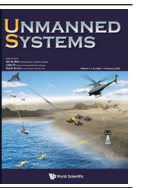
Unmanned Systems, Vol. 6, No. 1 (2018) 1–13
© World Scientific Publishing Company
DOI: 10.1142/S2301385018500012

UNMANNED SYSTEMS
http://www.worldscientific.com/worldscinet/us



Trajectory Planning and Control of a Quadrotor Choreography for Real-Time Artist-in-the-Loop Performances

Michael El-Jiz, Luis Rodrigues*

Department of Electrical and Computer Engineering, Concordia University, 1515 St. Catherine Street, EV5.139, Montreal, QC, Canada

This paper proposes a systematic methodology for the guidance, control, and navigation of a quadrotor to perform a choreographed dance in real-time as a function of, and interacting with, the music performed by an artist-in-the-loop. This methodology allows for a real-time interaction with improvized music by an artist based on the pitch of the acoustic signal being played without prior knowledge of the music. The four main components of a human choreography (namely, the notions of space, shape, time and structure) are analyzed and mathematically formulated for a robotic performance. A new approach for mapping music features to trajectory parameters is proposed, as well as the design of a trajectory shaping filter based on two coefficients that are set in real-time by an artist through a MIDI foot-pedal board. The proposed approach maps motion parameters and the music to trajectory motifs that are then switched in harmony with the chord structure. The overall system is validated in a hardware-in-the-loop simulation where the hardware will consist of the musical instrument and the foot pedals. In the simulation, the trajectory generator system is inverted to generate a sequence of music pitches from the actual trajectory of the quadrotor. The music generated by the quadrotor is then played back to the musician allowing for real-time interaction. The simulation results show that the proposed methodology yields an effective performance for a quadrotor choreography based on the real-time interaction with a musician. The proposed system was successfully used by an artist as can be seen in a video link to the work described in this paper and listed in the conclusions.

Keywords: Quadrotor choreography; trajectory planning; real-time control; artist-in-the-loop.

US

1. Introduction

Uninhabited Air Vehicles (UAVs) are flying vehicles without a human pilot aboard. Their current and planned future tasks include, but are not limited to, traffic surveillance, emergency response, forest fire monitoring, building inspections, and package delivery [1–3]. This paper will focus on fixed-rotor quadrotors, which are a sub-category of UAVs. The quadrotor is controlled by varying the thrust generated by its four propellers. Traditionally, the navigation parameters of a UAV are the heading angle, the altitude (or the height), and the speed. These parameters are typically chosen to achieve a given mission for the UAV and are fed to a control loop that changes the direction of the total thrust with respect to the world frame and thus actively changes the orientation of the quadrotor. Given the desired trajectories for heading angle, altitude and speed, different control algorithms can be applied to track these trajectories such as Proportional-Derivative (PD) controllers [4], integral backstepping [5], and Linear Quadratic Regulator (LQR) [6]. However, music features of improvised music have never been mapped to quadrotor parameters such as desired heading, height and speed so that a choreographic dance can be performed by UAVs.

A choreography is the sequence of steps and movements in a dance, usually performed by human dancers. By extension, a robotic choreography is the artistic motion of a robot [7]. An artistic robotic performance allows an artist to express his or her idea through the use of a robot. The artist controlling the robot is usually "twice removed" from the performance (artist to computer, computer to robot) and therefore the conveyability of emotions is reduced in a

Received 1 October 2016; Revised 7 September 2017; Accepted 8 September 2017; Published 3 November 2017. This paper was recommended for publication in its revised form by editorial board member, Youmin Zhang.
Email Address: *luisrod@encs.concordia.ca

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

traditional sense, while new and different freedoms of expression are granted to the artist [7]. Although robotic choreography has been studied at least for 20 years, the integration of flying UAVs in an artistic performance with a choreography has been a recent topic of research over the last six years. Previous work on robotic choreography has relied mainly on robotic reactions to an *a priori* selected music piece determined by features such as the beat or time signature [8–10]. By contrast, this work will focus on real-time response to the pitch of an improvized music piece. The pitch of a sound is associated with its fundamental frequency and is not a physical property, but rather a subjective psycho-acoustical attribute of the sound. Contrary to popular misconception, the pitch is not always the frequency component of the sound with the highest amplitude. All the different frequency components are integer multiples of the fundamental frequency [11]. The pitch can be extracted through frequency-based approaches [12–16] or time-based approaches [17–22]. Different pitch detection algorithms were compared in [23, 24].

Besides the pitch of the note being played, other features such as the rhythm or the beat of the music are also important. According to [25], rhythm is "the aspect of music concerned with the organization [of sounds and silences. . .] in time". Several beat-detecting and beat-tracking algorithms exist in the literature, such as for example the ones exposed in [26–30]. The beat of the music has been the main musical feature used in the pioneering work of [8], which makes a quadrotor sway to the beat of a musical piece. The problem addressed in [8] is not one of real-time mapping but rather a control problem of a quadrotor following a time-dependent periodic trajectory where time is associated to the beat of a previously selected piece of music. This limits the musical interaction and does not allow for musical improvization by a human player in real-time. Recent extensions to this work, presented in [9,10] describe a quadrotor choreography based on the beat of the music. The trajectories are generated through a Fourier series, and the coefficients are manipulated to create the desired periodic trajectories and the desired periodic choreography effects in the presence of input constraints and using collision avoidance algorithms. Although the work in [9, 10] has been quite pioneering in the use of quadrotors in choreographies, where several performances include pre-programmed quadrotor movements, it does not allow for human input and human-machine (or artist-robot) interaction. Other related work on quadrotor performances can be found in [31–34]. For quadrotor performances with several vehicles, collision avoidance techniques must be employed. A survey on such techniques for UAVs can be found in [35] and references therein.

The main contribution of this paper is the proposition of a systematic methodology for the control of a quadrotor robot based on acoustic musical signals in a real-time interactive artist-in-the-loop performance. The goal of the proposed approach is to give an artist the tools to control, choreograph, and interact with a quadrotor performance in real-time. To the best of the authors' knowledge, the robotic choreography problem has never been approached for a real-time interaction with a musician based on the pitch of the acoustic signal and without prior knowledge of the music. More specifically, the following are the detailed contributions of this paper:

(1) A novel approach for mapping music features to trajectory parameters is proposed together with the design of a trajectory shaping filter based on two coefficients that are set in real-time by an artist. The two coefficients are inspired by a mathematical description of acoustic signals. This is a key aspect that differentiates the proposed work from previous work and allows an artist to interact in real-time with the robotic choreography through the use of foot pedals.
(2) The four main components of a human choreography (namely, the notions of space, shape, time and structure) are analyzed and mathematically formulated for a robotic performance as a switched dynamical system. Although such components have been discussed before in the literature referenced in the introduction, the mathematical formulation of a robotic choreography as a switched system based on chord changes and pitch of the melody has not been considered before to the best of our knowledge. Furthermore, the shape of the dance is modeled in this paper by a shape filter whose parameters can be changed by the artist in real-time. This has also not been done in previous work. Finally, in contrast to the literature, the time parameter of the choreography is mathematically modeled by the vector field of the dynamics instead of being modeled just by the beat of the music. However, by proper choice of the inputs and the inclusion of a beat detection algorithm one can still have the vector field of the dynamic systems associated to the beat.
(3) The mathematical formulation of a quadrotor choreography performance is validated by a simulation programmed in Python with hardware-in-the-loop. An artist can control the simulation by playing a musical instrument and by using a set of foot pedals to change the quadrotor choreography in real-time. To validate the control system, the position of the quadrotor is compared with the desired position. To measure the effectiveness of the link between music and the position of the quadrotor, the trajectory generator system is inverted to generate a sequence of music pitches. This melodic phrase generated by the position of the quadrotor is then played back to the musician. Therefore,

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

a real-time musical interaction between the musician and the quadrotor occurs. This real-time musical interaction has also not been described previously in the literature to the best of our knowledge.

