2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

Synchronizing the Motion of a Quadrocopter to Music

Angela Schöllig, Federico Augugliaro, Sergei Lupashin and Raffaello D'Andrea

Abstract— **This paper presents a quadrocopter flying in rhythm to music. The quadrocopter performs a periodic side-to-side motion in time to a musical beat. Underlying controllers are designed that stabilize the vehicle and produce a swinging motion. Synchronization is then achieved by using concepts from phase-locked loops. A phase comparator combined with a correction algorithm eliminate the phase error between the music reference and the actual quadrocopter motion.**

Experimental results show fast and effective synchronization that is robust to sudden changes in the reference amplitude and frequency. Changes in frequency and amplitude are tracked precisely when adding an additional feedforward component, based on an experimentally determined look-up table.

I. INTRODUCTION

Controls for the synchronization of movement are indispensable in any robotics application where high performance, precision, and agility are required. Involving a coordination in time between two or more systems or events, controls for synchronized behavior fall into two broad categories: algorithms focused on coordinating internal signals (for example, from multiple dynamic subsystems), and algorithms focused on coordinating with external inputs (from other bodies or the environment).

Driven by the need to control the movement of complex industrial robots, early synchronized control algorithms dealt with coordinating internal signals from multiple dynamic subsystems. Examples include multi-axis machine tools [1], [2], parallel manipulators [3], and multi-robot assembling machines [4]. More recently, motion synchronization has led, for example, to the development of algorithms for spacecraft formation flying [5] and is used in mobile robots to reduce the velocity errors between driving wheels [6].

Control algorithms for synchronizing robots with external inputs have been largely developed within the field of humanoid robotics, where researchers aim for a lively interaction between robots and their environment. Examples of motion synchronization with external inputs have thus far dealt mostly with easily repeatable motion primitives like steps or up-and-down arm movements, including robots that step and sing along to music [7]–[10] or drum in tempo with an exogenous signal [11]–[13]. Periodic inputs and rhythmic robot movements are often featured in these applications for their simplicity, repetitiveness, aesthetic pleasure, and entertainment value.

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

Associated video at http://tinyurl.com/dancingquadro.

Fig. 1. The desired side-to-side motion.

The objective of this paper is to synchronize the motion of a quadrocopter to music. The vehicle's nonlinear and unstable dynamics present significant challenges in motion synchronization. Stabilizing control is required just to keep the vehicle in the air, and modeling errors, motor saturation, and communication delays have noticeable effects on the quadrocopter dynamics. An appropriate synchronization algorithm is indispensable to time precisely the response of the vehicle with the music reference. Note that this is not the case in most other approaches dealing with synchronizing rhythmic movement. In digital animation the motion of a character is not affected by mass inertia or system delay, but is directly dictated by the developer, eliminating the need for a synchronization algorithm. Examples of virtual dancing characters are found in [14], [15]. Other papers on dancing robots, as summarized in [16], do not focus on synchronization either, since they deal with systems that are better understood, less sensitive to disturbances, and, generally, easier to manage. The *Keepon Robot* [17] and *Ms DanceR* [18], which dance with humans, are two such examples.

In this paper, a simple motion primitive is chosen for studying the feasibility of our idea. The quadrocopter undergoes a planar side-to-side motion as depicted in Fig. 1, where at beat times the vehicle reaches the outermost points of the trajectory, either on the left or right. In a preliminary step, the music is *pre-processed* and the beat times obtained are transformed into a periodic signal which is used as a reference trajectory for the quadrocopter. Fig. 2 shows the overall control system. Synchronized rhythmic behavior is achieved if the music reference signal and the actual quadrocopter motion are in phase, cf. Fig. 3.

The general idea is borrowed from phase-locked loops (PLL). A phase-locked loop acts on the frequency of a controlled oscillator and matches its output to a periodic reference signal both in frequency and phase. PLLs are widely used in radio, telecommunications, computers, and many other electronic applications, cf. [19], [20]. Inspired by this concept, underlying controllers are designed that turn the unstable quadrocopter dynamics into an oscillating system behavior. A phase comparator detects the phase error between desired reference trajectory and quadrocopter motion.

