Robotica: (2020) volume 38, pp. 2189–2203. © Cambridge University Press 2020
doi:10.1017/S026357472000003X

User–Robot Interaction for Safe Navigation of a Quadrotor

L. F. Sanchez†‡, H. Abaunza† and P. Castillo†*

†Sorbonne Universités, Université de Technologie de Compiègne, CNRS, UMR 7253, Heudiasyc,
57 Avenue de Landshut, CS 60319, 60203 Compiègne cedex, France
E-mail: habaunza@hds.utc.fr
‡Instituto Tecnológico y de Estudios Superiores de Monterrey, Mexico
E-mail: luisanch@etu.utc.fr

(Accepted December 27, 2019. First published online: January 29, 2020)

SUMMARY
This paper introduces an intuitive and safe command approach for a quadrotor, where inertial and muscular gestures are used for semi-autonomous flight. A bracelet composed of gyroscopes, accelerometers, and electromyographic sensors is used to detect user gestures, then an algorithm is proposed to interpret the signals as flight commands. The main goal is to provide a wearable, easy-to-handle human–machine interface for users a to safely command this kind of vehicles, even for inexpert operators. Safety measures are incorporated in the scheme to further enhance the user's experience. Experimental tests are performed to validate the proposal.

KEYWORDS: Quadrotor; Quaternion; Gesture command; Unmanned aerial vehicles; User–Robot interaction.

1. Introduction

Unmanned aerial vehicles (UAVs) have become increasingly popular in recent years due to their versatility, agility, and their overall capability of performing a wide array of tasks, which include surveillance, film shooting, or search and rescue missions. However, more applications of these systems are being developed at an increasing rate, including complex task such as payload manipulation, exploring unstructured environments, and directly interacting with humans and other robots. The growing complexity of modern tasks that involve UAVs will require more advanced techniques to ensure the safety of surrounding humans, while avoiding work overload on operators.

Some works have explored techniques that safely interact with aerial vehicles in more intuitive ways, for example, a multimodal ground control station for UAVs composed of voice commands, tactile displays, gesture tracking, and haptic devices was introduced in ref. [1], whereas ref. [2] employed camera arrays, sound signals, graphical user interfaces, and an inertial joystick to control a group of three quadrotors. These works provide users with a more complete feedback of information, which enhances the situational awareness during missions; however, this does not directly change the commands given to the UAVs.

Meanwhile, methods for command-based voice teleoperation of mobile robots have also been explored, for instance in ref. [3], where a speech recognition system was developed over an acoustic voice model to identify a set of predefined commands to fly a quadrotor. Another speech-based algorithm for robot control was introduced in ref. [4], where a learning technique was added for adapting to user preferences and capabilities on the go.

Furthermore, since an important part of human communication comes from gestures, gesture recognition technologies have been also explored for robot control, providing more intuitive ways to interact with robotic systems. An early multimodal approach for a gestural human–machine interface

* Corresponding author. E-mail: castillo@hds.utc.fr

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

for disabled users to remotely manipulate familiar objects in unstructured environments is presented in ref. [5]. In ref. [6], researchers used inertial handheld sensors to identify arm gestures by measuring Euler angles sequences, which were then classified as user commands for flying a hypothetical UAV in ref. [7], whereas ref. [8] employed an armband to measure muscular gestures that trigger movements on a group of simulated agents.

Visual-based gestural interface for commanding drones was explored in ref. [9], where camera systems were used to track user gestures that were then interpreted as translational commands of a quadrotor. In ref. [10], frame-by-frame image processing and learning algorithms were used to identify body gestures made by the user, which were also interpreted as flight commands for a UAV. Facial recognition software has also been explored for engaging interaction with quadrotors, such that hand gestures can then be captured by cameras and used to command the vehicles.[11]

More recent proposals seek to exploit the capabilities of readily available devices like the *Microsoft Kinect* for recognizing gestures that define position references for a quadrotor.[12] Ref. [13] proposed a fuzzy algorithm to control the velocity of a quadrotor, following arm movements performed by a user. However, these kind of devices are very dependent on environmental conditions such as light intensity, or presence of other humans in the scenario, even if they provide an entertaining way to interact with UAVs, their actual implementation on quadrotor operation results unpractical.

Wearable and electromyographic (EMG) devices present a completely different approach for gesture recognition methods and viable control schemes for robotics applications. The use of noninvasive EMG techniques for commanding simulated robots has been explored by some researchers,[14] and their potential application in different fields including medical human–machine interfaces has also been studied,[15] proving the viability of this kind of interaction approaches. Some works have employed devices that combine EMG sensors with inertial measurement units (IMUs) in a wearable armband to track different kinds of human gestures.[16] In ref. [17], the authors used such devices to recognize hand postures using pattern recognition strategies. In ref. [18], an armband was used along other sensors to control a wheelchair by tracking the gestures of its user.

Applications of wearable motion detection sensors for aerial vehicles have been studied by some recent works. For instance, ref. [19] uses a hand motion sensor along other devices to detect gestures, interpreted as high-level commands for a quadrotor. Ref. [20] used the inertial sensor signals of a wearable smartwatch as attitude references for a commercial quadrotor. However, most of these works propose limited solutions to this problem, mainly using *out of the shelf* products with default settings as in refs. [21–25], which reduce the flexibility of the proposal. Furthermore, safety measures for limiting hazardous gestures are not considered, compromising the safety of operators, specially for novice users.

In this work, we propose an intuitive, user friendly, and portable approach for the navigation of a quadrotor based on the aforementioned capabilities of wearable armbands, equipped with EMG sensors and an IMU. Previous results from our team were presented in ref. [26], where a preliminary approach was proposed to pilot a quadrotor with a sensor-equipped armband. Our proposal interprets the user's arm orientation and muscular signals as flight commands for a drone and requires a minimal amount of gear for its operation. This scheme is composed of two parts; first, brute attitude reference signals will be acquired from the bracelet inertial sensors. These signals will be processed through nonlinear algorithms, such that they are attenuated in small values (commonly produced by involuntary movements), then amplified in the higher end of the operation range, and saturated and limited for greater and hazardous values such that the user has a more intuitive and safe interaction with the quadrotor. Second, the EMG sensors will be used along the inertial measurements to develop a gesture-based command, which seeks to exploit unconventional flight maneuvers of the drone. Finally, a quaternion-based control algorithm, which ensures the stability of the quadrotor dynamic equations, is used to track the proposed navigation references in real-time experiments. Furthermore, our gesture-based command improves the vehicle's usability due to its intuitive input for orientation commands and a mild learning curve for more complex tasks, requiring only a few minutes and a brief explanation to get a grasp of the interface.

The outline of this paper is the following: In Section 2, an approach to interpret inertial measurements from the user's forearm postures as safe quadrotor flight commands is introduced. Algorithms to interpret muscular gestures as flight instructions using EMG signals are presented in Section 3. A description of the safety measures put in place to assure the UAV's operation is explained in Section 4. A quaternion-based control strategy that commands the vehicle's attitude in closed loop

