

Feed-Forward Parameter Identification for Precise Periodic Quadcopter Motions

Angela P. Schoellig, Clemens Wiltsche and Raffaello D'Andrea

Abstract—This paper presents an approach for precisely tracking periodic trajectories with a quadcopter. In order to improve temporal and spatial tracking performance, we propose a feed-forward strategy that adapts the motion parameters sent to the vehicle controller. The motion parameters are either adjusted on the fly or, in order to avoid initial transients, identified prior to the flight performance. We outline an identification scheme that tunes parameters for a large class of periodic motions, and requires only a small number of identification experiments prior to flight. This reduced identification is based on analysis and experiments showing that the quadcopter's closed-loop dynamics can be approximated by three directionally decoupled linear systems. We show the effectiveness of this approach by performing a sequence of periodic motions on real quadcopters using the tuned parameters obtained by the reduced identification.

I. INTRODUCTION

The objective of the research presented in this paper is to have a quadcopter accurately track three-dimensional periodic motions, without incurring large transients at the beginning of the motion. This research is motivated by the ultimate goal of performing quadcopter choreographies along to music. To achieve precise synchronization to a given periodic (music) reference signal, and to achieve exact reference trajectory tracking, we concentrate on adapting the parameters of the feed-forward input signal sent to the vehicle controller. An example of the 3D periodic motion considered in this paper is pictured in Fig. 1.

Trajectory tracking with quadcopters is typically achieved using feedback control approaches. Methods range from classical PID control, backstepping techniques and nonlinear control [1]–[4], to LQ optimal solutions and model predictive control, e.g. [5]. Such controllers, however, are not usually able to achieve high-performance trajectory tracking with zero phase lag for arbitrary periodic quadcopter motions of varying angular frequencies. Perfect temporal accuracy can only be achieved by using different controllers (or controller parameters) for different motions and motion frequencies. Moreover, feedback control is inherently causal because the control actions depend only on past measurements. Causality, imperfect initial conditions and model errors effect an initial transient phase, in which the tracking errors are substantial.

For choreographies in which motions are changed in quick succession, designing separate controllers for different motions is impractical, and the transient behavior as described

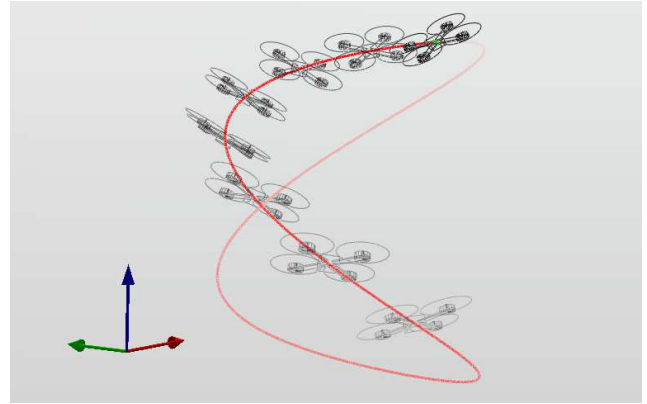


Fig. 1. An example of a 3D periodic motion (with $\omega_d^{x,y} = 2\omega_d^z = \pi$ rad/s, $A_d^{x,y} = A_d^z/2 = 0.4$ m, $\theta_d^x = \pi/2$).

above is highly undesirable. Switching between different controllers may even cause instability.

In this paper, we propose a control strategy that builds upon the same basic trajectory-tracking controller for periodic motions at all frequencies, but adapts the parameters of the actual input to the controller in order to guarantee precise tracking and synchronization. These parameters can be identified prior to the flight performance to effectively reduce transient time and tracking errors.

Other research on motion synchronization with external inputs has focused largely on real-time interaction between robots and the environment. This research mostly deals with real-time signal processing and synchronization schemes, and typically features humanoid robots that perform rhythmic motions such as dancing, drumming and singing in tempo with an exogenous signal [6]–[8]. In contrast, our work features aerial robots and *a priori* known reference signals.

The contribution of this paper is a feed-forward strategy that avoids the large transients, preserves the shape of the periodic motion and maintains high temporal accuracy, even in the first period of the motion. It is possible to adapt only the amplitude and phase of the motion, which is done either online, or offline prior to the actual flight performance. We show that, for directionally decoupled linear systems, the identification of offline parameters requires only a small number of experiments that can be stored concisely in a table. The general idea is based on first identifying the linear closed-loop transfer function and then using it to compensate for the steady-state errors in advance [9]. This ‘black box’ approach allows the strategy presented herein to be used for any (approximately) linear system with independent directions.

