pipelines, dams, and power lines [4], the creation of ad hoc antenna networks or arrays [5], as well as disaster coordination [6].

The capability of quadrocopters to perform highly dynamic, complex, and precise motions has been demonstrated repeatedly in recent years (see, for example, Mellinger et al. [7], Michael et al. [8], Muller et al. [9], Ritz et al. [10]). In order to execute such high-performance motions, the commonly used approach consists of using a first-principles model of the quadrotor dynamics to design the nominal maneuver, and a model-based feedback control law to ensure tracking of the nominal trajectory.

Such traditional feedback controllers however have important limitations in high-performance quadrotor applications. While the first-principles models used to design the controllers capture the near-hover behavior of quadrocopters well, secondary effects become increasingly important when maneuvering speed increases. Examples of such effects are the complex drag and lift behavior of rotary wings under unsteady inflow conditions [11],

* Corresponding author.
E-mail address: hehnm@alumni.ethz.ch (M. Hehn).

http://dx.doi.org/10.1016/j.mechatronics.2014.09.013
0957-4158/

the aerodynamic effects of a vehicle moving through the turbulent wake of its propellers [12], and external influences such as wind or ground and wall effects when operating in proximity to the environment [13]. Such effects are not typically accounted for in the maneuver and controller synthesis stage in order to make the design process tractable. The execution then heavily relies on the feedback controller to compensate for potentially significant effects not captured by the nominal dynamics.

In order to improve the tracking performance of quadrocopters under feedback control, a number of researchers have proposed learning schemes. Examples of such schemes include those based on reinforcement learning techniques [14,15] and neural networks [16,17], which are designed to automatically find well-performing control policies, and adaptive control methods [18–20] that adapt parameters that are based on modeled disturbances such as payloads, center of mass shifts and external disturbances.

When a motion is to be executed repeatedly, a further opportunity to improve tracking performance may arise: Many of the disturbances that degrade tracking performance will be similar each time the vehicle performs the motion. These disturbances can then be compensated for non-causally using data from past executions. Control strategies that exploit available data from past executions in order to improve tracking performance were first proposed in the late 1970s and early 1980s [21,22] for applications in motion control and power supply control. Since then, active research in this field, covering numerous applications and problem formulations (see e.g. Wang et al. [23], Bristow et al. [24], Cuiyan et al. [25], and references therein), has shown it to be a powerful approach for high-performance reference tracking. In extensions to these learning methods, several authors have shown the application of learning control methods to systems with underlying feedback control loops (e.g. [26,27]). In such scenarios, the powerful capability of learning control to non-causally compensate repeatable disturbances is combined with real-time feedback control to correct for non-repetitive noise.

While the application of learning algorithms, and specifically non-causal strategies, to stationary systems (such as rotating machinery and robotic arms [23]) is well-established, its use for the compensation of complex aerodynamic effects in flying vehicles is less mature and has been actively researched during recent years. Several high-performance maneuvers for multi-rotor vehicles have been demonstrated with the use of learning algorithms. Broadly speaking, the learning approaches used can be categorized in two groups:

The first group is characterized by its ability to learn motions that are parameterized. The motion is thus described by a (finite) set of design parameters, chosen by the user. After the execution of the motion, these parameters are adapted to compensate for disturbances. The direction and magnitude of the correction may be model-based, or based on the user's intuition. A discussion on the importance of choosing 'good' design parameters may be found in Lupashin and D'Andrea [28], where a learning algorithm for this kind of parameterized motions is demonstrated for multiple flips and fast translations with quadrocopters. A further demonstration of this class of learning algorithms is provided in Mellinger et al. [29]. The ability to shape the tracking performance strongly depends on the number of parameters that are optimized; in the above examples, the objective is to minimize the error at specific time instants ('key frames'), and a relatively small number of parameters is sufficient to do so. This makes the methods computationally lightweight.

The second group of learning approaches considers more generic motions that need not be specified by parameters. The system dynamics are considered in discrete time, and the correction consists of correction values (typically control inputs or set points) for each discrete time step. After execution of the motion, a numerical optimization over the correction values is performed in order to minimize a metric related to the tracking error. In this optimization, a model of the system dynamics provides the mapping from corrections to the tracking error. This approach is commonly known as a form of iterative learning control [24], and its application to high-performance quadrocopter flight has been demonstrated [30–33].

The delimitation between the two groups is not strict. Indeed, the second group of learning approaches could be seen as using a very large number of values to parameterize the correction.

The algorithm presented in this paper can be characterized to be a form of repetitive control [23] in that it is a technique for non-causally compensating repeated tracking errors in the execution of periodic motions. Algorithms of this form have previously shown good performance when applied to related problems where aerodynamic disturbances are considered, in particular the rejection of periodic wind disturbances on wind turbines [34,35].

Similar methods can also be found in the field of time waveform replication, as commonly applied to vibration testing systems (e.g., [36] and references therein). In such applications, the first-principles models guiding the iterative learning process are often replaced by experimentally identified frequency response functions.

Similar to the second group of learning algorithms, we do not assume a parameterized motion. However, we reduce the dimensionality of the corrections that we intend to learn by assuming that they are periodic. This allows us to parameterize the corrections as the coefficients of truncated Fourier series. The order of the Fourier series provides a means to trade off computational complexity and the ability to compensate for temporally local or high-frequency disturbances. Furthermore, the approach can be considered to be conceptually similar to the one presented by Lupashin and D'Andrea [28], which presents an adaptation strategy to correct for state errors at discrete points in time of parameterized motion primitives. However, we consider periodic errors (instead of errors at specific points in time), and do not require parameterization of the maneuver.

The contribution of this paper to the field of quadrocopter control lies in the application of methods from the fields of repetitive control and iterative learning control to quadrocopters. A general framework for arbitrary periodic motions is presented. We demonstrate how a feedback controller can be leveraged to shape the closed-loop dynamics of the quadrotor system, and show that a linear time-invariant approximation of the closed-loop dynamics suffices to guide the learning process. Using statistical properties of the disturbance, measurement noise, and the influence of nonlinearities, we derive the optimal inter-execution learning update step size. The validity of the approach and its performance is investigated through experiments in the ETH Flying Machine Arena with a quadrocopter under position control.

Furthermore, this paper introduces a novel method that extends the applicability of the repetitive control approach when the reference trajectory is too fast to be learnt directly, for example because the initial execution fails entirely. The core idea here lies in providing an improved initial guess of the disturbances degrading tracking performance. This typically enables learning of the trajectory because the errors are sufficiently small for the first-principles model to provide reliable information on how to compensate. To find the improved initial guess, we introduce a time scaling method that allows initial learning to occur at reduced maneuvering speeds and the transfer of learnt corrections from the reduced-speed execution to full speed. This method may also be applied to other complex dynamic systems where it is necessary to limit initial tracking errors in order to avoid the system failing. The time-scaling method provides an interesting alternative to methods that rely on aborting trials when the errors grow too large [31] in that

the tracking error is always learnt over the entirety of the maneuver, and over more general methods to extend the motion [30] due to its computational simplicity.

Preliminary results of this method were presented at international conferences [37,38], and this paper extends these results through the computation of the optimal learning rate for given statistical properties of the process, a novel way to predict disturbances when transferring knowledge between different execution speeds, as well as providing an in-depth experimental analysis of the method.

The remainder of this paper is structured as follows: We introduce the model of the quadrocopter and the used feedback law in Section 2. The learning algorithm is then presented and analyzed in Section 3. Section 4 presents experimental results highlighting the performance of the algorithm. Section 5 discusses advantages and restrictions of the learning algorithm, and Section 6 provides a conclusion.

2. Quadrocopter dynamics and closed-loop control

This section first introduces the first-principles model of a quadrocopter, along with a discussion of the accuracy of the model. Furthermore, we introduce the input–output linearizing feedback controller used to control the vehicle. The combination of vehicle and feedback controller form the closed-loop dynamics that the iterative learning algorithm is then applied to. For ease of notation, vectors are expressed as n-tuples $(x_1, x_2, \ldots)$ where convenient, with dimension and stacking clear from context.