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

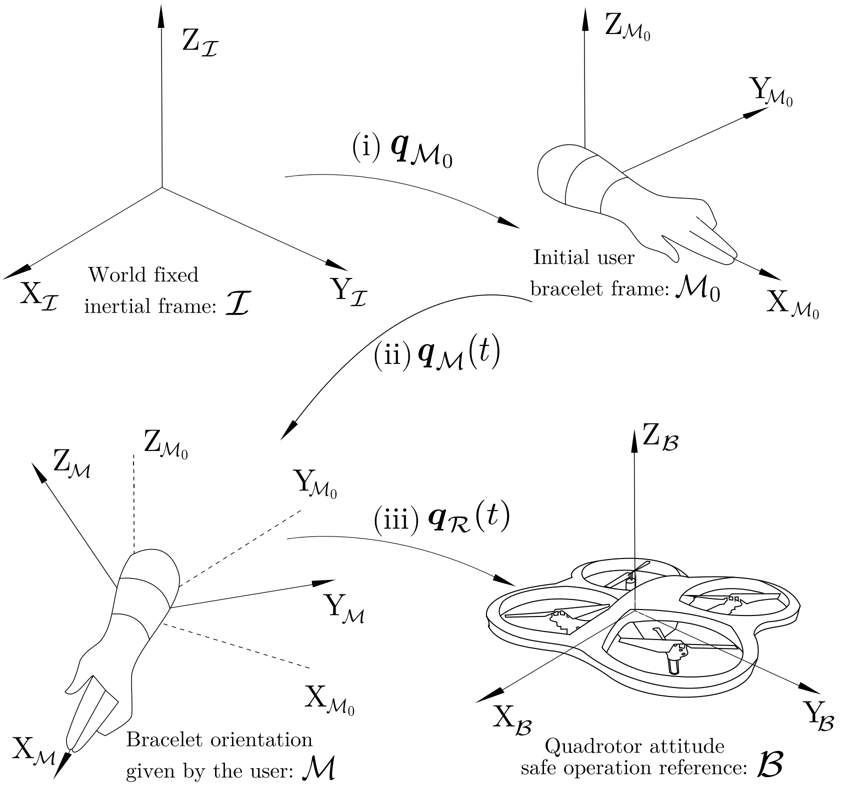
$Z_{\mathcal{I}}$
$X_{\mathcal{I}}$
$Y_{\mathcal{I}}$
World fixed inertial frame: $\mathcal{I}$
(i) $\boldsymbol{q}_{\mathcal{M}_0}$
$Z_{\mathcal{M}_0}$
$Y_{\mathcal{M}_0}$
$X_{\mathcal{M}_0}$
Initial user bracelet frame: $\mathcal{M}_0$
(ii) $\boldsymbol{q}_{\mathcal{M}}(t)$
$Z_{\mathcal{M}}$
$Z_{\mathcal{M}_0}$
$Y_{\mathcal{M}_0}$
$Y_{\mathcal{M}}$
$X_{\mathcal{M}_0}$
$X_{\mathcal{M}}$
Bracelet orientation given by the user: $\mathcal{M}$
(iii) $\boldsymbol{q}_{\mathcal{R}}(t)$
$Z_{\mathcal{B}}$
$X_{\mathcal{B}}$
$Y_{\mathcal{B}}$
Quadrotor attitude safe operation reference: $\mathcal{B}$

Fig. 1. Reference frames for quadrotor attitude reference.

is described in Section 5. Flight tests developed to demonstrate the performance of the proposed scheme are detailed in Section 6. Finally, conclusions and future works are discussed in Section 7.

2. User Commands for Attitude References

Consider a scenario where a quadrotor is piloted using arm gestures from its pilot with the objective of giving them an intuitive and safe flight experience. If the vehicle follows blindly all the movements from the input device, aggressive unintended rotations could arrive from an unexperienced user, which might be dangerous for the vehicle and people around it.

In order to address this problem, algorithms will be introduced to put constraints on the vehicle such that it operates under safe flight conditions.

2.1. Attitude gestures

Let four coordinate systems be represented as $\mathcal{I}$, $\mathcal{M}_0$, $\mathcal{M}$, and $\mathcal{B}$, which locate, respectively, at the global fixed frame, the initial bracelet pose, the rotating coordinates of the pilot's forearm, and the quadrotor body frame, see Fig. 1, such that the rotation between $\mathcal{I}$ and $\mathcal{M}_0$ is given by a constant quaternion $\boldsymbol{q}_{\mathcal{M}_0}$ defined as

$$\boldsymbol{q}_{\mathcal{M}_0} = \left\{ \cos\frac{\psi_0}{2}, \begin{bmatrix} 0 \\ 0 \\ \sin\frac{\psi_0}{2} \end{bmatrix} \right\} \otimes \left\{ \cos\frac{\theta_0}{2}, \begin{bmatrix} 0 \\ \sin\frac{\theta_0}{2} \\ 0 \end{bmatrix} \right\} \otimes \left\{ \cos\frac{\phi_0}{2}, \begin{bmatrix} \sin\frac{\phi_0}{2} \\ 0 \\ 0 \end{bmatrix} \right\}, \tag{1}$$

where θ_0, ϕ_0, and ψ_0 represent the forearm's initial pitch, roll, and yaw angles. Then, the rotation from $\mathcal{M}_0$ to $\mathcal{M}$ is defined by

$$\boldsymbol{q}_{\mathcal{M}}(t) = \left\{ \cos\frac{\psi_{\mathcal{M}}}{2}, \begin{bmatrix} 0 \\ 0 \\ \sin\frac{\psi_{\mathcal{M}}}{2} \end{bmatrix} \right\} \otimes \left\{ \cos\frac{\theta_{\mathcal{M}}}{2}, \begin{bmatrix} 0 \\ \sin\frac{\theta_{\mathcal{M}}}{2} \\ 0 \end{bmatrix} \right\} \otimes \left\{ \cos\frac{\phi_{\mathcal{M}}}{2}, \begin{bmatrix} \sin\frac{\phi_{\mathcal{M}}}{2} \\ 0 \\ 0 \end{bmatrix} \right\}, \tag{2}$$

where $\phi_{\mathcal{M}}$, $\theta_{\mathcal{M}}$, and $\psi_{\mathcal{M}}$ symbolize the time variant Euler angles from $\mathcal{M}_0$ to $\mathcal{M}$.

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

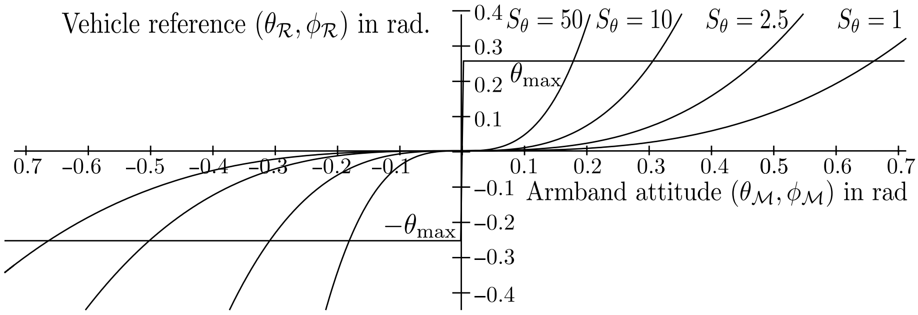

Fig. 2. From left to right, greater values $\theta_{\mathcal{M}}$ and $\phi_{\mathcal{M}}$ are required to reach tilt limit a as sensitivity values S_θ and S_ϕ decrease.

The quadrotor attitude commands are defined as a function of the angular variations experienced on the user's forearm $\boldsymbol{q}_{\mathcal{M}}(t)$, that is, if the user tilts his arm, the quadrotor should undergo a correspondent motion along the same axis. This allows the user to get a more natural feel on how the drone will behave since it will essentially mimic the user's arm orientation.

However, a direct correspondence between the armband orientation $\boldsymbol{q}_{\mathcal{M}}(t)$ and that one of the drone could result in undesired behaviors. Any person (specially inexperienced and nervous operators) is susceptible of doing involuntary small movements while bending their arm to interact with the quadrotor. Furthermore, excessive arm tilts could result in vertical poses, which would turn perpendicularly the drone, crashing it to the ground.

