2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Feeling the True Force in Haptic Telepresence for Flying Robots

Alexander Moortgat-Pick*[1], Anna Adamczyk*[1], Teodor Tomić[1,2], Sami Haddadin[1]

***Abstract*— Haptic feedback in teleoperation of flying robots can enable safe flight in unknown and densely cluttered environments. It is typically part of the robot's control scheme and used to aid navigation and collision avoidance via artificial force fields displayed to the operator. However, to achieve fully immersive embodiment in this context, high fidelity force feedback is needed. In this paper we present a telepresence scheme that provides haptic feedback of the external forces or wind acting on the robot, leveraging the ability of a state-of-the-art flying robot to estimate these values online. As a result, we achieve true force feedback telepresence in flying robots by rendering the actual forces acting on the system. To the authors' knowledge, this is the first telepresence scheme for flying robots that is able to feedback real contact forces and does not depend on their representations. The proposed event-based teleoperation scheme is stable under varying latency conditions. Secondly, we present a haptic interface design such that any haptic interface with at least as many force-sensitive and active degrees of freedom as the flying robot can implement this telepresence architecture. The approach is validated experimentally using a Skydio R1 autonomous flying robot in combination with a ForceDimension sigma.7 and a Franka Emika Panda as haptic devices.**

I. Introduction

Algorithmic and computational advances are increasingly enabling autonomous flying robots (drones) to operate in densely cluttered environments that are potentially inaccessible to humans. Such robots are typically equipped with vision- or laser-based onboard mapping and motion planning based obstacle avoidance capabilities, allowing safe operation over wireless links with time-varying latency. While some missions can be carried out fully autonomously by the flying robot, operating in uncertain and a-priori unknown environments typically requires a human operator in the loop. State of the art teleoperation schemes for flying robots focus on using haptic feedback for navigation and obstacle avoidance, utilizing e.g. onboard video. With the advancements in state of the art flying robots, now already providing obstacle mapping and avoidance functionality, novel teleoperation schemes are possible and needed. In this paper, we propose a novel tele*presence* scheme for flying robots over communication channels with time-varying latency, that provides feedback about the *external forces* acting on the robot, allowing the operator to *feel* wind and contact forces. The true physical interaction forces acting on the robot, as opposed to artificial forces, are displayed to the operator via a haptic interface. The flying robot used in the scheme must satisfy two assumptions. First, it is able to safely execute given motion commands and the obstacle avoidance behaviour does not rely on the telepresence scheme. Second, the robot must be able to estimate or sense external forces and torques acting on its body.

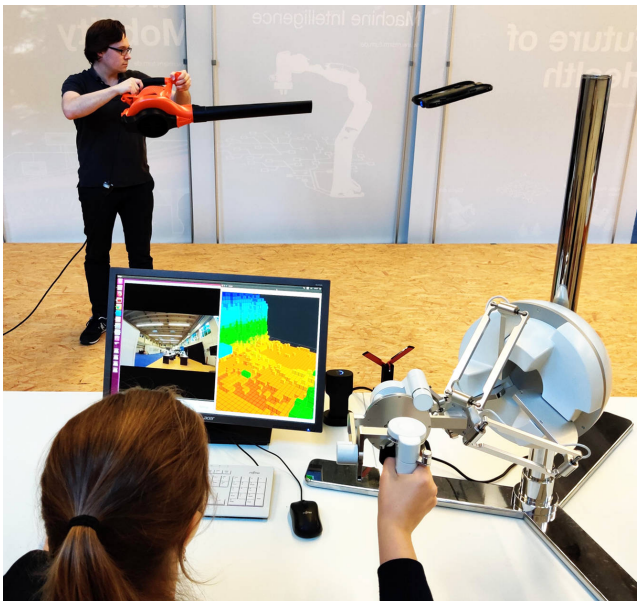

Fig. 1: Haptic feedback from external forces acting on a flying robot (Skydio R1), generated by a leafblower, is provided to a human operator using a Force Dimension sigma.7 haptic device. The event-based telepresence relies on a faster dynamics stable robotic system with onboard obstacle avoidance for operation under time-varying latency.[3]

Experimental results are provided using a Skydio R1 flying robot in conjunction with a Force Dimension sigma.7, a haptic device with 7 degrees of freedom (DoF), and a Franka Emika Panda 7-DoF robot arm.