2.1. Quadrocopter 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 Fig. 1.

The quadrotor vehicle incorporates four actuators, consisting of motors with fixed-pitch propellers. Each motor produces a thrust force and a drag torque, and the resulting rotational dynamics in the body-fixed coordinate frame V are

$$I\dot{\omega} = \begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ \zeta(F_1 - F_2 + F_3 - F_4) \end{bmatrix} - \omega \times I\omega \tag{1}$$

where $\omega = (\omega_x, \omega_y, \omega_z)$ is the rotational rate of the vehicle, I is its rotational inertia, L the arm length of the vehicle, ζ the drag-to-thrust ratio of the propeller, and F_1 to F_4 are the individual thrust forces of each propeller. The total mass-normalized thrust produced by the four propellers is

$$a = \frac{1}{m}(F_1 + F_2 + F_3 + F_4) \tag{2}$$

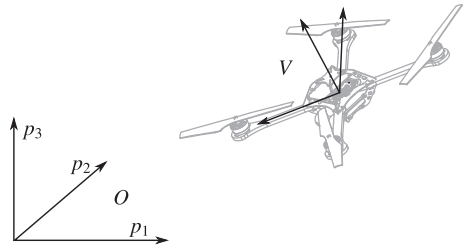

Fig. 1. The inertial coordinate system O and the vehicle coordinate system V, used to describe the dynamics of the quadrocopter.

where m denotes the mass of the vehicle. The rotational kinematics are given by the first-order differential equation of the rotation matrix

$${}^O_V\dot{R} = {}^O_V R \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{3}$$

The translational dynamics of the vehicle, expressed in the inertial coordinate system O, are

$$\begin{bmatrix} \ddot{p}_1 \\ \ddot{p}_2 \\ \ddot{p}_3 \end{bmatrix} = {}^O_V R \begin{bmatrix} 0 \\ 0 \\ a \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{4}$$

where g denotes gravitational acceleration.

This first-principles model of the quadrocopter rotational and translational dynamics is commonly used to design and analyze algorithms such as feedback control laws and path planners (see, for example, Mahony et al. [39] and references therein), and captures near-hover dynamics well. When higher maneuvering speeds are reached, however, a multitude of additional – mainly aerodynamic – effects become more significant. A number of these effects have been identified and incorporated into more refined models; for example, they include induced translational and rotational rotor drag [11], blade flapping [40], and dominant propellers tip vortices during vertical descent flight [12]. Furthermore, the model neglects potential interactions of the vehicle with the environment. For example, it is well known that flight dynamics are influenced significantly by ground and wall effects [13], fast maneuvers can cause the vehicle to fly through its own wake, and external disturbances such as wind effects [41] can cause large disturbances.

While the modeling of such effects has provided valuable insight, their incorporation into control strategies has generally been slow due to the highly complex nonlinear models making controller design significantly more difficult, and the added difficulty of identifying the parameters of such models. Instead, most control laws treat such effects as disturbances, relying on feedback control to account for them implicitly. In this paper, we will follow a similar approach in that we do not model the effects, but we will compensate for them non-causally during the repeated execution of periodic motions.

2.2. 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 was described and analyzed in more detail in Schoellig et al. [42], and is taken as a given in this paper. It consists of cascaded feedback linearizing control loops for position, attitude, and rotational rates as follows:

2.2.1. Position control

For all three degrees of freedom, a feedback control law determines the desired acceleration $\hat{\ddot{p}}_i$ 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:

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

where $\hat{p}_i$ is the commanded position.

With R_{xy} denoting the xth element of the yth column of ${}^O_V R$, the thrust is computed to enforce the desired vertical acceleration according to (4):

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

2.2.2. Reduced attitude control

The desired rotation matrix entries $\widehat{R}_{13}$ and $\widehat{R}_{23}$ for the given desired accelerations are computed from (4)–(6) to be

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

The attitude control loop is shaped such that the rotation matrix entries R_{13} and R_{23} (which govern the translational dynamics (4)) react in the manner of a first-order system with time constant τ_{rp} by computing the desired derivative of the rotation matrix elements:

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

Inverting the rotational kinematics (3), this is converted to the commanded rotational rates about the first two body axes of the vehicle:

$$\begin{bmatrix} \hat{\omega}_x \\ \hat{\omega}_y \end{bmatrix} = \frac{1}{R_{33}} \begin{bmatrix} R_{21} & -R_{11} \\ R_{22} & -R_{12} \end{bmatrix} \begin{bmatrix} \hat{\dot{R}}_{13} \\ \hat{\dot{R}}_{23} \end{bmatrix} \tag{9}$$

The commanded rotational rate about the third body axis, $\hat{\omega}_z$, 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.

2.2.3. Body rate control

Using the desired body rates $\hat{\omega}_x$, $\hat{\omega}_y$, $\hat{\omega}_z$ as commands, the body rate controller is designed to follow the commands in the fashion of three decoupled first-order systems. In order to achieve this, the rotational dynamics (1) are inverted:

$$\begin{bmatrix} L(F_2 - F_4) \\ L(F_3 - F_1) \\ \zeta(F_1 - F_2 + F_3 - F_4) \end{bmatrix} = I \begin{bmatrix} (\hat{\omega}_x - \omega_x)/\tau_{pq} \\ (\hat{\omega}_y - \omega_y)/\tau_{pq} \\ (\hat{\omega}_z - \omega_z)/\tau_r \end{bmatrix} + \omega \times I\omega \tag{10}$$

with the time constants τ_{pq} for the first two axes, and τ_r for the third axis. The above equation, in combination with the total thrust constraint resulting from combining equations (2) and (6), define the four individual propeller forces F_1 to F_4, and thereby complete the feedback control design.

2.2.4. Trajectory feed-forward

The presented controller can readily be augmented to apply feed-forward velocities, accelerations, rotation rates, and rotational accelerations for known input trajectories by extending Eqs. (5), (8), (9), and (10) with the corresponding feed-forward terms derived from the derivatives of the nominal position trajectory $\hat{p}(t)$. A discussion of the effects and performance benefits thereof can be found in Mueller et al. [32], where it is shown that feed-forward commands can improve tracking performance, though large systematic errors remain. This can be explained by the fact that these feed-forward terms are model-based, and the unmodeled effects discussed in Section 2.1 cause significant model mismatch that is not accounted for.

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 almost entirely. Indeed, the learning algorithm can be considered to be providing the necessary feed-forward signal to make the system track the reference trajectory accurately, and thereby also captures conventional feed-forward terms.

2.3. Approximate closed-loop system dynamics

The learning algorithm that will be introduced in Section 3 relies on an approximation of the system dynamics in the form of a linear time-invariant (LTI) system.

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., $\tau_{rp} \ll \tau_{12}$ and $\tau_{pq} \ll \tau_{rp}$) 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 LTI second-order systems:

$$\ddot{p}_i \approx \frac{1}{\tau_i^2}(\hat{p}_i - p_i) - 2\frac{\zeta_i}{\tau_i}\dot{p}_i \quad \text{for } i = \{1,2,3\} \tag{11}$$

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 as long as very high frequencies are not considered.

3. Learning algorithm

This section introduces the learning algorithm that is applied to the quadrocopter system in order to compensate for systematic disturbances that deteriorate flight performance during the execution of periodic motions. The fundamental idea is to use data from past executions in order to characterize the tracking errors, and to then compensate for them in a non-causal manner during following executions. For this compensation, we leverage the prior knowledge of the dominant dynamics of the quadrocopter under feedback control, and combine this knowledge with measurement data from experiments in order to determine appropriate compensations to apply during the next execution. Compared to the use of pure feedback control, this scheme can improve the tracking performance because repeated disturbances are compensated for non-causally, whereas pure feedback control is limited to causal corrections.

The basic system structure used in the learning algorithm is depicted in Fig. 2. The closed-loop dynamics of the quadrocopter under feedback control remain unmodified by the learning algorithm. We use the approach of adapting the set point of the controller, also called a serial architecture [24] or indirect

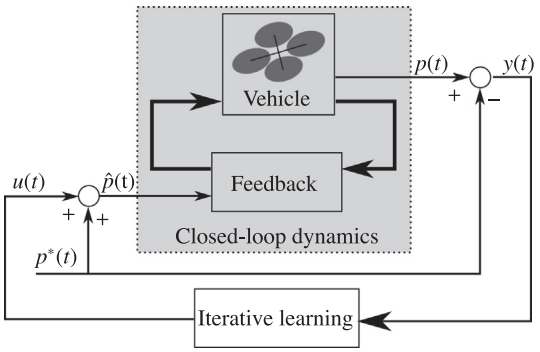

Fig. 2. The learning architecture. The iteration-domain feedback law uses the tracking error as input, and its output is a set point shift to the real-time feedback law.

learning-type control [23,43]. The controller set point is augmented by adding a correction input $u(t)$ to it. We assume that we can derive a linear time-invariant (LTI) approximative model of the closed-loop dynamics between the position tracking error and the correction input (as done in Section 2.3 for the presented feedback control law). In comparison to modifying the control inputs of the quadrocopter directly, the serial architecture offers the advantage that the dynamics from a change in the set point to a change in the tracking error output are those of the closed loop system, which are designed to be approximately LTI. Due to the periodic nature of the motions addressed in this paper, we use Fourier series [44] to characterize correction input and tracking error output signals of the system.

