2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

An Iterative Learning Scheme for High Performance, Periodic Quadrocopter Trajectories

Markus Hehn and Raffaello D'Andrea

Abstract— **Quadrocopters allow the execution of high-performance maneuvers under feedback control. However, repeated execution typically leads to a large part of the tracking errors being repeated. This paper evaluates an iterative learning scheme for an experiment where a quadrocopter flies in a circle while balancing an inverted pendulum. The scheme permits the non-causal compensation of periodic errors when executing the circular motion repeatedly, and is based on a Fourier series decomposition of the repeated tracking error and compensation input. The convergence of the learning scheme is shown for the linearized system dynamics. Experiments validate the approach and demonstrate its ability to significantly improve tracking performance.**

I. Introduction

The capability of quadrocopters to perform highly dynamic, complex, and precise motions has been demonstrated repeatedly in recent years (see, for example, [1]–[4]). Such motions are commonly executed by using a first-principles model of the vehicle dynamics to determine nominal control inputs, and a feedback control law to ensure tracking of the nominal trajectory.

In order to account for model mismatches, a number of learning schemes have been developed. Examples include those based on sliding mode control and reinforcement learning techniques [5], neural networks [6], and adaptive control [7].

When performing the same motion repeatedly, a further opportunity to improve tracking performance may arise because many of the disturbances that degrade tracking performance will be similar each time the vehicle performs the motion. It is thus possible to non-causally correct for these repeatable disturbances: By using data from previous executions to characterize them, model-based correction inputs can be computed before executing the motion again. Ideally, these correction inputs are able to fully compensate for the repeatable disturbances, such that the feedback controller is only required to compensate for non-repeatable disturbances.

Such iteration-based learning approaches have been successfully demonstrated for multi-rotor vehicles performing high-performance maneuvers. Broadly speaking, the learning approaches may be separated into 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 [8], 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 [9]. 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 [10], and its application to high performance quadrocopter flight has been demonstrated [11], [12].

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.

This paper evaluates a technique for non-causally compensating repeated, periodic tracking errors. Specifically, we consider a motion where the linearized dynamics around the nominal motion are time-invariant under an appropriate coordinate transformation. Similar to the second group of learning algorithms, we do not assume a parameterized

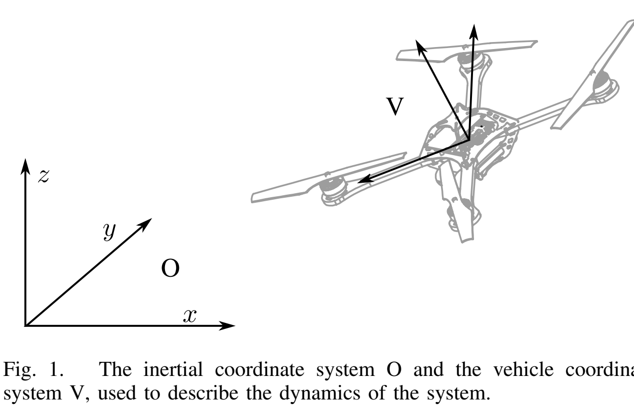

Fig. 1. The inertial coordinate system O and the vehicle coordinate system V, used to describe the dynamics of the system.

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

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 a 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.

The specific motion considered in this paper consists of a quadrocopter balancing an inverted pendulum and tracking a circular trajectory [13]. This problem is an interesting platform for learning algorithms for multiple reasons: Due to the highly dynamic nature of the motion and the agility required to balance the pendulum, the full range of the vehicle dynamics is used. Due to relatively high flight speeds and fast vehicle attitude changes, unmodeled aerodynamic effects (see e.g. [14]) significantly influence the dynamics of the vehicle. Furthermore, the control system employs a time-varying coordinate system transformation, the errors in which potentially introduce further periodic errors in the feedback control loop.

The remainder of this paper is structured as follows: We explain our motivation for developing the learning algorithm in Chapter II by introducing the flying inverted pendulum experiment and recapitulating experimental results that highlight its performance without learning. We then describe the iterative learning strategy in Chapter III, and show convergence of repeatable errors. Chapter IV shows experimental results from the application of the algorithm to the experiment. Finally, we summarize and discuss potential research directions in Chapter V.