The authors are with the Institute for Dynamic Systems and Control, ETH Zurich, 8092 Zurich, Switzerland. (aschoellig, wclemens, rdandrea)@ethz.ch

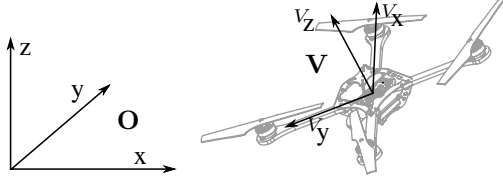


Fig. 2. The inertial coordinate system \mathbf{O} and the vehicle coordinate system \mathbf{V} defining the vehicle attitude.

This feed-forward adaptation scheme was applied to quadcopter music performances. Videos are available at <http://tiny.cc/MusicInMotion>.

The 3D periodic motion primitives considered in this paper are presented in Sec. II. The quadcopter dynamics and the trajectory-tracking controller are introduced in Sec. III and IV, respectively. The quadcopter response to the motion primitives is investigated in Sec. V, where an online parameter adaptation strategy is introduced and relevant system properties are derived. The system properties are then used in Sec. VI to develop an offline parameter adaptation strategy. We show the effectiveness of the offline strategy by performing a sequence of motion primitives with a quadcopter. Experiments are conducted in the Flying Machine Arena, an indoor test environment for quadcopters. For a detailed description of the experimental setup, refer to [10].

II. PERIODIC MOTIONS

We present a framework for periodic motion primitives in three dimensions, generalizing from the one-dimensional side-to-side motion in our previous work [11]. We specify motion primitives on the vehicle's position in the inertial coordinate system \mathbf{O} , see Fig. 2. The heading of the quadcopter (that is, the yaw angle in Z-Y-X Euler attitude representation) is held at zero. The remaining rotational degrees of freedom are defined by the quadcopter dynamics, cf. [10] and Sec. III. The desired position of the quadcopter at time t , $s_d(t) = (x_d(t), y_d(t), z_d(t))$ is given by

$$\begin{bmatrix} x_d(t) \\ y_d(t) \\ z_d(t) \end{bmatrix} = \begin{bmatrix} \delta_d^x \\ \delta_d^y \\ \delta_d^z \end{bmatrix} + \begin{bmatrix} A_d^x \cos(\omega_d^x t + \theta_d^x) \\ A_d^y \cos(\omega_d^y t + \theta_d^y) \\ A_d^z \cos(\omega_d^z t + \theta_d^z) \end{bmatrix}, \quad (1)$$

where $(\delta_d^x, \delta_d^y, \delta_d^z)$ is the desired center position of the motion primitive. The shape of the primitive is adjustable by a set of nine motion parameters: in each direction $i \in \{x, y, z\}$, we specify the desired amplitude A_d^i , frequency ω_d^i and phase θ_d^i . By varying the nine parameters, a wide range of different motions can be expressed, such as side-to-side motions, bounces, ellipses, eights and spirals, see for example Fig. 1.

Not all parameter values result in periodic motion primitives that can be followed by the quadcopter. If the amplitudes or frequencies are too high, the motion becomes infeasible due to thrust limitations of the propellers and limited sensor range. This is investigated in [10]. The motions considered in this paper are assumed to be feasible.

One of our goals is to synchronize the motion to an external reference signal (for example, the beat of a music piece),

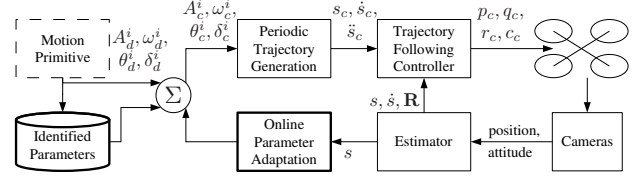


Fig. 3. The control configuration shows the online and offline motion parameter adaptation (bold boxes), and the underlying trajectory-following controller.

which sets the motion frequencies ω_d^i . Ultimately, motion primitives can be arranged in a choreographic sequence and be timed to music [12].

III. QUADROPTER DYNAMICS

The translational motion of a quadcopter in the inertial frame \mathbf{O} is described by

$$\begin{bmatrix} \ddot{x}(t) \\ \ddot{y}(t) \\ \ddot{z}(t) \end{bmatrix} = \mathbf{R}(t) \begin{bmatrix} 0 \\ 0 \\ c(t) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \Leftrightarrow \begin{cases} \ddot{x} = c b^x \\ \ddot{y} = c b^y \\ \ddot{z} = c b^z - g \end{cases}, \quad (2)$$

where $\mathbf{R}(t)$ is the rotation matrix from the body frame \mathbf{V} to the inertial frame \mathbf{O} , $c(t)$ is the collective thrust of the four propellers, and g is the acceleration due to gravity. The values (b^x, b^y, b^z) correspond to the third column of the rotation matrix, namely $(\mathbf{R}_{13}, \mathbf{R}_{23}, \mathbf{R}_{33})$, and represent the direction of the collective thrust in the inertial frame \mathbf{O} .

