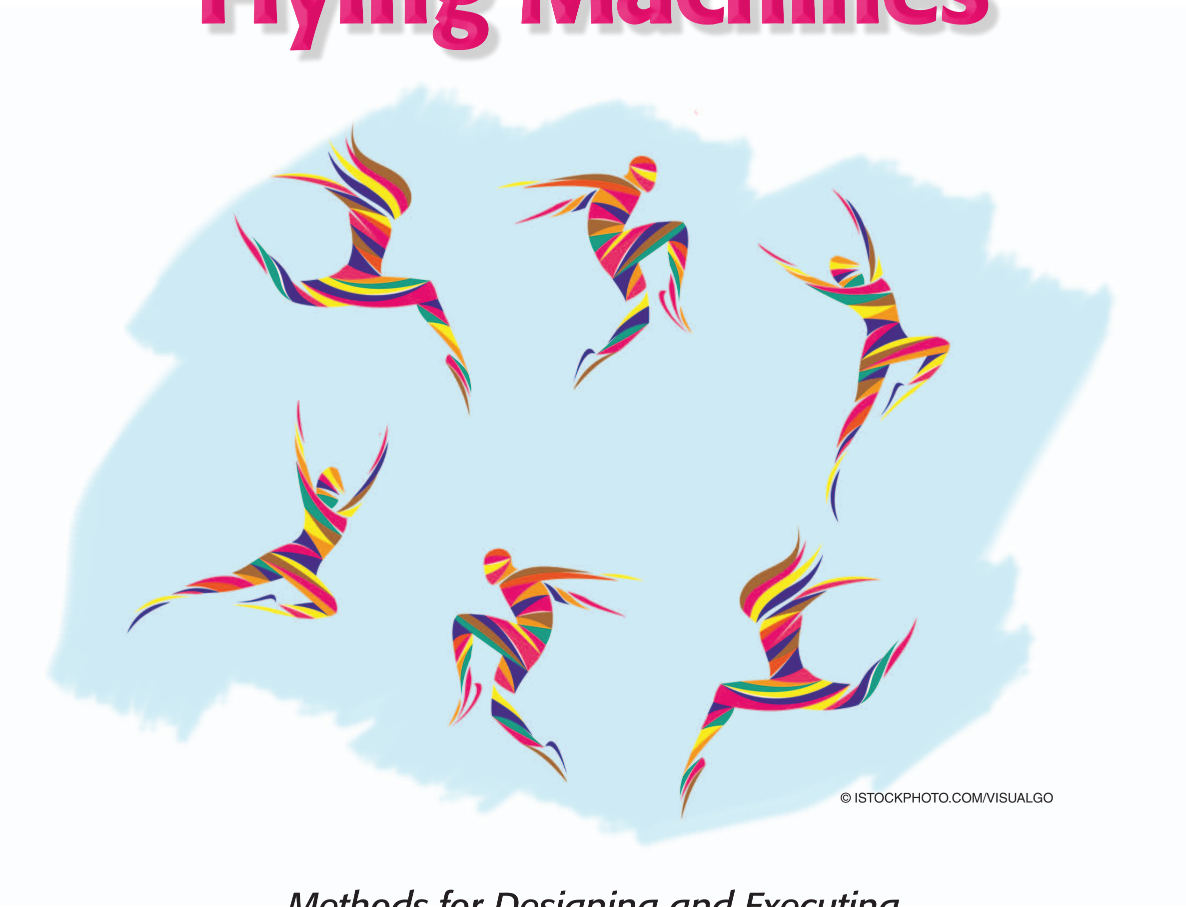
Dance of the Flying Machines



Methods for Designing and Executing an Aerial Dance Choreography

By Federico Augugliaro, Angela P. Schoellig, and Raffaello D'Andrea

Imagine a troupe of dancers flying together across a big open stage, their movement choreographed to the rhythm of the music. Their performance is both coordinated and skilled; the dancers are well rehearsed, and the choreography well suited to their abilities. They are no ordinary dancers, however, and this is not an ordinary stage. The performers are quadrocopters, and the stage is the ETH Zurich Flying Machine Arena, a state-of-the-art mobile testbed for aerial motion control research (Figure 1).

Digital Object Identifier 10.1109/MRA.2013.2275693
Date of publication: 30 October 2013

Quadrocopters are exceptionally agile vehicles, but it takes more than agility to successfully perform a coordinated flight choreography to music. From trajectory generation and synchronization to motion feasibility and collision-free flight, this article describes our efforts to create a robotic musical flight performance that is both viable and convincing.

Let us begin, however, with a discussion on the emergence of a new field of research that has grown out of the overlap between music and robotics. The workshop held at the 2010 IEEE/Robotics Society of Japan (RSJ) International Conference on Intelligent Robots and Systems (IROS) titled "The First IROS Workshop on Robots and Musical Expressions" [1] bridged the worlds of robotics, music information

1070-9932/13/$31.00©2013IEEE

retrieval, and the cognitive sciences. The workshop was evidence that the overlap of these once-distinct fields is becoming a prominent area of research for musicians and roboticists alike and has resulted in both novel human–robot interplay and unique musical experiences. Research on musical robots from the past decade can, for the most part, be characterized by the background of the researchers. Musicians, composers, and music technologists have generally sought to develop innovative forms of musical expression and sound production that overcome the limitations of human music generation and traditional musical instruments. Recently, this group has been behind the development of perceptual music robots, some of which facilitate music collaboration with human musicians [2], [3].