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

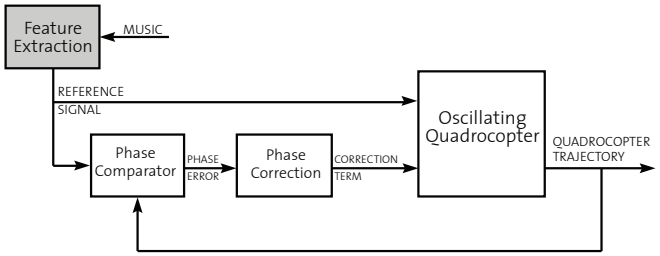

Fig. 2. The overall control system transforming music into an appropriate quadrocopter motion.

After having determined the phase error, it is compensated for by closing the loop on the phase error, similar to [9] and [12]. The efficacy of the proposed synchronization algorithm is experimentally studied. The derived algorithm proves to be able to accurately coordinate the movements of the flying vehicle with the desired music reference. The accompanying video shows the 'dancing' quadrocopter.

In the following sections, the overall control system is presented. Section II explains how the reference signal is generated from the music signal and introduces the quadrocopter dynamics. Section III describes how the required oscillating system is realized by using controllers for the side-to-side motion, while Section IV presents the phase detection and correction step which cause the desired synchronized behavior. Experimental results complete this work.

II. SYSTEM REPRESENTATION

This section introduces the desired quadrocopter motion as extracted from the music and presents a two-dimensional model of the quadrocopter that is used for the controller design.

A. Periodic Motion

The goal of this work is to show a quadrocopter flying in rhythm to music, where the main focus lies on solving the underlying synchronization problem. A simple motion primitive was selected as reference trajectory for the vehicle with the goal of being able to visualize the existing phase error and successful phase locking. The quadrocopter performs a planar side-to-side motion where beats occur at the outermost positions, see Fig. 1. The amplitude and frequency of this lateral motion are modulated by the music's melody and beat intervals, respectively.

For the derivations, a constant amplitude and constant beat interval are considered; that is, music beats occur with a constant frequency. This is a reasonable assumption for many types of music. The efficacy of the derived algorithm for more complicated scenarios is shown in Section V-C.

Analyzing a a piece of music yields a desired beat interval T and amplitude A_d. These values define the reference trajectory that is fed to the quadrocopter. As depicted in Fig. 1, the corresponding desired vehicle trajectory is a sinusoidal side-to-side motion in the $xz-$plane,

$$\begin{aligned} x_d(t) &= A_d \cos(\omega_d t) \\ z_d(t) &= z_d = \text{constant}, \end{aligned} \tag{1}$$

with $\omega_d = \pi/T$. Fig. 3 illustrates the beat-motion relation. Beats occur at the peaks of the trajectory, i.e. two times per period. As an example, if the music tempo is 120 beats per minute, the desired frequency of the oscillating trajectory is $\omega_d = 2\pi$ rad/s. The altitude of the quadrocopter is stabilized at a given height z_d.

B. Quadrocopter Model

The side-to-side motion (1) is defined in the $xz-$plane. In-plane and out-of-plane dynamics are thus decoupled and additional degrees of freedom are separately stabilized. A simplified two-dimensional model of the quadrocopter is depicted in Fig. 4. The equations governing the dynamics of the system are given by

$$\ddot{z}(t) = f(t)\cos\theta(t) - g \tag{2}$$

$$\ddot{x}(t) = f(t)\sin\theta(t) \tag{3}$$

$$\dot{\theta}(t) = u(t), \tag{4}$$

where g is the gravitational constant and $\theta(t)$ is the pitch angle. The inputs to the system are the normalized thrust $f(t)$ in m/s^2 and the pitch rate $u(t)$ in rad/s.

III. CONTROLLER DESIGN

The oscillating quadrocopter motion is achieved by a cascaded controller design: the z-direction is stabilized first and, assuming a constant height, the trajectory-tracking controller for the x-direction is designed.