The rotation matrix \mathbf{R} evolves according to

$$\dot{\mathbf{R}}(t) = \mathbf{R}(t) \begin{bmatrix} 0 & -r(t) & q(t) \\ r(t) & 0 & -p(t) \\ -q(t) & p(t) & 0 \end{bmatrix}, \quad (3)$$

where (p, q, r) represent the quadcopter angular body velocities around the body (V_x, V_y, V_z) axes, see Fig. 2 and [13]. The quadcopter is controlled by four inputs: a collective thrust command c_c , and commanded angular body velocities (p_c, q_c, r_c) , see Fig. 3 and Sec. IV-A.

The controller design introduced below is based on the above dynamics model. For a more detailed quadcopter model that includes rotational dynamics refer to [10].

IV. QUADROPTER CONTROL

The overall control configuration of our approach is outlined in Fig. 3. Below we describe the trajectory-following controller (TFC) that is the basis of our approach, and analyze its properties with particular regard to periodic motions as described in Sec. II.

A. Approach

The TFC accepts position, velocity, and acceleration commands, denoted by $s_c(t)$, $\dot{s}_c(t)$ and $\ddot{s}_c(t)$ respectively, and attempts to maintain the quadcopter on this specified trajectory. Control is based on the estimated quadcopter position $s = (x, y, z)$, velocity \dot{s} and attitude \mathbf{R} , see Fig. 3. The TFC outputs the commands c_c and (p_c, q_c, r_c) to the vehicle. The TFC consists of three separate loops for altitude, horizontal position, and attitude, see Fig. 4. While the TFC

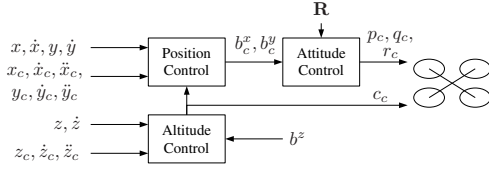


Fig. 4. Cascaded control loops of the trajectory-following controller (TFC).

operates in discrete time, the controller design is based on the continuous-time system dynamics representation.

The altitude control is designed such that it responds to altitude errors ($z - z_c$) like a second-order system with time constant τ_z and damping ratio ζ_z ,

$$\ddot{z} = -\frac{2\zeta_z}{\tau_z}(\dot{z} - \dot{z}_c) - \frac{1}{\tau_z^2}(z - z_c) + \ddot{z}_c. \quad (4)$$

It uses the collective thrust to achieve this. With (2) and (4), we obtain

$$c_c = (\ddot{z} + g)/b^z. \quad (5)$$

Similarly, the two horizontal position control loops are shaped based on (2) with c_c from (5), resulting in the commanded rotation matrix entries b_c^x and b_c^y . The attitude control is shaped such that the two rotation matrix entries b^x, b^y react in the manner of a first-order system with time constant τ_{rp} ; that is, for x : $\dot{b}_c^x = (b^x - b_c^x)/\tau_{rp}$. The values \dot{b}_c^x, \dot{b}_c^y are directly mapped to the commanded angular body velocities (p_c, q_c) using (3) and the estimated attitude \mathbf{R} ,

$$\begin{bmatrix} p_c \\ q_c \end{bmatrix} = \frac{1}{\mathbf{R}_{33}} \begin{bmatrix} \mathbf{R}_{21} & -\mathbf{R}_{11} \\ \mathbf{R}_{22} & -\mathbf{R}_{12} \end{bmatrix} \begin{bmatrix} \dot{b}_c^x \\ \dot{b}_c^y \end{bmatrix}. \quad (6)$$

Vehicle yaw control can be considered separately, since rotations around the body V_z axis do not affect the above dynamics. The yaw controller is a proportional controller and the resulting yaw angle rate is mapped to r_c using the kinematic relations of Euler angles. The innermost loop, on board the quadcopter, controls the angle rates (p, q, r) to the calculated set points (p_c, q_c, r_c).

