Tactile Human-Quadrotor Interaction: *MetroDrone*

Marc Lieser  
marc.lieser@hs-rm.de  
RheinMain University of Applied Sciences  
Wiesbaden, Germany

Ulrich Schwanecke  
ulrich.schwanecke@hs-rm.de  
RheinMain University of Applied Sciences  
Wiesbaden, Germany

Jörg Berdux  
joerg.berdux@hs-rm.de  
RheinMain University of Applied Sciences  
Wiesbaden, Germany

Figure 1: Tactile human-quadrotor interaction sequence with *MetroDrone*: The user taps the frame of the airborne vehicle several times to the beat of the music. Depending on the frequency and direction detected in the change of acceleration measured by the quadrotor’s onboard accelerometer, a trajectory is generated that makes the quadrotor move to the beat.

**ABSTRACT**

Aerial robots such as quadrotors enjoy ever-increasing popularity and emerge in everyday applications that require user interaction. At immediate proximity, physical control of the quadrotor by touch may be desired or even necessary. In this paper we present a tactile 3D touch interaction scenario with a quadrotor by introducing virtual buttons whose operation is detected in the accelerometer data of the built-in Inertial Measurement Unit (IMU) of the quadrotor. By dispensing with additional sensors, we are able to keep the size of the used quadrotor to a minimum and thus address the problem of users being discouraged from interaction with quadrotors at immediate proximity. As an example for the proposed interaction scenario, we introduce *MetroDrone*, a quadrotor responding to repeated user taps to virtually defined buttons by flying trajectories according to the beat and operated button. This introduces a minimalist interaction technique that requires no intermediary devices and strengthens human-robot connections through shared musical experience.

**CCS CONCEPTS**

- Human-centered computing → User interface design; Interaction techniques;  
- Computer systems organization → External interfaces for robotics.

**KEYWORDS**

human-quadrotor interaction, non-verbal communication, tactile interaction, quadrotor companions, social robots

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

TEI ’21, February 14–17, 2021, Salzburg, Austria

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8213-7/21/02...

https://doi.org/10.1145/3430524.3440649

ACM Reference Format:

1 INTRODUCTION

Quadrotors, Unmanned Aerial Vehicles (UAVs) with four rotors, are becoming increasingly popular due to their low cost and mechanical simplicity. Some of them are now fully autonomous, providing first aid for victims in Search and Rescue (SAR) scenarios, delivering medical equipment and parcels or document the endeavors of athletes with aerial videos. It is likely that in future scenarios we will share our homes with more robots, ground-based or of aerial nature, each for different purposes. Just as smartphones already do today on Earth and as robot assistants Astrobee [Bualat et al. 2018] and CIMON [DLR 2019] do on the International Space Station (ISS), small UAVs could support us in the future in our private homes. These scenarios require new interaction techniques that feel more natural than the use of classic radio remote controls or smartphones.

For natural communication at greater distances, several gesture control approaches were proposed [Cauchard et al. 2015; Naseer et al. 2013]. When it’s not possible for cameras to capture humans in their entirety and thus pose estimation fails, interaction scenarios from an even closer proximity may be desired. For these close-range scenarios, investigations were made for control by facial expression [Nagi et al. 2014] and voice [Krishna et al. 2015; Ng and Sharlin 2011] as well as multimodal interaction including hand gestures [Cacace et al. 2016; Suárez Fernández et al. 2016]. Most of these contact-free scenarios require additional higher level sensors like depth cameras or motion sensors, either mounted to the quadrotor’s frame, permanently installed in controlled environments or attached to the user’s body [Calella et al. 2016; Gromov et al. 2019]. At close proximity, direct touch interaction may be required to directly interact with flying interfaces or hovering programmable matter [Gomes et al. 2016]. Also physical buttons have been attached to the frame of a quadrotor in order to enable interaction at immediate proximity [Rajappa et al. 2017].