A. Height Stabilization

In order to stabilize the quadrocopter at a desired altitude z_d, the thrust input $f(t)$ in (2) is chosen such that a linear second-order system is obtained. With

$$f(t) = \frac{1}{\cos\theta(t)}\Big(g - 2\delta_z\omega_z\,\dot{z}(t) - \omega_z^2\,(z(t) - z_d)\Big), \tag{5}$$

the closed-loop dynamics are given by

$$\ddot{z}(t) + 2\delta_z\omega_z\,\dot{z}(t) + \omega_z^2\,z(t) = \omega_z^2\,z_d. \tag{6}$$

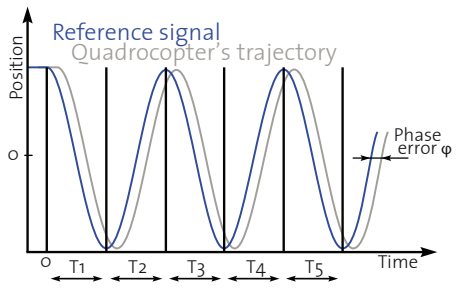

Fig. 3. The desired rhythmic side-to-side motion of the quadrocopter. Vertical lines represent beat times.

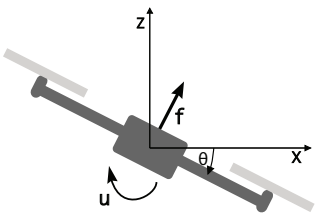

Fig. 4. Schematic of the 2D quadrocopter model.

In most cases, a damping ratio δ_z between 0.7 (underdamped case) and 1 (critically damped case) is a reasonable choice, see [21]. The only remaining design parameter is the natural frequency ω_z.

B. Trajectory Tracking

Building upon the above control scheme and assuming a constant height z_d, the x-dynamics (3) reduce to

$$\ddot{x}(t) = g \tan\theta(t). \tag{7}$$

In addition, the pitch angle $\theta(t)$ is assumed to be small. This is a good approximation for reference trajectories with a small frequency ω_d as compared to the desired amplitude A_d. To first order, $\tan\theta(t) = \theta(t)$, resulting in a linear approximation for the sideways dynamics,

$$\dddot{x}(t) = g\,\dot{\theta}(t) = g\,u(t), \tag{8}$$

relating the position $x(t)$ directly to the angle rate input $u(t)$.

With the aim of following the desired sinusoidal side-to-side motion (1), the input $\overline{u}(t) = gu(t)$ is composed of a feedforward component,

$$\overline{u}_1(t) = \dddot{x}_d(t) = A_d\omega_d^3 \sin(\omega_d t), \tag{9}$$

and an additional feedback term to correct for errors,

$$\begin{aligned}\overline{u}_2(t) = \alpha\,\big(\ddot{x}_d(t) - \ddot{x}(t)\big) + \beta\,\big(\dot{x}_d(t) - \dot{x}(t)\big) \\ + \gamma\,\big(x_d(t) - x(t)\big),\end{aligned} \tag{10}$$

where the control parameters $\alpha,\ \beta$, and γ are defined through

$$\alpha = \omega_x(1+2\delta_x), \quad \beta = \omega_x^2(1+2\delta_x), \quad \gamma = \omega_x^3 \tag{11}$$

and act on the acceleration, velocity, and position errors, respectively. With the choice (11), the characteristic polynomial of the closed-loop system is

$$(s+\omega_x)(s^2 + 2\delta_x\omega_x\, s + \omega_x^2) = 0. \tag{12}$$

The damping ratio δ_x is again chosen to be a value between 0.7 and 1, and w_x remains the only parameter to be chosen in order to achieve satisfactory tracking performance. Finally, the input

$$u(t) = \frac{1}{g}\left(\overline{u}_1(t) + \overline{u}_2(t)\right) \tag{13}$$

is applied to the quadrocopter.

IV. SYNCHRONIZATION

When applying the input $u(t)$ as defined in (13), a phase shift is noticed between the reference trajectory of the sideways motion and the actual quadrocopter trajectory, illustrated in Fig. 3. (Corresponding experimental results are shown in Fig. 6 and Fig. 7.) This phenomenon results mainly from unmodeled dynamics (for example communication delays and the propeller dynamics), which were neglected in the controller design presented in Section III. (We refer to Section V-A for more details of the vehicle dynamics.)