II. The Flying Inverted Pendulum Experiment

This chapter introduces the flying inverted pendulum [13], an experiment that demonstrates the performance and agility of quadrocopters. This experiment is the motivation behind the algorithm presented in Chapter III, and is the basis for the experimental results in Chapter IV. The objective of this experiment is to balance an inverted pendulum on a quadrocopter while the vehicle flies horizontal circles. The derivations from [13] are reproduced here in abbreviated form for the purpose of completeness; the reader is referred to the previously published paper for a more thorough discussion.

A. Dynamics

The quadrocopter is modeled as a rigid body with six degrees of freedom: Its position (x, y, z) in the inertial coordinate system O, and its attitude, represented by the rotation between the inertial coordinate system O and the body-fixed coordinate system V, as shown in Figure 1. The rotation is parameterized by three Euler angles, representing rotations about the z-axis (α), the y-axis (β) and the x-axis(γ), executed in this order:

$$ {}^{\mathrm{O}}_{\mathrm{V}}R(\alpha,\beta,\gamma) = R_z(\alpha)\,R_y(\beta)\,R_x(\gamma)\ . \tag{1}$$

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 body z-axis, (a; in units of acceleration). It follows that the differential equations governing the vehicle motion are

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = {}^{\mathrm{O}}_{\mathrm{V}}R(\alpha,\beta,\gamma)\begin{bmatrix}0\\0\\a\end{bmatrix} + \begin{bmatrix}0\\0\\-\mathrm{g}\end{bmatrix} \tag{2}$$

$$\begin{bmatrix}\dot{\gamma}\\ \dot{\beta}\\ \dot{\alpha}\end{bmatrix} = \begin{bmatrix}\cos\beta\cos\gamma & -\sin\gamma & 0\\ \cos\beta\sin\gamma & \cos\gamma & 0\\ -\sin\beta & 0 & 1\end{bmatrix}^{-1}\begin{bmatrix}\omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{3}$$

where g denotes gravitational acceleration.

The inverted pendulum is modeled as a point mass with two degrees of freedom, chosen to be the horizontal position of the pendulum center of mass relative to its base in O (r along the x-axis, s along the y-axis). Using Lagrangian mechanics, the nonlinear equations of motion of the pendulum can be derived. In the interest of compactness, we state only that the pendulum acceleration depends on its position and velocity, and the vehicle acceleration:

$$\begin{bmatrix}\ddot{r}\\ \ddot{s}\end{bmatrix} = \mathrm{h}\left(r, s, \dot{r}, \dot{s}, \ddot{x}, \ddot{y}, \ddot{z}\right)\ . \tag{4}$$

B. Coordinate Transformation for Circular Trajectories

The objective of the experiment is to fly circular trajectories while balancing the pendulum. We introduced a transformation into rotating coordinate systems for the vehicle position (C) and attitude (W) in [13]. It is then possible to transform the equations of motion such that, for circular flight, the nominal states and the linearized dynamics about them can be described in a time-invariant manner:

$$\begin{bmatrix}x\\y\\z\end{bmatrix} =: \begin{bmatrix}\cos\Omega t & -\sin\Omega t & 0\\ \sin\Omega t & \cos\Omega t & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}u\\v\\w\end{bmatrix} \tag{5}$$

$$ {}^{\mathrm{O}}_{\mathrm{V}}R(\alpha,\beta,\gamma)\begin{bmatrix}0\\0\\1\end{bmatrix} =: {}^{\mathrm{O}}_{\mathrm{W}}R(\Omega t,\mu,\nu)\begin{bmatrix}0\\0\\1\end{bmatrix} \tag{6}$$

$$\begin{bmatrix}r\\s\end{bmatrix} =: \begin{bmatrix}\cos\Omega t & -\sin\Omega t\\ \sin\Omega t & \cos\Omega t\end{bmatrix}\begin{bmatrix}p\\q\end{bmatrix}\ . \tag{7}$$