Therefore, it is necessary to implement some safety motion restrictions to prevent the drone from performing a risky rotations. Introducing a quaternion reference as

$$\boldsymbol{q}_{\mathcal{R}}(t) = \left\{ \cos\frac{\psi_{\mathcal{R}}}{2}, \begin{bmatrix} 0 \\ 0 \\ \sin\frac{\psi_{\mathcal{R}}}{2} \end{bmatrix} \right\} \otimes \left\{ \cos\frac{\theta_{\mathcal{R}}}{2}, \begin{bmatrix} 0 \\ \sin\frac{\theta_{\mathcal{R}}}{2} \\ 0 \end{bmatrix} \right\} \otimes \left\{ \cos\frac{\phi_{\mathcal{R}}}{2}, \begin{bmatrix} \sin\frac{\phi_{\mathcal{R}}}{2} \\ 0 \\ 0 \end{bmatrix} \right\}, \tag{3}$$

the reference pitch, roll, and yaw angles can be designed and follow a cube relation with the original orientation values, such that, for small angles, the resulting orientation will be attenuated, while for greater inclinations, the vehicle will perform more reactive and agile maneuvers. Hence, their values can be computed according to

$$\theta_{\mathcal{R}}(\theta_{\mathcal{M}}) = \begin{cases} \theta_{\mathcal{M}}^3 S_\theta \; ; \; \theta_{\mathcal{M}} \leq \sqrt[3]{\frac{\theta_{\max}}{S_\theta}} \\ \theta_{\max} \quad ; \; \theta_{\mathcal{M}} > \sqrt[3]{\frac{\theta_{\max}}{S_\theta}} \end{cases}, \tag{4}$$

$$\phi_{\mathcal{R}}(\phi_{\mathcal{M}}) = \begin{cases} \phi_{\mathcal{M}}^3 S_\phi \; ; \; \phi_{\mathcal{M}} \leq \sqrt[3]{\frac{\phi_{\max}}{S_\phi}} \\ \phi_{\max} \quad ; \; \phi_{\mathcal{M}} > \sqrt[3]{\frac{\phi_{\max}}{S_\phi}} \end{cases}, \tag{5}$$

$$\psi_{\mathcal{R}}(\psi_{\mathcal{M}}) = \psi_{\mathcal{M}}, \tag{6}$$

being $\theta_{\max}$ and $\phi_{\max}$ the maximum tilt allowed on the reference angles (θ_R and ϕ_R), respectively, while S_θ, $S_\phi < 1$ are tunning parameters that improve the sensitivity of the command along them and must be selected such that

$$\sqrt[3]{\frac{\phi_{\max}}{S_\phi}}, \; \sqrt[3]{\frac{\theta_{\max}}{S_\theta}} < 1. \tag{7}$$

Note from (4) and (5) that cubic relationships are followed by the attitude reference with respect to the forearm rotation, nevertheless (7) implies that the attitude gesture signals are cubically attenuated, see Fig. 2.

These cubic expressions also mean that higher values for these parameters translate into the quadrotor performing more aggressive movements in response to the user's arm motion; while lower

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

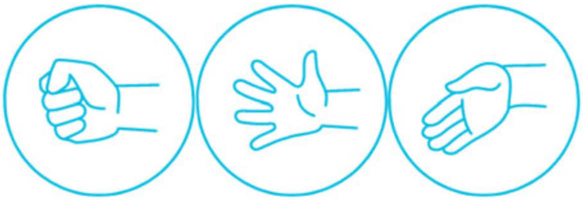

Fig. 3. Examples of poses, respectively: a fist, fingers spread, folded wrist. Image via https://support.getmyo.com.

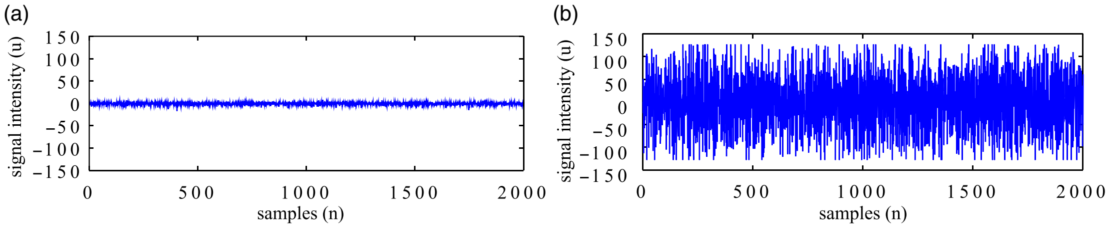

Fig. 4. Resting and tense muscular EMG measurements for one sensor. (a) A single EMG sensor for a resting arm. (b) EMG signal when the muscle is tighten.

values will cause the quadrotor to perform slower in response to the same arm motion, the tilting will also be diminished as the sensitivity values approach zero until the drone remains totally unresponsive.

The range of rotation and the velocity at which tilt limits $\theta_{\max}$ or $\phi_{\max}$ are reached, depend on the sensitivity parameters S_θ and S_ϕ, as can be seen in Fig. 2. These parameters are tuned to enhance the user's comfort and experience: advanced users might prefer higher sensitivity values for sharper, faster maneuvers, whereas beginners may prefer lower values for slower, easier to follow movements.

3. Electrical References from Skeletal Muscles

For this work, a pose will be defined as any combination in the fingers and wrist disposition, see Fig. 3.

Consider the armband contains eight EMG sensors, such that each one detects electrical activity in the superficial layers of the arm muscles and translates its intensity into an 8-bit value ranging from −127 to 128, see Fig. 4. The combination and interpretation of these measurements can be used to detect five manufacturer predefined poses (wave in, wave out, double tap, fist, rest, and spread fingers), or be used as raw data as well.

Some of the predefined poses can be, at times, triggered incorrectly, mainly when the armband is passed from one user to another. The *double-tap* predefined pose, which is performed by tapping the middle finger and thumb in quick succession, was deemed as the most reliable of these; since it was always triggered when the user intended to. A second pose was required for this proposal, having ran out of reliable predefined poses a *custom pose* was proposed.

3.1. Custom gesture

The custom pose in Fig. 5 works by measuring and adding up the electrical raw muscle activity in the user's arm or fingers collected from all eight sensors of the array. Once it surpasses a certain threshold, it is checked to ascertain the intentionality of the command and then the signal is sent, this is the command used to set the custom rotation reference, which is mostly used when the user drifts too much apart from his starting position.

The raw measurement of the total muscular activity in the subject's arm can be computed by

$$\sigma_{EMG}(t) = \sum_{i=1}^{8} |EMG_i(t)|, \tag{8}$$

where $EMG_i(t)$ is the measure for each electrode.

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

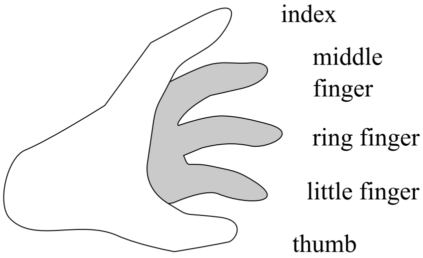

Fig. 5. Example of the custom pose, fingers are to be held tightly in place as if firmly holding a spherical object.

Let b symbolize the muscular activity threshold and

$$f_1(t) = \frac{1 + \text{sign}(\sigma_{EMG}(t) - b)}{2} \tag{9}$$

be the function that verifies whether or not the raw muscular activity in the user's arm has surpassed threshold b.

Define c and d as parameters that will verify the intentionality of the gesture. c is defined as a whole number greater than zero, and d as a rational number such that $0 < d < 1$, by checking c number of samples from (9) at different instants of time, that is

$$f_2(t) = \sum_{i=0}^{c} f_1(t - i) \tag{10}$$

Besides, parameter d effectively refers to the least amount of times that threshold b has to be surpassed within c samples in order to be considered as a valid pose. This can be explained in the following:

$$f_3(t) = \text{sign}(f_2(t) - dc), \tag{11}$$

signifying $f_3(t) = 1$ that the pose has been successfully triggered and $f_3(t) = -1$ or $f_3(t) = 0$ means that the pose has not been triggered.

3.2. Input based on a gesture sequence

Once the bracelet signals are processed, the next goal is to apply them for controlling the aerial robot. The transition between the user's arm orientation and drone attitude reference is relatively natural; however, if the user wants to perform more specific tasks, it is necessary to device a different set of gestures.

At Heudiasyc lab, in collaboration with CINVESTAV Saltillo, a nonlinear controller was developed to perform quadrotor pirouettes in closed loop.[27] The aim in the arm command algorithm is to indicate the drone when to perform acrobatic maneuvers using the user gestures.

A sequence triggered by a *double-tap* pose was chosen, consisting of two stages: After the double-tap gesture, the user has L_{t1} milliseconds to either perform the follow-up for a single, double, or triple quadrotor pirouette, see Fig. 6.

If the user begins a gesture sequence during L_{t1} an additional L_{t2} milliseconds are provided during which the rest of the sequence is to be performed, this amounts $L_{t1} + L_{t2}$ milliseconds during which the user has to perform the whole sequence.