The paper is organized as follows. Section 2 presents the overall description of the system and the required background in music, choreography composition, and dance. Section 3 presents the mathematical formulation of the basic elements of a choreography, leading to the trajectory generation, the dynamics of the quadrotor, and the controllers based on the linearized dynamics. Section 4 discusses the simulation setup and the validation of the results. Section 5 concludes the paper.

2. Overall System Description and Background Material

Figure 1 presents an overview of the proposed methodology. The system consists of a musician interacting with a quadrotor. During an artistic performance, the musician plays an instrument to generate acoustic signals. The required acoustic features are extracted from this music. This paper will focus on controlling a quadrotor based on the pitch of the acoustic music signal. A trajectory generator and a trajectory filter transform the extracted music features into a trajectory, which is then forwarded to the quadrotor control system. Each choreography component is represented by a different color in Fig. 1. The overall system will be validated in a hardware-in-the-loop simulation where the hardware will consist of the musical instrument and the foot pedals.

The proposed combination of art and engineering allows for a stable, dynamic, and eloquent performance while providing artists the tools to control quadrotors, design a choreography for quadrotors, and to integrate them in a real-time artistic performance.

2.1. *Musical notes*

Music is an art form based on sound. It has been theorized that music has been around since the Paleolithic era (around 35,000 years ago); archaeologists found flutes carved from bones in which lateral holes had been pierced [36]. According to [37], music is an inherent and universal human quality. The pitch generated by a musical instrument is the fundamental frequency of a sound. The pitch is usually associated to a note. Notes go from A to G in what are called tone intervals, except for the intervals $B-C$ and $E-F$ which are semitones. An octave contains 12 notes in total. Geometrically, the pitch space of an octave can be represented by a circle, which is a one dimensional manifold (see Fig. 2 and [38]). The same notes repeat themselves every octave, but at fundamental frequencies that are doubled (if the octave is higher) or halved (if the octave is lower). For example, the note A_4 (the note A on the 4th octave) has a fundamental frequency of 440 Hz. The fundamental frequency of A_5, the A an octave higher, is 880 Hz. A melodic phrase is a combination of notes or pitches held for different times in a specific order.

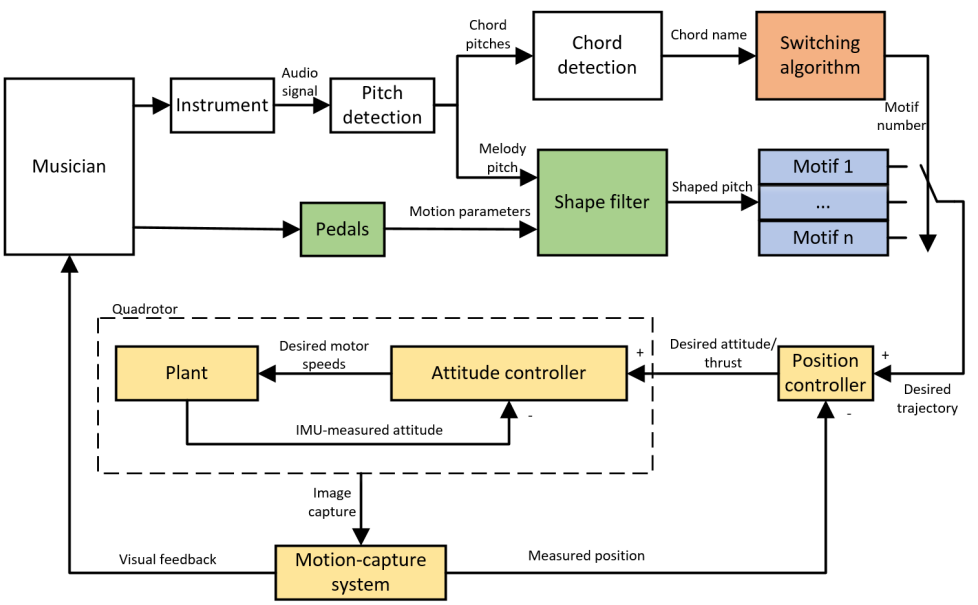


Fig. 1. System overview. The musician block is the supervising artist-in-the-loop.

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

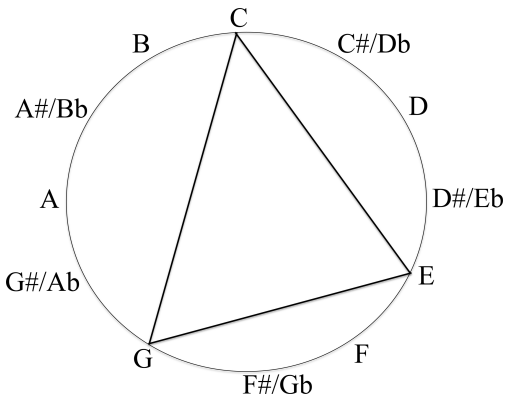


Fig. 2. Circle of notes, in which the notes of a C-major chord are connected through black lines.

2.2. *Pitch extraction*

An acoustic signal can be represented by the sum of its frequency components called the frequency spectrum. The pitch or fundamental frequency of a sound is the first frequency component to rise above the noise level of the frequency spectrum. To compute the frequency spectrum, the discrete Fourier transform can be used as [39]

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N}, \tag{1}$$

where N is the number of samples of the acoustic signal $x(n)$. The fast Fourier transform (FFT) is a method that efficiently computes the discrete Fourier transform [40]. There are several publicly available approaches for pitch extraction, such as the MIDI Merlin patch on Max/MSP [41], or the PredominantMelody function of the C++ open-source audio library Essentia [42]. However, we have developed the simple algorithm in Table 1 that better matches our real-time application.

Table 1. Pitch extraction algorithm.

FFT: Compute absolute value z of the FFT of signal x with N samples
Set Noise Factor: $\beta = 5$
Set Noise Level: $L = (max(z) - mean(z))/\beta$
Initialize Running Index: $n = 2$
while $abs(z(n) - z(n-1)) < L$ and $n < N$:
set $n := n + 1$
end while
if $n = N$ **then** no significant pitch was detected above noise level
else pitch is given by $p = nF_s/N$
endif

2.3. *The MIDI Format and musical representation*

The Musical Instrument Digital Interface (MIDI) is a popular standard, used to connect electronic musical instruments and computers [43]. In the MIDI format, the pitch of each note is transmitted by 7 data bits (a positive integer) and the musical information can be stored and processed digitally. The lowest note that can be represented is a C_{-1}, five octaves below middle C. This note has a fundamental frequency of 8.176 Hz. Let P be an integer representing a note, where $P = 0$ represents the lowest note. The highest note reached by MIDI is a G_9, at 12 544 Hz, represented by $P = 127$. Equation (2) converts the frequency of a sound to the integer P as [44]

$$P = 69 + 12 \log\left(\frac{f}{440}\right). \tag{2}$$

Let O be the octave to which the played note belongs and let $\tilde{P}$ be the pitch class of the note. The mod operator returns the remainder of a division by a certain number. The expressions for the pitch class $\tilde{P}$ and octave O are

$$\begin{aligned} \tilde{P} &= P \bmod 12, \\ O &= (P - \tilde{P})/12, \end{aligned} \tag{3}$$