A horizontal circle of radius R, flown at a constant rate Ω and at constant height, is described by $u = R$, $v = 0$, and $\dot{w} = 0$. The quadrocopter and pendulum equations of motion (2)-(4) can be rewritten in the rotating coordinate system using the above transformations. It was shown that a nominal circular trajectory for pendulum and quadrocopter is then given by the constant nominal state

$$\bar{\mu} = \arctan\left(-\frac{\Omega^2 R}{\mathrm{g}}\right) \tag{8}$$

$$\bar{\nu} = 0 \tag{9}$$

$$\bar{a} = \sqrt{\mathrm{g}^2 + \left(\Omega^2 R\right)^2} \tag{10}$$

$$\bar{q} = 0 \tag{11}$$

$$\Omega^2(R+\bar{p}) + \frac{\mathrm{g}\bar{p}}{\sqrt{L^2-\bar{p}^2}} = 0 \tag{12}$$

where L represents the length of the pendulum from its base to its center of mass. The vehicle yaw angle, α, is not defined by the motion, and may be chosen separately. In our experiments, we chose a constant yaw angle $\alpha = 0$.

C. Feedback Control Design

About this constant nominal state, the dynamics of the quadrocopter-pendulum system in the rotating C-W coordinate system were approximated by a first-order Taylor expansion. We denote the system state under the coordinate transformation (5)-(7) by

$$\mathrm{x} := (u, v, w, \dot{u}, \dot{v}, \dot{w}, \alpha, \mu, \nu, p, q, \dot{p}, \dot{q}) \tag{13}$$

and use the transformed control input $\mathrm{v} := (a, \dot{\mu}, \dot{\nu}, \dot{\alpha})$. The true rotational rate control inputs $(\omega_x, \omega_y, \omega_z)$ can be recovered for known values of v and x through Equations (6) and (3). The linear dynamics resulting from a Taylor expansion about the nominal trajectory are then time-invariant:

$$\dot{\tilde{\mathrm{x}}} = \frac{\partial \mathrm{f}(\bar{\mathrm{x}}, \bar{\mathrm{v}})}{\partial \mathrm{x}} \tilde{\mathrm{x}} + \frac{\partial \mathrm{f}(\bar{\mathrm{x}}, \bar{\mathrm{v}})}{\partial \mathrm{v}} \tilde{\mathrm{v}} \tag{14}$$

where a tilde denotes small deviations from the nominal trajectory, and $\mathrm{f}(\bar{\mathrm{x}}, \bar{\mathrm{v}})$ denotes the system dynamics consisting of Equations (2)-(4) under the coordinate transformation (5)-(7). An infinite-horizon linear quadratic regulator [15] was designed to stabilize the system around the nominal trajectory. We denote the resulting time-invariant state feedback law by $\tilde{\mathrm{v}} = K\tilde{\mathrm{x}}$.

D. Flight Performance

Results from the previous experiment [13] demonstrate the ability of the presented control system design to reliably stabilize the vehicle-pendulum system during circular flight. However, significant trajectory tracking errors occur, as can be seen for an exemplary experiment in Figure 2. Note that the errors contain two clearly distinguishable components: A mean error, and an error that oscillates at the rate at which the circle is flown (Ω). A possible explanation for these oscillating errors is the rotating coordinate system transformation (5)-(7), which would map a constant error in the inertial coordinate system to such an oscillating error in the rotating coordinate system.

Figure 2 also shows that the tracking error largely repeats for every round of the circle that is flown: The experiment may be considered to be a periodic motion with relatively large repeatable disturbances deteriorating the tracking performance of the feedback control law. As discussed in Chapter I, the repeatability of the disturbances offers the opportunity to non-causally compensate for them. We develop a learning algorithm for this purpose in the following chapter, and will revisit the above experiment to present experimental results thereafter.

III. Learning Algorithm

The approach presented in this paper is conceptually similar to [8], 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. We propose a correction strategy, and show that, under the assumption of linear time-invariant system dynamics, this strategy indeed fully corrects for such recurring errors.