The quadrotor will perform the intended mission immediately thereafter, or, it will remain hovering if the sequence fails to be performed correctly.

The method used to define each sequence is the following: first, since the gesture is performed by tilting the forearm upward and downward a vertical boundary $b_{\theta_{\mathcal{M}}}$ was set around the initial frame $\mathcal{M}_0$. Gestures can then be identified by the way the user's arm orientation transitions between these regions by the following

$$f_{G1}(t) = \frac{1 + \text{sign}(\theta_{\mathcal{M}}(t) - b_{\theta_{\mathcal{M}}})}{2} - \frac{1 - \text{sign}(\theta_{\mathcal{M}}(t) + b_{\theta_{\mathcal{M}}})}{2}. \tag{12}$$

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

Table I. Gesture assignment to detection function.

ρ value	**Gesture**	**Number of loops**
$\rho = 0$	No gesture detected	One loop
$\rho = 2$	Up, down, up	Two loops
$\rho = -2$	Down, up, down	Three loops
Otherwise	Unassigned gesture	One loop

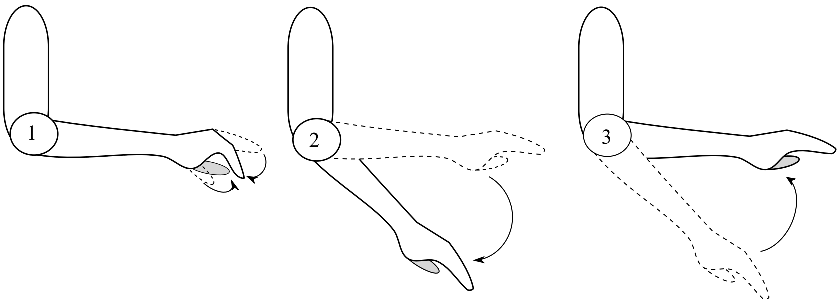

Fig. 6. Example of the gesture sequence to be performed in order to trigger a triple loop command.

Define a scalar function, which depends on discrete evaluations of (12) as

$$\rho = \sum_{i=0}^{2} f_{G1}(t_i) + f_{G1}(t_{i-1}), \tag{13}$$

where t_i is the time instant where the value of $f_{G2}(t)$ changes, marking the transition between any two adjacent poses. The value of (13) is finally assigned to specific tasks for the drone according to Table I:

Other missions can also be performed by these gestures if required.

4. Safe Human–UAV Interaction

Human error is always a possibility in semi-autonomous navigation. In order to decrease its impact, safety measures were developed to allow more intuitive commands and a more enjoyable user experience. These also ensure safety by avoiding unintended maneuvers that might result in crashes.

4.1. IMU-based safety measures

An additional orientation-based safety measure is implemented by defining two boundaries ϕ_{lim} and θ_{lim} in the event when the user would drop his arm. Should this happen, the orientation reference for $\theta_{\mathcal{R}}$ and $\phi_{\mathcal{R}}$ will be set to 0 to ensure safe hovering, this is achieved by enhancing (4), (5), and (6) as

$$\theta_{\mathcal{R}}(\theta_{\mathcal{M}}) = \begin{cases} \theta_{\mathcal{M}}^3 S_\theta \; ; & \theta_{\mathcal{M}} \leq \sqrt[3]{\frac{\theta_{\max}}{S_\theta}} \\ \theta_{\max} \quad ; & \sqrt[3]{\frac{\theta_{\max}}{S_\theta}} < \theta_{\mathcal{M}} \leq \theta_{\text{lim}} \\ 0 \quad ; & \theta_{\text{lim}} < \theta_{\mathcal{M}} \end{cases}, \tag{14}$$

$$\phi_{\mathcal{R}}(\phi_{\mathcal{M}}) = \begin{cases} \phi_{\mathcal{M}}^3 S_\theta \; ; & \phi_{\mathcal{M}} \leq \sqrt[3]{\frac{\phi_{\max}}{S_\phi}} \\ \phi_{\max} \quad ; & \sqrt[3]{\frac{\phi_{\max}}{S_\phi}} < \phi_{\mathcal{M}} \leq \phi_{\text{lim}} \\ 0 \quad ; & \phi_{\text{lim}} < \phi_{\mathcal{M}} \end{cases}, \tag{15}$$

$$\psi_{\mathcal{R}}(\psi_{\mathcal{M}}) = \psi_{\mathcal{M}}, \tag{16}$$

where ϕ_{lim} and θ_{lim} are thresholds where the user's arm is considered to be beyond the acceptable, see Figs. 7 and 8.

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

Table II. Predefined poses and correspondent action.

$\mathcal{P}$ **numeric value**	**Pose name**	**Pose action**
0	Resting position	Neutral reference
1	Fist	Unused
2	Wave in	Unused
3	Wave out	Unused
4	Spread fingers	Unused
5	Double tap	Unlock device, trigger loop
6	Unknown	Lock device if not worn

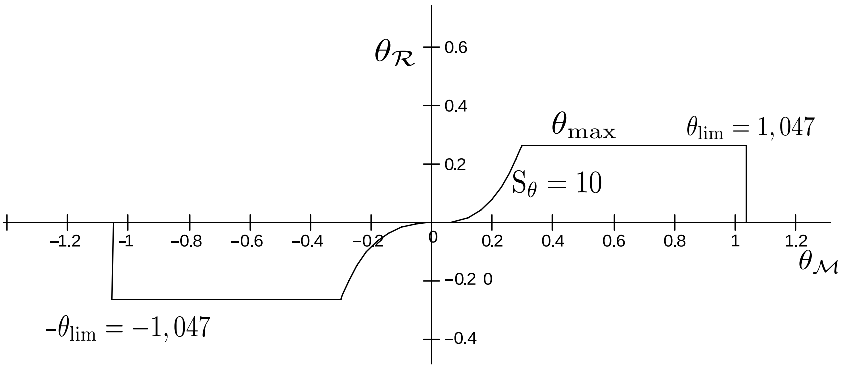

Fig. 7. $\phi_{\mathcal{R}}$ with safety limit put in 1.047 rad (60°).

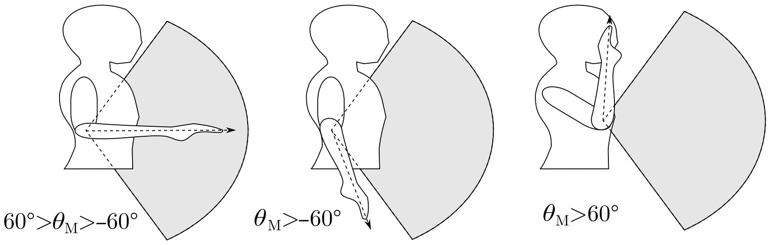
60°>θ_{M}>-60°
θ_{M}>-60°
θ_{M}>60°

Fig. 8. $\theta_{\mathcal{M}}$ orientation references considered within the accepted range.

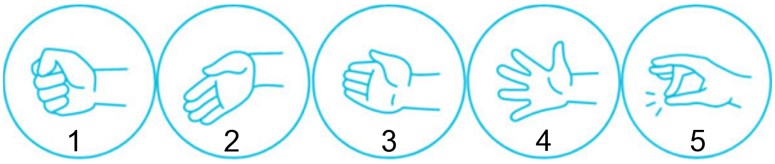

Fig. 9. Default preprogrammed poses. From left to right poses 1 through 5, respectively. Image via https://support.getmyo.com.

4.2. EMG-based safety measures

The EMG capabilities of the device were also used to make it safer. As previously mentioned, there are 7 preprogrammed muscular poses p each assigned a default numeric value, see Fig. 9 and Table II.

Furthermore, the bracelet can detect whether or not is it being worn by a user, and it will remain locked $l(t)$ until the unlocking pose is performed (double tap pose while its locked). For this proposal: 6 (unknown) will determine whether or not the bracelet is being worn, 5 (double tap) will be used as the unlocking pose (as well as begin loop command), and 0 (resting position) will be used a neutral reference. Therefore, the chosen poses for our application are: $\mathcal{P} = \{0, 5, 6\}$.