Roboticists, on the other hand, have sought to use music to establish a new dimension of human–robot communication, interaction, and collaboration. Research from this second community has been largely focused on the development of humanoid robots capable of imitating human musical behavior: robots that perform human-inspired rhythmic motions, such as dancing, drumming, stepping, and singing along to music [4]–[7]. Common to both communities is the desire to create an audio-visual performance with both aesthetic and entertainment values. So far, most of the research on dancing robots has focused on humanoid robots. Only recently have two artistic performances featured quadrocopters on stage with human actors/dancers [8], [9]; these focused on the interplay between humans and machines and had skilled human operators controlling the quadrocopters.

This article describes how to generate a choreography for flying vehicles and considers the composition and autonomous execution of multivehicle rhythmic flight. The music is analyzed ahead of time to obtain a description of the music's structure. The pre-extracted music beat is chosen as the basic unit of the music structure and serves as a reference for rhythmic movements. The trajectories are predesigned and parameterized, providing a flexible and customizable library of motion primitives. These basic elements are then concatenated and assigned to different music sections by the user, using the underlying music description to guide the choreography design. This process is done with the help of a customized software tool and is conceptually shown in Figure 2. To complement the design process, we also developed a numerical test that assesses the feasibility of desired trajectories. This article focuses on quadrocopters, but many components may be applied to other robots and flying vehicles.

In this article, we first explain the key methods and strategies used to generate an aerial choreography. We then highlight specific issues related to the use of quadrocopters and multiple flying vehicles for dance performances. This is followed by a tutorial-like explanation that guides the reader through the generation of a choreography for multiple vehicles. We conclude with a summary of our experimental results. A video featuring multiple quadrocopters dancing to music can be seen as a complement to this article in IEEE *Xplore*.

Dancing Quadrocopters

Quadrocopters are exceptionally agile aerial vehicles. Capable of vertical takeoff and landing, these vehicles can perform fast motions in the translational and rotational degrees of freedom. These characteristics, and the vehicles' inherent symmetric design, make quadrocopters a suitable platform for executing an *aerial dance*, where expressiveness in motion and temporal variety is the key. However, unlike human dancers or humanoid robots, quadrocopters do not have limbs: their expressiveness is based solely on their position and attitude in space over time. Consequently, finding motion patterns that are convincingly expressive to the human eye is not trivial.

Quadrocopters are exceptionally agile vehicles, but it takes more than agility to successfully perform a coordinated flight choreography to music.

Four fundamental choreographic elements: 1) space, 2) time, 3) weight, and 4) flow are commonly used by choreographers and dance teachers to build choreography with interest, dynamics, and aesthetic appeal [10], [11]. These parameters provide a framework for meaningful quadrocopter choreography and are described as follows: 1) *space* refers to the area the dancer is performing in and also relates to how the dancer moves through the area, as characterized by the direction and path of a movement, as well as its size, level, and shape; 2) *time* encompasses rhythm, tempo, duration, and phrasing of movements; 3) *weight* relates to the quality of movement, as some types of choreographies are soft and smooth whereas others are sharp and energetic; and 4) *flow* represents the organization of movement sequences into larger concepts: the combination and variation of movements using recurring elements, contrast, and repetition.

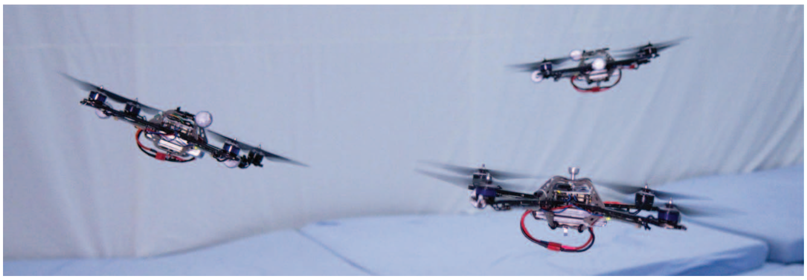

FEDERICO AUGUGLIARO

Figure 1. A dance performance of multiple quadrocopters in the Flying Machine Arena.

In the following sections, we explain how we arrive at a choreographed quadrocopter dance performance. We first present our motion library, from which we extract viable motions for the performance. We then introduce a basic description of music that serves as a reference when designing a new choreography. Finally, we explain the strategies we use for linking the motion to the music.

Motion Library

Motion primitives are the key building blocks of our choreographies: they represent well-defined motions of a quadrocopter during a finite period of time. The concatenation of motion primitives defines the vehicles' trajectories during the whole performance. The motion primitives are parameterized, categorized, and stored in the motion library to be later combined into a variety of choreographic designs.

Quadrocopters possess six degrees of freedom. However, since their dynamics are differentially flat, all states and inputs of the system can be expressed in terms of a set of outputs and their derivatives [12]. One of these sets consists of the three-dimensional (3-D) position of the vehicle and its heading, given by the yaw angle [13]. Therefore, the motion primitives considered in this article are described by their position trajectory in 3-D space and by a yaw trajectory profile, which can be independently specified. These terms fully specify the vehicle's position and attitude.

The trajectories are input to an underlying trajectory-following controller that computes the necessary inputs to the vehicle based on its current state. Adaptation techniques are used to synchronize the quadrocopter trajectory to the music beat and to achieve precise trajectory tracking (see the "Synchronization" section). The overall system setup is outlined in Figure 3.