2.4. *Chord detection*

A chord is a combination of multiple pitches played at the same time. The most basic type of chords are triads, chords created by three pitches. The different types of triads are Major, Minor, Augmented and Diminished chords. An A-Major chord will have a difference of 4 semi-tones between the first note (an A) and the second note (a C#), 3 semi-tones between the second note (C#) and third note (E), and 5 semi-tones between the third note (E) and the first note of the chord on the next octave (also an A). Based on this information we can form what are called difference matrices (Δ) of a chord. For example, the A-major chord is represented by a semi-tone difference matrix of $[4\ 3\ 5]$. A C-Major chord would have the pitch classes $C - E - G$, and would have the same difference matrix than the A-major chord. A minor chord, for example, would have a difference matrix of $[3\ 4\ 5]$. Note that the sum of all entries in the matrices is equal to 12, which is the total number of semi-tones in an octave. The difference matrices for the four types of triads are given by

$$\begin{aligned} \Delta_{\text{major}} &= [4\ 3\ 5], \\ \Delta_{\text{minor}} &= [3\ 4\ 5], \\ \Delta_{\text{augmented}} &= [4\ 4\ 4], \\ \Delta_{\text{diminished}} &= [3\ 3\ 6]. \end{aligned} \tag{4}$$

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

The use of difference matrices enables triad chord recognition (even for the case where the base note is not the lowest note played, called inverted chords). In order to recognize chords, a difference matrix is generated based on the notes played and then compared to the template difference matrices. A graphical representation of the difference matrices would be the triangles that form triads. For example, the notes that constitute any major chord are connected by a triangle identical to the C-major chord in Fig. 2. Chord detection algorithms can obviously be extended to chords with more than three pitches.

2.5. *Choreography*

This section is based on [45]. A choreography, derived from the greek words *choria* (circular dance) and *graphia* (writing) is the art of designing a sequence of movements. There are four main notions a choreographer takes into consideration when designing a human dance: space, shape (or energy), time, and structure (or dynamics) [45, 46]. This subsection will introduce the main components of a human choreography.

The space of a choreography refers to the space used by the dancer(s) during the performance. The space can be studied from a symmetry point of view, by analyzing if the dancers move in a symmetrical or asymmetrical motion. The dancers can move in a straight line, in curved motions, and other geometric forms. These geometric forms are also studied by the choreographer. The scale of the occupied space is also an important aspect of a dance — it refers to the space occupied by the dancer in comparison to the allowable space.

The shape of a choreography refers to the shape of the dancer and the motion of his/her body. Symmetry and asymmetry of the dancer's body shape can be used to infer stability or instability. The scale of the dance can be explained by the type of motion — such as the contrast between angular, curved, or straight shapes, or the contrast between shrinking and expanding motions, or the difference between contracted and extended body shapes. The shape could also refer to the energy behind specific dance moves.

The time or timing of the choreography usually accompanies that of the music the dancers are following. The tempo, the meter and the dynamics of the dance are important aspects of the choreography as they usually relate the music to the dance.

Similar to a song, a dance has different motions, or phases. The different movements of the dancers along the different spaces create a structure through combinations, variations and repetitions. Reference [45] describes the flow of a dance by the changes in phases of the dance as a function of the main timeline.

The four elements of a choreography will be mathematically formulated in the next section.

3. Mathematical Formulation of Quadrotor Choreography Components

In Fig. 1, the blocks that correspond to each choreography component have been highlighted in a specific color: the space is in blue, the shape is in green, the structure is in orange, and the time is in yellow. The overall objective is to use the control inputs (time component) to make the quadrotor follow a trajectory as a function of the music played by a musician. The trajectories are generated along the path motifs (space) and following the motion specifications (shape) dictated by the artist. The path motifs can be switched (structure switching) throughout a performance, in order to add structure to the choreography. The contribution of this paper at the level of guidance, navigation and control is to propose a novel mapping of parameters extracted from musical features (pitch or pitch class of the note, and octave) to guidance setpoints on height, and heading, which are two parameters typically used for UAV guidance, navigation, and control. This will be described in detail in the next section.

3.1. *Mathematical formulation of space*

Path motifs are one-dimensional manifolds embedded in a three-dimensional space, parameterized by a musical feature. A path motif serves as a basis for the trajectory generation system. The desired trajectory of the quadrotor is generated from the equations of a specific path motif. Any point of the desired trajectory will satisfy the equations of the motif it is based on. The motif designation stems from a quote by philosopher of mind and of art Susanne Langer [45]: "The fundamental forms which occur in the decorative arts of all ages and races — for instance the circle, the triangle, the spiral, the parallel — are known as motifs of design. They are not art 'works', not even ornaments, themselves, but they lend themselves to artistic creation. The word motifs bespeaks this function: *motifs are organizing devices that give the artist's imagination a start, and so* '*motivate*' *the work. They drive it forward and guide its progress*." The desired trajectory is along the path motif. However, it is noteworthy to mention that the quadrotor might not be always moving along the desired path: the quadrotor control system is designed to track the desired trajectory and guarantee convergence, but there are always tracking errors.

This subsection describes different parametrized path motifs to represent the space the dancing quadrotor will use. In all the motif figures the path motif is the red line, and the red points are the waypoints for integer pitches. The

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

following musical and physical parameters will be used in the mathematical description:

- P is the pitch ($P \in [0; 127]$) and $\tilde{P}$ is the pitch class
- N is the highest pitch an instrument can play
- Q is the maximum number of octaves an instrument can reach (assumed to be $Q = 11$)
- the number of semitones in an octave is 12
- the allowable maximum flight height is denoted as $h_{\max}$ (5 meters will be used)
- the minimum flight height is denoted by $h_{\min}$ (0.3 meters will be used)

This work considers three path motifs: the straight line, the circle, and the helix.

3.1.1. *Straight line path motif*

In this music-to-motif mapping the path motif is a vertical line, and the pitch of the current note being played by the musician is associated to a particular height. As the melody goes higher in pitch, the quadrotor will hover higher, and conversely the quadrotor will hover lower if the pitch goes lower. The straight line path motif is illustrated in Fig. 3 and is parameterized as

$$z = (h_{\max} - h_{\min})\frac{P}{N} + h_{\min}. \tag{5}$$

3.1.2. *Circular path motif*

A circular trajectory parallel to the $x - y$ plane and centered at $(0, 0, z_0)$ is parameterized by

$$\begin{aligned} x &= R\cos(\theta), \\ y &= R\sin(\theta), \\ z &= z_0, \end{aligned} \tag{6}$$

where x and y are the Cartesian coordinates of the horizontal plane, R is the radius of the circle, θ is the heading angle, and z_0 is the height. The proposed mapping assumes that a zero heading angle corresponds to the note C according to Fig. 2. The trajectories are parameterized by the fixed radius R and by the heading angle defined by

$$\theta = 2\pi\tilde{P}/12. \tag{7}$$

Figure 4 is an image from the simulation of the circle path motif, with $R = 2$ m, and $z_0 = 2$ m.