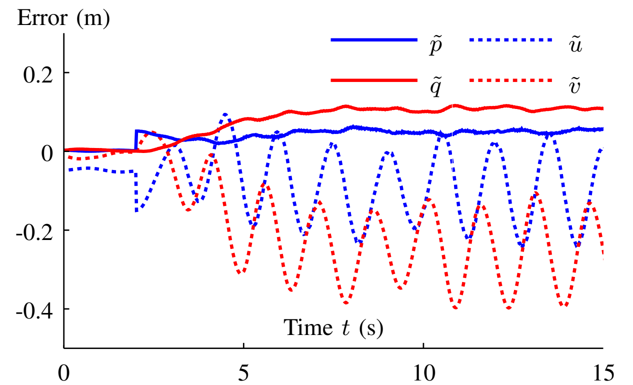

Fig. 2. Errors in the rotating coordinate system (from [13]): Pendulum position error $(\tilde{p}, \tilde{q})$ and quadrocopter position error $(\tilde{u}, \tilde{v})$. At $t = 2\,\mathrm{s}$, the controller is switched from a constant nominal position to a circular trajectory with $R = 0.1\,\mathrm{m}$. Large repeating errors can be seen.

A. Core Concept

The core idea of the adaptation law presented herein is to measure the tracking error over the period of the maneuver, and approximate it as a truncated Fourier series. In order to compensate for recurring errors, a correction input – also consisting of a truncated Fourier series – is applied to the system. For a known tracking error, the coefficients of the correction input Fourier series are computed by inverting the linear time-invariant dynamics of the system about the nominal trajectory. In order to increase the robustness against non-repeatable disturbances, the adaptation law uses a step size, permitting a trade-off between speed of convergence and noise rejection.

B. Adaptation Control Input

In addition to the state feedback control input, we apply an adaptation control input $\hat{\mathrm{v}}$ to the system. Using A and B to denote the derivatives with respect to the state and input, respectively, we can then rewrite Equation (14) as

$$\dot{\tilde{\mathrm{x}}} = A\tilde{\mathrm{x}} + B\left(K\tilde{\mathrm{x}} + \hat{\mathrm{v}}\right) \tag{15}$$

$$= \underbrace{(A + BK)}_{=:\bar{A}} \tilde{\mathrm{x}} + B\hat{\mathrm{v}} \tag{16}$$

where $\bar{A}$ represents the closed-loop linearized dynamics of the system.

C. Propagation of Fourier Series Inputs Through the System

Now we will use the adaptation control input $\hat{\mathrm{v}}$ to non-causally compensate for repeated disturbances. We parameterize this compensation by a Fourier series of order N and fundamental frequency ω:

$$\hat{\mathrm{v}} = r_0 + \sum_{k=1}^{N} r_k \cos(k\omega t) + \sum_{k=1}^{N} s_k \sin(k\omega t) \ . \tag{17}$$

It is straightforward to show that, to first order, the perturbation state $\tilde{\mathrm{x}}$ in reaction to this adaptation control input will also be a Fourier series of order N [16]:

$$\tilde{\mathrm{x}} = a_0 + \sum_{k=1}^{N} a_k \cos(k\omega t) + \sum_{k=1}^{N} b_k \sin(k\omega t) \tag{18}$$

and that the coefficients relate to the coefficients of the input by

$$0 = \bar{A} a_0 + B r_0 \tag{19}$$

$$-k\omega a_k = \bar{A} b_k + B s_k \tag{20}$$

$$k\omega b_k = \bar{A} a_k + B r_k \tag{21}$$

for $k = 1 \ldots N$.

D. Iteration-Domain Feedback Law

Consider a tracking error output that is a linear combination of the states:

$$\tilde{\mathrm{y}} = C \tilde{\mathrm{x}} \, . \tag{22}$$

Our objective is to eliminate the effects of errors described by the Fourier series (18) on the output. Due to the truncation of the Fourier series, higher-frequency components are not considered, and the order of the series would have to be increased in order to compensate for them.

The dimension of $\tilde{\mathrm{y}}$ is not defined by the problem, and may be chosen by the user as the set of error outputs that should be minimized. If the dimension of $\tilde{\mathrm{y}}$ is larger than the dimension of the adaptation control input $\hat{\mathrm{v}}$, a full correction of all errors cannot be expected.