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

Once communication between the ground station and the bracelet is enabled, the bracelet must be unlocked by performing the double tap pose according to (17).

$$l(t_0) = \begin{cases} 0 & \text{for} \quad \mathcal{P} = 5 \\ 1 & \text{for} \quad \mathcal{P} \neq 5 \end{cases}, \qquad l(t) = \begin{cases} 0 & \text{for} \quad \mathcal{P} \neq 6 \\ l(t_0) & \text{for} \quad \mathcal{P} = 6 \end{cases}. \tag{17}$$

If the user suddenly removes the bracelet, data transmission is stopped, then θ_R and ϕ_R angles are set to zero while ψ_R remains in the last reference recorded such that

$$\theta_{\mathcal{R}}(t) = \begin{cases} l(t)\theta_{\mathcal{M}}^3 S_\theta \ ; & \theta_{\mathcal{M}} \leq \sqrt[3]{\frac{\theta_{\max}}{S_\theta}} \\ l(t)\theta_{\max} \quad ; & \sqrt[3]{\frac{\theta_{\max}}{S_\theta}} < \theta_{\mathcal{M}} \leq \theta_{\lim} \\ 0 \quad ; & \theta_{\lim} < \theta_{\mathcal{M}} \end{cases}, \tag{18}$$

$$\phi_{\mathcal{R}}(t) = \begin{cases} l(t)\phi_{\mathcal{M}}^3 S_\theta \ ; & \phi_{\mathcal{M}} \leq \sqrt[3]{\frac{\phi_{\max}}{S_\phi}} \\ l(t)\phi_{\max} \quad ; & \sqrt[3]{\frac{\phi_{\max}}{S_\phi}} < \phi_{\mathcal{M}} \leq \phi_{\lim} \\ 0 \quad ; & \phi_{\lim} < \phi_{\mathcal{M}} \end{cases}, \tag{19}$$

$$\psi_{\mathcal{R}}(t) = l(t)\psi_{\mathcal{M}} + (1 - l(t))\psi_{\mathcal{M}}(t-1). \tag{20}$$

4.3. Additional safety measures

As a required safety measure to ensure the laboratory material integrity, a joystick command device is always kept at hand by a more experienced operator who can, at a moments notice, override the armband navigation commands in order to deliver the quadrotor from hazard.

This safety layer is included by defining the actual reference quaternion $\boldsymbol{q}_d$, which will be introduced to an attitude controller in the next section.

$$\boldsymbol{q}_d(t) = \begin{cases} \boldsymbol{q}_{\text{joy}}, & \text{override on}, \\ \boldsymbol{q}_{\mathcal{R}}, & \text{override off}, \end{cases} \tag{21}$$

where $\boldsymbol{q}_{\text{joy}}$ represents a quaternion computed from a dual-joystick device, following remote piloting conventions.

5. Inner Quaternion Attitude Control

Quaternion control has been proved to be robust avoiding singularities points in quadrotor flights. An inner attitude control based on quaternions is used to control the quadrotor vehicle. The vehicle attitude dynamic model is considered as

$$\frac{d}{dt}\begin{bmatrix} \boldsymbol{q} \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{q} \otimes \omega \\ J^{-1}\ (\tau - \omega \times J\,\omega) \end{bmatrix} \tag{22}$$

where $\boldsymbol{q}$ describes the vehicle orientation, ω denotes the angular velocity, J introduces the inertia matrix with respect to the body-fixed frame, and τ is the torque vector caused by the action of the motors. The control law that stabilizes (22) to the desired reference is then defined as

$$\tau = -J(K_i\, 2\ln(\boldsymbol{q} \otimes \boldsymbol{q}_d^*) + K_i\,\omega) + \omega \times J\,\omega, \tag{23}$$

where $K_i \in \mathbf{R}^3$ denotes the control gains, and $\boldsymbol{q}_d$ defines the desired attitude quaternion representing the user references from (21). More details can be found in ref. [28]. The reference quaternion can be computed as

$$\boldsymbol{q}_d = \left\{q_{d0}, [q_{d1} \quad q_{d2} \quad q_{d3}]^T\right\} s_z, \tag{24}$$

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

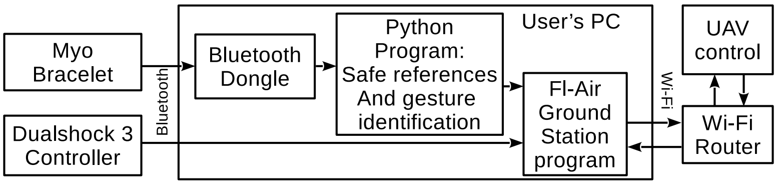

Fig. 10. Communication process involved in the armband-UAV interface.

with

$$\begin{aligned} q_{d0} &= \cos\left(\frac{\psi_d}{2}\right)\cos\left(\frac{\theta_d}{2}\right)\cos\left(\frac{\phi_d}{2}\right) + \sin\left(\frac{\psi_d}{2}\right)\sin\left(\frac{\theta_d}{2}\right)\sin\left(\frac{\phi_d}{2}\right),\\ q_{d1} &= \cos\left(\frac{\psi_d}{2}\right)\cos\left(\frac{\theta_d}{2}\right)\sin\left(\frac{\phi_d}{2}\right) - \sin\left(\frac{\psi_d}{2}\right)\sin\left(\frac{\theta_d}{2}\right)\cos\left(\frac{\phi_d}{2}\right),\\ q_{d2} &= \cos\left(\frac{\psi_d}{2}\right)\sin\left(\frac{\theta_d}{2}\right)\cos\left(\frac{\phi_d}{2}\right) + \sin\left(\frac{\psi_d}{2}\right)\cos\left(\frac{\theta_d}{2}\right)\sin\left(\frac{\phi_d}{2}\right),\\ q_{d3} &= \sin\left(\frac{\psi_d}{2}\right)\cos\left(\frac{\theta_d}{2}\right)\cos\left(\frac{\phi_d}{2}\right) - \cos\left(\frac{\psi_d}{2}\right)\sin\left(\frac{\theta_d}{2}\right)\sin\left(\frac{\phi_d}{2}\right), \end{aligned} \tag{25}$$

quaternions are known to be singularity free and continuous; however, discontinuities can be passed on to the control law by the original Euler representation of the armband, causing the drone to behave erratically. s_z represents a term that makes a sign change for $\boldsymbol{q}_d$ to avoid such discontinuities, and is defined as

$$s_z = \begin{cases} -1 \text{ if } & |\psi(t) - \psi(t - t_\delta)| > \pi, \\ 1 \ \text{ otherwise}, \end{cases} \tag{26}$$

where t_δ symbolizes the signal sampling period.

The quadrotor pirouette maneuvers are performed autonomously thanks to the quaternions-based control law developed in ref. [27].

6. Experimental Results

The internal control algorithm for quadrotor navigation was programmed using the Fl-AIR (Framework libre AIR) C++ libraries[29] and uploaded to a Parrot AR drone 2 with a completely re-flashed firmware, also based on the Fl-AIR framework.

The brute attitude commands were measured by a sensor-equipped bracelet.[30] This device is equipped with three main sensors: a gyroscope, an accelerometer, and an array of eight EMG sensors. The safety signal treatment and gesture interpretation algorithms were coded on Python2, with a communication layer based on the PyoConnect library.[31]

The armband communicates its sensor data via bluetooth into a ground station. This data is handled by the Python program which processes the attitude references and the EMG measures interpretation routines.

Data output from this algorithm is handed over to the internal UAV controller via User Datagram Protocol (UDP) using wireless communications. This program handles the commands for the drone attitude and contains the control laws needed for its operation, see Fig. 10.

6.1. Attitude safety restrictions

The first flight test consisted of controlling the vehicle attitude using arm gestures, at some moments, the user performed high inclinations that could result in dangerous movements if managed incorrectly. However, the proposed algorithm restricts such rotations by following (18) and (19), where the angular limits were set at $\theta_{\text{lim}} = \phi_{\text{lim}} = 15°$.