3.1.3. *Helicoidal trajectory motif and variations*

Three different helicoidal motifs will be proposed. In the first musical mapping, the concept of "higher pitch" and the circle of notes are brought together. Different circles of notes are generated at different heights: the heading angle depends on the pitch, and the height depends on the octave in which the pitch lies. This path motif can be described by the following equations

$$\begin{aligned} x &= R\cos(2\pi P/12), \\ y &= R\sin(2\pi P/12), \\ z &= (h_{\max} - h_{\min})\frac{(P - \tilde{P})}{12Q} + h_{\min}. \end{aligned} \tag{8}$$

This particular motif is discontinuous: the jumps in height are due to the jump in octaves. In a second mapping, the different circles of notes are connected in a more smooth way. The height is dependent on the pitch instead of the octave. The parameterization is described by the

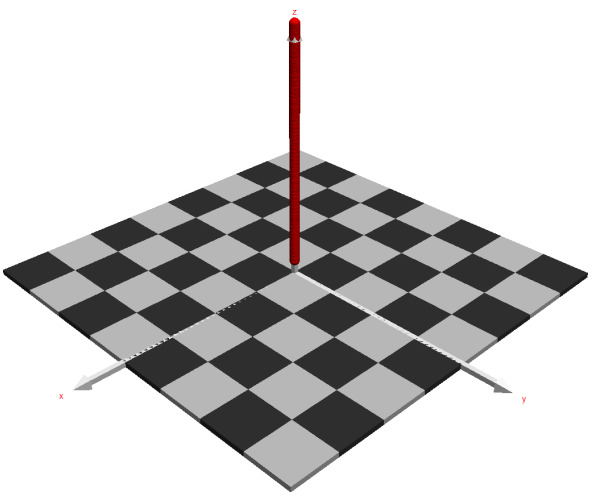


Fig. 3. Simulation of the straight line path motif.

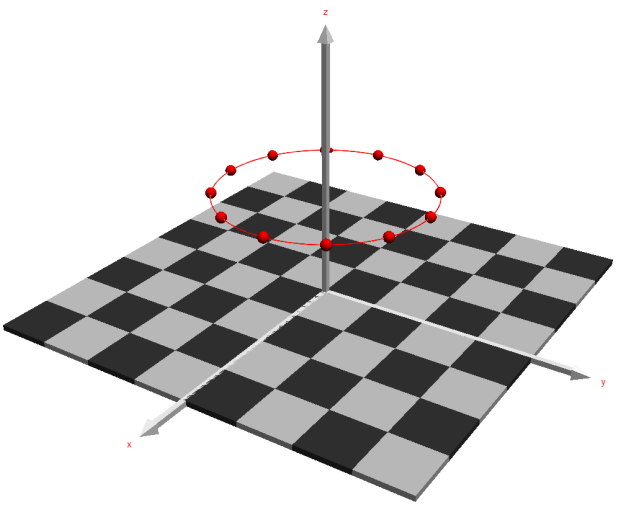


Fig. 4. Simulation of the circle path motif.

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

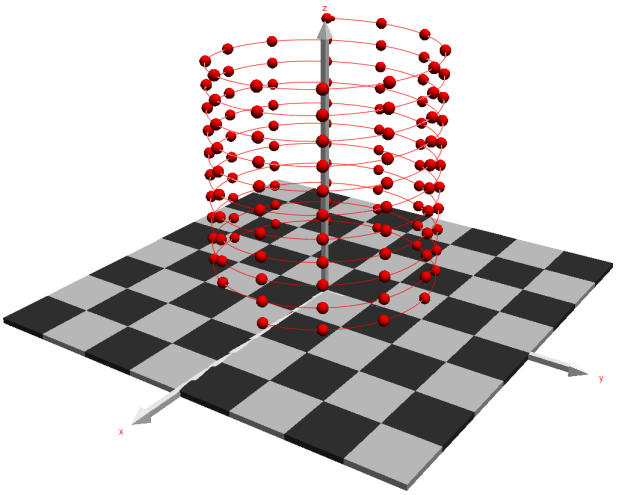


Fig. 5. Simulation of the second helicoidal path motif, with $R = 2$ m.

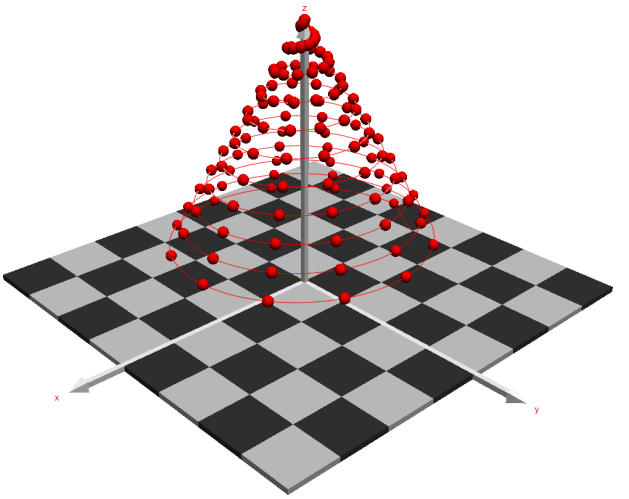


Fig. 6. Simulation of the third helicoidal path motif, with $\alpha = 0.02$.

following equations (see Fig. 5):

$$\begin{aligned} x &= R\cos(2\pi P/12), \\ y &= R\sin(2\pi P/12), \\ z &= (h_{\max} - h_{\min})\frac{P}{N} + h_{\min}. \end{aligned} \tag{9}$$

In a third mapping, the circles of notes are smoothly connected since the height is a function of the pitch instead of the octave. However, the radius of the circles is not constant. The radius R is made to decrease as the pitch increases to create a set of circles that form a cone-shaped path. This can be seen in Fig. 6 and is described by the following equations, where α is fixed

$$\begin{aligned} x &= \alpha(N - P)\cos(2\pi P/12), \\ y &= \alpha(N - P)\sin(2\pi P/12), \\ z &= (h_{\max} - h_{\min})\frac{P}{N} + h_{\min}. \end{aligned} \tag{10}$$

The path motif possibilities are endless, and their design is only limited by the creativity of the artist.

3.2. *Mathematical formulation of shape*

The quadrotor is a rigid body, and cannot alter its shape. However, the shape of the quadrotor refers to the motion of the quadrotor. The desired trajectory of the quadrotor is within the flying space by generating specific trajectories along a path motif. Any point along the generated desired trajectory of the quadrotor will belong to the path motif. However, the path motifs and the extracted music features generate discrete geographic waypoints corresponding to discrete notes that are played. The transition from one point to another as the note being played changes corresponds to a step discontinuity. In order to generate a continuous trajectory, the pitch is filtered by a shape filter (see Fig. 7) to generate a 'modified pitch' continuous signal. The proposed filter combines the ideas of frequency and decay of a sound and is a second-order system that filters the pitch values to generate a continuous signal. The transfer function of the system is

$$H(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n + \omega_n^2} \tag{11}$$

where ω_n is the undamped natural frequency, and ζ is the damping ratio.

A higher ω_n will result in a faster response, and conversely a lower ω_n results in a slower response. The damping ratio ζ controls the overshoot of the oscillations — a higher damping ratio will result in a more damped response with smaller overshoot. These two parameters are made available to the artist through two MIDI pedals. Each MIDI pedal generates values between 0 and 127, depending on the position of the pedal. These values are linearly mapped to the allowable range of damping ratio ($\zeta \in [\zeta_{\min}, \zeta_{\max}]$) and the allowable range of natural frequency ($\omega_n \in [\omega_{n,\min}, \omega_{n,\max}]$) according to

$$\begin{aligned} \omega_n &= (\omega_{n,\max} - \omega_{n,\min})\frac{E_1}{127} + \omega_{n,\min}, \\ \zeta &= (\zeta_{\max} - \zeta_{\min})\frac{E_2}{127} + \zeta_{\min}, \end{aligned} \tag{12}$$

where E_i is the MIDI value of the ith pedal ($i = 1, 2$), which is a positive number. Leaving the parameters tunable enables artists to shape the trajectory of the quadrotor in real-time along the path motif.