The learning algorithm builds upon this LTI model when interpreting the measurement data of an execution in order to determine the appropriate correction input signal for the next execution, as detailed in Section 3.2. Using the statistical properties of the relevant disturbance signals, it is possible to derive the optimal correction input signal in a least-squares sense (Section 3.3). Safer learning of high-performance maneuvers can be achieved by reducing the execution speed during the initial learning phase, and using the disturbance characteristics identified at lower speeds to provide an improved disturbance prediction at high speeds (Section 3.4).

3.1. System model

This section introduces the quantities used by the learning algorithm. Because the maneuvers considered herein are of periodic nature, time signals are parameterized as Fourier series with fundamental frequency $\Omega_0 = 2\pi/T$, with T being the period of the maneuver. Furthermore, the notation $(\cdot)^i$ is used to denote quantities of iteration i of the learning algorithm. Unless stated otherwise, we assume that time signals are periodic with period T.

The error output measurement $y(t)$ is the signal capturing the tracking errors which should be eliminated by the learning algorithm. The dimension of $y(t)$ is not defined by the problem, and may be chosen by the user as the set of error outputs that should be minimized. For the specific implementation considered in this paper, it consists of the three-dimensional deviation of the vehicle position from its reference trajectory $p^*(t)$, as seen in Fig. 2. We assume that the measured output of an experiment can be decomposed as follows, where the components will be explained in the following:

$$y^i(t) = d^i(t) + h^i(t) + n^i(t) \tag{12}$$

The first component $d^i(t)$ represents the systematic tracking error that should be compensated. It is not known a priori, but we will later assume that a statistical description of $d^0(t)$ is available for the derivation of the optimal learning step size in Section 3.3.3. Specifically, we will then assume that its mean is zero, and that it is stationary with a known autocorrelation function. The evolution of the tracking error over several iterations is modeled as

$$d^{i+1}(t) = d^i(t) + \gamma^i(t) \tag{13}$$

where $\gamma^i(t)$ models slight changes to the disturbance from iteration to iteration as a trial-uncorrelated sequence of zero-mean stationary noise, where we will again assume a known autocorrelation function in Section 3.3.3.

The second component, $h^i(t)$, is the response of the closed-loop dynamic system to the correction input $u^i(t)$, characterized in the frequency domain by

$$H^i(\omega) = G(\omega)U^i(\omega) \tag{14}$$

with $G(\omega)$ being the transfer function [45] of the LTI approximation of the closed-loop system, and $H^i(\omega)$ and $U^i(\omega)$ being the frequency domain representations of $h^i(t)$ and $u^i(t)$, respectively. The final component of the tracking error measurement, $n^i(t)$, represents trial-uncorrelated, non-periodic noise. It is assumed to vary for each execution of the maneuver, and captures effects such as non-repeatable disturbances to the vehicle (for example wind gusts), but also measurement noise. The non-repeatable disturbances are assumed to be zero mean; a non-zero mean would represent a repeatable disturbance and would therefore be accounted for by $d^i(t)$. Furthermore, we assume stationarity, and for the derivation of the optimal learning step size a known autocorrelation function.

The goal of the learning algorithm is to choose the correction input $u^i(t)$ such that the output $y^i(t)$ is minimized. The desired output is assumed to be periodic with period T, and the algorithm will use measurements of previous executions $y^{i-1}(t), y^{i-2}(t), \ldots$ in order to determine $u^i(t)$. Intuitively, this task implies finding the correction input $u^i(t)$ such that $h^i(t) + d^i(t)$ is minimized, meaning that the unwanted repeated disturbance is canceled out in the output (12) as well as possible.

Note that the dimensions of $u(t)$ and $y(t)$ are not necessarily given by the system model, but can be chosen by the user to indicate which signals are relevant to the tracking task at hand, and which quantities can be modified. We will, however, assume that the dimension of $u(t)$ is no less than that of $y(t)$, implying that we have at least as many compensation inputs available as we have error quantities. If this is not the case, full tracking can typically not be achieved, although it can be shown that the tracking error can still be reduced [37]. In the specific implementation of the algorithm considered in this paper, $u(t)$ is a three-dimensional additive correction to the position control reference point and $y(t)$ is the three-dimensional deviation of the vehicle from the reference trajectory, as seen in Fig. 2.

3.2. Learning update law

Due to the periodic nature of the maneuvers considered herein, we will leverage Fourier series decompositions of correction input and error output signals. We parameterize the correction input by a truncated Fourier series of order N and fundamental frequency Ω_0:

$$u^i(t) = r_0^i + \sum_{k=1}^{N} r_k^i \cos(k\Omega_0 t) + \sum_{k=1}^{N} s_k^i \sin(k\Omega_0 t) \tag{15}$$

The frequency domain representation of $u(t)$ is then

$$U^i(0) = r_0^i \tag{16}$$

$$U^i(k\Omega_0) = r_k^i - j s_k^i \quad \text{for } k = 1, 2, \ldots, N \tag{17}$$

The system response $h^i(t)$ to this Fourier series $u^i(t)$ is also a Fourier series of order N [44], which is defined in the frequency domain through the relationship

$$H(k\Omega_0) = G(k\Omega_0)U(k\Omega_0) \quad \text{for } k = 0, 1, \ldots, N \tag{18}$$

We will now invert this relationship in order to use it as a learning feedback law that compensates a given disturbance. Assuming that $G(k\Omega_0)$ is full rank, let $G^+(k\Omega_0)$ denote the Moore–Penrose pseudoinverse [46] of $G(k\Omega_0)$ (in the special case that G is square, $G^+(k\Omega_0) = G^{-1}(k\Omega_0)$ holds).

Assume that we have executed the trajectory for iteration i, and measured the tracking error $y^i(t)$. Let $Y^i(k\Omega_0)$ denote the Fourier series that coincides with $y^i(t)$ for $t \in [0\ T]$. The iteration feedback law is then given by the correction Fourier series

$$U^{i+1}(k\Omega_0) = U^i(k\Omega_0) - \mu_k^i G^+(k\Omega_0) Y^i(k\Omega_0) \tag{19}$$

for $k = 0, \ldots, N$, with the step size μ_k^i controlling the adaptation rate. The time signal $u^{i+1}(t)$ is then constructed from $U^{i+1}(k\Omega_0)$ and applied to the system in the next iteration.

This learning update law corresponds closely to plant inversion methods in iterative learning control methods with an additional step size parameter, the properties of which are discussed in Bristow et al. [24]. The limitation of the learning process to the Fourier series order N can be considered as the inclusion of an ideal low pass filter, similar to the ones used in plant inversion methods to manage high-frequency uncertainty.

3.3. Learning step size

The remaining degree of freedom in the learning update law (19) is the step size μ_k^i. Depending on the application scenario and, more specifically, the availability of the statistical properties of the disturbance signals $d^0(t)$, $n^i(t)$ and $\gamma^i(t)$, the step size μ_k^i may be considered to be a tuning parameter that is directly set by the user, or its optimal value may be computed from the available statistical properties.

3.3.1. Noise influence on Fourier coefficients

In practice, the Fourier series coefficients of $y^i(t)$ are estimated from a set of discrete-time observations of the continuous-time signal, which are used to compute a discrete Fourier Transform [47]. The influence of noise such as $n^i(t)$ on the coefficients of a discrete Fourier Transform has been studied in the past [48]: It was shown that, for a sufficiently large number of samples of $y^i(t)$, the noise on the Fourier coefficients will be additive, approximately Gaussian regardless of the distribution of the noise, and zero mean. A more rigorous description of the statistical properties of the coefficients can be found in Schoukens and Renneboog [48], and the variance of the individual coefficients can be computed from the correlation function of $n^i(t)$. The specific equations are not repeated here, but it is sufficient for our purposes to state that the variance of the coefficients is known, and that they are approximately uncorrelated for a sufficiently large number of discrete-time observations.