A. Phase Comparator

The phase shift $\varphi(t)$ between the quadrocopter trajectory and the desired motion (1) is determined by multiplying the quadrocopter output separately with two different sinusoidal reference signals and integrating the product. Define the reference signals,

$$r_{cos}(t) = \cos(\omega_d t) \tag{14}$$

$$r_{sin}(t) = \sin(\omega_d t)\,, \tag{15}$$

where $r_{cos}(t)$ is the same frequency and phase as the desired motion (1). Under the assumption that the vehicle dynamics are linear, the response of the controlled quadrocopter system (derived in Section III) to the periodic reference signal $x_d(t)$, see (1), is a sinusoidal signal with the same frequency but possibly shifted phase and different amplitude,

$$x(t) = A\ \cos\left(\omega_d t + \varphi(t)\right). \tag{16}$$

Multiplying the signals (14) and (15) with the vehicle output (16) and using trigonometric identities lead to

$$q_{cos}(t) = x(t)r_{cos}(t) = \frac{A}{2}\big[\cos\varphi(t) + \cos\big(2\omega_d t + \varphi(t)\big)\big]$$

$$q_{sin}(t) = x(t)r_{sin}(t) = \frac{A}{2}\big[-\sin\varphi(t) + \sin\big(2\omega_d t + \varphi(t)\big)\big].$$

Integrating these signals over one period $T_d = 2\pi/\omega_d$ and assuming a constant phase shift during that time interval

$$\varphi(\tau) = \varphi_t = \text{constant}, \quad t - T_d \le \tau \le t, \tag{17}$$

results in

$$\eta_1(t) = \frac{1}{T_d}\int_{t-T_d}^{t} q_{cos}(t)\ dt = \frac{A}{2}\cos\varphi_t \tag{18}$$

$$\eta_2(t) = \frac{1}{T_d}\int_{t-T_d}^{t} q_{sin}(t)\ dt = -\frac{A}{2}\sin\varphi_t. \tag{19}$$

The value φ_t can be interpreted as the mean value of the phase shift $\varphi(t)$ during the last period, and when exciting a linear system with a periodic input, the phase shift is in fact constant (after a transient phase). Therefore, (17) is a valid assumption in steady state. Finally, the phase shift φ_t is obtained by

$$\varphi_t = -\arctan\left(\frac{\eta_2(t)}{\eta_1(t)}\right). \tag{20}$$

Note that in steady state, integration over several periods improves the robustness of the estimate of φ_t.

B. Phase Correction

The phase error φ_t is corrected by a feedback technique borrowed from PLL design [19]. The reference signal $x_d(t)$ in (1) is shifted by a correction term $e(t)$,

$$x_d^s(t) = A_d \cos\big(\omega_d t + e(t)\big), \tag{21}$$

which is defined as

$$e(t) = -k \int_0^t \varphi_t \, dt. \tag{22}$$

Similarly, the derivatives of $x_d(t)$ are shifted in phase by $e(t)$. Replacing the reference signal $x_d(t)$ and its derivatives in the controller equations (9) and (10) by the shifted values,

$$x_d^s(t), \ \dot{x}_d^s(t), \ \ddot{x}_d^s(t), \ \text{and} \ \dddot{x}_d^s(t), \tag{23}$$

produces a new input $u(t)$, cf. (13), which compensates for the phase error. With the feedback integrator term $e(t)$, precise and robust phase locking is achieved. Convergence behavior is controlled by tuning the gain factor k.

V. RESULTS

The developed synchronization scheme is demonstrated on small quadrocopters of about 30 cm diameter operated in the ETH Flying Machine Arena, a $10 \times 10 \times 10$ m indoor flight-test facility.

A. Experimental Setup

The setup is similar to [22]: An 8-camera Vicon motion capture system provides pose data for any vehicle in the space at 200Hz with a latency of about 25 ms. The localization data is sent to a PC, which runs the control algorithm, and which in turn sends commands to the quadrocopters, delivered with a latency between 30 to 60 ms. The flying vehicles are modified commercially available quadrocoptor described in [23]. Each vehicle accepts a collective thrust command and three angular rate commands at 50 Hz. An onboard 1 kHz feedback controller uses rate gyros to track the given commands. More details about this test environment may be found in [24] and [25].