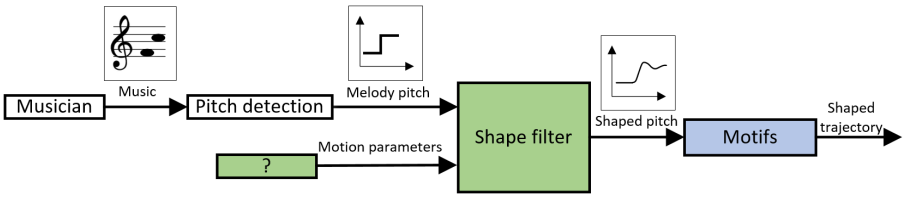


Fig. 7. Shape filter with graphical representations of the signals.

3.3. *Mathematical formulation of structure*

The structure of the quadrotor choreography is the switching of path motifs as a function of chord sequences. The chord difference matrices were introduced in Sec. 2.4. Path motifs will be associated with musical phrases and the measures of the music being played. A measure is a structural element of music that consists of a specific number of beats. Figure 8 is an example of a simple musical structure consisting of three measures with four chords each. The first two measures have the same chord sequence ($C_{\text{maj}} - G_{\text{maj}} - A_{\text{min}} - F_{\text{maj}}$); they are therefore given the same path motif. The third measure could indicate a change in the structure of the song, such as, for example, a change from a verse to the chorus. The chord sequence of the third measure is different ($G\#_{\text{maj}} - Bb_{\text{maj}} - C_{\text{maj}} - C_{\text{maj}}$). This change in the chord sequence triggers a switching in the motif. This switching is represented by an arrow in Fig. 8. The chord switching algorithm starts when a chord is detected and has the flow chart drawn in Fig. 9. This algorithm is pre-programmed and it corresponds to an autonomous switching. As an alternative a manual activated switch is provided so that the artist can control the switching through MIDI button pedals where each path motif is assigned to a button pedal.

3.4. *Mathematical formulation of time*

A quadrotor choreography, as a whole, can be mathematicaly modeled as a switched system. A switched system is a system composed of two different mathematical objects:

- Object 1: Differential equations that describe the flow of the continuous dynamics, the quadrotor dynamics in this case, depending on a continuous control input U.

Measure 1				Measure 2				Measure 3			
C_{maj}	G_{maj}	A_{min}	F_{maj}	C_{maj}	G_{maj}	A_{min}	F_{maj}	$G\#_{maj}$	Bb_{maj}	C_{maj}	C_{maj}
Chord sequence 1				Chord sequence 1				Chord sequence 2			
1.1	1.2	1.3	1.4	1.1	1.2	1.3	1.4	2.1	2.2	2.3	2.4
Path motif 1				Path motif 1				Path motif 2			

Switch

Fig. 8. Representation of the structure of a choreography as a function of the structure of the music.

- Object 2: A switching rule that orchestrates among different subsystems corresponding to a different value of the discrete-event input σ, which for the quadrotor orchestrates the changes in path motifs as a function of the changes in music chords.

In the previous subsection, the switching algorithm was associated with the structure of the dance. In this subsection, the timing will refer to the time evolution of the closed-loop dynamic system as described by the solution of a set of differential equations. The quadrotor can move along the x-, y- and z-axes of the inertial reference frame. Rotations are described by ϕ for roll, θ for pitch and ψ for yaw. The rest of this section follows closely classical linear control design as presented in for example [47] and used in [10] for quadrotors.

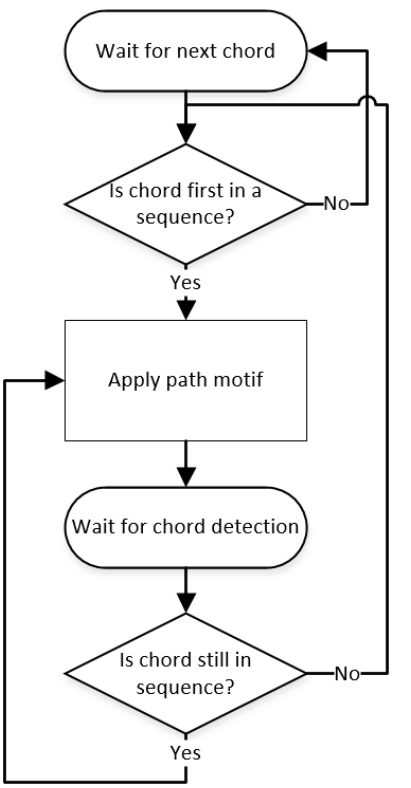


Fig. 9. Flow chart of chord switching algorithm.

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

The linearized translational dynamics in the inertial reference frame can then be described by (see [48–50]):

$$\begin{aligned}\ddot{x} &= (\cos(\psi_0)\theta + \sin(\psi_0)\phi)\frac{U_1}{m},\\ \ddot{y} &= (\sin(\psi_0)\theta - \cos(\psi_0)\phi)\frac{U_1}{m},\\ \ddot{z} &= -g + \frac{U_1}{m},\end{aligned} \tag{13}$$

where m is the mass (kg) of the quadrotor and g is the magnitude of the gravitational acceleration on Earth ($9.81\,\mathrm{m/s^2}$).

Following a similar approach to [48–50], the linearized attitude dynamics are described by

$$\ddot{\phi} = \frac{U_2}{I_{xx}}, \quad \ddot{\theta} = \frac{U_3}{I_{yy}}, \quad \ddot{\psi} = \frac{U_4}{I_{zz}} \tag{14}$$

and the control inputs U_i are defined as

$$\begin{aligned}U_1 &= T_1 + T_2 + T_3 + T_4,\\ U_2 &= l(T_2 - T_4),\\ U_3 &= l(-T_1 + T_3),\\ U_4 &= l(T_1 - T_2 + T_3 - T_4),\end{aligned} \tag{15}$$

where T_i represents the thrust generated by the corresponding ith motor, and l is the distance in meters from the center of rotation of the motor to the center of mass of the quadrotor. Equation (15) represents the vertical, roll, pitch, and yaw control inputs, respectively.

The controller design to make the quadrotor follow the path motifs will now be described. The control structure consists of a position controller and an attitude controller. These two controllers provide a nested closed-loop solution, where the position control is the outer-loop, and the attitude control is the inner-loop as described in the bottom of Fig. 1. The position controller is designed such that the closed-loop dynamics are described by a second-order stable system as follows

$$\begin{aligned}0 = {} & \omega_z^2(z_{\text{desired}} - z) + 2\zeta_z\omega_z(\dot{z}_{\text{desired}} - \dot{z})\\ & + (\ddot{z}_{\text{desired}} - \ddot{z}),\end{aligned} \tag{16}$$

Replacing $\ddot{z}$ from Eq. (13) in Eq. (16) yields the control law for U_1 as

$$\begin{aligned}\frac{U_1}{m} = {} & \omega_z^2(z_{\text{desired}} - z) + 2\zeta_z\omega_z(\dot{z}_{\text{desired}} - \dot{z})\\ & + \ddot{z}_{\text{desired}} + g.\end{aligned} \tag{17}$$

Solving Eqs. (13) at hover conditions ($U_1 = mg$) for the desired values of ϕ and θ yields

$$\begin{aligned}\phi_{\text{desired}} &= \frac{1}{g}(\ddot{x}\sin(\psi_0) - \ddot{y}\cos(\psi_0)),\\ \theta_{\text{desired}} &= \frac{1}{g}(\ddot{x}\cos(\psi_0) + \ddot{y}\sin(\psi_0)).\end{aligned} \tag{18}$$