Using $N^i(k\Omega_0)$ to denote the additive zero-mean coefficient noise caused by $n^i(t)$, the tracking error Fourier series coefficients at iteration $i+1$ can be written as

$$Y^{i+1}(k\Omega_0) = D^{i+1}(k\Omega_0) + H^{i+1}(k\Omega_0) + N^{i+1}(k\Omega_0) \tag{20}$$

Dropping the argument $k\Omega_0$ for notational convenience (all quantities in the following are parameterized by $k\Omega_0$), we expand the above using Eqs. (13), (18), and (19):

$$\begin{aligned} Y^{i+1} &= D^{i+1} + GU^{i+1} + N^{i+1} && (21)\\ &= D^i + \Gamma^i + G\left(U^i - \mu_k^i G^+ Y^i\right) + N^{i+1} && (22)\end{aligned}$$

where Γ^i, analogously to N^{i+1}, is zero-mean coefficient noise caused by the disturbance change $\gamma^i(t)$ as defined in Eq. (13).

3.3.2. Stability

We first consider μ_k^i a tuning parameter, and derive stability conditions for constant values $\mu_k^i = \bar{\mu}_k$. Assuming an accurate model of the closed-loop dynamics, G^+ (constructed according to Section 3.2 and used in the learning update law) is the right inverse of the closed-loop dynamics G. We rewrite Eq. (22) using Equations (18), (20), and this property:

$$\begin{aligned} Y^{i+1} &= D^i + \Gamma^i + GU^i - \bar{\mu}_k Y^i + N^{i+1} && (23)\\ &= \Gamma^i + (1-\bar{\mu}_k)Y^i - N^i + N^{i+1} && (24)\end{aligned}$$

The expected value of the Fourier coefficients is then

$$E[Y^{i+1}] = (1-\bar{\mu}_k)E[Y^i] \tag{25}$$

since the noise terms are zero mean as discussed in Section 3.3.1. Note that

$$E[Y^0] = 0 \tag{26}$$

since it is assumed that D^0 is zero mean and the expected value of the Fourier coefficients Y^i is therefore zero for all i. If the algorithm is incorrectly initialized or D^0 is not zero mean, the tracking error still converges to zero in expectation for step sizes $0 < \bar{\gamma}_k \leqslant 1$. Furthermore, the variance of the Fourier coefficients can be computed by rewriting Eq. (23) using (18) and (20):

$$Y^{i+1} = (1-\mu_k^i)(D^i + H^i) + \Gamma^i - \mu_k^i N^i + N^{i+1} \tag{27}$$

Taking the variance and using the fact that $(D^i + H^i)$, Γ^i, N^i and N^{i+1} are independent, it follows that

$$\begin{aligned} \mathrm{Var}[Y^{i+1}] &= (1-\bar{\mu}_k)^2\mathrm{Var}[D^i+H^i] + \mathrm{Var}[\Gamma^i] + \bar{\mu}_k^2\mathrm{Var}[N^i] + \mathrm{Var}[N^{i+1}] && (28)\\ &= (1-\bar{\mu}_k)^2\left(\mathrm{Var}\left[D^i+H^i\right] + \mathrm{Var}\left[N^i\right]\right)\\ &\quad + \mathrm{Var}\left[\Gamma^i\right] + (2\bar{\mu}_k - 1)\mathrm{Var}\left[N^i\right] + \mathrm{Var}\left[N^{i+1}\right] && (29)\\ &= (1-\bar{\mu}_k)^2\mathrm{Var}\left[Y^i\right] + \mathrm{Var}\left[\Gamma^i\right] + (2\bar{\mu}_k - 1)\mathrm{Var}\left[N^i\right] + \mathrm{Var}\left[N^{i+1}\right] && (30)\end{aligned}$$

from which it follows that the variance does not diverge for step sizes $0 < \bar{\mu}_k \leqslant 1$, assuming bounded variance of N and Γ.

Because the Fourier series coefficients are approximately Gaussian, their mean and variance fully describe their approximate distribution. Assuming that the approximation as a Gaussian distribution is sufficiently accurate, the learning update law results in the Fourier series coefficients of the tracking error being zero mean with bounded variance for constant step sizes $0 < \bar{\mu}_k \leqslant 1$.

This result only holds for the Fourier series coefficients $k = 0, 1, \ldots, N$ of $Y(k\Omega_0)$. No adaptation occurs for higher-frequency components, and the coefficient dynamics for $k > N$ are therefore simply those of Eq. (22) without the input:

$$Y^{i+1} = D^i + \Gamma^i + N^{i+1} \tag{31}$$

3.3.3. Minimum mean square error step size

We now consider time-varying step sizes μ_k^i, and derive the step size sequence that minimizes the trace of the tracking output variance, also called the minimum mean square estimate [49]. Computing the trace of the variance analogously to Eq. (30) from Eq. (27), and setting the derivative with respect to μ_k^i to zero, it follows that the optimal step size $\overset{*}{\mu}{}_k^i$ is

$$\overset{*}{\mu}{}_k^i = \frac{\mathrm{Tr}\left[\mathrm{Var}\left[Y^i(k\Omega_0)\right]\right] - \mathrm{Tr}\left[\mathrm{Var}\left[N^i(k\Omega_0)\right]\right]}{\mathrm{Tr}\left[\mathrm{Var}\left[Y^i(k\Omega_0)\right]\right]} \tag{32}$$

where $\mathrm{Tr}[\cdot]$ denotes the trace operator, and the argument $(k\Omega_0)$ is stated explicitly in order to highlight that the optimal step size may differ for each multiple of the fundamental frequency. The output variance $\mathrm{Var}[Y^i]$ can be computed recursively from Eq. (27), starting with the statistical properties of $d^0(t)$ and $n^0(t)$ to determine $\mathrm{Var}[Y^0]$. Note that the step size sequence $\overset{*}{\mu}{}_k^i$ only depends on the statistical properties of the random signals $d^0(t)$, $\gamma^i(t)$ and $n^i(t)$ and not on their actual realizations, and can therefore be entirely precomputed before starting experiments if desired.

3.4. Time scaling for high performance maneuvers

When initializing the learning of a maneuver, an initial guess of the compensation input $u^0(t)$ is used to execute the first iteration. When no other information is available, a typical 'naive' choice

would be to simply choose $u^0(t) = 0$. However, for high performance motions that approach the feasibility limits of the system (e.g., due to actuator saturation), the naive initial guess may be so poor that it becomes impossible to apply the learning algorithm successfully. This may be caused, for example, by the tracking errors growing 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 (18) used to compute the correction input no longer accurately predict the behavior of the closed-loop control system, leading to instabilities in the learning algorithm.

To allow the algorithm to be applied to such maneuvers, we extend it with a new time scaling method: The execution speed of the maneuver is controlled by introducing a speed scaling factor λ. The core idea is that many motions become 'easier' to execute when the motion duration is increased (some examples of this were given in Schoellig et al. [50] and include constant-altitude side-to-side and circular motions), where easier loosely refers to the amplitude of the required control inputs.[1] The speed scaling factor allows the motion to be initially executed and improved at relatively low speeds, where there is no danger of collisions and the learning algorithm works reliably, and to then use the learnt compensation inputs to generate an improved initial guess of the compensation input $u(t)$ for higher execution speeds. Since the initial tracking errors should be lower due to the better initial guess, it is more likely for the learning algorithm to successfully compensate for errors as the initial execution is more likely to be successful, and the approximate dynamics (18) are then only used to compensate for relatively small errors.

We define the scaled maneuver duration

$$T_\ell = \frac{T}{\lambda_\ell} \tag{33}$$

and assume that we have successfully learnt the motion for the execution speed λ_1, i.e. the tracking error output $y_1(t)$ associated with the execution speed λ_1 is sufficiently small for all $t_1 \in [0\ T_1]$. The objective is then to use the learnt compensation input at speed λ_1, which we denote $u_1(t)$, to initialize the compensation input for the (typically higher) execution speed λ_2, which we analogously denote $u_2(t)$.