For the experiments described below, two degrees of freedom (collective thrust and desired pitch rate) were controlled by the algorithm, while the other two degrees of freedom were handled by a linear controller described in [24]. The measurements $x(t)$ and $\theta(t)$ are provided directly by the Vicon system while velocity is obtained by approximately differentiating position data. The acceleration in the x-direction used in the tracking controller (10) is obtained from (2) and (3) by assuming the acceleration in the z-direction is negligible, and by using the measured pitch angle $\theta(t)$. In the referenced video, the music was pre-processed and the beat times were stored on the same PC that runs the control algorithm and sends the commands to the vehicle.

B. System Characteristics

To begin with, the control laws (5) and (13) without synchronization are applied to the real system, and the behavior of the resulting oscillating quadrocopter motion is studied. In this case the desired amplitude A_d and frequency ω_d, cf. (1), can be interpreted as the inputs to the oscillating quadrocopter, Fig. 2, whereas the actual quadrocopter trajectory is the output. Since (2)-(4) represent a simplified model of the actual vehicle dynamics that neglects aerodynamic effects, delays in the system, the onboard controller, etc., the control laws (5) and (13) do not achieve synchronized trajectory tracking.

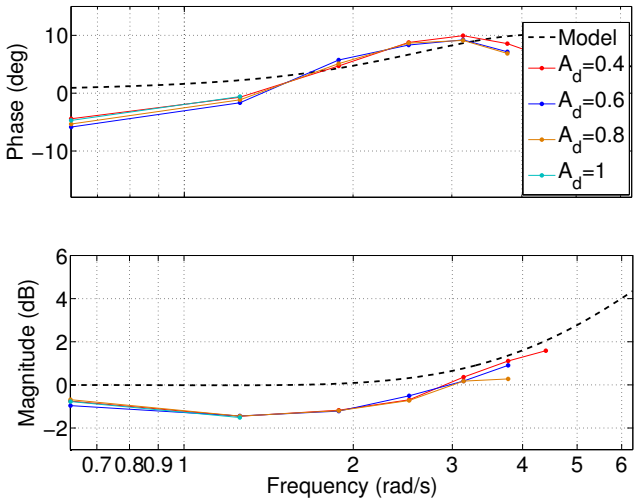

Fig. 5. Experimentally determined Bode diagram for the closed-loop oscillating quadrocopter system *without* phase correction. The dashed line shows the theoretically derived transfer function taking communication delays and propeller dynamics into account.

The transfer function of the closed-loop oscillating quadrocopter system (without the phase error compensation) was studied by varying the input frequency ω_d and amplitude A_d. The closed-loop response exhibits linear behavior with the frequency of the quadrocopter response equal to the input frequency. However, the motion is shifted in time, and depending on the frequency ω_d the amplitude is attenuated or amplified. Fig. 5 depicts the experimentally determined Bode plot for the frequency range of interest. Note that the Bode plot is independent of the amplitude A_d, as expected for a linear system.

The observed behavior can be explained by the non-idealities in the system. In particular, the delayed information exchange affects the overall system behavior. In addition, the dynamics of both the onboard controllers and the propeller motors are neglected in the system description (2)-(4). A quadrocopter model including realistic latency values (45 ms on sending commands and of 25 ms on receiving position data, see Section V-A) and motor dynamics modeled as first-order system,

$$\ddot{\theta}(t) = \frac{1}{T_\theta}\Big(u(t) - \dot{\theta}(t)\Big), \quad T_\theta = 25\,\text{ms}, \tag{24}$$

produces the dashed-line behavior in Fig. 5. For deriving the transfer function, the simple relation (8) between the x-position and the pitch rate input was used. This approximate

model accounts for the general trend in the experimental data.

Besides providing a better understanding of the closed-loop oscillating system, the experimental data can be incorporated as feedforward phase and amplitude compensation in the proposed synchronization scheme. The results are shown in the subsequent section, supporting the idea of an online identification of the values shown in Fig. 5 and constructing a look-up table. The look-up table might be continuously adapted to changing conditions in the environment (for example caused by worn out propellers). This allows a later implementation of highly agile maneuvers and fast changes between different motion primitives, cf. Fig. 8 and Fig. 9.