The feedback loop closed by the TFC is responsible for maintaining the quadcopter on a trajectory, which is provided by the periodic trajectory generation (PTG). The PTG is based on the motion primitives in Sec. II and implements (1) with motion parameters $\delta_c^i, A_c^i, \omega_c^i, \theta_c^i$, where the subscript 'c' stands for 'commanded'. The commanded parameters are one of the following: simply the desired parameters (no adaptation); adapted online (see Sec. V); or adapted both online and offline (see Sec. VI). Fig. 3 illustrates the three options.

B. Analysis

To highlight the key characteristics of the above control architecture, we analyze the closed-loop dynamics under the following simplifying assumptions: (i) the commanded collective thrust can be changed instantaneously, that is, $c(t) = c_c(t)$; (ii) the estimated rotation matrix entry b^z corresponds to the actual one; and (iii) we have direct control over the other two rotation matrix entries, namely

$b^x(t) = b_c^x(t)$ and $b^y(t) = b_c^y(t)$. Then, (2) can be written as

$$\begin{aligned} \ddot{x} &= u_x, & \text{with } u_x &= f_x(t, b_c^x) = c(t)b_c^x(t) \\ \ddot{y} &= u_y, & \text{with } u_y &= f_y(t, b_c^y) = c(t)b_c^y(t) \\ \ddot{z} &= u_z, & \text{with } u_z &= f_z(t, c_c) = c_c b^z(t) - g, \end{aligned} \quad (7)$$

where u_i , $i \in \{x, y, z\}$, represent the flat inputs resulting from a feedback linearization [14]. Such a transformation between the virtual inputs u_i and b_c^x, b_c^y, c_c was applied in the previous controller equations, cf. (5), allowing us to use techniques from linear feedback control design. Eq. (7) decouples the three directions and shows linear system behavior in each direction. In the ideal case where the quadcopter dynamics correspond to the model (2), the combination of feedback linearization with velocity and acceleration feed-forward (see \dot{z}_c, \ddot{z}_c in (5)) results in perfect trajectory tracking.

Assumptions (i) and (ii) are good approximations due to the fast motor dynamics (with motor time constants more than six times faster than the controller time constants, cf. [15]) and due to precise attitude estimates that are based on high-accuracy camera measurements and that include a prediction step to compensate for system latencies. Assumption (iii) is true only if the system has zero rotational inertia (cf. [10]), which is not the case. In reality, exceptionally high angular accelerations (cf. [15]) can be achieved and rotational inertia terms are small. Thus, the overall closed-loop dynamics may still be expected to be approximately linear. The approximate directional independence and linear dynamics behavior is exploited in Sec. V and further investigated using experimental data, since additional effects of latencies, time discretization, on-board dynamics, and modeling errors are difficult to predict.

C. Results

In a first attempt, the desired periodic trajectory is directly fed to the vehicle controller (TFC); that is, $s_c(t) := s_d(t)$. The quadcopter response is a sinusoidal motion with a (after the transient phase) constant change in amplitude, phase and center position. Phase shift and amplitude error are observed in each translational direction and are not necessarily equal in size. The frequency of the quadcopter motion corresponds to the commanded one. This suggests that the quadcopter controlled by the TFC can be regarded as a linear system, which explains the phase offset and amplitude amplification.

Fig. 5 (top figure) shows the result for a planar side-to-side motion. The amplitude error of the quadcopter response (black solid line) is obvious, whereas the phase error between the reference trajectory and the actual quadcopter response is hardly noticeable. In actual experiments, however, small phase errors are visible and audible when, for example, quadcopter choreography is timed to music, as a phase shift causes a misalignment between the flight trajectory and the music beat. For the side-to-side motion, music beats may occur at the outermost points of the trajectory. In this case, humans are particularly sensitive to non-zero vehicle velocity at beat times. Correspondingly, the bottom plot of

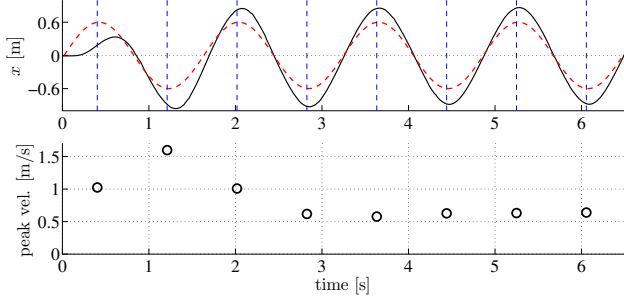


Fig. 5. Side-to-side motion (with $\omega_d^x = 3.25$ rad/s, $A_d^x = 0.6$ m): *no motion parameter adaptation*. **Top:** quadcopter response (solid) for a desired oscillation in the x direction (dashed). **Bottom:** corresponding peak velocities, i.e. absolute value of vehicle velocity at the peaks of the desired trajectory. High peak velocities imply a large phase error.