Three different kinds of motions can be used to describe the motion trajectories in this article: 1) periodic motions, 2) acrobatic motions, and 3) transition motions. Periodic motions are strictly linked to the music beat, whereas acrobatic motions are thought to be aesthetic highlights. Finally, transition motions allow for smooth transitions between different motion primitives.

Periodic Motion Primitives

Periodic motion is a natural human response to a recurring beat: we often clap, sway, or tap our feet when we hear music. Thus, we want the flying vehicle to mimic this behavior. For our purpose, the trajectory of a periodic motion primitive is represented by a sum of sines and cosines

$$x_0 + \sum_{k=1}^{N} c_k \cos(k(\Omega t + \varphi)) + d_k \sin(k(\Omega t + \varphi)), \qquad (1)$$

where the frequency Ω is related to the music beat. The center point x_0, the amplitudes c_k and d_k (all vectors in $\mathbb{R}^3$), and the phase φ are design parameters. Trajectories with different frequency components are combined if N is chosen to be greater than 1.

The above formulation is mathematically redundant but allows us to easily define a multitude of periodic motion primitives using a set of intuitive design parameters. For example, a circle of radius r in the horizontal plane can be

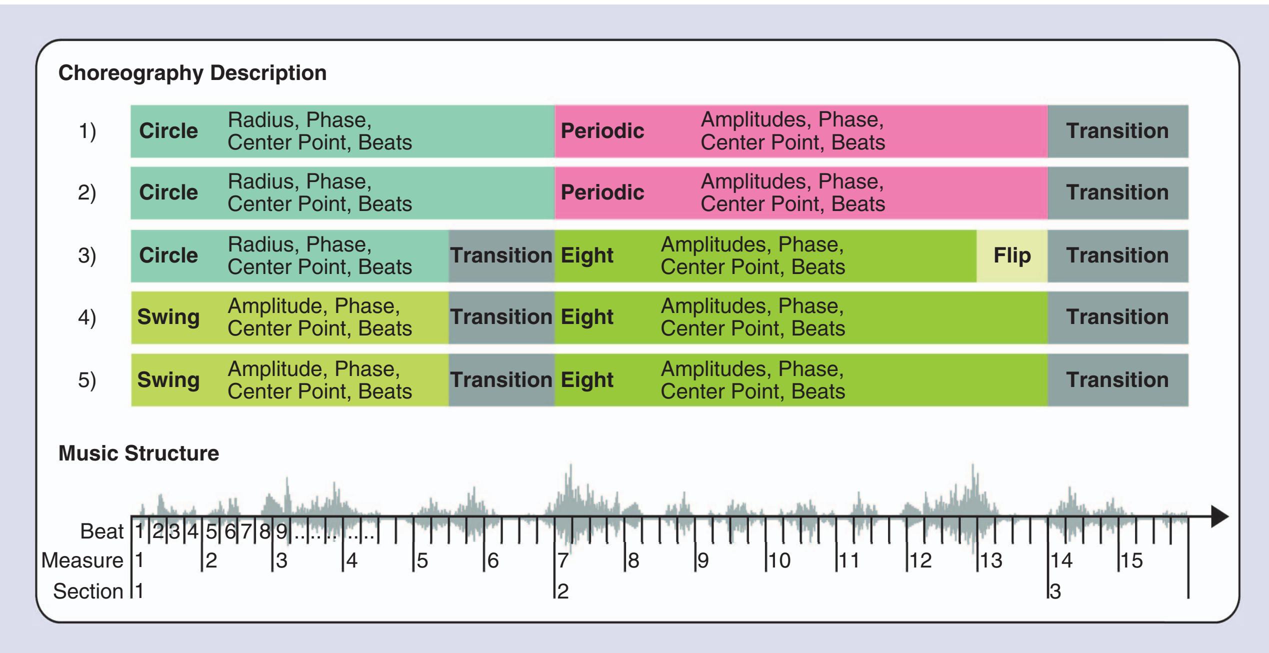


Figure 2. The choreography design for five vehicles. Motion primitives from the motion library are assigned to different sections of the music and follow the music's time structure. Parameters such as amplitude and center point are specified for each motion. Some periodic motions (such as the swing motion, the circle, and the figure eight) have their own specialized interface. A general framework allows the definition of any periodic motion. Periodic motions are synchronized to the music beat.

defined by choosing $c_1 = (r,0,0)$ and $d_1 = (0,r,0)$, whereas φ sets the starting position. Notice that these design parameters can be directly linked to the fundamental choreographic elements described above. The variables c_k, d_k, x_0, and φ refer to the space, and Ω refers to the time. The number N is related to the weight: the higher the value, the more energetic and sharp are the possible motions.

To speed up and ease the choreography design phase, some commonly used periodic motions (e.g., circle, swing, and figure eight) are included separately in the motion library with a convenient customized interface, as shown in Figure 2.

Acrobatic Motions

To increase the variety of our motions in the performance, we include choreographic highlights such as flips, loopings, or bang-bang-type transitions, some of which require cutting-edge research in areas such as aerial control and adaptation/learning [14]. These motions add aesthetic and technical varieties to the performance even though they are not strictly related to the music beat.

Transition Motions

A choreography is created by concatenating motion primitives from the motion library. However, to avoid jumps in the quadrocopter's attitude, the trajectory must be continuous up to the acceleration. Smooth transitions between consecutive motion primitives are therefore essential for a proper execution. Although assigning individual motion primitives to many vehicles without creating intersections is relatively simple (for example, placing them on the same circle by varying the phase parameter), transitioning between multiple motion primitives while avoiding collisions is more complicated. Therefore, we developed a method for generating collision-free trajectories for multiple vehicles from a set of initial states (specified by position, velocity, and acceleration) to given final states. The trajectories satisfy various constraints, including the physical limits of the vehicle, space boundaries, and a minimum distance between vehicles. The general idea of the transition algorithm is described in the section "Collision-Free Motions."