The values of $\ddot{x}$ and $\ddot{y}$ are then obtained from Eq. (16). The attitude controller takes as inputs U_1, ϕ_{desired} and θ_{desired} and generates the control inputs responsible for changing the orientation of the quadrotor. More specifically, the controller receives the desired roll, pitch, and yaw angles from the position controller and computes the control inputs U_2, U_3, and U_4. The control input U_2 can be designed such that the closed-loop dynamics for roll behave like a second-order system with damping ratio ζ_ϕ and natural frequency ω_ϕ as

$$0 = \omega_\phi^2(\phi_{\text{desired}} - \phi) + 2\zeta_\phi\omega_\phi(\dot{\phi}_{\text{desired}} - \dot{\phi}) + (\ddot{\phi}_{\text{desired}} - \ddot{\phi}). \tag{19}$$

Including the dynamics for $\ddot{\phi}$ from equation (14) in the desired second-order response (19) yields

$$U_2 = I_{xx}[\omega_\phi^2(\phi_{\text{desired}} - \phi) + 2\zeta_\phi\omega_\phi(\dot{\phi}_{\text{desired}} - \dot{\phi}) + \ddot{\phi}_{\text{desired}}]. \tag{20}$$

where ϕ_{desired} is given by the first equation of the pair of equations (18).

The pitch and yaw controllers are designed following the same procedure and will therefore be omitted due to space constraints.

4. Simulation

Figure 10 is an image of a hardware-in-the-loop simulation of the quadrotor choreography system, where the laptop, the pedal board, and the MIDI keyboard are clearly visible. The parameters used to simulate the model are presented in Table 2.

4.1. *Hardware*

There are three hardware components in the simulation implementation. The following is a list of the components (see Fig. 10):

- The simulation is run on a Samsung Series 9 laptop on Windows 10 Technical Preview. The laptop has 8 GB of RAM and an Intel i5 1.7 GHz CPU. However, there are no restrictions to the operating system: the simulation is written in Python, and the MIDI standard can be used on all operating systems.

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

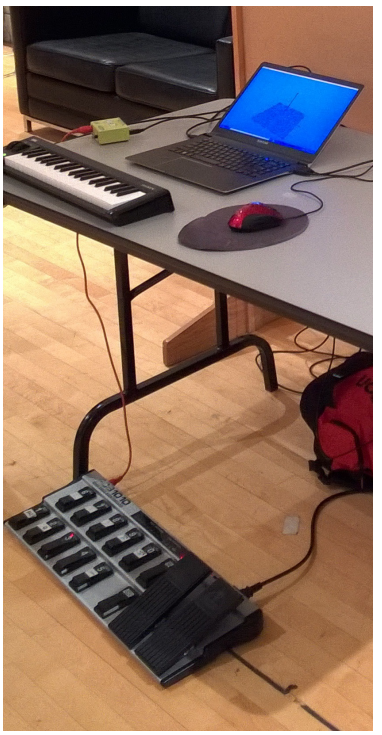


Fig. 10. Quadrotor simulation with hardware-in-the-loop.

Table 2. Simulation parameters.

Parameter	Value
m (kg)	0.431
I_{xx} (kg·m^2)	0.0021
I_{yy} (kg·m^2)	0.0018
I_{zz} (kg·m^2)	0.0027
$l(m)$	0.19
k_ψ	0.1

- A MIDI keyboard is used to play music to the quadrotor. The Korg microKEY-37 MIDI keyboard [51] connects to the main computer through a USB cable.
- The Behringer FCB1010 MIDI pedal board has two expressive pedals to control the damping ratio and the natural frequency of the shape filter, and 10 button pedals to control the path motif. The physical output on the pedal is a MIDI connector (a converter is required to connect the pedal to a computer through USB).

4.2. *Software*

The system is simulated in a Python environment [52]. Two open-source Python add-ons have been used:

- VPython is a graphical library which gives programmers the tools to draw 3D objects [53].
- The Pygame library [54] is a set of modules designed for programming video games. One of the modules reads MIDI hardware input events and plays MIDI music [54].

The simulation is run at a frequency of 100 Hz. The controllers and the shape filter are transformed to a discrete time equivalent system through the bilinear transform

$$s = \frac{2}{T}\frac{z-1}{z+1}, \tag{21}$$

where T is the sampling period, s is the derivative operator, and z is the one step advance operator. The bilinear transform discretizes continuous systems so that the left half-plane of the Laplace domain is mapped to the unit circle of the z-domain, therefore preserving stability [47].

4.3. *Quadrotor music playback*

When generating MIDI music, multiple parameters can be modified to better design the sound. There are two main parameters:

- the pitch of the sound,
- the amplitude of the sound.

The tracking error between the quadrotor's position and the path motif can be characterized by, incidentally, two parameters as well:

- the shortest distance between the path motif and the quadrotor,
- the closest point to the quadrotor which belongs to the path motif.

In what follows, we will consider that the pitch associated with the position of the quadrotor is P and that the total number of pitches is 128. The distance d between the path motif and the quadrotor is mapped to the amplitude A of the sound generated by the music motif as

$$A = \begin{cases} (A_{\min} - A_{\max})d + A_{\max} & d \in [0;1] \\ 0 & d > 1. \end{cases} \tag{22}$$

Note that the amplitude will decrease with the distance from the motif, and the amplitude will be 0 if the quadrotor is further than one meter away from the path motif. The MIDI standard is used in this work to play the music generated by the quadrotor by knowing the pitch and amplitude of the sound (note that the maximum and minimum amplitudes are 127 and 0, respectively [44]). Analytically retrieving the pitch from a point along the manifold involves inverting the equations of the path motifs, such that a point along the path motif is mapped to a pitch. If the manifold consists of several copies of itself this inverse mapping does not lead to an unique point. For example, for the circle of notes, there are several pitches in different octaves that

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

correspond to the same position. The determination of the pitch will now be detailed for the straight line and for the circular path motifs to illustrate the process of inverse mapping.

4.3.1. *Straight line path motif*

The pitch can be extracted from the height of the quadrotor as

$$P = N\frac{z - h_{\min}}{h_{\max} - h_{\min}}. \tag{23}$$

The distance between the quadrotor and the z-axis can be calculated as

$$d = \sqrt{x^2 + y^2}. \tag{24}$$

4.3.2. *Circular path motif*

The heading angle of the position of the quadrotor is obtained from

$$\theta = \tan^{-1}(y/x). \tag{25}$$

The pitch class of the music motif is then extracted using the equation

$$\tilde{P} = \frac{12\theta}{2\pi}. \tag{26}$$

The pitch is not unique unless one knows the octave that is being played since $P = \tilde{P} + 12O$. Finally, the shortest distance between a circle of radius R parallel to the $x-y$ plane and with center along the z-axis at $z = z_{\text{center}}$ and any point (x, y, z) is

$$d = \sqrt{\left(\sqrt{x^2 + y^2} - R\right)^2 + (z - z_{\text{center}})^2}. \tag{27}$$

4.4. *System validation*

During validation, the parameters of the mathematical formulation of the choreography and the music are changed as a function of time. The path motif switching element will also be time dependent. The shape filter parameters ζ and ω are changed twice for each motif. As an example, Fig. 11 presents the conic helix path motif drawn in red and is a screen capture of the quadrotor simulation implementation. It can be seen that the quadrotor is following very closely the path motif. The hardware-in-the-loop simulation proposed in this paper was successfully used by an artist. A video of the performance can be seen in [55]. This reference is a recording of the first quadrotor performance designed by an artist, entitled *Geometry of Curves: Prelude on a Ramped Spiral*. In this performance, the music played by the artist is muted, and the quadrotor music playback sounds are generated in Ableton Live, a software for music production [56]. The MIDI information is sent to Ableton Live through a virtual port, which is managed by ipMIDI.