Fig. 5 illustrates the velocity of the quadcopter at beat times, i.e. when the reference trajectory reaches its maximum or minimum value. Note that if different directions exhibit different a phase shift or amplitude error, the shape of the motion primitive can even be changed, see Fig. 6.

In the following section, we correct for the amplitude error and phase shift, and investigate the closed-loop behavior in more detail by analyzing the steady-state correction terms. Because the commanded amplitude is often amplified by the closed-loop system (rendering feasible commanded trajectories into infeasible quadcopter motions), the correction is done first.

V. ONLINE CORRECTION

A. Approach

The goal is to accurately track the desired trajectory $s_d(t)$ by minimizing the deviation from the estimated trajectory $s(t)$. To this end, the motion parameters in the commanded trajectory $s_c(t)$ are adjusted by directionally decoupled integral controllers, see Fig. 3. The approach is based on our previous work, see [11].

For notational convenience, we drop the superscripts $i \in \{x, y, z\}$ indicating the direction throughout this section. The motion parameters of the commanded trajectory are set to

$$\begin{aligned}\theta_c(t) &= \theta_d + \theta_{\text{on}}(t), & A_c(t) &= A_d + A_{\text{on}}(t), \\ \delta_c(t) &= \delta_d + \delta_{\text{on}}(t),\end{aligned}$$

where the subscript ‘on’ indicates the online correction terms. They are updated in real time, during the flight.

We first determine the additive error in amplitude A_t , phase θ_t and position δ_t of the quadcopter response

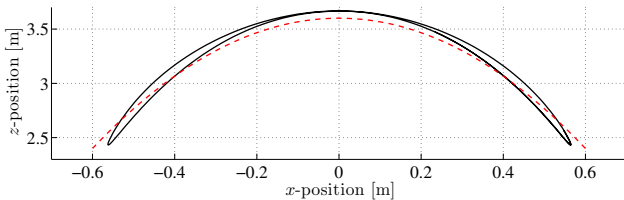


Fig. 6. Vertical bounce motion (with $\omega_d^{x,y} = \omega_d^z/2 = 1.56$ rad/s, $A_d^{x,y} = A_d^z/2 = 0.6$ m): *no motion parameter adaptation*. The vehicle's response (solid) can differ in shape from the desired trajectory (dashed).

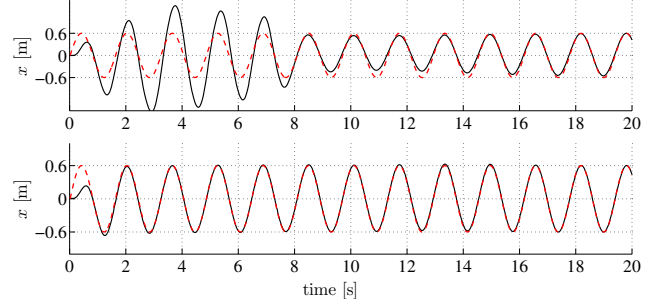


Fig. 7. Side-to-side motion. **Top:** online motion parameter adaptation only, quadcopter response (solid) for a desired oscillation in the x direction (dashed). **Bottom:** offline motion parameter adaptation, with online motion parameter adaptation turned on after 2 periods.

at time t . We multiply the position estimate $s(t)$ and with the two reference signals, $r_{\cos}(t) = \cos(\omega_d t + \theta_d)$ and $r_{\sin}(t) = \sin(\omega_d t + \theta_d)$, and integrate the result over N periods, that is $T = \frac{2\pi N}{\omega_d}$. We assume that the errors stay constant over the interval $[t - T, t]$. This yields

$$\begin{aligned}\eta_1(t) &= \frac{1}{T} \int_{t-T}^t s(t) r_{\cos}(t) dt = \frac{A_t + A_d}{2} \cos(\theta_t) \\ \eta_2(t) &= \frac{1}{T} \int_{t-T}^t s(t) r_{\sin}(t) dt = \frac{A_t + A_d}{2} \sin(\theta_t),\end{aligned}\quad (8)$$

and finally

$$\begin{aligned}A_t &= 2\sqrt{\eta_1(t)^2 + \eta_2(t)^2} - A_d \\ \theta_t &= -\arctan(\eta_2(t)/\eta_1(t)).\end{aligned}\quad (9)$$

The position error of the sinusoidal response is $\delta_t = \int_{t-T}^t (\delta_d - s(t)) dt$. Since an insufficient number of measurements is available at the beginning of a motion, the integrals deliver reliable values only after several periods of the motion primitive. The online correction terms are calculated by integrating the errors according to

$$A_{\text{on}}(t) = k_A \int_0^t A_\tau d\tau, \quad \theta_{\text{on}}(t) = k_\theta \int_0^t \theta_\tau d\tau, \quad (10)$$

and similarly for $\delta_{\text{on}}(t)$. The gains k_θ , k_A , k_δ are chosen to ensure converge of the online correction terms to the steady-state values $\theta_{\text{on},\infty}$, $A_{\text{on},\infty}$ and $\delta_{\text{on},\infty}$, respectively. Note again that the above online parameter adaptation strategy is implemented for each direction separately.