Haptic communication is a branch of nonverbal communication which refers to the ways people and animals communicate and interact through the sense of touch. Since touch is the earliest sense to develop in the fetus [Coren et al. 1999] it may be the most intuitive way of interaction. Motivated by an interaction method that does not rely on additional hardware, we utilize a novel approach to control quadrotors: 3D touch interaction realized by virtual buttons. Quadrotors are equipped with an Inertial Measurement Unit (IMU), an electronic device that combines several instruments – usually accelerometers, gyroscopes and magnetometers – to measure the force, angular velocity and orientation of the UAV. The IMU is an essential part of the UAV, as it could not fly without it. In this work we utilize the accelerometer data of the built-in IMU for physical human-quadrotor interaction without intermediary devices. By dispensing with additional sensors, we are able to use miniature quadrotors that would otherwise not be able to fly due to the additional weight. With the ability to realize 3D touch interaction, for example, the behavior of hitting the trunk of a vehicle twice after unloading to signal the driver that he can drive off could be transferred to last-centimeter UAV delivery: The vehicle has been unloaded and is ready for departure. Communicating acknowledgement of a satisfactory done job by patting someone’s shoulder could also be transferred to flying robots, especially since it has been observed that people tend to interact with UAVs as with a person or a pet [Cauchard et al. 2015].

Another problem that we address by using data from existing sensors to keep the size of the robot to a minimum is the higher level of mental stress that people experience when interacting with mechanical-looking robots in such a close proximity [Walters et al. 2005], especially when they are of aerial nature [Acharya et al. 2017]. This results in an inhibition threshold caused by rotor blades and the overall mechanical appearance [Acharya et al. 2017; Yeh et al. 2017]. To counteract safety concerns about exposed rotor blades, they can be fitted with rotor guards resulting in a less mentally demanding interaction [Abtahi et al. 2017]. Furthermore, people tend to encounter quadrotors skeptically due to privacy or security reasons [Chang et al. 2017; Kim et al. 2016]. However, recent research shows positive evidence for having social or service aerial robots in human environments [Yeh et al. 2017], and several use cases of social drone companions for home [Karjalainen et al. 2017] and outdoor applications [Deng et al. 2018] have been proposed.

Apart from anthropomorphism, which is not always an option to make the robot’s actions interpretable for humans and thus lowering the inhibition threshold, research focuses on utilizing the trajectories of the quadrotor in order to convey intent or the internal state of the quadrotor. Affect and emotions are communicated through their trajectories [Cauchard et al. 2016; Sharma et al. 2013] and Disney’s principles of animation are applied in order to avoid surprises by anticipating the quadrotor’s motion [Lieser et al. 2020].

Socially expressive ground robots are well accepted because their movements can be well interpreted by humans. Music listening companions like the Sony Rolly [Kim et al. 2009] or the Keepon [Kozima et al. 2009] are used for research, therapy and entertainment. Research has shown that robots responding to musical experience causes humans to attribute human-like traits to the robot and to rate the robot as more similar to themselves [Hoffman and Vanunu 2013]. The shared experience of emotions evoked by music also strengthens the bond between humans [Koelsch 2013]. In this work we present a scenario to replicate exactly this emotional bond between a human and a quadrotor. Fig. 1 shows a short sequence of the proposed interaction scenario, where the user taps four times to the quadrotor’s frame and its response by flying a short trajectory matching the beat.

Instead of using a software to detect the song or the beat, this work aims at the interactive experience where human and quadrotor enjoy music together. This utilizes the proposed tactile immediate proximity interaction without intermediary devices in order to teach the quadrotor a song’s beats per minute (bpm), a common measurement of tempo in music. In response, the quadrotor performs a short dance and thus appears to be emotionally responding to music, making it a little less mechanic and intimidating in order to help lowering the inhibition threshold.

In Section 2, the method used to detect tap interaction in the acceleration data of the quadrotor’s IMU is presented. Section 3 gives a general outline on how trajectories are generated. In Section 4, the design and implementation of the proposed human-quadrotor interaction scenario MetroDrone, where user taps initiate dance moves, is explained. Finally, our work is summarized in Section 5 and an outlook on future research is given.
2 TAP DETECTION

Jerk, the third time derivative of the position vector or the change of an object’s acceleration with respect to time, is minimized in most trajectory generators to achieve smooth motion of a vehicle. Thus, without external forces acting on the vehicle, jerk is uncommon under usual flight conditions and is what the method described in the following detects in the quadrotor’s accelerometer data.