C. Synchronization Behavior

The proposed synchronization algorithm is applied to the quadrocopter and the resulting performance of the vehicle is analyzed. A reference signal with a frequency $\omega_d = 1.2\pi$ rad/s and an amplitude $A_d = 0.4$ m was chosen, cf. (1). Fig. 6 shows the quadrocopter response for three cases: (i) no phase correction, i.e. $k = 0$ in (22), (ii) phase error compensation with $k = 0.28$, and (iii) feedforward phase and amplitude correction based on the pre-determined values depicted in Fig. 3. After a short transient phase, the phase comparator (introduced in Section IV-A) outputs a constant phase error in the case of no phase correction, see Fig. 7. Perfect phase-locking is achieved when adding phase compensation through either a feedback or feedforward component. Note that the proposed phase comparator needs at least one full period to converge to the correct phase error. While the phase error between the reference trajectory and the actual quadrocopter response is hardly noticeable in Fig. 6, small phase errors are very visible and audible in actual experiments. In particular, humans expect zero vehicle velocity at beat times. Correspondingly, Fig. 7 plots the velocity of the quadrocopter at beat times, i.e. when the reference trajectory reaches its maximum or minimum value. The sign of every second velocity value is altered such that a constant positive phase delay is represented by positive velocity values, cf. Fig 7.

While assuming a constant reference frequency and amplitude in the theoretic derivations, frequency and amplitude changes are followed closely when compensating for phase and amplitude errors with a feedforward term, see Fig. 8 and Fig. 9. In fact, this better reflects the properties of real music, where beat intervals T_i, cf. Fig. 3, might vary over time. For changing frequencies, the reference is still given by a signal as depicted in Fig. 3, but corresponding to the next beat interval T_i a different frequency ω_d is used for each half period of the sinusoidal signal (1).

A 'dancing' choreography was performed on the song *Beat goes on* from Dj Ross Vs Dy. A video of the corresponding quadrocopter motion is available at `http://tinyurl.com/dancingquadro` and also accompanying this paper. Throughout all experiments $\delta_z = \delta_x = 1$, $\omega_z = 1.25$, and $\omega_x = 2.5$, although the results are not sensitive to changes in these parameters.

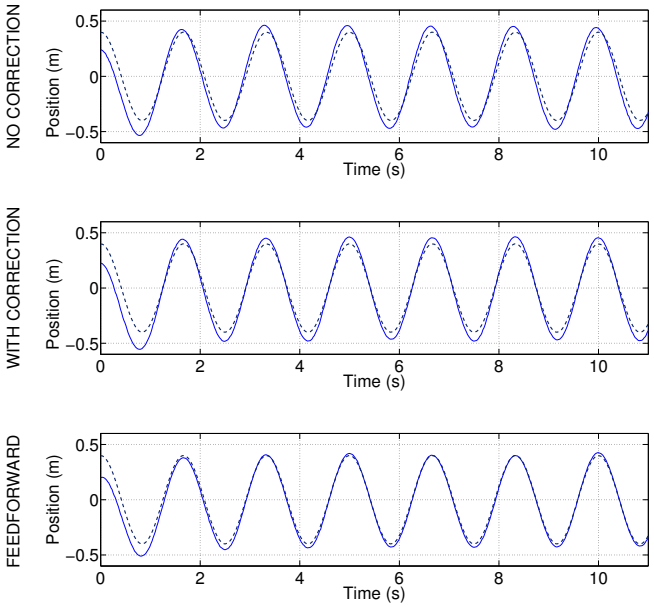

Fig. 6. Quadrocopter response to the dashed-lined input signal for the case of no phase correction (top), phase error compensation (center), and feedforward phase and amplitude correction (bottom). The solid blue line represents the vehicle trajectory.