Ideally, the values for $u_2(t)$ should result in a tracking error at the new execution speed $y_2(t)$ that is as small as $y_1(t)$. This would require knowing how the disturbance $d(t)$ changes with the time scaling, such that $u(t)$ can be chosen accordingly as seen in Eq. (12). However, due to $d(t)$ capturing unmodeled disturbances, a model of its dependence on execution speed is not readily available.

An obvious choice for the transfer between two maneuver speeds is to keep the learnt input corrections coefficients and simply re-map them to the corresponding frequencies (i.e., $U_2(k\lambda_2\Omega_0) = U_1(k\lambda_1\Omega_0)$). However, the varying sensitivity of the closed-loop transfer function at the two different frequencies, as well as the changing disturbances, could potentially lead to large errors.

In order to account for this, we use a linear extrapolation method to predict the correct value of the compensation term $H(\lambda k\Omega_0)$ in Eq. (20) from past time scaling changes. The first-order prediction of $H(\lambda_3 k\Omega_0)$ for the execution speed λ_3 from the past execution speeds λ_1 and λ_2 is then

$$H(\lambda_3 k\Omega_0) = H(\lambda_2 k\Omega_0) + \frac{\lambda_3 - \lambda_2}{\lambda_2 - \lambda_1}\left(H(\lambda_2 k\Omega_0) - H(\lambda_1 k\Omega_0)\right) \tag{34}$$

which we can expand using Eq. (14) to find the initial guess for the correction Fourier series coefficients:

$$U(\lambda_3 k\Omega_0) = G^+(\lambda_3 k\Omega_0)\left(G(\lambda_2 k\Omega_0)U(\lambda_2 k\Omega_0)\left(1 + \frac{\lambda_3 - \lambda_2}{\lambda_2 - \lambda_1}\right) - G(\lambda_1 k\Omega_0)U(\lambda_1 k\Omega_0)\frac{\lambda_3 - \lambda_2}{\lambda_2 - \lambda_1}\right) \tag{35}$$

The prediction requires the storage of the learnt correction series coefficients of the past two execution speeds ($U(\lambda_1 k\Omega_0)$ and $U(\lambda_2 k\Omega_0)$), as well as the respective execution speeds (λ_1 and λ_2). For the first prediction, only one past set of learnt series coefficients is available, and we therefore resort to a zeroth-order prediction by eliminating the second summand in Eq. (34).

3.5. Design parameters

The learning algorithm presented in this section comprises a number of parameters that are user-defined, and can be modified to influence the learning performance:

1. The order of the compensation Fourier series N fundamentally determines the frequencies of the error output $y(t)$ that are compensated for since the compensation input $u(t)$ is characterized by a truncated Fourier series of order N, and no adaptation occurs at higher frequencies. While an upper bound for N is in principle only given by the discrete-time measurements used to estimate the coefficients of $y(t)$, the high-frequency components of the tracking error are often inherently small and can therefore be neglected by choosing a lower order. If the approximate model (18) only captures the underlying closed-loop dynamics well for low frequencies, it may also be beneficial to avoid learning at higher frequencies by limiting the series order.
2. The iteration- and order-dependent learning step size μ_k^i can be treated as a tuning parameter chosen by the user, or its optimal value may be derived from the statistical properties of $n(t)$, $d^0(t)$, and $\gamma(t)$. While the explicit statistical properties of these signals may be difficult to identify for an experimental platform, they can also be considered design parameters. In this context, the properties of $\gamma(t)$ capture the likely change of the disturbance from iteration to iteration due to unmodeled dynamics (for example, nonlinearities of the underlying model), and can therefore be used to encode model uncertainty. A typical choice would be to assume the disturbance changes to be large initially to account for significant changes in the correction input that could highlight nonlinear behavior, and smaller changes as the algorithm converges. The properties of the non-repetitive noise term $n(t)$ encode measurement noise and the level of repeatability of the experiment, and large values thereof enforce smaller step sizes (as seen in Eq. (32)), reducing the level of 'trust' in the measurements. The statistical properties of $d^0(t)$ capture the confidence in the initial guess of what the systematic disturbances are, and thereby influences the step size during the initial phase of the learning. Assuming little prior knowledge implies the coefficients of $D^0(k\Omega_0)$ having very large variance, which in turn implies that the first step size $\overset{*}{\mu}{}^0$ will approach 1 according to Eq. (32).
3. Finally, when time scaling is applied, the time scale sequence $\lambda_0, \lambda_1, \ldots$ remains to be chosen. This is typically done based on past experiments, where the performance with no initial compensation input determines λ_0, and the following time scales are determined based on the performance of the predictor (34). Note that, if the optimal step sequence size $\overset{*}{\mu}{}_k^i$ is used, $\gamma(t)$ should be chosen to have significant influence after a change in execution speed in order to account for modeling mismatches caused by the change of execution speed.

[1] While this property holds for many motions, it is straightforward to find counterexamples, such as motions with very large negative vertical accelerations.

4. Experiments

Experiments demonstrating the use of the learning algorithm presented herein for quadrocopter flight are described in this section. After introducing the experimental setup, we present the flight task to which the algorithm was applied, discuss its performance, and demonstrate the influence of a selection of the tuning parameters introduced in Section 3.5.

4.1. Experimental setup

The flight experiments were carried out in the Flying Machine Arena, an aerial vehicle research platform at ETH Zurich [51]. The vehicles used for the experiments are custom-built quadrocopters that are based on Ascending Technologies 'Hummingbird' vehicles [52]. The custom electronics [28] mounted on-board each vehicle provide inertial measurements and implement the body rate control law (10) based on the filtered measurements. The desired rotational rates $\hat{\omega}_x$, $\hat{\omega}_y$, $\hat{\omega}_z$ and the collective thrust command a (as defined in Eq. (2)) are communicated to the vehicle from a desktop computer through a low-latency wireless communication channel at a rate of 50 Hz.

A commercial motion capture system [53] provides position and attitude information, which is filtered by a Luenberger observer. The filtered full state information is used on the desktop computer to implement the feedback controller (5)–(9), and the filtered position information is used in the learning algorithm.

The feedback control system used in these experiments introduces additional, albeit small, disturbances to the system in addition to the dynamic effects described in Section 2.1. For example, the wireless radio link between the quadrocopter and the desktop computer as well as the image processing required by the motion capture system introduce variable delays into the feedback loop, approximately 1–5% of the feedback control commands are lost in transmission over the wireless channel, and additional dynamics are introduced by the state observer.

4.2. Learning implementation

The learning algorithm was implemented on the desktop computer that also executes the feedback control law, and uses the position data provided by the Luenberger observer to compute the Fourier coefficients of the tracking error. In all experiments, the tracking error output $y(t)$ was the three-dimensional deviation of the position from the reference trajectory and the control input $u(t)$ was a set point correction to the feedback controller, as seen in Fig. 2. Each iteration of the learning algorithm execution then consisted of the following steps:

1. Measure the tracking error for at least one period T. Note that, by measuring for more than one iteration, the variance of $N(k\Omega_0)$ can be reduced [48], therefore allowing the use of larger step sizes μ_k^i.
2. Apply the learning update law (19), using appropriate step sizes chosen as discussed in Section 3.5.
3. Wait for the system to converge under the new control inputs. Note that the instantaneous change of the correction input in Step 2 represents a non-periodic excitation of the system, making this step necessary.

4.3. Figure-eight maneuver

The motion we consider is a periodic figure-eight maneuver flown at high speed in the horizontal plane around two obstacle points, as shown in Fig. 3. This motion is used to demonstrate the basic working of the learning algorithm, as well as the influence of a number of learning parameters.

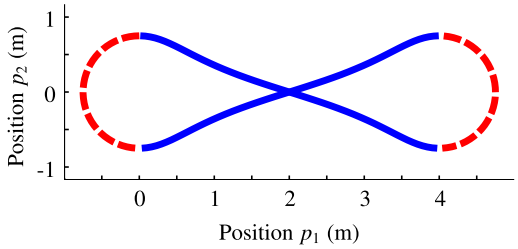

Fig. 3. The reference trajectory of the figure-eight maneuver, highlighting the composition from two half-circles (dashed red) and two connecting splines (solid blue). The period of the maneuver is $T = 3.3$ s, and the maximum speed reached is 6 m s^{-1}. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

To execute the maneuver under closed-loop control, the feedback linearizing control law presented in Section 2.2 is used, and the transfer function $G(\omega)$ required for the iteration-domain feedback law (19) was derived from the approximate dynamics (11). The three degrees of freedom are decoupled in the approximate dynamics, and the transfer function matrix $G(\omega)$ therefore only has non-zero entries on its diagonal.