Music Description

Most music pieces have a well-defined time structure, and because beat times are a key characteristic of this structure, we use them as the basic unit of the choreography (Figure 2). Using publicly available beat extraction software, we analyze the music to determine beat locations. Periodic quadrocopter motions are then automatically synchronized to the music beat. Measure information follows from the music time signature. In future iterations, this simple time structure may be extended by considering additional features of music, such as its mood or its relative loudness (dynamic range). This will ease the choice of motion primitives (and their parameters) and will eventually lead to an automatic generation of music-driven trajectories, where the elements of the motion library are shaped and concatenated according to the music's features.

Roboticists, on the other hand, have sought to use music to establish a new dimension of human–robot communication, interaction, and collaboration.

Linking Motion and Music

Connecting vehicle motion to music is key to a rhythmic flight performance, as it is this link that transforms movement into dance. We face two main obstacles as we seek to link motions from the library with each section of music.

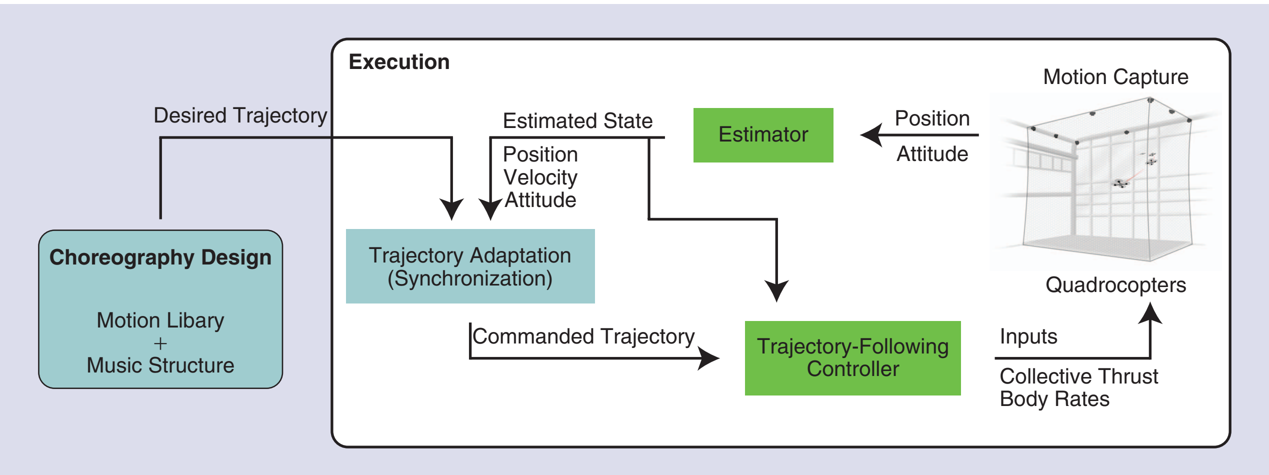


Figure 3. The high-level system architecture used to design and execute dance choreographies with quadrocopters. The key blocks (gray) are the choreography design stage and the trajectory adaptation techniques. The other blocks depict commonly used infrastructure of our testbed.

Periodic Motion Primitives and Music Beat

To achieve a recognizable dance behavior, periodic motion primitives are linked to the music beat. The frequency Ω from (1) is directly given by the music beat or a multiple of it. A simple example is shown in Figure 4: the quadrocopter performs a planar swing motion where, at beat times, the vehicle reaches the outermost points of the trajectory, either on the left or right.

The concatenation of motion primitives defines the vehicles' trajectories during the whole performance.

However, experimental results have shown that, at steady state, the quadrocopter response to a periodic sinusoidal input is a sinusoidal motion of the same frequency with a constant error in amplitude and phase, resulting in asynchrony, as shown in Figure 5(a). To achieve precise temporal and spatial tracking, we adapt the parameters of the commanded trajectory sent to the trajectory-following controller with correction terms (Figure 3). A more detailed explanation of this approach is provided in the "Synchronization" section.

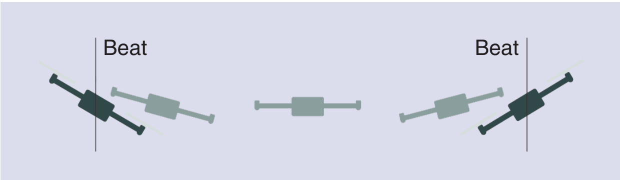


Figure 4. A periodic swing motion. The goal is to have the music beats occur at the outermost points of the movement.

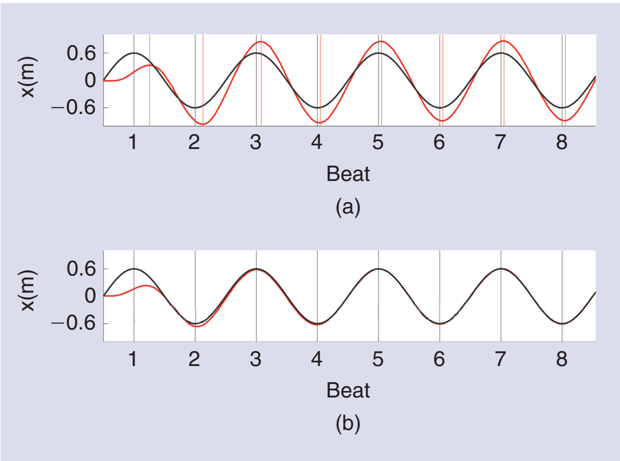


Figure 5. The unidimensional swing motion with a frequency of 0.62 Hz. The quadrocopter starts at zero velocity. (a) The quadrocopter response (in red) to a desired oscillation in the *x*-direction (in black). The vertical lines indicate the peak values and show the phase error. Amplitude errors in the quadrocopter response are also observed. (b) The quadrocopter trajectory when applying the adaptation methods described in the "Synchronization" section. Phase and amplitude errors are compensated for.