In order to reject noise, a low-pass filter is applied to the accelerometer data, i.e.
\[
\dddot{x}_k = a \dddot{x}_k + (1 - a) \dddot{x}_{k-1}, \quad a = \frac{dt}{\tau + dt},
\]
where \( \dddot{x}_k \) is the raw acceleration vector at discrete time \( k \) and \( \tau \) is an empirically determined time constant for two consecutive measurements being separated in time by \( dt \). The jerk vector \( \dddot{x}_k \) at time \( k \) is derived from the filtered acceleration vector by
\[
\dddot{x}_k = \frac{\dddot{x}_k - \dddot{x}_{k-1}}{dt}.
\]

Now, a potential tap is detected, when the length of the jerk vector exceeds a given threshold, i.e. if
\[
\|\dddot{x}_k\| > t_1.
\]

Since a jerk is followed by a jerk along the opposite direction that can also lead to a peak in noisy data, a time period \( t_2 \) for which the detection is suspended, prevents the algorithm from double detection.

If just the jerk and its direction are detected, the quadrotor is able to react, e.g., by moving away from the tap source. To identify multiple tap directions in order to respond differently, we introduce a virtual button metaphor by a set of \( M = 1 \) unit vectors given in the quadrotor’s body frame
\[
\mathcal{R} = \{r_i \in \mathbb{R}^3 : \|r_i\| = 1, i = 0, \ldots, M\}.
\]

An example of virtual buttons along with acceleration data caused by interacting with them is shown in Fig. 2.

![Figure 2: Schematic view of a quadrotor with four virtual buttons illustrated by the direction of their operation and the raw accelerometer data of multiple operations shown in corresponding colors visualizing the virtual button metaphor that defines a button by its normal vector.](image)

The operated virtual button is detected by maximizing the cosine similarity between the normalized, detected jerk direction vector \( \dddot{x}_k = \dddot{x}_k / \|\dddot{x}_k\| \) and the set of virtual buttons, i.e.
\[
\mathbf{r}_k = \arg \max_{\mathbf{r} \in \mathcal{R}} \mathbf{r} \cdot \dddot{x}_k.
\]

If in addition \( \mathbf{r}_k \cdot \dddot{x}_k \) exceeds a given similarity threshold \( t_3 \), a tap with intensity of \( \|\dddot{x}_k\| \) is detected along \( \mathbf{r}_k \). In the evaluation for virtual button presses the intensity does not matter, but if it is desired to react to the intensity of the button operation, e.g., by wider movements, the magnitude of the jerk can be taken into account.

3 TRAJECTORY GENERATION

We describe trajectories as piecewise polynomial functions (see for example [Corke 2013]). Quintic (fifth-order) polynomials are a common choice for trajectories, since their first and second time derivatives — velocity and acceleration — are continuous and thus result in smooth trajectories. Boundary conditions, such as position, velocity, acceleration and time, can easily be set. A scalar trajectory quintic polynomial and its first and second order time derivatives, given as

\[
\begin{align*}
  s(t) &= at^5 + bt^4 + ct^3 + dt^2 + et + f \\
  \dot{s}(t) &= 5at^4 + 4bt^3 + 3ct^2 + 2dt + e \\
  \ddot{s}(t) &= 20at^3 + 12bt^2 + 6ct + d
\end{align*}
\]

result in a linear system
\[
(2)
\]
with system matrix
\[
A_T c = s_{[0,T]}
\]

\[
A_T = \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 2 & 0 & 0 \\
5T^4 & 4T^3 & 3T^2 & 2T & 1 & 0 \\
20T^3 & 12T^2 & 6T & 2 & 0 & 0
\end{bmatrix}
\]

polynomial coefficient vector \( c = (a, b, c, d, e, f)^T \) and condition vector \( s_{[0,T]} = (s(0), s(T))^T \), which combines two boundary states \( s(t) = (s(t), \dot{s}(t), \ddot{s}(t)) \) with start time \( t = 0 \) and end time \( t = T \), the duration of the trajectory segment. For \( T \neq 0 \) the linear system (2) can be uniquely solved for the polynomial coefficient vector \( c \).