Related work. Feedback in robotic teleoperation has evolved from using only visual data to haptic telepresence [1]. This approach has been extended to flying robots, where a virtual force based on distance to obstacles is fed back to a haptic joystick in order to facilitate collision-free navigation of mobile robots in cluttered environments [2], [3] [4], [5]. In [2] it was proposed to map the robot proximity to objects to repulsive forces that are used as haptic information in a closed-loop autonomous system with the goal to develop a collision avoidance system. In order to facilitate collision-free teleoperation in [3] virtual environmental forces from a

* The authors contributed equally to this paper.

[1] {alexander.moortgat-pick, anna.adamczyk, haddadin}@tum.de, Chair of Robotics and Systems Intelligence, Technical University of Munich (TUM), Heßstraße 134, 80797 Munich, Germany. Please note that S. Haddadin may have a potential conflict of interest as being a shareholder of Franka Emika.

[2] teo@skydio.com, Skydio, 114 Hazel Ave, Redwood City, CA 94061, USA

[3] Please refer to the accompanying video showing bilateral haptic telepresence of a flying robot with true force feedback.

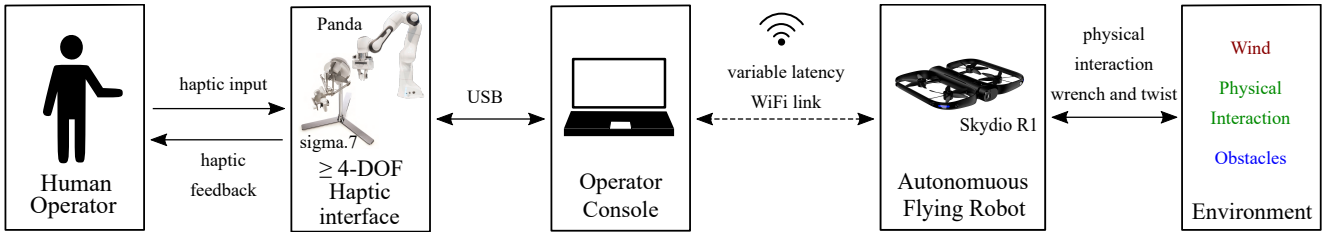

Fig. 2: Hardware setup of our flying robot telepresence system. A human operator interacts with the flying robot's environment through the haptic interface and a variable latency WiFi link. The haptic interface reflects the true physical interaction forces acting on the flying robot.

mobile robot are fed back to a haptic joystick and closed-loop controller. The virtual forces are based on optical flow computed from onboard wide-angle cameras. A virtual force based on optic and telemetric flow of the vehicle is fed back to a haptic joystick in [4]. In [5] virtual potential forces resulting from the distance to objects are mapped to a haptic joystick of which the impedance and admittance configuration is compared w.r.t performance and perceptual differences in teleoperation. Another body of literature aims to improve haptic feedback for maneuvering purposes. In [6] an algorithm was proposed that additionally renders haptic feedback from virtual forces by potential wall functions more rigidly. The concept of a virtual vehicle as a virtual follower is introduced in [7] and [8]. Therein, virtual forces from potential fields are combined with deviation from a desired vehicle state and the virtual force is fed back as haptic information about the environment to the pilot or control loop with the goal of collision avoidance and safe operation in cluttered 3D environments. We may conclude that the main goal of haptic feedback in literature has been to facilitate collision avoidance, which was necessary for safe teleoperation due to the lack of onboard motion planning capabilities as well as force and contact sensing. Thus, collision avoidance and path planning had to be performed by the human operator.