Design Parameters

Any of the library's motion primitives can be readily shaped to better fit the underlying music by adapting the motion parameters, e.g., x_0, c_k, d_k, and φ for periodic motion primitives (1). However, the resulting trajectories must satisfy vehicle feasibility constraints, e.g., the propeller forces required to fly a trajectory must not exceed the vehicle's motor capabilities. The choice of the design parameters is thus not completely free. We have developed various methods for assessing the feasibility of a trajectory, the underlying principles of which are summarized in the "Motion Feasibility" section.

Some Insights

This section presents algorithms and methods that are fundamental to the generation of a multiple-quadrocopters dance performance. Interested readers can find detailed explanations in [15], [16], and [18]–[20].

Synchronization

As introduced previously, the quadrocopter's response to a desired periodic sinusoidal input exhibits a constant phase shift and amplitude error. As a result, the quadrocopter's motion is perceived as out of tempo with the music.

The phase lag of the quadrocopter's response was first identified and corrected for in [15] for the unidirectional case of a swing motion (Figure 4). Figure 5(a) shows the actual quadrocopter trajectory (in red) and the desired one (in black). The amplitude error of the quadrocopter response is obvious; the phase error between the desired trajectory and the actual quadrocopter response seems negligible. However, small phase errors are perceivable to the human eye in the actual experiments. Depending on the motion frequency, we observe delays of up to 115 ms [16]. Cognitive science literature states that audio–video asynchrony can be perceived when the audio leads the video by as little as 29 ms [17].

Our strategy for coping with the aforementioned constant phase shift and amplitude error is to adjust the motion parameters of the trajectory commanded to the underlying trajectory-following controller. This means, for example, that if the amplitude of the quadrocopter motion is larger than the desired one, we reduce the commanded amplitude. Similarly, if the vehicle motion is lagging, we shift the commanded trajectory by increasing the phase. The proposed adaptation strategy entails feedforward correction terms, which are obtained from an offline identification and a method for correcting for residual errors online while performing the motion. With preidentified feedforward correction terms, the tracking errors during the initial transient phase become negligible; this is especially the case if the vehicle's initial state corresponds to the desired one. Thus, the vehicle tracking performance is convincing to the human observer. Figure 5(b) shows the quadrocopter trajectory when using this method: the periodic trajectory is tracked correctly. This approach is described in detail in [16].

Motion Feasibility

Not every desired trajectory is feasible. For quadrocopters, motor capabilities (minimum and maximum thrust) and sensor limitations (maximum rotational rates) constrain the range of feasible motion primitives. In [18], feasible parameter sets for parameterized motion primitives a priori to flight experiments are identified. Figure 6 shows the feasible parameter sets (radius and frequency) for a planar circle. This provides guidelines for the choreography generation step and will also facilitate a future automatic generation of motion primitives by providing the algorithm with a clear set of rules regarding the feasibility of parameterized motion primitives.

These results were recently extended to numerically assess the feasibility of arbitrary quadrocopter trajectories [19]. Before actual flight, the feasibility of a created motion sequence is evaluated. Using a first-principles model of the vehicle dynamics, a numerical method relates a desired trajectory to the required vehicle rotational rates and propeller forces. The obtained values must meet vehicle constraints caused by actuator and sensor limitations.

Collision-Free Motions

In the context of multivehicle flight, we developed an algorithm that moves a group of vehicles from one configuration to another one without collisions [20]. These trajectories are used to generate smooth transitions between consecutive motion primitives before actual flight and for safely coordinating multiple vehicles during takeoff, landing, and in unexpected situations. This method returns collision-free trajectories for multiple vehicles within a few seconds.

The goal is to generate trajectories for transitioning from a set of initial states (position, velocity, and acceleration of each vehicle) to a set of final states in a given time, while satisfying simple dynamic constraints and guaranteeing a minimal distance between vehicles, as shown in Figure 7 for a two-dimensional scenario. We formulate our multivehicle trajectory planning problem as a constrained optimization problem. We use an approach called *sequential convex programming* [21], which recursively solves the problem by approximating the nonconvex constraints (minimum distance between vehicles) by convex ones. A video demonstrating the algorithm can found online at http://youtu.be/wwK7WvvUvlI.

Generating a Choreography

In this section, we show how a choreography featuring multiple quadrocopters can be created using our existing setup. A video of the resulting flight performance can be found on the project's Web site [22].

Music Analysis

Our first step is to choose a piece of music and process its audio file with beat extraction software. We use Beatroot [23] for this purpose. The beat times are extracted automatically and serve as the basic unit of the music time structure. Sections are identified manually

Adaptation techniques are used to synchronize the quadrocopter trajectory to the music beat and to achieve precise trajectory tracking.