The different boundary states \( s_i(t_i) \) for prescribed times \( t_i \) can be considered as keyframes. A trajectory \( \mathcal{K} \) is then represented by a list of \( N + 1 \) of these keyframes, i.e. \( \mathcal{K} = (s_0(t_0), \ldots, s_N(t_N)) \) and thus consists of \( N \) trajectory segments \( s_{[i,i+1]} = (s_i(t_i), s_{i+1}(t_{i+1}))^T, i \in \{0, \ldots, N - 1\} \). The corresponding polynomial coefficients \( c_i, i \in \{0, \ldots, N-1\} \) for the respective trajectory segments are determined by solving (2) with \( T = (t_{i+1} - t_i) \), i.e.
\[
(3)
\]

\[
c_i = A_T^{-1} s_{[i,i+1]}
\]

which results in the polynomials \( s_i(t), i \in \{0, \ldots, N - 1\} \) for the \( N \) trajectory segments. These trajectory segments are then sampled at rate \( f_s \) by
\[
s_i(t_j) = \frac{j}{f_s} T, \quad j \in \{0, \ldots, f_s\}.
\]
For the multi-dimensional vector case, the piecewise polynomials are interpolated independently in each dimension and joined to obtain the final trajectory, that is passed to the controller.

4 METRODRONE

To demonstrate tactile drone interaction, we implemented MetroDrone, a quadrotor responding to interaction by flying different trajectories according to the identified button operated by the user and the detected beat. In the following we provide a short overview of our testbed and the hardware used. We also provide the details of generating a specific flight trajectory and further discuss tactile interaction with quadrotors.

4.1 Testbed

Our low-cost quadrotor testbed ICARUS [Lieser et al. 2020, 2017] was continuously expanded and improved over the past years and allows for safe test flights of arbitrary quadrotor models in a simulated environment but also controls real quadrotors. It employs a predictor-corrector estimator similar to the one described in [Lupashin et al. 2014]. Noise from pose estimation is filtered and the quadrotors’ states are latency-compensated by the implemented model of quadrotor dynamics summarized in [Lupashin et al. 2014; Michael et al. 2010]. The control loop runs at a frequency of 100 Hz. The hover controller used was presented in [Mellinger and Kumar 2011]. For trajectory control a Model-Predictive Controller (MPC) with a discretization time step $dt = 0.1$ s and time horizon $t_h = 2$ s is used. The implementation is based on open-source code [Falanga et al. 2018] and uses ACADO to set up the optimization problem and qpOASES to solve it. While the model parameters remain unchanged, the controller was tuned by experiment and varies from the original implementation: Costs for the horizontal and vertical position errors were left at 200 and 500 respectively, the costs for the attitude error were reduced from 50 to 25 and the costs for the velocity error from 10 to 4. The costs for the thrust, roll and pitch inputs were left at 1 and the costs for the yaw input was lowered from 1 to 0.1. The monocular pose estimation system [Tjaden 2019] multicasts UDP messages that consist of a six degrees of freedom pose and a timestamp. The estimator receives position and attitude from the tracking marker and an unmodified platform for comparison in the background. With this modification, the motor mounts expose to the top and are used as interaction points.

4.2 Tactile Interaction

In the provided example, interaction is restricted to taps that lie within the horizontal plane of the quadrotor. External, vertical forces applied to points that do not lie in the center of mass of the quadrotor result in tilting the thrust vector and thus attitude destabilization. As an example, applying a downward force to an arm of the frame would result in the quadrotor moving towards the user’s finger, which is of course not desirable. For MetroDrone, we propose and implement four virtual buttons in the horizontal plane of the quadrotor as shown in Fig. 2, along with the according, unfiltered accelerometer data, this interaction technique is based on. These four buttons are applicable to most quadrotors, but buttons can be defined virtually anywhere, provided a suitable frame or housing is available.