Figure 11 represents the quadrotor attitude behavior. The effects of the safety pitch and roll bounding, translated into quaternion references, can be appreciated in Fig. 11(b) and (c), while Fig. 11 indicates a complete rotation over the yaw angle by a continuous change of the q_3 value from 1 to -1.

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

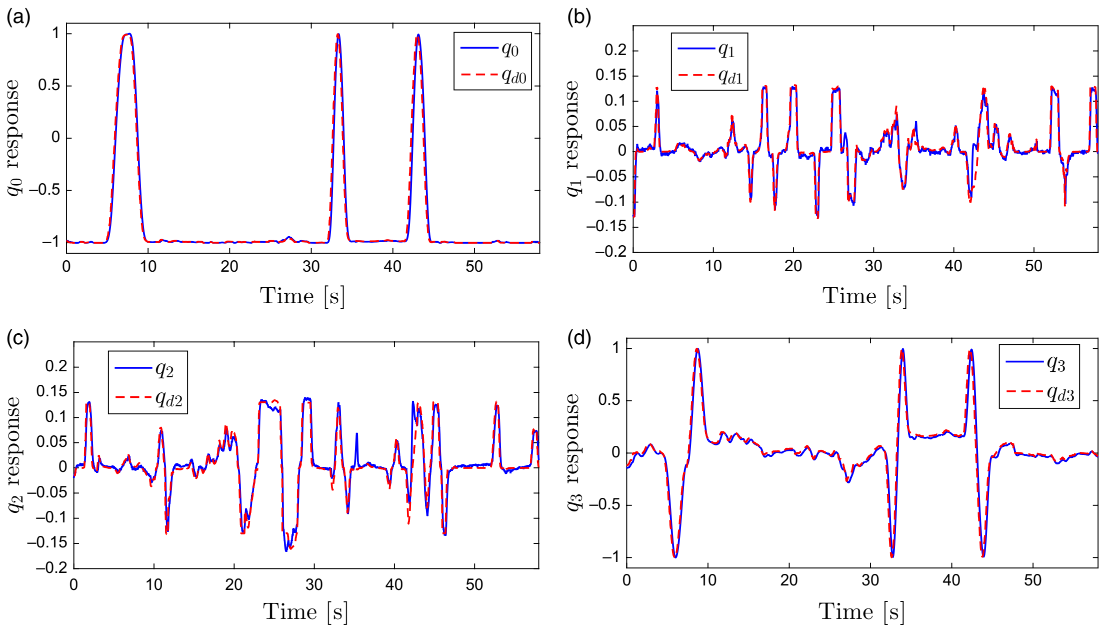

Fig. 11. Quadrotor attitude quaternion tracking. (a) First test q_0 quaternion component, switching its value from -1 to 1 each time a full 360° rotation is performed. (b) First test q_1, this represents the x component of the quaternion. (c) First test q_2, this represents the quaternion y component. (d) First test q_3, representing the vehicle rotation over its z axis.

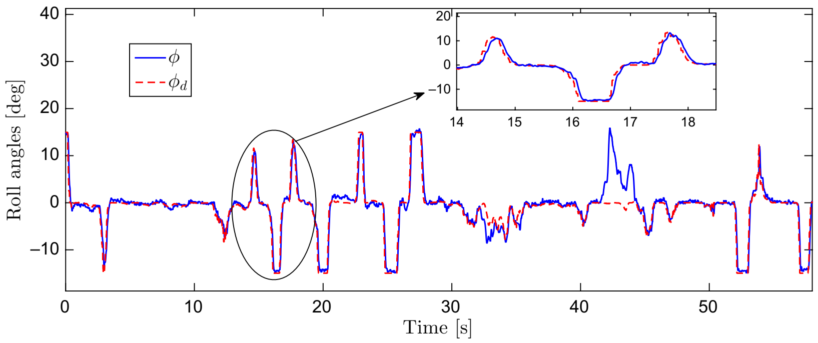

Fig. 12. Roll response using ϕ_d from the bracelet, reference signal is bounded to safe values between ±15°.

At the beginning and end of the test, rotations around the user's body axis were performed, resulting in multiple 360° yaw rotations, since the user movements are not restricted over the z axis, the given reference is tracked integrally.

To give a better illustration of the attitude behavior, the Euler representation for the same test can be observed in Figs. 12–14. Notice how the saturation occurs in θ_d and ϕ_d whenever the 15 degrees (0.261 radians) is reached. A close-up view of the pitch and roll signals is included to better illustrate such behavior.

Note from the detailed zoom on the pitch and roll signals that a smooth transition happens between near-zero poses and the maximal permitted attitude value. This behavior is an effect of the attenuation functions from (18) to (20).

Remark also that the desired values are thoroughly maintained very close to zero when the vehicle is desired to hover in a fixed position. This happens regardless of the operator's involuntary small movements, which occur due to natural oscillations of the human arm when trying to hold a fixed pose.

Discontinuities in the roll and pitch angles are never present due to the imposed constraints in the algorithms, as well as those from natural human anatomy. However, discontinuities in yaw movements using Euler representation appear whenever a full turn is performed, since the value of

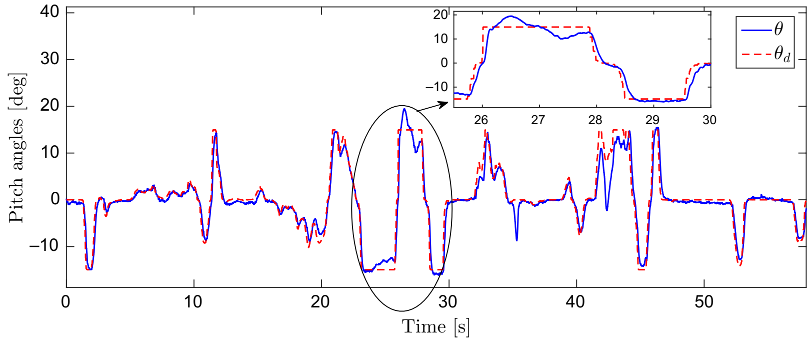

Fig. 13. Pitch response using θ_d from the bracelet, imposing restrictions at ±15° values.

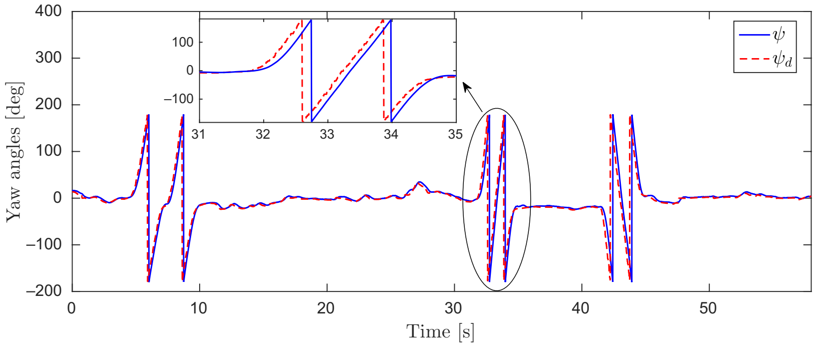

Fig. 14. Yaw response using ψ_d from the bracelet. Notice how discontinuities appear in yaw Euler representation when a full 360° turn is performed.

this angle suddenly changes from 180 to −180 deg, as can be seen in Fig. 14. However, when using quaternion representation, these discontinuities are not present, as seen in Fig. 11(d).

6.2. EMG commands

In the second test, the user performed muscular gestures corresponding to a triple-loop, a double-loop, and a single loop consequently, Fig. 15 illustrates attitude quaternion performance of this experiment.

In contrast, close-ups of the Euler angles equivalent responses of this experiment are depicted in Fig. 16; here, two undesired effects are revealed, since the pirouettes are performed over the y axis, Fig. 16(a) and (c) reveal a Gimball-lock effect, where multiple roll and yaw angles are possible when the pitch angle reaches a singular value. Figure 16(b) illustrates a triple pitch loop, passing through discontinuities three consecutive times.