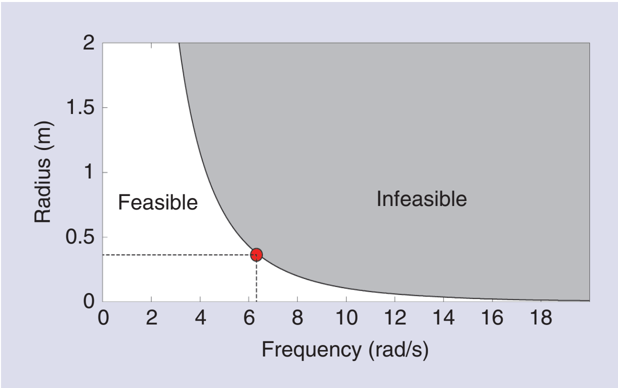


Figure 6. Feasible parameter sets for a circular motion primitive in the horizontal plane. Parameter combinations of the dark gray region are infeasible due to motor limitations. For example, if we want to perform a circle every second (6.28 rad/s), its maximum radius cannot exceed 0.38 m, as shown by the red circle. More details can be found in [18].

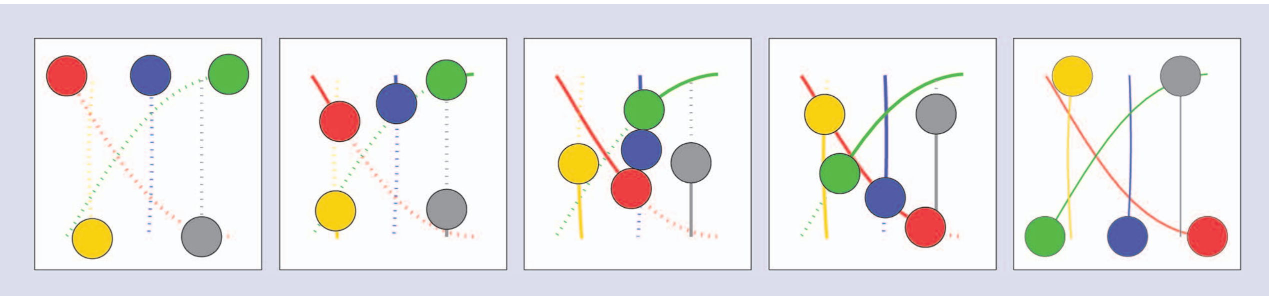


Figure 7. The collision-free transition motion demonstrated for a two-dimensional scenario. The circles show the positions of the vehicles (and the required minimum distance) in the *xy*-plane at subsequent times from left to right. The lines indicate the resulting trajectories.

by listening to the music. We use this description later when designing the choreography. We store the information in a text file, where $B1$ represents the first beat of the music piece and $S1$ the beginning of section number one. Each tag is time stamped, i.e., $B4$ may refer to $t = 2$ s of the music piece.

Choreography Design

We developed a software tool and an intuitive scripting language to ease the choreography design. In this section, we demonstrate its use by creating a choreography for four vehicles and present excerpts from the text file containing the scripted choreography.

The beat times are extracted automatically and serve as the basic unit of the music time structure.

First, we assign a swing motion to all quadrocopters, from beat 1 to the beginning of section 2 of the music piece. This motion uses the periodic motion framework described in (1), but has a specialized interface (Figure 2). We assign this motion to vehicle 1 with

```
B1-S2, 1, SWING, amp=0.6, center=
 [-1.5;-1;3], beats=4;.
```

The parameter `center` refers to x_0, `amp` indicates the amplitude of the motion. We want the vehicle to complete a full period of the motion every four beats (`beats=4`).

The swing motion is followed by a bounce motion for all vehicles. This motion is created by exploiting the general framework for periodic motion primitives. The matrices `C` and `D` contain the vectors c_k and d_k, `phi` refers to φ

```
S3-S4, 1, PERIODIC, C=[1 0;1 0;0 -0.5],
 D=[0 0;0 0;0 0], center=[1;1;5], phi=0,
 beats=8.
```

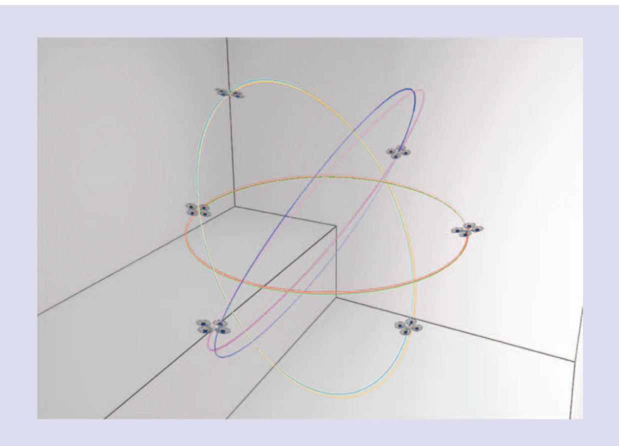


Figure 8. The Flying Machine Arena simulation environment, showing part of a choreography. The lines indicate the flown trajectories and help the user to visually assess the choreography.