Contribution. First, we propose a novel *true force feedback* telepresence scheme for force sensitive flying robots. It is built upon the external wrench estimation scheme for flying robots shown in [9]. The proposed telepresence scheme is event-based and utilizes the onboard estimation of external forces acting on the robot as well as the autonomous robot high-level motion planning system.

Second, we present a design and implementation principle for a flying robot haptic interface. The proposed telepresence provides feedback about the external forces acting on the robot, enabling the operator to feel e.g. wind and contact forces. Force feedback is displayed to the human via a haptic interface that can be either a dedicated haptic device or a force-sensitive robot with at least as many DoF in its task space as are actuated in the flying robot. In this paper, we focus on an underactuated flying robot with four actuated DoF. Please note that the approach generalizes to full screws, i.e. six DoF for the six-dimensional dual vector describing the spatial rigid body movement. For sake of clarity, we limit ourselves to the single angular motion case that allows rotation only around the vertical axis. We introduce two types of true feedback forces:

1) feedback from direct physical interaction forces perceived by the flying robot, using an external wrench estimator [10] that relies on the robot dynamics model and proprioceptive sensors only;
2) feedback from aerodynamic forces due to wind speed that is estimated by the flying robot, relying on an aerodynamics model [9], which allows the operator to directly experience the wind field around the robot.

This results in haptic feedback that simultaneously combines transient aerodynamic forces, physical interaction, and disturbance forces, including modeling errors [11]. Note that the haptic feedback is not part of the robot local control scheme or used by the operator for collision avoidance. Instead, it is a mere interface that lets the human experience the complex flying system through the sense of touch.

To validate the feasibility of true interaction forces as haptic feedback, experimental results are achieved using a Skydio R1 flying robot. Moreover, we show the integration of two haptic interfaces: a dedicated haptic device as well as a force-sensitive 7-joint robot arm, implementing the proposed interface design.

The remainder of the paper is organized as follows. We outline the haptic interface principle and the approach to haptic feedback in force sensitive flying robots in Section II. The proposed event-based control scheme is explained in Section III. In Section IV we outline the experimental hard- and software setup. The according results are described and discussed in Section V. Finally, we conclude in Section VI.

II. System Design

A. Haptic interface concept

Our telepresence system, illustrated in Fig. 2, provides a framework for a human operator to interact with a force sensitive flying robot. In order to achieve a bilateral connection between operator and robot, we propose impedance controlled rendering of force feedback on a gravity compensated haptic device. The underlying force controller used by the haptic device for force rendering is also used for generating operation commands. This results in a consistent experience for the human operator during telepresence, whether for operation or feedback.

For the proposed telepresence scheme, a gravity compensated force-sensitive robot or dedicated gravity compensated

haptic device is assumed as leader. Here, we discuss the proposed approach with a 4 DoFs quadrotor. Following screw theory, forces are represented as wrenches and velocities as twists. Since four DoFs are actively used by the flying robot and correspondingly by the haptic interface, the four parameters that define the spatial displacement of both, robot and haptic device, correspond to the three components of the translation vector and the Euler Angle around the z-axis. Hence, for sake of clarity, we limit ourselves to a notation of the translational vector paired with the rotation around the z-axis for wrenches and twists (x, y, z, γ). However, w.l.o.g. our approach generalizes to full 6 DoFs. Figure 3 depicts two introduced frames, each corresponding to either leader or follower, between which the operation and feedback are mapped:

1) The haptic device frame F^H. It is globally fixed in space and its origin defines the resting position of the end effector $\boldsymbol{x}_d = (x^H, y^H, z^H, \gamma^H)^T$. The haptic end effector pose $\boldsymbol{x}$ is defined with respect to this frame, where x_i are the respective linear and angular deflection along the spatial axes and yaw axis.
2) The flying robot vehicle frame F^V, which is the local haptic frame translationally shifted to the center of gravity (COG) of the robot. It can be rotated only around the z-axis (yaw), not pitched or rolled.