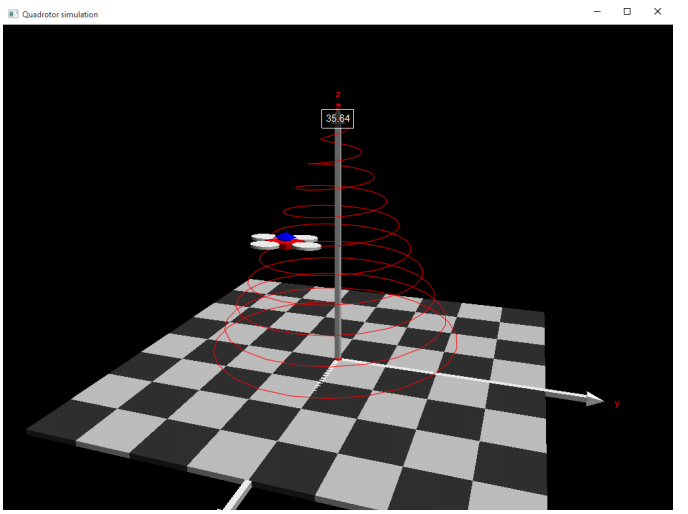


Fig. 11. Screen capture of the quadrotor simulation with the cone motif.

The system was also tested in hardware and the results can be seen in the video [57]. This video was presented in the Tedx Concordia talk [58].

5. Conclusions

This paper presented a novel systematic methodology to perform guidance navigation and control of a quadrotor in response to real-time music commands from an artist-in-the-loop based on the main components of a choreography. The methodology uses the pitch extracted from the music being played plus two parameters that can be modified in real-time by the artist-in-the-loop. All these parameters are used to send guidance commands to the quadrotor allowing an effective real-time interaction with the musician while the choreography is being performed. The four main components of a choreography (space, shape, structure and time) were mathematically formulated. For the space component, different path motifs parameterized by the music were presented: a straight line, a circle, and three helicoidal variations. For the shape component, the quadrotor is given movement characteristics by filtering the pitch through a second-order filter to generate a shaped pitch. The parameters of this filter can be modified in real-time by the artist-in-the-loop. The structure of the choreography is modified by a switching algorithm based on chord

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.

sequences, or by a pedal board and button-pedals that the artist can use. The time component refers to the dynamic equations describing the motion of the quadrotor and the associated closed-loop control system. The system proposed in this paper was successfully used by an artist in [55] where a video of the performance can be seen. The performance validates that the proposed system can effectively be used by an artist-in-the-loop.

Part of the material of this paper was presented in the Tedx Concordia talk [58].

Acknowledgments

The authors would like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT) for sponsoring this research. The authors would also like to thank and acknowledge Ms. Caroline Glass for composing the music and being the artist-in-the-loop for the performance that validated the results of this research. Ms. Glass has also supplied all the hardware needed for the performance.

References

[1] D. W. Casbeer, R. W. Beard, T. W. McLain, S. Li and R K. Mehra, Forest fire monitoring with multiple small UAVs, in *American Control Conf.*, June 8–10, 2005, pp. 3530–3535.

[2] F. Caballero, L. Merino, J. Ferruz and A. Ollero, A visual odometer without 3D reconstruction for aerial vehicles. Applications to building inspections, *IEEE Int. Conf. Robotics and Automation*, Barcelona, Spain, April 2005, pp. 4673–4678.

[3] J. R. Martinez-De Dios and A. Ollero, Automatic detection of windows thermal heat losses in buildings using UAVs, in *World Automation Congress (WAC)*, Budapest, Hungary, July 24–26 2006, pp. 1–6.

[4] T. Luukkonen, Modelling and control of quadcopter, Independent research project in applied mathematics, Aalto University, School of Science, August 22, 2011.

[5] S. Bouabdallah and R. Siegwart, Full control of a quadrotor, in *Int. Conf. Intelligent Robots and Systems*, San Diego, CA, USA, Oct 29–Nov 2, 2007, pp. 153–158.

[6] S. Bouabdallah, A. Noth and R. Siegwan, PID vs LQ control techniques applied to an indoor micro quadrotor, in *Int. Conf. Intelligent Robots and Systems*, Sep 28–Oct 2, 2004, Sendal, Japan, pp. 2451–2456.

[7] M. K. Apostolos, M. Littman, S. Lane, D. Handelman and J. Gelfand, Robot choreography: An artistic-scientific connection, *Comput. Math. Appl.* **32**(1) (1996) 1–4.

[8] A. Schollig, F. Augugliaro, S. Lupashin and R. D'Andrea, Synchronizing the motion of a quadrocopter to music, in *IEEE Int. Conf. Robotics and Automation*, Anchorage, Alaska, USA, 2010, pp. 3355–3360.

[9] F. Augugliaro, A. Schollig and R. D'Andrea, Methods for designing and executing an aerial dance choreography, *IEEE Robot. Autom. Mag.* **20**(4) (2013) 96–104.

[10] A. P. Schoellig, H. Siegel, F. Augugliaro and R. D'Andrea, So you think you can dance? Rhythmic flight performances with quadrocopters, in *Controls and Art*, A. LaViers and M. Egerstedt (eds.) (Springer, 2014), pp. 73–105.

[11] W. M. Hartmann, *Signals, Sound, and Sensation* (Springer, New York, 1998).

[12] H. Ding, B. Qian, Y. Li and Z. Tang, A method combining LPC-Based cepstrum and harmonic product spectrum for pitch detection, in *Int. Conf. Intelligent Information Hiding and Multimedia Signal Processing*, Pasadena, CA, USA, 2006.

[13] D. G. Childers, D. P. Skinner and R. C. Kemerait, "The cepstrum: A guide to processing, *Proc. IEEE* **65**(10) (1977) 1428–1443.

[14] A. V. Oppenheim and R. W. Schafer, From frequency to quefrency: A history of the cepstrum, *IEEE Signal Process. Mag.* **21**(5) (2004), pp. 95–106.

[15] M. R. Schroeder, Period histogram and product spectrum: New methods for fundamental frequency measurement, *J. Acoust. Soc. Am.* **43** (1968) 829.

[16] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing* (Prentice Hall, Englewood Cliffs, New Jersey, 1975).

[17] L. P. Rabiner, On the use of autocorrelation analysis for pitch detection, *IEEE Trans. Acoustics, Speech, and Signal Processing* **ASSP-25**(1) (1977) 24–33.

[18] W. B. Kuhn, A real-time pitch recognition algorithm for music applications, *Comput. Music J.* **14**(3) (1990) 60–71.

[19] S. Harbeck, A. Kiebetaling, R. Kompe, H. Niemann and E. No"th, *Robust Pitch Period Detection Using Dynamic Programming with an ANN Cost Function*, Univ. Erlangen-Nurnberg, Report 91, July 1995.

[20] E. Benetos and S. Dixon, Joint multi-pitch detection using harmonic envelope estimation for polyphonic music transcription, *IEEE J. Sel. Topics Signal Process.* **5**(6) (2011) 1111–1123.

[21] E. Benetos and S. Dixon, A temporally-constrained convolutive probabilistic model for pitch detection, in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2011, pp. 133–136.

[22] J. P. Kroeker, Resonant aperture detection of natural resonances, *Wave Reactive Network*, Available at https://drive.google.com/file/d/0B-XL4upneCeCdzhGNXJLNEtaMjQ/.