B. Results

Using the proposed online parameter adaptation strategy, the errors in amplitude, phase and center position are effectively regulated to zero, see Fig. 7. We observe a substantial transient phase before the online correction terms attain steady state, see Fig. 8. This is mainly due to the fact that the error identification scheme (8)-(9) only provides reliable values after several periods.

To draw further conclusions, we consider the steady-state values in the following form: the amplitude-normalized amplification factor,

$$\alpha_{\text{on},\infty}^i = (A_d^i + A_{\text{on},\infty}^i)/A_d^i, \quad i \in \{x, y, z\}, \quad (11)$$

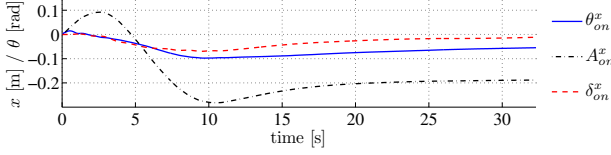


Fig. 8. Side-to-side motion: convergence of the online correction terms.

and the steady-state phase and offset, $\theta_{on,\infty}^i$ and $\delta_{on,\infty}^i$, as before.

We found that, when executing the *same* motion primitive multiple times, the standard deviations of the corresponding steady-state values are small. We call this the *intra-class variability*, which is a measure of the repeatability of the experiments. We then investigated the *inter-class variability*: the standard deviation of the steady-state values $\alpha_{on,\infty}^i$, $\theta_{on,\infty}^i$, and $\delta_{on,\infty}^i$ is evaluated for *different* motion primitives. We found that the inter-class variability of the obtained steady-state values for a given translational direction at a given directional frequency is of the same order of magnitude as the intra-class variability. Supported by the considerations in Sec. IV-B and by experiments shown below, this leads to the following conclusions:

- *Decoupled directions.* The steady-state values $\alpha_{on,\infty}^i$, $\theta_{on,\infty}^i$ in each direction are independent of the motion's components in the other directions. Moreover, as expected from the quadcopter's symmetry, the x and y directions exhibit the same behavior.
- *Linear behavior.* Considering the motion component in one direction i , the steady-state values depend only on the motion's frequency in this direction ω_d^i .

Fig. 9 depicts the amplification factor $\alpha_{on,\infty}^i$ and steady-state phase $\theta_{on,\infty}^i$ against the motion's directional frequency ω_d^i for the two directions $i \in \{x, y\}$. Plots are shown for ten different periodic motion primitives in 1D, 2D and 3D of various amplitudes up to 0.6 m and different relative phase shifts. The standard deviation of the steady-state terms is indicated by the vertical labels. In particular, the variability of the amplification factors translates, for the largest amplitude, to a residual deviation of ± 1.5 cm, which lies within the TFC hover accuracy of ± 2 cm. The variability of the phase translates to a residual time shift of ± 25 ms (at maximum), which is within the human audiovisual synchrony perception limits [16], and therefore likewise negligible. These results affirm the linearity and directional independence property. Note that when performing the identification run with the same motion primitives several times, the variability is of the same order of magnitude. Consequently, we do not lose accuracy when identifying $\theta_{on,\infty}^i$, $A_{on,\infty}^i$ for one motion primitive and later using it for another one as described in the next section.

The steady-state correction terms for the center position $\delta_{on,\infty}^i$ lie within the hover accuracy with a variability of the same magnitude. Thus, the value $\delta_{on,\infty}^i$ cannot be called repeatable and may either be neglected because of its small average size or identified each time when flying.