We introduce two mappings: Force-velocity mapping for robot operation and force-force mapping for feedback. In order to operate the flying robot, the human interacts with the impedance controlled haptic device by deflecting it from $\boldsymbol{x}_d$, thus exerting a simultaneously sensed force on it, which is converted into a desired twist $\boldsymbol{\nu}_d$ for the flying robot. The resting position resembles hovering. In feedback direction, external forces acting on the flying robot in its vehicle frame F^V are displayed on the haptic devices end effector as a force on the corresponding axis in the haptic device frame F^H, e.g., a contact force in positive x^V direction is displayed accordingly on the x^H-axis. If the force is not met with resistance by an operator, it manifests as a deflection corresponding to the impedance behaviour of the end effector along x^H.

Leveraging deflections generated from a clear design objective like impedance behaviour to generate forces has two advantages. First, the respective linear x^H, y^H, z^H or angular γ^H deflections can be exerted separately on the haptic device end effector and thus generate decoupled operational forces, which translate into robot velocities, e.g. only yaw, no linear movement. Such behaviour seems preferable, as it does not require the operator to perform frame transformations during operation, which we suspect to increase the operators cognitive load. Instead, the present end effector state corresponds to the present desired velocity state commanded to the robot. E.g., on an already forward moving robot, resulting from a positive linear deflection, an angular end effector deflection γ^H can be simply super-positioned onto, resulting in a combined forward and yaw movement. This way the operator has the feeling as if pushing, pulling and rotating the flying robot via the haptic interface in space.

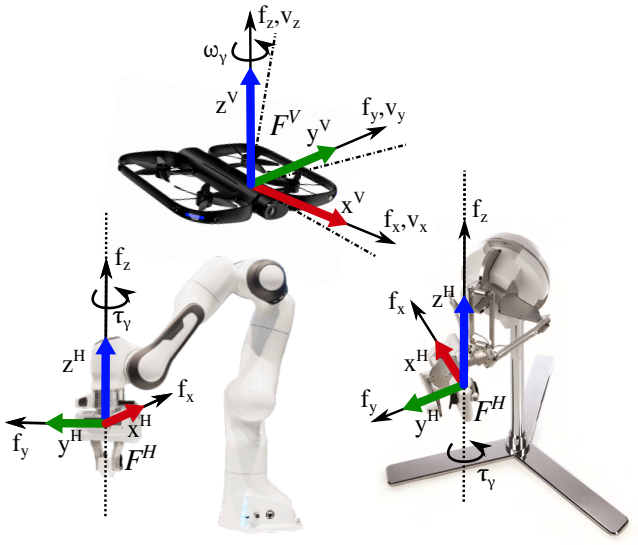

Fig. 3: Illustration of haptic frame F^H attached to the haptic device and F^V attached to the vehicle. Operational commands and feedback are mapped between the frames. Shown by example for Skydio R1 (top), Franka Emika Panda (left), and Force Dimension sigma.7 (right).

Furthermore, by mapping forces to vehicle velocities, the vehicle operation, having a theoretically unlimited workspace, is liberated from the haptic end effector workspace limitations.