A video showing the complete test can be watched at:
https://youtu.be/Qa_slursVvA

It is worthy of mention that, since this approach attenuates the effects of involuntary movements. Novice users (which are unaware of all the algorithms that run to provide this intuitive interaction) tend to require less time to grasp the movements required to interact with the vehicle, and also display less signs of mental fatigue and stress, compared to using a classical joystick.

7. Conclusion and Future Work

A safe and intuitive navigation approach for semi-autonomous flight of a quadrotor was proposed in this paper. The scheme is composed by two main components. First, a gesture-based attitude reference is computed to adapt orientation signals that ensure a safe operation of the quadrotor. This is achieved by distinguishing small involuntary movements from piloting gestures and limiting extreme rotations that could be hazardous.

Second, EMG gesture sequences were used to trigger commands for performing predefined tasks with the drone. Safety measures and customizable sensitivity parameters made the interface easier to

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

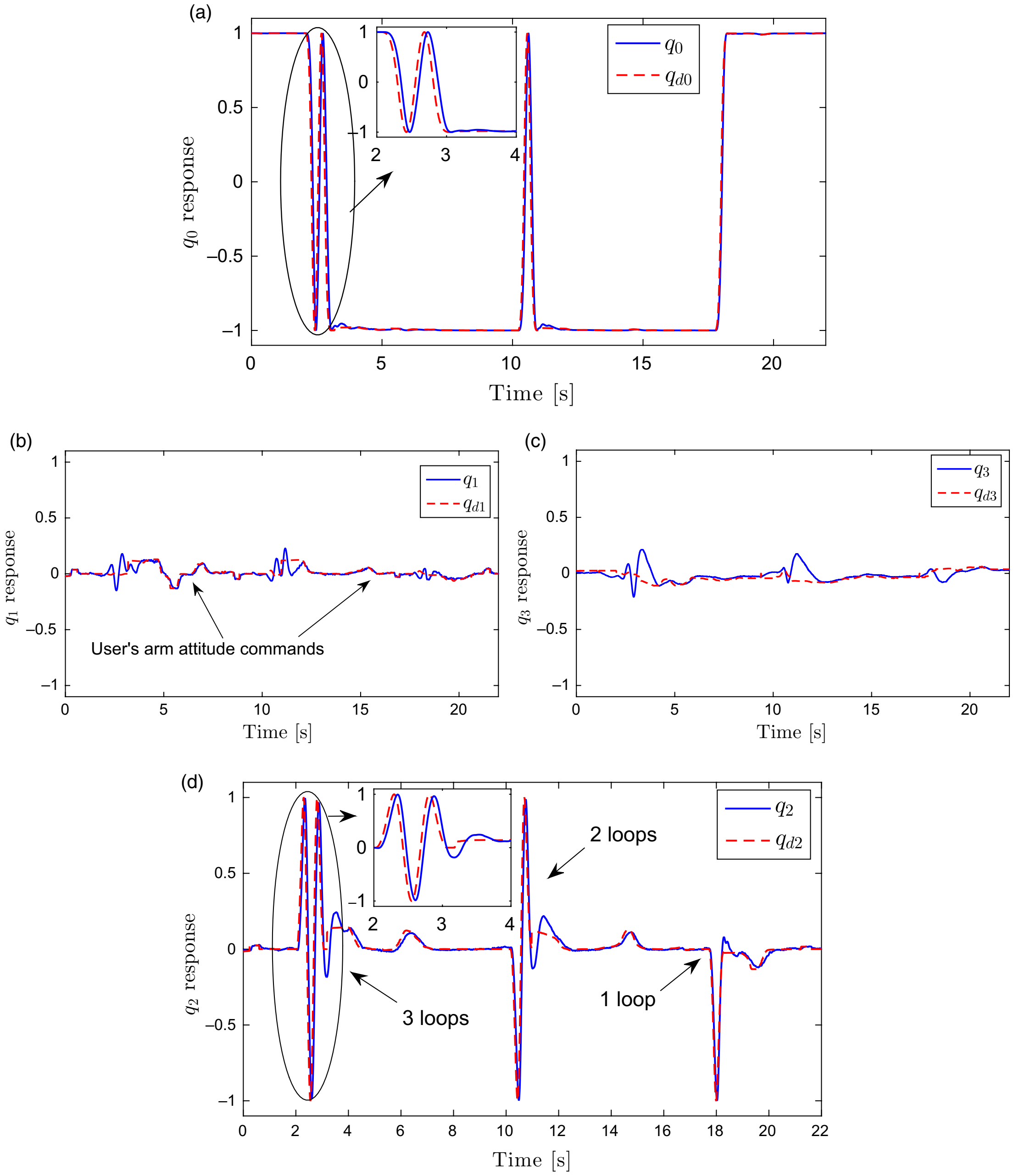

Fig. 15. Quadrotor attitude tracking during a looping experiment. (a) Notice how the sign of q_0 is inverted whenever a loop is performed. (b) Quaternion component q_1, following only the piloting references. (c) Quaternion component q_3, maintaining the bracelet reference. (d) q_2 component, sinusoidal oscillations indicate when a pirouette is performed.

handle for inexperienced users by reducing the learning curve and overall achieving a more intuitive command scheme.

The proposed algorithms were validated in experimental flight tests where a user commanded a quadrotor, tested the proposed safety measures, and performed a series of gestures, including multiple loop maneuevers.

Future work will consist of identified customized hand gestures using EMG signals that can be measured with this bracelet. A machine learning approach has been considered to identify a gestural profile for each individual user through computational intelligence. The aim will be to enable each operator to use the gestures that are more convenient for their personal preferences.

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press

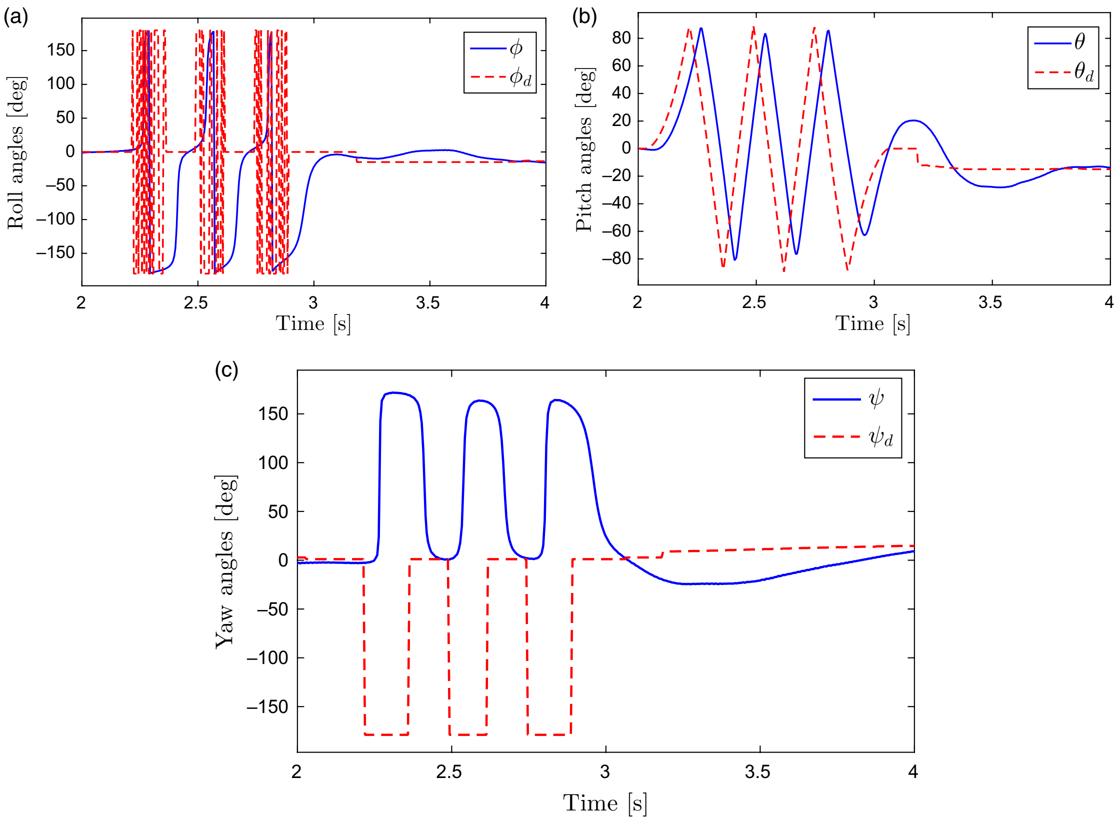

Fig. 16. Euler angles representation of the quadrotor attitude during looping tests. (a) Close up of the ϕ euler angle during the triple loop maneuver. Notice the discontinuities in ϕ and the Gimball lock in ϕ_d. (b) Close up of the θ angle triple loop maneuver, a triangular ramp behavior is the result of the quaternion to Euler angles conversion. (c) Close up of the ψ angle. Notice how ψ diverges from ψ_d due to the quaternion to Euler angles transformation.