The above equations (19)-(21) can be written in matrix form, and multiplied by C in order to describe the output tracking errors (22):

$$\begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} s_k \\ r_k \end{bmatrix} = \begin{bmatrix} -k\omega & -\bar{A} \\ -\bar{A} & k\omega \end{bmatrix} \begin{bmatrix} a_k \\ b_k \end{bmatrix} \tag{23}$$

$$\underbrace{\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} -k\omega & -\bar{A} \\ -\bar{A} & k\omega \end{bmatrix}^{-1} \begin{bmatrix} B & 0 \\ 0 & B \end{bmatrix}}_{=:J} \begin{bmatrix} s_k \\ r_k \end{bmatrix} = \underbrace{\begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} \begin{bmatrix} a_k \\ b_k \end{bmatrix}}_{=:e_k} . \tag{24}$$

The matrix J represents the linear mapping of input coefficients s_k, r_k to tracking error output coefficients e_k, and is the equivalent to the nominal maneuver Jacobian for parameterized motions described in [8].

Let J^+ denote the Moore-Penrose pseudoinverse [17] of J (in the special case that J is square, $J^+ = J^{-1}$ holds). Note that the existence of the inverse is not given for all systems. For the purpose of this paper however, we assume its existence and intend to investigate this question further in the future. Assume that we have executed the trajectory for iteration $i-1$, and measured the tracking error $\tilde{\mathrm{y}}^{i-1}$. Let e_k^{i-1} be the Fourier coefficients of the tracking error output of iteration $i-1$. The iteration-domain feedback law is

$$\begin{bmatrix} s_k \\ r_k \end{bmatrix}^i = \begin{bmatrix} s_k \\ r_k \end{bmatrix}^{i-1} - \gamma J^+ e_k^{i-1} \tag{25}$$

where γ is the step size parameter. The step size parameter can be used to trade off convergence of errors and noise rejection. This may be necessary because e_k^{i-1} usually contains components caused by non-repetitive process noise and measurement errors, and is therefore only an estimate of the true repeatable tracking error.

E. Convergence

Using the above feedback law, the tracking error Fourier coefficients e_k evolve as follows:

$$e_k^i = e_k^{i-1} + J \left(\begin{bmatrix} s_k \\ r_k \end{bmatrix}^i - \begin{bmatrix} s_k \\ r_k \end{bmatrix}^{i-1} \right) \tag{26}$$

$$= e_k^{i-1} - \gamma J J^+ e_k^{i-1} \, . \tag{27}$$

If J^+ is the right inverse of J (the interpretation of this is that there are not more tracking error outputs than adaptation control inputs), it follows that the tracking error converges to zero for $0 < \gamma \leq 1$:

$$e_k^i = (1-\gamma)\, e_k^{i-1} \, . \tag{28}$$

Note that if the tracking error dimension is lower than the input dimension, the correction term $J^+ e_k$ is the least-squares solution to an underconstrained set of equations [17], implying that the Euclidian norm of the Fourier coefficients is minimized. By Parseval's theorem [18], this is equivalent to minimizing the energy of the correction input signal $\hat{\mathrm{v}}$.

Now consider the case where J^+ is not the right inverse, implying that there are more tracking error outputs than inputs. In this case, the error dynamics are

$$e_k^i = \left(I - \gamma J J^+\right) e_k^{i-1} \tag{29}$$

$$= \left(I - \gamma J \left(J^T J\right)^{-1} J^T\right) e_k^{i-1} \tag{30}$$

where I denotes the identity matrix and $(\cdot)^T$ denotes the transpose of a matrix. Assuming that J is full rank, we apply a singular value decomposition [17]:

$$J = U \begin{bmatrix} \Sigma \\ 0 \end{bmatrix} V^T \tag{31}$$

$$e_k^i = \underbrace{\left(I - \gamma U \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} U^T\right)}_{=:M} e_k^{i-1} . \tag{32}$$

Using a similarity transformation [17], we show that the error does not diverge by computing the eigenvalues of M:

$$\mathrm{eig}(M) = \mathrm{eig}\left(U^T M U\right) \tag{33}$$

$$= \mathrm{eig}\left(I - \gamma \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}\right) \tag{34}$$

$$= \{1-\gamma, \ldots, 1-\gamma, 1, \ldots, 1\} \, . \tag{35}$$

This implies that, under the appropriate coordinate transformation, the tracking error coefficients either reduce to zero over time, or remain constant. Note that because this case implies fewer control inputs than tracking errors, it is to be expected that not all components of the tracking error Fourier series can be driven to zero.