B. True force feedback from external wrench and wind

Realism in telepresence needs feedback of meaningful and accurate information, here haptic feedback. In [12], [13] an external wrench estimation scheme for a flying robot was developed. It accomplishes simultaneous online estimation of aerodynamic and contact forces with no need for dedicated sensors but by exploiting proprioceptive standard sensing. The proposed telepresence is built upon this wrench estimation scheme [9] to bring haptic feedback to the operator that originates from true, as opposed to artificial, environmental interaction forces. The flying robot can e.g. switch into various interaction modes when contacts occur. Specifically, we utilize this monitoring signal as a feedback to render interaction between the flying robot and its environment to the operator. Being able to directly interact with the environment through a flying robot and feel the resulting interaction forces immediately on a haptic interface is a first step to approach realism in haptic telepresence for flying robots. Beside contacts, other real environmental forces, like wind drag, may also be of interest to the operator. Furthermore, discrimination of forces not only by the flying robot itself, but also on the operator side through the sense of touch is a cornerstone for applications, where flying robots are utilized as flying tools. As a first step, the stable transmission of the respective force to the operator under varying time delays is developed: haptic feedback from interaction forces and wind forces. Note, that in the scope of this paper we assume the estimated external wrench and estimated wind speed to be accurate.

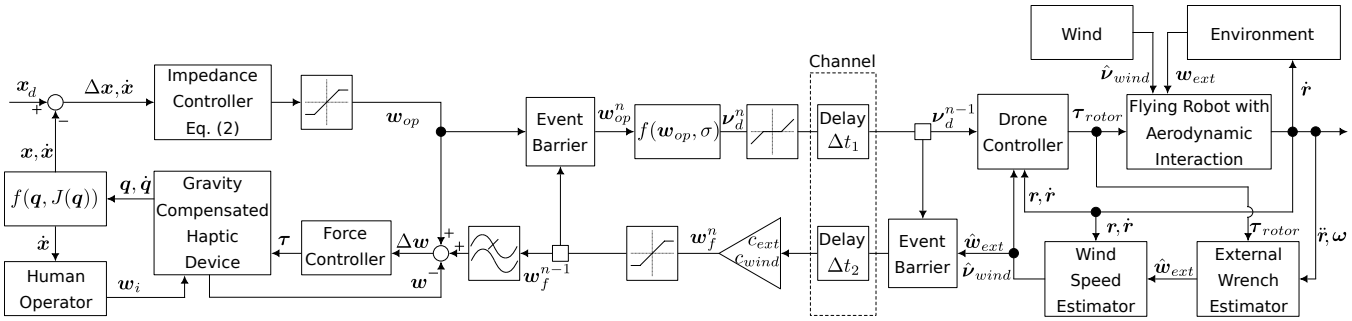

Fig. 4: Overview of the proposed control scheme for the bilateral telepresence controller. It contains three main loops, the haptic controller (left), the event loop (middle) and the flying robot controller (right).

III. Control Scheme

A. Event-based telepresence principle

Figure 4 depicts an overview of our overall controller paradigm, enabling telepresence in a flying robot with true force feedback rendered via a gravity compensated impedance controlled haptic device. The three main components are the haptic interface (left), the flying robot (right) and the event-based loop (middle). The leader – the haptic interface – and the follower – the flying robot – are connected via the event-based control loop. This event loop contains the communication channel and serves to connect and synchronize the haptic interface with the flying robot over the channel.

Typical communication channels for flying robots are based on the Internet Protocol (IP) stack, namely WiFi and Internet. They suffer from time-varying latencies, network buffering effects and brief disconnections. We aim to bypass the need for assumptions about these characteristics, as in other well-established telepresence architectures, by adopting a telepresence architecture, proposed in [14], which does not require a priori assumptions about the communication channel characteristics for stability: event-based telepresence.

In brief, a non-time reference is employed, in which the leader and follower are synchronized by an event-loop e_n instead of time t. As a result the challenges related to synchronizing leader and follower in a time reference frame through an IP channel are omitted. Additionally, a Theorem was proposed to prove the event-based telepresence controller's stability. Several studies evaluated the event-based approach [15], [16], [17].