4.3.1. Maneuver design

The nominal maneuver is designed based on the quadrocopter dynamics (1)–(4). 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 limitation of the collective thrust

$$a_{\min} \leqslant a \leqslant a_{\max} \tag{36}$$

and the limitation of the allowable body rate commands

$$|\omega_i| \leqslant \omega_{\max} \quad \text{for } i = \{x, y, z\} \tag{37}$$

The first constraint, limiting the collective thrust, is given by the minimum and maximum rotational speed of the propellers. The second constraint, limiting the rotational rate of the quadrocopter, is given by the limited range of the gyroscopic inertial sensors. Because the quadrocopter has low rotational inertia and high achievable torques due to the outwards mounting of the propellers, the body rate control loop (10) has very high bandwidth. We therefore do not explicitly limit rotational accelerations, and assume that the rate constraint (37), with $\omega_{\max}$ suitably chosen, along with (36) suffices to ensure feasibility.

Based on the constraints (36) and (37), we will now design the two components of the figure-eight maneuver, i.e. the half-circles about the obstacle points and the splines connecting them:

Semi Circles: A circular trajectory, covering an angle of 180°, surrounds each obstacle point at a user-defined radius R_s. The thrust a and rotational rates ω_x, ω_y along this circular trajectory are then computed [50], and the time required for each semi circle is chosen to be the fastest time for which the control input constraints (36) and (37) are satisfied.

Connecting Splines: The two semi circle trajectories are connected through polynomial trajectories. The trajectory is continuous in position, velocity, attitude, and rotational rate of the quadrocopter if the polynomial trajectories match the position, velocity, acceleration, and jerk of the circular trajectories at either end [54]. 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 (36) and (37) is found, using the algorithm from Schoellig et al. [50] to compute the inputs and verify feasibility.

The figure-eight maneuver is then fully defined through the concatenation of the first semi circle, the first spline, the second semi circle, and the second spline. Due to the continuity and feasibility conditions imposed on each component of the trajectory, the resulting trajectory is feasible with respect to the constraints (36), (37), and it is periodic.

An example of the figure-eight motion can be seen in Fig. 3. In this specific example, the parameters were chosen to be $L = 4$ m, $R_s = 0.75$ m, $a_{\max} = 1.8g = 17.65$ m s^{-2}, and $\omega_{\max} = 500^\circ$ s^{-1}. The resulting maneuver duration is $T = 3.3$ s, with an average speed of 4 m s^{-1} and a maximum speed of 6 m s^{-1}. The duration of each half circle is 0.71 s, and the duration of each connecting spline is 0.93 s. This example was used as the reference trajectory to be learnt in the following results.

4.3.2. Learning at fixed maneuver speed

As a first test case, we show the learning performance when the maneuver is executed at the fixed speed of $\lambda = 0.7$, i.e. 70% of the maneuver speed the motion was nominally designed for. The order of the correction input series was chosen to be $N = 10$, which showed to be a good compromise between the ability to compensate for temporally localized tracking errors and robustness to high-frequency uncertainty. We chose the learning step size series to be

$$\mu_k^i = \min\left(1, \frac{3}{i+1}\right) \quad \text{for all } k \tag{38}$$

which provides fast initial convergence with the first three steps being of size 1, and good robustness to non-repetitive noise since the steps reduce in size as the iteration number increases.

At this speed, the maneuver could be safely executed with no initial correction input (i.e. $u^0(t) = 0$), and the learning algorithm converged. Fig. 4 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. It can be seen that the deviations from the nominal trajectory after the learning process are minimal, and that the learning algorithm significantly improved the tracking performance.

Fig. 5 shows the evolution of the error coefficients over 22 iterations of the learning algorithm. It can be seen that the error coefficient magnitude quickly reduced from values in excess of 100 cm to values below 2 cm. The peak tracking error was reduced from approximately 200 cm to 5 cm. A significant outlier can be seen

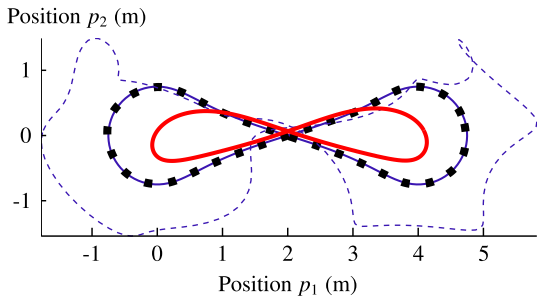

Fig. 4. Top-down view of the flown trajectory before (thick solid red) and after (thin solid blue) learning. The dotted black line denotes the nominal trajectory, and the dashed blue line shows the set point $\hat{p}(t)$ provided to the feedback control law (as seen in Fig. 2) after the learning is applied. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

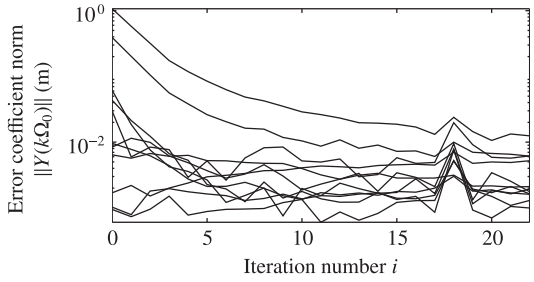

Fig. 5. Evolution of the Fourier series coefficients during learning at a fixed maneuver speed of $\lambda = 0.7$. The lines at iteration zero are from top to bottom: $k = \{1, 2, 4, 5, 0, 6, 3, 7, 10, 8, 9\}$. Note the significant outlier for iteration $i = 18$, the influence of which on subsequent iterations was relatively small because $\mu_k^{18} = 3/19$ for all k is relatively small.

at the 18th iteration, likely caused by a non-repeatable disturbance to the vehicle such as a temporary wireless communication failure. Because the step size $\mu_k^{18} = 3/19$ is relatively small, the large tracking error in this iteration was not strongly propagated to following iterations.

It should be noted that the non-learning controller design used in these experiments was not tuned to provide the best possible tracking performance without the adaptation; as discussed in Section 2.2.4, a straightforward improvement would be the inclusion of feedforward terms in the control law. What can, however, be seen from these results is that such improvements to the control strategy do not appear necessary because the adaptation law can largely eliminate repeatable tracking errors.

4.3.3. Influence of Fourier series order

In a further experiment, we demonstrate the effect of varying the order of the compensation input Fourier series N, as given in Eq. (15). The learning experiment was repeated at a constant speed of $\lambda = 0.8$ (approximately 14% faster than the previously presented experiment), and the learning process was executed with Fourier series of the orders $N = \{0, 1, 2, 5, 8, 9, 10\}$. The resulting progression of the root mean square (RMS) position tracking error (computed over one motion period) over 25 iterations is shown in Fig. 6. Note that the Fourier series of order $N = 0$ provided no significant improvement as it consists only of a constant set point

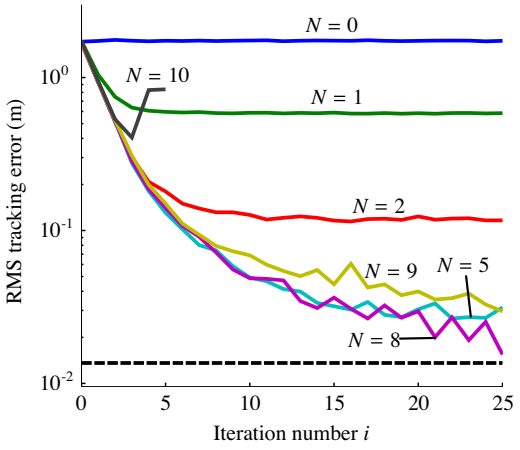

Fig. 6. Learning performance (as measured by the RMS position tracking error) for the high-performance figure-eight maneuver executed at a speed of $\lambda = 0.8$, with varying order N of the correction Fourier series. The dashed black horizontal line represents the estimated repeatability limit of 1.4 cm. Note that the relatively large uncertainty at higher frequencies causes the learning performance to decrease for $N = 9$, and the vehicle collided with the floor during the sixth iteration for $N = 10$.

shift. Increases in the Fourier series order consistently improved the learning performance initially. The orders $N = 5$ and $N = 8$ provided nearly identical performance, showing that the predominant tracking error components were sufficiently covered. The error then increased for $N = 9$, and the vehicle collided with the floor after five iterations for $N = 10$. This was caused by the insufficient accuracy of the approximate system dynamics (11) in the high-frequency range, and a possible remedy would be to reduce the step size for higher frequencies. Note that the order $N = 10$ could successfully be used for the experiments in Section 4.3.2 because the speed scaling factor λ was chosen lower, thereby mapping each component of the series to a lower frequency.