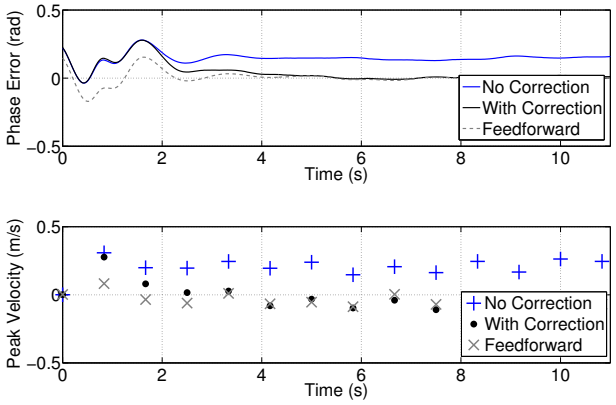

Fig. 7. Top: Phase error of the vehicle trajectories shown in Fig. 6 (detected by the phase comparator). Bottom: Vehicle velocity at the peak values of the corresponding reference signal (using altered signs, see text for details).

VI. CONCLUSIONS

This paper presents a control and synchronization scheme that enables flying vehicles to perform rhythmic movements. The proposed algorithm synchronizes the side-to-side motion of a quadrocopter with the beat from an arbitrary piece of music. A feedback scheme is used to adjust the phase of the oscillating quadrocopter to the music reference signal that was deduced from the music song in a pre-processing step. Based on prior experimental data, a feedforward component can be added that achieves fast adaptation to changes in the reference frequency or amplitude. The derived control and synchronization techniques generalize to any other periodic vehicle motion.

The presented results are a first step towards developing the algorithms capable of controlling multiple quadrocopters

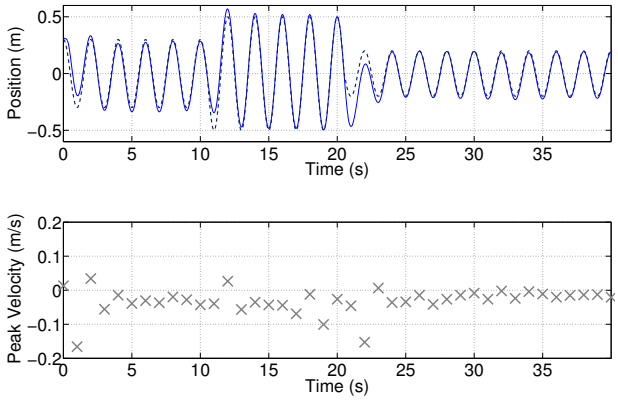

Fig. 8. Top: Quadrocopter response (solid blue line) to an input sequence with changing amplitudes (dashed line) using feedforward phase and amplitude correction. Bottom: Vehicle velocity at peak values of the corresponding reference signal (using altered signs).

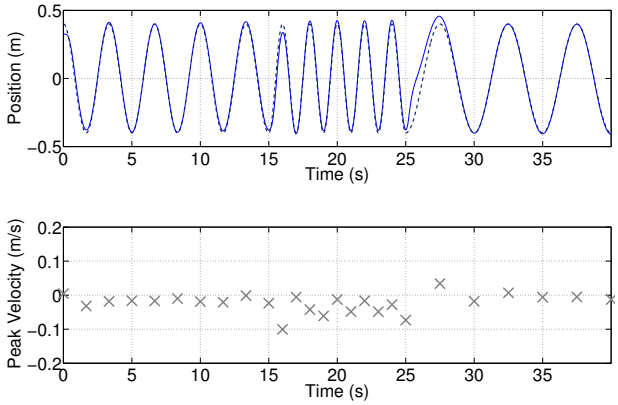

Fig. 9. Top: Quadrocopter response (solid blue line) to an input sequence with changing frequencies (dashed line) using feedforward phase and amplitude correction. Bottom: Vehicle velocity at peak values of the corresponding reference signal (using altered signs).

as they perform sophisticated aerobatic maneuvers timed to music. An aerobatic dance performance of a group of quadrocopters is envisioned in the Flying Machine Arena, where the music's features, like beat, volume, and melody, are reflected in the movement of the quadrocopters.

VII. ACKNOWLEDGEMENTS

This work would never have been possible without the prior work of Guillaume Ducard and Felix Althaus. Their contributions are gratefully acknowledged.