Note, however, that the swing motion and the bounce motion take place at different locations in space. To smoothly concatenate the two motion primitives and, at the same time, ensure collision-free trajectories for every vehicle, we use the algorithm illustrated in the "Collision-Free Motions" section. A collision-free transition is planned from the final states of the previous motion primitive (swing) to the initial states of the next one (bounce), guaranteeing a smooth acceleration profile and safe trajectories

```
S2-S3, 1, TRANSITION;
S2-S3, 2, TRANSITION;
S2-S3, 3, TRANSITION;
S2-S3, 4, TRANSITION;.
```

The same algorithm is also used to transition to the next motion of our example choreography, a circle. We take advantage of the parameters `phi` to place multiple vehicles on the same circle, by varying the phase φ only

```
S5-S6, 1, CIRCLE, radius=2.5, center=
       [0;0;3], phi=0, beats=8;
S5-S6, 2, CIRCLE, radius=2.5, center=
       [0;0;3], phi=1.57, beats=8;
S5-S6, 3, CIRCLE, radius=2.5, center=
       [0;0;3], phi=3.14, beats=8;
S5-S6, 4, CIRCLE, radius=2.5, center=
       [0;0;3], phi=-1.57, beats=8;.
```

Finally, we transition to the final motion, a flip, which is taken from the *acrobatic motion* primitives set. The total vertical displacement for this motion is indicated by

```
S8-B70, 1, FLIP, h=3;.
```

The multimedia accompanying this article in IEEE *Xplore* shows a video of the actual flight performance.

Assess the Feasibility

Numerical Test

Once the choreography has been designed, the feasibility of the trajectories must be assessed. Thus, we check numerically whether the desired trajectories satisfy motor and sensor constraints (see the "Motion Feasibility" section). We also verify that the trajectories are inside our flyable space and that a minimum distance between quadrocopters is ensured throughout the performance to prevent collisions.

Simulation

In addition to checking the desired trajectories for feasibility, we also simulate the flight behavior with our standard simulator. This encompasses additional dynamic effects (e.g., motor dynamics) and various noise sources, and considers the dynamic response of the trajectory-following controller. The simulator also provides a 3-D environment for evaluating the choreography, as shown in Figure 8.

MARKUS HEHN

Figure 9. The ETH Zurich Flying Machine Arena at the Hannover Messe in Germany, the world's largest industrial fair (April 2012).

Fly

The very same infrastructure, software, and network communication is then used to actually fly the designed choreography in the ETH Zurich Flying Machine Arena. The straightforward transition from simulation to actual experiments is enabled by the software architecture of the testbed and by the accuracy of the simulation.

Performances in the Flying Machine Arena

Experimental Testbed

We demonstrate our algorithms on small custom quadrocopters operated in the ETH Zurich Flying Machine Arena, a $10 \times 10 \times 10$ m^3 mobile testbed for quadrocopter research. The system features a motion capture system that provides precise vehicle position and attitude measurements. The localization data is sent to a personal computer, which runs control algorithms and sends commands to the quadrocopters at $\sim$50 Hz. A suite of software tools, algorithms, and components (such as the underlying trajectory-following controller) support the testbed, allowing for a rapid transition of algorithms from simulation to the real vehicles. More details on the testbed can be found in [14] and at www.FlyingMachineArena.org.

Choreographies

Since the start of the project, several choreographies have been designed and regularly demonstrated. The following list presents the choreographies that are featured in the Flying Machine Arena with the song name, the singer or composer, the number of quadrocopters, and their respective design year:

- *Please Don't Stop the Music,* Rihanna, one quadrocopter, 2009
- *Pirates of the Caribbean,* Hans Zimmer, two quadrocopters, 2009
- *Rise Up,* Yves Larock, three quadrocopters, 2010
- *From the Clouds,* Jack Johnson, four quadrocopters, 2011
- *Armageddon,* Prism, five quadrocopters, 2011
- *Dance of the Flying Machines,* Victor Hugo Fumagalli, six quadrocopters, 2013.

These choreographies have been performed not only at ETH Zurich, where we conduct our research, but also at exhibitions such as Google I/O (June 2012) and the Hannover Messe (April 2012) (Figure 9). A video showing six quadrocopters dancing to *Dance of the Flying Machines* demonstrates the system capabilities and can be seen as a complement to this article in IEEE *Xplore*.

Implementation

The resulting performance is completely preprogrammed. However, to allow for a robust and reliable execution, adaptation schemes are used at several points. Because of the collision-free method presented in the "Collision-Free Motions" section, takeoff and landing with multiple vehicles can be executed from arbitrary vehicle positions. When executing a performance, the quadrocopters are put into the flying space at random locations, and the algorithm plans collision-free trajectories to the starting points of the choreography within seconds. Furthermore, as explained in the "Synchronization" section, residual phase and amplitude errors in the quadrocopter response are compensated for during the performance by adapting the commanded trajectory online. The online adaptation allows us to synchronize the motion of vehicles with slightly different dynamic properties (e.g., shifted center of mass and degraded propellers).

Conclusions

This article demonstrated the design and execution of dance performances with multiple quadrocopters. It provided an overview of the various components we use to design choreographies that are timed to music. We proposed a trajectory design based on the concatenation of different motion primitives, and introduced transition motions for smooth concatenations and for the collision-free coordination of multiple vehicles. Methods for precisely synchronizing the quadrocopter motion to the music beat, and for assessing the feasibility of the desired trajectories were also presented. Finally, the "Generating a Choreography" section guided the reader through the generation of an actual choreography. This article includes a video in IEEE *Xplore* featuring multiple quadrocopters dancing to music.

It is our hope that these methods and tools will facilitate the development of a (semi)automatic generation of choreographies, where motion primitives are chosen automatically to reflect the music's character [24]. The automatic choice of appropriate motion primitives parameters is still open: they will be selected according to a richer description of the music from a set of feasible parameters given by our feasibility assessment tools. Additional future steps include the expansion of the motion library and the automatic incorporation of audience feedback during the choreography design phase.