Fig. 6 also shows the repeatability limit of our experimental system during the tests presented herein. In order to determine this limit, the maneuver was executed multiple times with an identical correction input, and the RMS deviation from the average flight trajectory was computed for each execution. With identical execution parameters as used for the other data in Fig. 6, the average RMS repeatability error over fifteen executions was 13.6 mm, with a standard deviation of 2.2 mm. This can be taken to be a measure of the level of non-repeatable noise, as captured by $n(t)$, in the system. For the flight trajectory learnt using a Fourier correction series of order $N = 8$, the average RMS tracking error over the last five iterations (i.e., $i = 21,\ 22,\ 23,\ 24,\ 25$) was 24.2 mm (with a standard deviation of 4.5 mm), indicating that the compensations by the learning algorithm corrected for almost all systematic errors. In other words, if an input signal $u(t)$ exists that would entirely eliminate all systematic tracking errors given the true dynamics of the quadrocopter, then it would allow the tracking error to be reduced by a further approximately 11 mm compared to the learning algorithm presented here.

4.3.4. Transfer to increasing speeds

When initializing the learning algorithm with no correction signal for the full maneuver speed ($\lambda = 1$), the maneuver could not be learnt successfully. This is likely due to the maneuver being designed to use the full dynamic envelope of the quadrocopter, and the LTI system approximation not being sufficiently accurate when considering large initial deviations. Fig. 7 demonstrates the use of the time scaling method (introduced in Section 3.4) to learn the maneuver at full speed. We initialized the speed factor to $\lambda_0 = 0.8$, and then applied the update law

$$\lambda_{\ell+1} = \lambda_\ell + 0.05 \tag{39}$$

after every five learning iterations of the maneuver. We used the same learning step size sequence (38) as before, but at iterations where time scaling occurs, we reset the iteration counter to $i = 0$ in order to compensate for the significant model uncertainty after a change of the time scale. The order of the correction Fourier series was chosen to be $N = 5$.

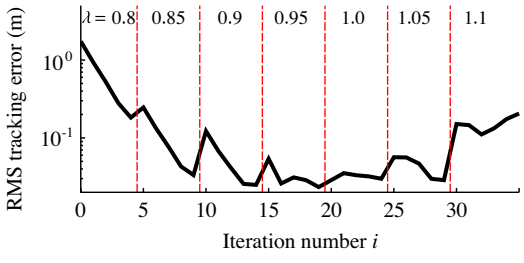

Fig. 7. Logarithmic plot of the root mean square position tracking error during execution of the figure-eight maneuver with time scaling. The maneuver execution speed (shown at the top of the graph) is increased every five iterations. Note that $\lambda = 1.0$ denotes the nominal maneuver speed, and values $\lambda > 1.0$ are not nominally feasible.

In Fig. 7, the results of the experiment show that the RMS tracking error was initially reduced from 1.7 m to 0.18 m over the first four iterations, which were executed at $\lambda = 0.8$. The RMS error increased by values of 0.005 m to 0.09 m during the first four speed changes. Beginning with the 25th iteration, the execution speed was $\lambda > 1$, i.e. the maneuver is faster than was computed to be feasible according to Section 4.3.1. It can be seen that the tracking error increased as one would expect, and the learning algorithm ultimately failed to reduce the error at an execution speed of $\lambda = 1.1$ due to significant input saturations. These experimental results correspond well to the predicted feasibility.

5. 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 that the learning is performed on the closed-loop system of the quadrocopter under feedback control, which makes repeated maneuver executions highly repeatable, and therefore well-suited for the non-causal correction of tracking errors.

The parametrization of the correction input $u(t)$ 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. Relatively low Fourier series orders often suffice to significantly improve the tracking performance, as seen in our experiments. 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 [31]. Furthermore, inaccuracies of the dynamic model at high frequencies can be circumvented by limiting the learning to lower frequencies, or can be modeled through the characteristics of the noise models used to capture changing dynamics and non-repeatable effects. Through the appropriate choice of these noise characteristics, the learning algorithm can be tuned to quickly compensate for disturbances at frequencies where an accurate model of the system is available, and to perform more cautious adaptation at frequencies where significant model uncertainty exists.

It can also be seen from our experiments that highly simplified models of the closed-loop dynamics, though only capturing a relatively rough approximation of the true behavior, suffices to guide the learning process. Similar results have been demonstrated for other learning algorithms, e.g. reinforcement learning [55]. The simplified model, and therewith the uncomplicated derivation of such a model, make the application of the algorithm to changing conditions (such as specific tasks, differing controllers, or different vehicles) a straightforward undertaking. An example of applying this method to a more complex controller that stabilizes an inverted pendulum on a quadrocopter [37] highlights this.

The algorithm presented particularly lends itself to applications where computational power and memory are limited. During the execution of the maneuver, the algorithm only requires the estimation of the coefficients of the tracking error Fourier series up to order N, and the control input update can be reduced to a matrix multiplication according to Eq. (19) by pre-computing the step size sequence μ_k^i and the inverse transfer function $G^+(k\Omega_0)$ for the considered frequencies $k = 0, 1, \ldots, N$.

A limitation of our method is that it is not trivial to incorporate additional constraints due to the simplified dynamic model, the serial architecture, and the frequency domain approach in the learning algorithm, Examples of potentially useful constraints are the explicit consideration of input saturations, or the penalization

of control effort. Their inclusion can improve the learning performance because they help capture the underlying dynamics more accurately, or can help to shape the outcome of the learning process. Such constraints are easier to include in time-domain-based learning methods, in particular when the learning process is posed as an optimization problem (e.g. [31]).

Furthermore, maneuvers exhibiting very large initial tracking errors highlight the limitations of using a simplified model in the learning process. The system dynamics may not be captured with sufficient accuracy for large errors, causing the learning algorithm to fail. The time-scaling method presented herein can offer a viable solution to this problem when it is possible to initially reduce the execution speed of the maneuver: The frequency domain approach to iterative learning 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.

6. Conclusion

This paper presents the use of a frequency-domain iterative learning scheme for periodic quadrocopter flight. The iteration-domain feedback law leverages an underlying feedback control law that stabilizes the vehicle and makes its motion highly repeatable. The nominally linear time-invariant closed-loop dynamics of the quadrocopter feedback system are used to determine correction values from observed errors in a straightforward manner. By parameterizing the non-causal tracking error compensation as Fourier series, the algorithm is computationally lightweight and easy to adapt. Uncertainties, such as measurement noise, inaccuracies of the approximate transfer function, or model uncertainty can be accounted for in the learning step size, and the optimal step size for each frequency component can be computed from the statistical properties thereof. The approach also allows the learning of high-performance maneuvers by executing them at a reduced speed initially and then transferring learnt corrections to higher speeds, where a disturbance prediction scheme is used to initialize the learning at higher speeds.

The approach was experimentally verified for a quadrocopter executing a high-performance motion. The experimental results highlighted the advantages of learning at reduced speeds, with a fast motion only being learnt successfully at full speed when using learnt parameters from lower speeds to initialize the learning process.