REFERENCES

[1] Y. Koren, "Cross-coupled biaxial computer control for manufacturing systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 102, no. 4, pp. 265–272, 1980.

[2] M. Tomizuka, J. S. Hu, T. C. Chiu, and T. Kamano, "Synchronization of two motion control axes under adaptive feedforward control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 114, no. 2, pp. 196–203, 1992.

[3] L. Ren, J. K. Mills, and D. Sun, "Trajectory tracking control for a 3-DOF planar parallel manipulator using the convex synchronized control method," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, pp. 613–623, 2008.

[4] D. Sun and J. K. Mills, "Adaptive synchronized control for coordination of multirobot assembly tasks," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 498–510, 2002.

[5] H.-T. Liu, J. Shan, and D. Sun, "Adaptive synchronization control of multiple spacecraft formation flying," *Journal of Dynamic Systems, Measurement, and Control*, vol. 129, no. 3, pp. 337–342, 2007.

[6] L. Feng, Y. Koren, and J. Borenstein, "Cross-coupling motion controller for mobile robots," *IEEE Control Systems Magazine*, vol. 13, no. 6, pp. 35–43, 1993.

[7] K. Murata, K. Nakadai, 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 scatting and singing," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2008, pp. 2459–2464.

[8] ——, "A robot singer with music recognition based on real-time beat tracking," in *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, 2008, pp. 199–204.

[9] K. Yoshii, K. Nakadai, T. Torii, Y. Hasegawa, H. Tsujino, K. Komatani, T. Ogata, and H. G. Okuno, "A biped robot that keeps steps in time with musical beats while listening to music with its own ears," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2007, pp. 1743–1750.

[10] C. A. A. Calderon, M. R. Elara, C. Zhou, L. Hu, and B. Iniya, "Neural oscillator for rhythmic motion control of biped robot," in *Proceedings of the International Conference on Signal Processing, Communications and Networking (ICSCN)*, 2008, pp. 450–453.

[11] K. Shin'ya and S. Schaal, "Synchronized robot drumming by neural oscillator," *Journal of the Robotics Society of Japan*, vol. 19, pp. 116–123, 2001.

[12] D. Pongas, A. Billard, and S. Schaal, "Rapid synchronization and accurate phase-locking of rhythmic motor primitives," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp. 2911–2916.

[13] C. Crick, M. Munz, and B. Scassellati, "Synchronization in social tasks: Robotic drumming," in *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication (ROMAN)*, 2006, pp. 97–102.

[14] G. Kim, Y. Wang, and H. Seo, "Motion control of a dancing character with music," in *Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science (ICIS)*, 2007, pp. 930–936.

[15] T. Kim, S. I. Park, and S. Y. Shin, "Rhythmic-motion synthesis based on motion-beat analysis," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 392–401, 2003.

[16] J.-J. Aucouturier, "Cheek to chip: Dancing robots and AI's future," *IEEE Intelligent Systems*, vol. 23, no. 2, pp. 74–84, 2008.

[17] M. P. Michalowski, S. Sabanovic, and H. Kozima, "A dancing robot for rhythmic social interaction," in *Proceedings of the ACM/IEEE international Conference on Human-Robot Interaction (HRI)*, 2007, pp. 89–96.

[18] K. Kosuge, T. Hayashi, Y. Hirata, and R. Tobiyama, "Dance partner robot – Ms DanceR," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 4, 2003, pp. 3459–3464.

[19] W. F. Egan, *Phase-Lock Basics*, 2nd ed. Wiley-Interscience, 2008.

[20] G.-C. Hsieh and J. C. Hung, "Phase-locked loop techniques. A survey," *IEEE Transactions on Industrial Electronics*, vol. 43, no. 6, pp. 609–615, 1996.

[21] K. Ogata, *Modern Control Engineering*, 4th ed. Prentice Hall, 2002.

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

[23] 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 *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2007, pp. 361–366.

[24] G. Ducard and R. D'Andrea, "Autonomous quadrotor flight using a vision system and accommodating frames misalignment," in *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, 2009, pp. 261–264.

[25] S. Lupashin, A. Schöllig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2010.