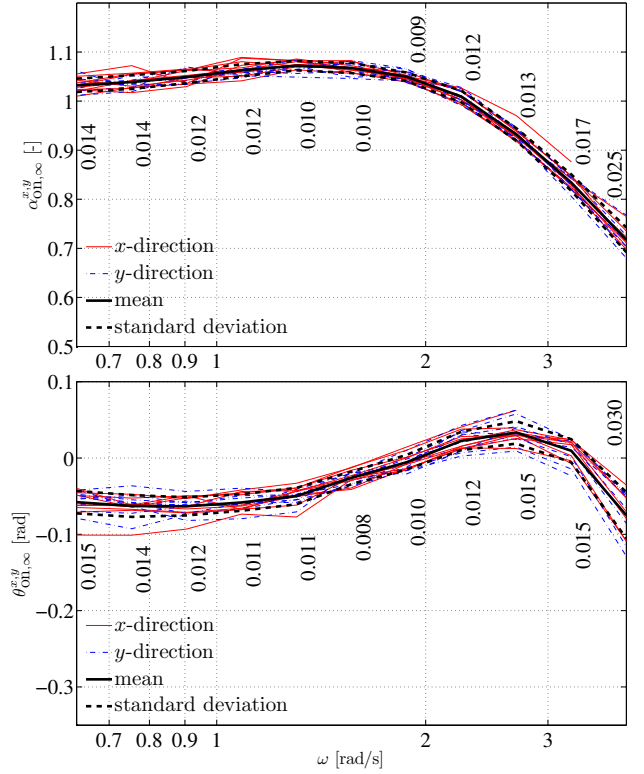


Fig. 9. Steady-state correction values in x and y for ten different motions primitives in 1D, 2D and 3D. Here, $\omega = \omega_d^x = \omega_d^y$. The vertical labels provide the standard deviation for the samples. **Top:** steady-state amplification factor $\alpha_{on,\infty}^{x,y}$. **Bottom:** steady-state phase $\theta_{on,\infty}^{x,y}$.

VI. OFFLINE IDENTIFICATION

A. Approach

The previous section showed that the steady-state values obtained from the online correction are repeatable. Consequently, relevant steady-state values can be extracted once, and later applied to improve the transient performance. We employ offline identified parameters in addition to the online adaptation. Again, we drop the superscripts indicating the direction. The parameters of the commanded trajectory are set to

$$A_c(t) = \alpha_{\text{off}} A_d + A_{\text{on}}(t), \quad \delta_c(t) = \delta_d + \delta_{\text{on}}(t), \\ \theta_c(t) = \theta_d + \theta_{\text{off}} + \theta_{\text{on}}(t),$$

where the subscript 'off' indicates the offline motion parameters identified prior to the experiment. Note that there is no offline parameter for the center point δ_c , because the errors are small in size and less repeatable and, thus, more efficiently handled by the online adaptation strategy. The offline parameters are selected at the start of a motion on the basis of the desired motion primitive and stay constant for the duration of the executed motion. For a given motion, the offline correction terms are set to the steady-state values obtained from the online correction, see Sec. V-B:

$$\alpha_{\text{off}} = \alpha_{on,\infty}, \quad \theta_{\text{off}} = \theta_{on,\infty}.$$

We make use of the directional independence and linearity property derived above to efficiently identify the offline cor-

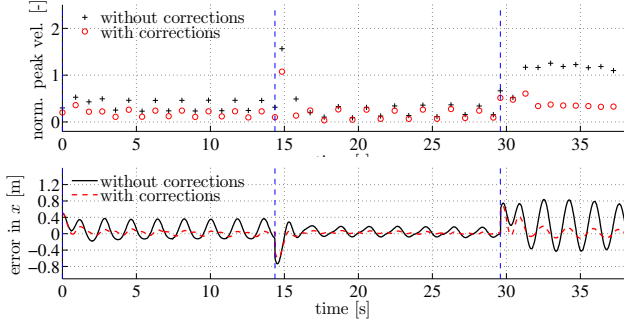


Fig. 10. Sequence of motions (comprising a circular motion in 3D, a swing motion in 3D and a horizontal circle): *with and without feed-forward corrections*. Offline correction terms were obtained from a *reduced identification*. **Top:** absolute value of velocities at the peaks of the desired trajectory normalized by dividing by $A_d^x \omega_d^x$. **Bottom:** errors between desired trajectory and vehicle response ($s_d(t) - s(t)$) are plotted.

rection terms for all periodic motions that can be expressed in our framework (1). We perform a *single* identification run with fixed amplitudes A_d^i , phases θ_d^i and center point δ_d^i and vary only the frequency $\omega_d^i = \omega$. Moreover, since the x and y direction exhibit the same dynamics, an identification run with a 2D motion primitive in x or y , and z is sufficient to completely identify all necessary feed-forward parameters.