Acknowledgements

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

References

[1] Siegwart R, Nourbakhsh IR, Scaramuzza D. Autonomous mobile robots. 2nd ed. The MIT Press; 2011.
[2] Pounds P, Mahony R, Corke P. Modelling and control of a large quadrotor robot. Control Eng Pract 2010;18(7):691–9.
[3] Gibb AS. Droning the story. Master thesis. Simon Fraser University; 2011.
[4] Sinha AK, Atyeo S, Schauenburg A, Rathore A, Quinn B, Christoffersen T. Investigation of the application of UAV technology in power line inspection. In: International UAV systems conference; 2006.
[5] Tonetti S, Hehn M, Lupashin S, D'Andrea R. Distributed control of antenna array with formation of UAVs. In: IFAC world congress; 2011.
[6] Airrobot GmbH & Co KG,. AirRobot product information; 2007. <http://www.airrobot-us.com/images/Airrobot_info.pdf>.
[7] Mellinger D, Michael N, Kumar V. Trajectory generation and control for precise aggressive maneuvers with quadrotors. Int J Robot Res 2012;31(5):664–74.
[8] Michael N, Mellinger D, Lindsey Q, Kumar V. The GRASP multiple micro UAV Testbed. IEEE Robot Autom Mag 2010;17(3):56–65.
[9] Müller M, Lupashin S, D'Andrea R. Quadrocopter ball juggling. In: International conference on intelligent robots and systems; 2011.
[10] Ritz R, Müller MW, Hehn M, D'Andrea R. Cooperative quadrocopter ball throwing and catching. In: International conference on intelligent robots and systems; 2012.
[11] Bristeau PJ, Martin P, Salaün E, Petit N. The role of propeller aerodynamics in the model of a quadrotor UAV. In: European control conference; 2009.
[12] Bangura M, Mahony R. Nonlinear dynamic modeling for high performance control of a quadrotor. In: Australasian conference on robotics and automation; 2012.
[13] Powers C, Mellinger D, Kushleyev A, Kothmann B. Influence of aerodynamics and proximity effects in quadrotor flight. In: International symposium on experimental robotics; 2012.
[14] Waslander SL, Hoffmann GM, Jang JS, Tomlin CJ. Multi-agent quadrotor testbed control design: integral sliding mode vs. reinforcement learning. In: International conference on intelligent robots and systems; 2005.
[15] Barros dos Santos SR, Givigi SN, Nascimento CL. Autonomous construction of structures in a dynamic environment using reinforcement learning. In: Systems conference; 2013.
[16] Dunfied J, Tarbouchi M, Labonte G. Neural network based control of a four rotor helicopter. In: International conference on industrial technology; 2004.
[17] Dierks T, Jagannathan S. Output feedback control of a quadrotor UAV using neural networks. IEEE Trans Neural Netw 2010;21(1):50–66.
[18] Nicol C, Macnab C, Ramirez-Serrano A. Robust adaptive control of a quadrotor helicopter. Mechatronics 2011;21(6):927–38.
[19] Palunko I, Fierro R. Adaptive control of a quadrotor with dynamic changes in the center of gravity. In: IFAC world congress; 2011.
[20] Antonelli G, Arrichiello F, Chiaverini S, Robuffo Giordano P. Adaptive trajectory tracking for quadrotor MAVs in presence of parameter uncertainties and external disturbances. In: International conference on advanced intelligent mechatronics; 2013.
[21] Arimoto S, Kawamura S, Miyazaki F. Bettering operation of dynamic systems by learning: a new control theory for servomechanism or mechatronic systems. In: Conference on decision and control; 1984.
[22] Inoue T, Iwai S, Nakano M. High accuracy control of a proton synchrotron magnet power supply. In: IFAC world congress; 1981.
[23] Wang Y, Gao F, Doyle FJ. Survey on Iterative learning control, repetitive control, and run-to-run control. J Process Control 2009;19(10):1589–600.
[24] Bristow DA, Tharayil M, Alleyne AG. A survey of iterative learning control. Control Syst Mag 2006;26(3):96–114.
[25] Cuiyan L, Dongchun Z, Xianyi Z. A survey of repetitive control. In: International conference on intelligent robots and systems; 2004.
[26] Chin I, Qin SJ, Lee KS, Cho M. A two-stage iterative learning control technique combined with real-time feedback for independent disturbance rejection. Automatica 2004;40(11):1913–22.
[27] Barton K, Mishra S, Xargay E. Robust iterative learning control: L1 adaptive feedback control in an ILC framework. In: American control conference; 2011.
[28] Lupashin S, D'Andrea R. Adaptive fast open-loop maneuvers for quadrocopters. Auton Robot 2012;33(1–2):89–102.
[29] Mellinger D, Michael N, Kumar V. Trajectory generation and control for precise aggressive maneuvers with quadrotors. In: International symposium on experimental robotics; 2010.
[30] Purwin O, D'Andrea R. Performing and extending aggressive maneuvers using iterative learning control. Robot Auton Syst 2011;59(1):1–11.
[31] Schoellig AP, Mueller FL, D'Andrea R. Optimization-based iterative learning for precise quadrocopter trajectory tracking. Auton Robot 2012;33(1–2):103–27.
[32] Mueller FL, Schoellig AP, D'Andrea R. Iterative learning of feed-forward corrections for high-performance tracking. In: International conference on intelligent robots and systems; 2012.
[33] Pipatpaibul PI, Ouyang PR. Application of online iterative learning tracking control for quadrotor UAVs. ISRN robotics 2013; 2013 [Article ID 476153].
[34] Houtzager I, van Wingerden JW, Verhaegen M. Rejection of periodic wind disturbances on a smart rotor test section using lifted repetitive control. IEEE Trans Control Syst Technol 2013;21(2):347–59.
[35] Barlas T, van Kuik G. Review of state of the art in smart rotor control research for wind turbines. Prog Aerosp Sci 2010;46(1):1–27.
[36] Gang S, Zhen-Cai Z, Lei Z, Yu T, Chi-fu Y, Jin-song Z, et al. Adaptive feed-forward compensation for hybrid control with acceleration time waveform replication on electro-hydraulic shaking table. Control Eng Pract 2013;21(8):1128–42.
[37] Hehn M, D'Andrea R. An iterative learning scheme for high performance, periodic quadrocopter trajectories. In: European control conference; 2013.
[38] Hehn M, D'Andrea R. A frequency domain iterative feed-forward learning scheme for high performance periodic quadrocopter maneuvers. In: International conference on intelligent robots and systems; 2013.
[39] Mahony R, Kumar V, Corke P. Multirotor aerial vehicles: modeling, estimation, and control of quadrotor. IEEE Robot Autom Mag 2012;19(3):20–32.
[40] Pounds P, Mahony R, Corke P. Modelling and control of a quad-rotor robot. In: Australasian conference on robotics and automation; 2006.
[41] Alexis K, Nikolakopoulos G, Tzes A. Constrained optimal attitude control of a quadrotor helicopter subject to wind-gusts: experimental studies. In: American control conference; 2010.

[42] Schoellig A, Wiltsche C, D'Andrea R. Feed-forward parameter identification for precise periodic quadrocopter motions. In: American control conference. 2012.
[43] Wang Y, Doyle FJ. Stability analysis for set-point-related indirect iterative learning control. In: Conference on decision and control; 2009.
[44] Hsu HP. Schaum's outline of signals and systems. McGraw-Hill; 1995.
[45] Chen CT. Linear system theory and design. 3rd ed. Oxford University Press; 1998.
[46] Bernstein DS. Matrix mathematics. Princeton University Press; 2005.
[47] Oppenheim AV, Schafer RW. Discrete-time signal processing. 3rd ed. Pearson Prentice Hall; 2010.
[48] Schoukens J, Renneboog J. Modeling the noise influence on the Fourier coefficients after a discrete Fourier transform. IEEE Trans. Instrum Meas 1986;IM-35(3):278–86.
[49] Anderson BDO, Moore JB. Optimal filtering. Dover Publications; 2005.
[50] Schoellig A, Hehn M, Lupashin S, D'Andrea R. Feasibility of motion primitives for choreographed quadrocopter flight. In: American control conference; 2011.
[51] Lupashin S, Schoellig AP, Hehn M, D'Andrea R. The flying machine arena as of 2010. In: International conference on robotics and automation; 2011.
[52] Gurdan D, Stumpf J, Achtelik M, Doth KM, Hirzinger G, Rus D. Energy-efficient autonomous four-rotor flying robot controlled at 1 kHz. In: International conference on robotics and automation; 2007.
[53] Volgoi T, Roberts H, Gerber L, Aswege N. Tracking the future. In: The standard. Vicon Motion Systems Ltd.; 2011. p. 6–7.
[54] Mellinger D, Kumar V. Minimum snap trajectory generation and control for quadrotors. In: International conference on robotics and automation; 2011.
[55] Kolter JZ, Ng AY. Policy search via the signed derivative. In: Robotics: science and systems; 2009.