Acknowledgments

We would like to acknowledge the contributions of the current and former Flying Machine Arena team members, in particular, Markus Hehn, Sergei Lupashin, Mark W. Mueller, and Michael Sherback. We also would like to thank Marc-Andre Corzillius, Carolina Flores, Hans Ulrich Honegger, and Igor Thommen for the technical support and Philipp Reist and Hallie Siegel for their suggestions on the manuscript. This research was supported in part by the Swiss National Science Foundation.

References

[1] The First IROS 2010 Workshop on Robots and Musical Expressions. (2010). *Proc. IROS Workshop Robots Musical Expressions* [Online]. Available: http://winnie.kuis.kyoto-u.ac.jp/RMEWS/proc/

[2] G. Hoffman and G. Weinberg, "Interactive improvisation with a robotic marimba player," *Auton. Robots*, vol. 31, nos. 2–3, pp. 133–153, 2011.

[3] A. Lim, T. Mizumoto, T. Ogata, and H. G. Okuno, "A musical robot that synchronizes with a coplayer using non-verbal cues," *Adv. Robot.*, vol. 26, nos. 3–4, pp. 363–381, 2012.

[4] D. Grunberg, R. Ellenberg, I. H. Kim, J. H. Oh, P. Y. Oh, and Y. E. Kim, "Development of an autonomous dancing robot," *Int. J. Hybrid Inform. Technol.*, vol. 3, no. 2, pp. 33–44, 2010.

[5] J. Oliveira, L. Naveda, F. Gouyon, L. Reis, P. Sousa, and M. Leman, "A parameterizable spatiotemporal representation of popular dance styles for humanoid dancing characters," *EURASIP J. Audio, Speech, Music Processing*, vol. 18, pp. 1–14, June 2012.

[6] 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 *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Sept. 2008, pp. 2459–2464.

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

[8] R. Murphy, D. Shell, A. Guerin, B. Duncan, B. Fine, K. Pratt, and T. Zourntos, "A Midsummer Night's Dream (with flying robots)," *Auton. Robots*, vol. 30, no. 2, pp. 143–156, 2011.

[9] P. Catton. (2011, July). Culture city online: Dances with robots [Online]. Available: http://blogs.wsj.com/metropolis/2011/07/15/culture-city-online-dances-with-robots/

[10] J. Newlove and J. Dalby, *Laban for All*. London: Nick Hern Books, 2003.

[11] A. LaViers and M. Egerstedt, "Style based robotic motion," in *Proc. American Control Conf.*, June 2012, pp. 4327–4332.

[12] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[13] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robotics Automation*, May 2011, pp. 2520–2525.

[14] S. Lupashin and R. D'Andrea, "Adaptive fast open-loop maneuvers for quadrocopters," *Auton. Robots*, vol. 33, nos. 1–2, pp. 89–102, 2012.

[15] A. P. Schoellig, F. Augugliaro, S. Lupashin, and R. D'Andrea, "Synchronizing the motion of a quadrocopter to music," in *Proc. IEEE Int. Conf. Robotics Automation*, May 2010, pp. 3355–3360.

[16] A. P. Schoellig, C. Wiltsche, and R. D'Andrea, "Feed-forward parameter identification for precise periodic quadrocopter motions," in *Proc. American Control Conf.*, June 2012, pp. 4313–4318.

[17] S. van de Par and A. Kohlrausch, "Sensitivity to auditory-visual asynchrony and to jitter in auditory-visual timing," in *Proc. SPIE 3959, Human Vision Electronic Imaging V*, June 2000, pp. 234–242.

[18] A. P. Schoellig, M. Hehn, S. Lupashin, and R. D'Andrea, "Feasibility of motion primitives for choreographed quadrocopter flight," in *Proc. American Control Conf.*, June 2011, pp. 3843–3849.

[19] F. Augugliaro, "Dancing quadrocopters—Trajectory generation, feasibility and user interface," M.S. thesis, Inst. Dyn. Syst. Control, ETH Zurich, Zurich, Switzerland, 2011.

[20] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Systems*, Oct. 2012, pp. 1917–1922.

[21] S. Boyd. (2008). Sequential convex programming (Lecture Notes), Stanford Univ., Stanford, CA. [Online]. Available: http://www.stanford.edu/class/ee364b/lectures/seq_slides.pdf

[22] (2013). Music in Motion. Inst. Dyn. Syst. Control, ETH Zurich, Zurich, Switzerland. [Online]. Available: http://www.idsc.ethz.ch/Research_DAndrea/Music_In_Motion

[23] S. Dixon, "Evaluation of the audio beat tracking system Beatroot," *J. New Music Res.*, vol. 36, no. 1, pp. 39–50, 2007.

[24] G. Xia, J. Tay, R. Dannenberg, and M. Veloso, "Autonomous robot dancing driven by beats and emotions of music," in *Proc. 11th Int. Conf. Autonomous Agents Multiagent Systems*, June 2012, pp. 205–212.

Federico Augugliaro, Institute for Dynamic Systems and Control, Department of Mechanical and Process Engineering, ETH Zurich, Switzerland. E-mail: faugugliaro@ethz.ch.

Angela P. Schoellig, Institute for Aerospace Studies, University of Toronto, Canada. E-mail: schoellig@utias.utoronto.ca.

Raffaello D'Andrea, Institute for Dynamic Systems and Control, Department of Mechanical and Process Engineering, ETH Zurich, Switzerland. E-mail: rdandrea@ethz.ch.