Conceptually, the offline identification strategy results in a pair of maps

$$\Gamma_{xy} : \omega_d^{x,y} \mapsto (\alpha_{\text{off}}^{x,y}, \theta_{\text{off}}^{x,y}), \quad \Gamma_z : \omega_d^z \mapsto (\alpha_{\text{off}}^z, \theta_{\text{off}}^z),$$

where the superscript x, y indicates that this map is used for both, the x and y direction. The values are stored in a table with rows $[\omega_d^{x,y} \ \omega_d^z \ \theta_{\text{off}}^{x,y} \ \theta_{\text{off}}^z \ \alpha_{\text{off}}^{x,y} \ \alpha_{\text{off}}^z]$. We use linear interpolation between the offline parameters, which are only obtained at a discrete set of frequencies.

B. Results

As compared to using only online parameter adaptation, the proposed offline identification substantially decreases the transient phase, see Fig. 7. The offline parameters are effective from the start of the motion primitive. When combining online and offline correction, the former is used only after several periods.

In order to show the effectiveness of the reduced identification scheme, we perform a sequence of periodic 3D motions with offline parameters obtained from an oscillatory motion in 2D ($A_d^x = A_d^z = 0.4$ m, $\omega_d^x = \omega_d^z = \omega$). Fig. 10 shows that the quadcopter's deviation from the desired trajectory is clearly reduced when using the offline parameter adaptation strategy. As a consequence, the corresponding peak velocities (cf. Fig. 5) are also small indicating that the phase error is reduced. Note that the transient performance can be further improved by starting the vehicle with the appropriate velocity and acceleration.

The videos at <http://tiny.cc/MusicInMotion> show examples of quadcopter choreographies timed to music.

VII. CONCLUSION

In this paper we studied a feed-forward parameter tuning strategy that improves the tracking performance of periodic motion primitives, as compared to pure feedback control, especially during transients. With pre-identified correction terms, the tracking converges to a level virtually imperceptible to a human observer within a single period. The parameter correction terms depend only on the motion's 3D directional frequencies. The translational directions are independent, allowing for an efficient offline identification of correction values. Due to the directional independence, the approach presented in this paper can be applied even to non-periodic 3D motions that are composed of periodic motions in each translational direction.

REFERENCES

- [1] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005, pp. 2247–2252.
- [2] S. Al-Hiddabi, "Quadrotor control using feedback linearization with dynamic extension," in *Proceedings of the International Symposium on Mechatronics and its Applications (ISMA)*, 2009, pp. 1–3.
- [3] T. Lee, M. Leoky, and N. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425.
- [4] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 2520–2525.
- [5] C. Castillo, W. Moreno, and K. Valavanis, "Unmanned helicopter waypoint trajectory tracking using model predictive control," in *Proceedings of the Mediterranean Conference on Control Automation*, 2007, pp. 1–8.
- [6] K. Murata, K. Nakada, K. Yoshii, R. Takeda, T. Torii, H. G. Okuno, Y. Hasegawa, and H. Tsujino, "A robot uses its own microphone to synchronize its steps to musical beats while scattering and singing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 2459–2464.
- [7] J.-J. Aucouturier, "Cheek to chip: dancing robots and AI's future," *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 74–84, 2008.
- [8] D. Grunberg, R. Ellenberg, I. H. Kim, J. H. Oh, P. Y. Oh, and Y. E. Kim, "Development of an autonomous dancing robot," *International Journal of Hybrid Information Technology*, vol. 3, no. 2, 2010.
- [9] P. Allen, "Feed-forward compensated high switching speed digital phase-locked loop frequency synthesizer," *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 371–374, 1999.
- [10] A. P. Schoellig, M. Hehn, S. Lupashin, and R. D'Andrea, "Feasibility of motion primitives for choreographed quadcopter flight," in *Proceedings of the American Control Conference (ACC)*, 2011, pp. 3843–3849.
- [11] A. P. Schoellig, F. Augugliaro, S. Lupashin, and R. D'Andrea, "Synchronizing the motion of a quadcopter to music," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 3355–3360.
- [12] A. P. Schoellig, F. Augugliaro, and R. D'Andrea, "A platform for dance performances with multiple quadcopters," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) - Workshop on Robots and Musical Expressions*, 2010, pp. 1–8.
- [13] P. C. Hughes, *Spacecraft Attitude Dynamics*. John Wiley & Sons, 1986.
- [14] A. Isidori, *Nonlinear control systems*. Springer Verlag, 1995.
- [15] M. Hehn and R. D'Andrea, "Quadcopter trajectory generation and control," in *IFAC World Congress*, vol. 18, 2011, pp. 1485–1491.
- [16] R. Arrighi, D. Alais, and D. Burr, "Perceptual synchrony of audiovisual streams for natural and artificial motion sequences," *Journal of Vision*, vol. 6, no. 3, pp. 260–268, 2006.