IV. Experimental results

This chapter presents the application of the learning algorithm to the flying inverted pendulum experiment introduced in Chapter II. We start by presenting the experimental setup, then discuss the implementation of the learning algorithm, and finally show results obtained.

A. Experimental Setup

Experiments were carried out in the Flying Machine Arena, an aerial vehicle development platform at ETH Zurich [19]. The quadrocopters used in the experiments are modified Ascending Technologies 'Hummingbird' vehicles [20] equipped with custom electronics to allow greater control over the low-level control algorithms. A small cup-shaped pendulum mounting point is attached to the top of the vehicle, and the pendulum's center of mass is $59\,\mathrm{cm}$ away from its base. A photograph of the experimental setup is shown in Figure 3. An infrared motion tracking system measures the position and attitude of the vehicle, as well as the position of the pendulum. A Luenberger observer is used to filter the sensory data and provide full state information to the controller. The observer also compensates for systematic latencies occurring in the control loop by using the control inputs to project the system state into the future.

B. Implementation of the Learning Algorithm

The learning algorithm was implemented such that adaptation of the control inputs occurs without interrupting the circling motion of the vehicle. Each iteration of the learning algorithm consists of three steps:

1) Measure the tracking error $\tilde{\mathrm{y}}$ for multiple circles. While measuring the error over one circle would suffice, averaging it over several circles improves the rejection of non-repetitive disturbances. In our experiments, the tracking error was measured and averaged over three circles.
2) Compute the updated input correction Fourier series $\hat{\mathrm{v}}$ according to Equation (25).
3) Wait for several circles to allow the system to converge under the new input correction. Our experiments showed that this step is important in order to correctly measure the systematic tracking error in the next iteration. For the experiments presented in this paper, we chose to wait for two circles before starting the error measurement for the next iteration.

We chose the tracking error output to be the vehicle position error in the rotating coordinate system: $\tilde{\mathrm{y}} = (\tilde{u}, \tilde{v}, \tilde{w})$. This implies that the objective is for the vehicle to correctly track the horizontal circle in size, phase, and height. The nominal trajectory was set to a circle radius R of $0.3\,\mathrm{m}$. Errors in the pendulum position $\tilde{p}, \tilde{q}$ are not penalized in these experiments.

Fig. 3. Photograph of experiments: The vehicle flying a circle while balancing the inverted pendulum.

The update rate γ was chosen to be 0.3, a value that showed a good trade-off between noise rejection and speed of convergence in our experiments. The order of the error and input Fourier series, N, was varied during experiments. It was found that orders higher than $N = 1$ had little impact on the tracking performance during our experiments.

C. Results

The circular motion of the pendulum was started with all correction coefficients set to zero, i.e. $\hat{\mathrm{v}} = 0$. Figure 4 shows the Euclidian norm of the error Fourier series coefficients over ten iterations. It can be seen that initial tracking performance is relatively poor with peak tracking errors of $64\,\mathrm{cm}$. The errors are then quickly reduced over the first three iterations, after which non-repeatable disturbances cause them to vary from iteration to iteration while small improvements are made. After ten iterations of the learning

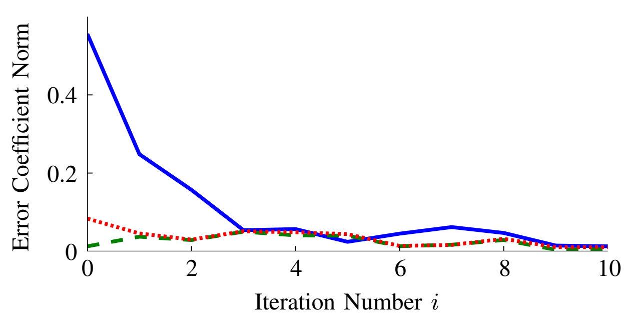

Fig. 4. Evolution of the tracking error output Fourier coefficients during learning. The lines shown represent the Euclidian norm of the Fourier coefficients: $||Ca_0||$ (solid blue), $||Ca_1||$ (dashed green) and $||Cb_1||$ (dotted red). Higher order terms are significantly smaller, and have been omitted in this figure.