[23] P. de la Cuadra, A. Master and C. Sapp, Efficient pitch detection techniques for interactive music, *Int. Computer Music Conf.*, La Habana, Cuba, 2001.

[24] L. R. Rabiner, M. J. Cheng, A. E. Rosenberg and C. A. McGonegal, A comparative performance study of several pitch detection algorithms, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-24**(5) (1976) 399–418.

[25] D. M. Randel, *The Harvard Dictionary of Music* (Harvard University Press Reference Library, 2003).

[26] K. Murata, K. Nakadai, R. Takeda, H. G. Okuno, T. Torii, Y. Hasegawa and H. Tsujino, A beat-tracking robot for human-robot interaction and its evaluation, in *IEEE-RAS Int. Conf. Humanoid Robots*, Daejeon, Korea, 2008, pp. 79–84.

[27] 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 *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Nice, France, 2008, pp. 2459–2464.

[28] E. W. Large and J. F. Kollen, Resonnance and the perception of musical meter, *Connect. Sci.* **6** (1994) 177–208.

[29] E. D. Scheirer, Tempo and beat analysis of acoustic musical signals, *J. Acoust. Soc. Am.* **103**(1) (1998) 588–601.

[30] M. Goto and Y. Muraoka, An audio-based real-time beat tracking system and its applications, in *Proc. 2001 Int. Computer Music Conference*, Ann Arbor, MI, 1998, pp. 17–20.

[31] V. Kumar, Robot quadrotors perform James Bond theme, University of Pennsylvania, General Robotics, Automation, Sensing and Perception

(GRASP) Lab, Feb 2012, Available at http://www.youtube.com/watch?v=_sUeGC-8dyk

[32] Cirque du Soleil, ETH Zurich, and Verity Studios, *SPARKED: A Live Interaction Between Humans and Quadcopters*, available at https://www.youtube.com/watch?v=6C8OJsHfmpI. [accessed on September 22, 2014].

[33] J. Santana, X. Smith and P. Saatchi, *Meet Your Creator — Quadrotor Show*, Available at https://www.youtube.com/watch?v=cseTX_r-W3uM.

[34] S. F. de Sousa Junior and M. F. M. Campos, Shall we dance? A music-driven approach for mobile robots choreography, in *Int. Conf. Intelligent Robots and Systems (IROS)*, San Francisco, CA, 25–30 Sep. 2011, pp. 1974–1979.

[35] I. Mahjri, A. Dhraief and A. Belghith, A review on collision avoidance systems for unmanned aerial vehicles, in *International Workshop on Communication Technologies for Vehicles. Nets4Cars/Nets4Trains/Nets4Aircraft 2015*: Communication Technologies for Vehicles M. Kassab, M. Berbineau, A. Vinel, M. Jonsson, F. Garcia and J. Soler (eds.). Lecture Notes in Computer Science, Vol. 9066, pp. 203–214 (Springer, Switzerland).

[36] N. J. Conard, M. Malina and S. C. Munzel, New flutes document the earliest musical tradition in southwestern Germany, *Nature*, **460**, June (2009), pp. 737–740.

[37] J. Blacking, *How Musical is Man?* (University of Washington Press, 1974).

[38] D. Tymoczko, *A geometry of Music: Harmony and Counterpoint in the Extended Common Practice* (Oxford University Press, 2011).

[39] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Second edn. (Prentice Hall, Upper Saddle River, New Jersey, 1998).

[40] J. W. Cooley and J. W. Tukey, An algorithm for the machine calculation of complex fourier series, *Math. Comput.* **19**(90) (1965) 297–301.

[41] R. George, *Project 326: MIDI Merlin*, 2012, Available at http://cycling74.com/project/midi-merlin/.

[42] Music Technology Group of Universitat Pompeu Fabra, *Algorithm Reference: PredominantMelody — Essentia 2.0.1 Documentation*, 2012.

[43] D. M. Huber, *The MIDI Manual: A Practical Guide to MIDI in the Project Studio*, 3rd edn. (Focal Press, 2007).

[44] MIDI Manufacturers Association. *MIDI 1.0 Detailed Specification*, Document Version 4.2, 1995, Los Angeles, CA.

[45] J. M. Smith-Autard, *Dance Composition: A Practical Guide to Creative Success in Dance Making*, 6th edn. (Bloomsbury Methuen Drama, London, 6th 2010).

[46] P. Sofras, *Dance Composition Basics: Capturing the Choreographer's Craft* (Human Kinetics, USA, 2006).

[47] G. F. Franklin, J. David Powell and A. Emami-Naeini. *Feedback Control of Dynamic Systems*, 5th edn. (Pearson Prentice Hall).

[48] M. Moarref, C. Ossa-Gomez and L. Rodrigues, Multiloop controller design for a fly-by-wireless UAV quadrotor based on a multi-rate sampled-data model, *IEEE Trans. Aerosp. Electron. Syst.* **51**(3) (2015) 2319–2331.

[49] T. Bresciani, Modelling, Identification and Control of a Quadrotor Helicopter, M.S. thesis, Dept. Automatic Control, Lund Univ., Lund, Sweden, 2008.

[50] B. Etkin and L. Duff Reid, *Dynamics of Flight, Stability and Control*, 3rd edn. (McGraw-Hill, 1996).

[51] KORG, *MicroKEY USB-Powered Keyboard — MIDI Controllers — KORG*, Available at http://www.korg.com/us/products/controllers/microkey/.

[52] Python Software Foundation, *What is Python? Executive Summary — Python.org*, Available at https://www.python.org/doc/essays/blurb/.

[53] Visual Python, *VPython*, Available at http://vpython.org/.

[54] Pygame, pygame.midi — Pygame v1.9.2 documentation, Available at https://www.pygame.org/docs/ref/midi.html.

[55] C. Glass, Quadrotor Choreography performance by Caroline Glass, June 2015, Available at https://youtu.be/wVw2M_RSegg.

[56] Ableton, *Music Production with Live 9 and Push — Ableton*, Available at https://www.ableton.com/.

[57] B. Carvalho, M. El-Jiz, M. G. Rodrigues, L. Rodrigues, Quadrotor Hardware Performance, October 2016, Available at https://youtu.be/QOo-0690GE4.

[58] L. Rodrigues, "Drones, Art and Music", Concordia Tedx Talk, Place des Arts, Montreal, September 24, 2017.

Michael El-Jiz graduated from Concordia University with a Bachelor's Degree in Electrical and Computer Engineering in 2013. Merging his appreciation for robots and music he worked on a Capstone project entitled "Bothoven", a robot that reads sheet music and plays it on a physical piano. For his Masters of Science thesis Michael performed research on the integration of drones and music. He is currently working as a Software Engineer in Seattle.

Luis Rodrigues is a professor at the Department of Electrical and Computer Engineering of Concordia University, Canada and a member of the Concordia Institute of Aerospace Design & Innovation. Dr. Rodrigues obtained his Ph.D. in Aeronautics and Astronautics from Stanford University in 2002. He earned his "licenciatura" and his M.Sc. degrees in Electrical and Computer Engineering from IST, Technical University of Lisbon. Dr. Rodrigues worked as a consultant in speech modelling and recognition for Eliza Corporation in USA and as a flight simulation project manager for Ydreams in Portugal before joining Concordia University in 2003. His research interests lie in the areas of optimal, switched, and hybrid control with applications to aerospace, automotive, manufacturing, biological, and artistic systems.

Un. Sys. 2018.06:1-13. Downloaded from www.worldscientific.com
by NATIONAL UNIVERSITY OF SINGAPORE on 07/06/18. Re-use and distribution is strictly not permitted, except for Open Access articles.