Acknowledgments

This work was done in the context of a PhD thesis supported by the French M.E.S.R.I, and the Mexican CONACYT program for postgraduate scholarships. The authors would also like to thank the French National Network of Robotics Platforms (ROBOTEX, ANR-10-EQPX-44).

Supplementary Material

To view supplementary material for this article, please visit https://doi.org/10.1017/S026357472000003X.

References

1. I. Maza, F. Caballero, R. Molina, N. Peña and A. Ollero, “Multimodal interface technologies for uav ground control stations,” *J. Intell. Robot. Syst.* **57**(1–4), 371–391 (2010).
2. M. Orsag, T. Haus, D. Tolić, A. Ivanovic, M. Car, I. Palunko and S. Bogdan, “Human-in-the-Loop Control of Multi-Agent Aerial Systems,” *Control Conference (ECC), 2016 European* (IEEE, 2016) pp. 2139–2145.
3. A. Poncela and L. Gallardo-Estrella, “Command-based voice teleoperation of a mobile robot via a human-robot interface,” *Robotica* **33**(1), 1–18 (2015).
4. H. Medicherla and A. Sekmen, “Human–robot interaction via voice-controllable intelligent user interface,” *Robotica* **25**(5), 521–527 (2007).
5. Z. Kazi and R. Foulds, “Knowledge driven planning and multimodal control of a telerobot,” *Robotica* **16**(5), 509–516 (1998).
6. J.-C. Lementec and P. Bajcsy, “Recognition of Arm Gestures using Multiple Orientation Sensors: Gesture Classification,” *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)* (IEEE, 2004) pp. 965–970.
7. M. Urban, P. Bajcsy, R. Kooper and J.-C. Lementec, “Recognition of Arm Gestures using Multiple Orientation Sensors: Repeatability Assessment,” *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No. 04TH8749)* (IEEE, 2004) pp. 553–558.

8. D. V. Redrovan and D. Kim, "Hand Gestures Recognition using Machine Learning for Control of Multiple Quadrotors," *2018 IEEE Sensors Applications Symposium (SAS)* (IEEE, 2018) pp. 1–6.
9. V. Nahapetyan and V. Khachumov, "Gesture recognition in the problem of contactless control of an unmanned aerial vehicle," *Optoelectr. Instrument. Data Process.* **51**(2), 192–197 (2015).
10. G. Costante, E. Bellocchio, P. Valigi and E. Ricci, "Personalizing Vision-Based Gestural Interfaces for HRI with UAVs: A Transfer Learning Approach," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2014) pp. 3319–3326.
11. V. M. Monajjemi, J. Wawerla, R. Vaughan and G. Mori, "HRI in the Sky: Creating and Commanding Teams of UAVs with a Vision-Mediated Gestural Interface," *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2013) pp. 617–623.
12. T. O. Costalonga, L. M. Ávila, L. Muniz and A. S. Brandão, "Gesture-Based Controllers to Guide a Quadrotor using Kinect Sensor," *2014 Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol* (IEEE, 2014) pp. 109–112.
13. P. Vadakkepat, T. C. Chong, W. A. Arokiasami and X. Weinan, "Fuzzy Logic Controllers for Navigation and Control of AR. Drone using Microsoft Kinect," *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)* (IEEE, 2016) pp. 856–863.
14. N. Vitiello, U. Olcese, C. Oddo, J. Carpaneto, S. Micera, M. Carrozza and P. Dario, "A Simple Highly Efficient Non Invasive EMG-Based HMI," *2006 International Conference of the IEEE Engineering in Medicine and Biology Society* (IEEE, 2006) pp. 3403–3406.
15. K. Zimenko, A. Margun and A. Kremlev, "EMG Real-Time Classification for Robotics and HMI," *2013 18th International Conference on Methods & Models in Automation & Robotics (MMAR)* (IEEE, 2013) pp. 340–343.
16. Y. Xu, C. Yang, P. Liang, L. Zhao and Z. Li, "Development of a Hybrid Motion Capture Method using MYO Armband with Application to Teleoperation," *2016 IEEE International Conference on Mechatronics and Automation* (IEEE, 2016) pp. 1179–1184.
17. K. Akhmadeev, E. Rampone, T. Yu, Y. Aoustin and E. Le Carpentier, "A testing system for a real-time gesture classification using surface EMG," *IFAC-PapersOnLine* **50**(1), 11498–11503 (2017).
18. G. Kucukyildiz, H. Ocak, S. Karakaya and O. Sayli, "Design and implementation of a multi sensor based brain computer interface for a robotic wheelchair," *J. Intell. Robot. Syst.* **87**(2), 247–263 (2017).
19. R. A. S. Fernandez, J. L. Sanchez-Lopez, C. Sampedro, H. Bavle, M. Molina and P. Campoy, "Natural User Interfaces for Human-Drone Multi-Modal Interaction," *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2016) pp. 1013–1022.
20. Y. Choi, I. Hwang and S. Oh, "Wearable Gesture Control of Agile Micro Quadrotors," *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)* (IEEE, 2017) pp. 266–271.
21. J. DelPreto, A. F Salazar-Gomez, S. Gil, R. M Hasani, F. H Guenther and D. Rus, "Plug-and-play supervisory control using muscle and brain signals for real-time gesture and error detection", **In:** *Robotics: Science and Systems* (2018).
22. B. Muresan and S. S. Esfahlani, "Autonomous Flight and Real-Time Tracking of Unmanned Aerial Vehicle," *Science and Information Conference* (Springer, 2018) pp. 945–956.
23. U. Côté-Allard, D. St-Onge, P. Giguère, F. Laviolette and B. Gosselin, "Towards the Use of Consumer-Grade Electromyographic Armbands for Interactive, Artistic Robotics Performances," *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (IEEE, 2017) pp. 1030–1036.
24. T. Sun, S. Nie, D.-Y. Yeung and S. Shen, "Gesture-Based Piloting of an Aerial Robot using Monocular Vision," *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017) pp. 5913–5920.
25. E. Peshkova, M. Hitz, D. Ahlström, R. W. Alexandrowicz and A. Kopper, "Exploring Intuitiveness of Metaphor-Based Gestures for UAV Navigation," *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)* (IEEE, 2017) pp. 175–182.
26. L. F. Sanchez, H. Abaunza and P. Castillo, "Safe Navigation Control for a Quadcopter using User's Arm Commands," *2017 International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2017) pp. 981–988.
27. F. Oliva-Palomo, A. Sanchez-Orta, P. Castillo and H. Alazki, "Nonlinear ellipsoid based attitude control for aggressive trajectories in a quadrotor: Closed-loop multi-flips implementation," *Control Engineering Practice*, vol 77 (Elsevier, 2018), pp. 150–161
28. J. Carino, H. Abaunza and P. Castillo, "Quadrotor Quaternion Control," *2015 International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2015) pp. 825–831.
29. G. Sanahuja, Fl-Air Framework Libre Air. https://uav.hds.utc.fr/software-flair/. Accessed: 2019-03-10.
30. Myo Gesture Control Armband. https://support.getmyo.com/. Accessed: 2019-03-10.
31. F. Consentino, Pyoconnect. http://www.fernandocosentino.net/pyoconnect/. Accessed: 2019-03-10.

https://doi.org/10.1017/S026357472000003X Published online by Cambridge University Press