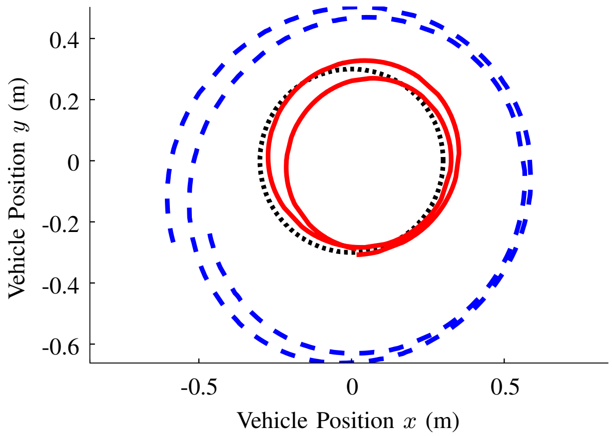

Fig. 5. Horizontal flight trajectory of the quadrocopter over two rounds of the circular trajectory. The dashed blue line shows the flight path before learning, the solid red line shows flight after ten iterations of learning. The nominal trajectory is shown in dotted black. Errors in height are not shown; they were reduced from a maximum error of $0.42\,\mathrm{m}$ to a maximum of $0.01\,\mathrm{m}$.

algorithm, the peak tracking error is reduced to $11\,\mathrm{cm}$, and subsequent iterations do not improve tracking performance significantly.

Figure 5 shows the flight path of the vehicle in the horizontal plane before learning and after the tenth learning iteration. Note that the initial flight path shows large errors, with the flown circle being much too large, shifted from the desired centre point, and warped. After ten learning iterations, the tracking performance has improved considerably, although remaining, largely unrepeatable, disturbances still prevent the vehicle from following the trajectory perfectly.

V. Conclusion and Outlook

This paper evaluated an iterative adaptation scheme that improves tracking performance when periodic disturbances cause poor tracking under feedback control. We have derived convergence properties for the presented method, and have shown that our approach greatly improves performance in an experiment where a quadrocopter balances an inverted pendulum while flying circles.

The method was presented with a focus on the specific problem of trajectories for quadrocopters, and a number of assumptions were made in order to simplify derivations. As is, the method is limited to dynamic systems where the linearization around the nominal trajectory is – under an appropriate coordinate transformation – time-invariant. We intend to investigate transferability to more general system descriptions, and hope to perform more experiments to verify its applicability to other problems.

One open question is the choice of the order of the Fourier series used to represent tracking errors and correction inputs. While we chose the order manually for our experiments, more systematic approaches could be considered.

Furthermore, it would be worthwhile to investigate whether learnt correction values could be transferred to similar maneuvers, as proposed for Iterative Learning Control in [11]. This would be particularly applicable to our experiment because attempts to fly large circles with the inverted pendulum currently lead to crashes before learning permits correct tracking of the trajectory.

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. Mellinger, N. Michael, and V. Kumar, “Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors,” *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[2] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The GRASP Multiple Micro UAV Testbed,” *IEEE Robotics and Automation Magazine*, vol. 17, no. 3, pp. 56–65, 2010.

[3] M. Muller, S. Lupashin, and R. D’Andrea, “Quadrocopter Ball Juggling,” in *International Conference on Intelligent Robots and Systems*, 2011.

[4] R. Ritz, M. W. Müller, M. Hehn, and R. D’Andrea, “Cooperative Quadrocopter Ball Throwing and Catching,” in *International Conference on Intelligent Robots and Systems*, 2012.

[5] 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.

[6] 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.

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

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

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

[10] 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.

[11] 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.

[12] 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.

[13] M. Hehn and R. D’Andrea, “A Flying Inverted Pendulum,” in *International Conference on Robotics and Automation*, 2011.

[14] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, “Aerodynamics and Control of Autonomous Quadrotor Helicopters in Aggressive Maneuvering,” in *International Conference on Robotics and Automation*, 2009.

[15] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. I*, 3rd ed. Athena Scientific, 2005.

[16] C.-T. Chen, *Linear System Theory and Design*, 3rd ed. Oxford University Press, 1998.

[17] D. S. Bernstein, *Matrix Mathematics*. Princeton University Press, 2005.

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

[19] 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.

[20] 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.