For jerk detection, the low-pass filter time constant $\tau = 0.0025$ and the jerk peak threshold of $t_1 = 400$ m/s$^3$ were chosen by experiment. A detection suspension threshold of $t_2 = 0.2$ s allows the algorithm to detect beats of up to 300 bpm. The cosine similarity threshold was set to $t_3 = \cos (\pi/8)$. The implemented beat detector collects the timestamps of detected taps (jerk value above peak threshold $t_1$) along the four specified normal directions that point toward the quadrotor’s center of mass. The differences between consecutive timestamps are converted into beats per minute. If the standard deviation of the bpm values within a discrete time window is below a given threshold, the beat detector initiates one of the four dance trajectories shown in Fig. 4. For the examples, we chose a standard deviation of 3 bpm and a window size of four to match 4/4 time signatures, where the first number corresponds to the number of beats in a measure in musical notation. Four user taps thus correspond to the four beats in a measure, and the dance trajectory is then initiated in time for the beginning of the following measure.

A basic, scalar periodic dance trajectory $X_0$ that implements a side-to-side motion at $f = 2$ Hz (120 bpm) can be seen in Fig. 5 and is constructed from the keyframes

$$s_i(t_i) = \begin{pmatrix} (-1)^i \cdot d \\ 0 \\ 0 \end{pmatrix}, \quad t_i = \frac{i}{f}, \quad i \in \{0, \ldots, N\}.$$

with displacement $d = 0.1$ m.
The according interaction sequence can be seen in Fig. 1. For more in-depth work on synchronizing quadrotors to music see [Schöllig et al. 2010].

Since the quadrotor should come to rest at the keyframes to emphasize the beat, velocity and acceleration are set to zero. The trajectory consists of $N = 8$ segments in order to be a multiple of the 4/4 time signature and is sampled at a rate of $f_s = 100$ Hz to match the update rate of the flight controller. The other trajectories shown in Fig. 4 and their projections (Fig. 5) are constructed similarly by using different keyframes.

As the fourth tap is detected, the trajectory is generated and executed just in time for the next beat. The time trajectory calculation takes, is subtracted so synchronize the start of the trajectory. The according interaction sequence can be seen in Fig. 1. For more in-depth work on synchronizing quadrotors to music see [Schöllig et al. 2010].

## 5 CONCLUSION & FUTURE WORK

In this work, we presented a novel technique for human-quadrotor interaction based on the basic human instinct, touch. We demonstrated how we refrain from the use of additional intermediary devices by resorting to the accelerometer data of a quadrotor’s built-in IMU and provided a method to detect user tap interaction based on peaks in the signal. In addition, we have introduced virtual buttons and explained how to distinguish them. As an example of implementation, we presented MetroDrone, a quadrotor responding to recurring user taps to its frame by initiating different dance trajectories. This way we provided a possible method to lower the inhibition threshold in human-quadrotor interaction by playfully creating a bond between human and robot through shared musical experience. In this interaction scenario physical communication is desired. But at immediate proximity, where sensors that are commonly used for user interaction fail, touch interaction may even be necessary, e.g., as a safety fallback measure to stop a quadrotor from further approaching. Although people tend to feel safer when a quadrotor is equipped with rotor guards, we have deliberately refrained from such features in order to emphasize our minimalist approach. Furthermore, any additional weight considerably reduces the flight time of a miniature-sized quadrotor. Because of the minimalist approach, which repurposed the IMU instead of adding physical buttons or capacitive sensors, the presented method can be implemented on almost any quadrotor system.

The presented interaction possibilities provide the proof of concept for basic physical interaction with quadrotors. Future work includes further elaboration of the proposed interaction by, e.g., incorporating tap intensities. This way the quadrotor could respond accordingly with wider or more reserved movements. Furthermore, adding a spherical housing would provide a sense of safety for the user and at the same time overcome the limitation of virtual buttons restricted to the horizontal plane allowing for arbitrary tap directions. The beat detection method could be extended to work with different time signatures and to learn rhythms and thus would enable a more comprehensive experience. It would also be conceivable to identify the playing song via appropriate services and to query databases for the genre in order to generate genre-specific dance trajectories. In addition to the implemented directional tap interaction, which is limited to taps towards the center of mass of the quadrotor, MetroDrone could be further improved by implementing the detection of rotational forces applied by the user based on data of the onboard IMU. With the help of this electronic device, we have taught the robot a sense of touch. Another topic we are currently working on, is to provide it with the sense of hearing by detecting quasiperiodic waves resulting from direct exposure of the quadrotor to a music source.