The event-based principle is adapted to our telepresence controller as follows (see Fig. 5). The non-time reference – the event-loop e_n – synchronizes the communication between leader – the haptic interface – and follower – the flying robot. Each event-loop e_n has the same principal sequence steps j:

0) The previous event loop e_{n-1} is completed, which allows the next event-loop e_n to start.
1) Start of event-loop e_n at a t_{j+1}: the current command $C_n(t_{j+1})$ is measured by the haptic interface and sent into the communication channel.
2) The command C_n reaches the flying robot and is set as the desired state for the robot.
3) A feedback F_n is taken in immediate temporal succession $(t_{j+3} > t_{j+2} > t_{j+1})$ to the arrival of command C_n and send into the communication channel.
4) The event loop e_n is completed when the haptic interface receives the corresponding feedback F_n.
5) The event-loop e_n is completed, thus a new event-loop e_{n+1} with a new command C_{n+1} can start.

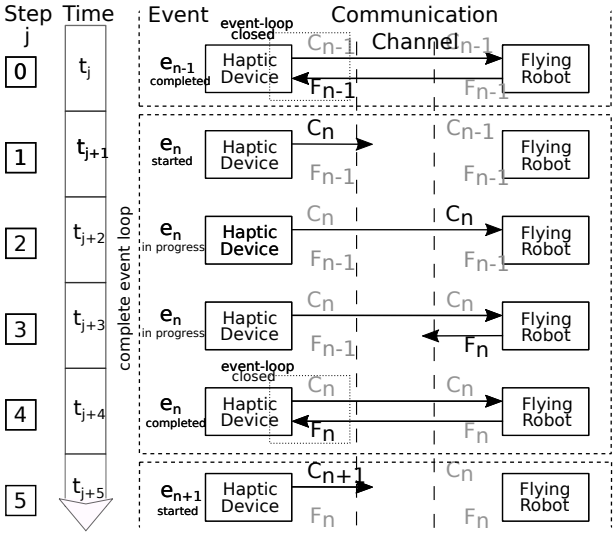

Fig. 5: Principle of the event-based telepresence control loop.

Consequently, at any time there is only one command C_n or respective feedback F_n in the communication channel. No subsequent signal, command or feedback, enters this channel until the event loop e_n is closed.

Evidently, time is still passing while an event-loop is executed. The total duration of an event loop e_n is determined by the round trip delay in the communication channel $\Delta t_n = t_{j+4} - t_{j+1}$. Although no assumptions about time delay are needed, for the purpose of efficient and transparent telepresence one condition should be considered: the duration Δt_n of an event-loop e_n must be sufficiently small, such that the telepresence can still be considered real-time.

Although the event-based principle was applied earlier to robot teleoperation, to the best of our knowledge it was only applied to wheeled robots moving in a plane.

While [14] shows stability of the event-based architecture, the results cannot be directly transferred to our architecture. In contrast to the scheme in [14], where the feedback closing the event-loop is a message from follower to leader, confirming that a certain leader-desired state has been obtained, e.g., a path has been covered, we feed back a real physical force for loop closure. Furthermore, this particular force feedback is subsequently rendered to a human operator via a haptic interface. We practically ensure stability of our system based on two following characteristics: First, the individual subsystems, here the haptic interface and the flying robot, are stable on their own even under influence of the environment or a human operator. Second, the control loop frequency of both subsystems is kept at least a magnitude higher than the event frequency, thus allowing both subsystems to converge independently during events.

While the initial experimental results show the overall functioning and indicate stability of the system, developing the full theoretical framework for event-based telepresence with true force feedback is obviously subject to our future work.

B. Bilateral telepresence with force feedback

Starting the control loop illustrated in Fig. 4 with the haptic interface, its end effector has a rest position $\boldsymbol{x}_d$ located in F^H origin, resembling hovering. In order to control the flying robot, the human exerts an interaction wrench w_i on the haptic devices end effector, moving the end effector $\boldsymbol{x}$. The operator-induced deflections are counteracted by the impedance controller (2) that intents to restore the end effector rest position. This restoring wrench is the negative operational wrench $\boldsymbol{w}_{op}$

$$\boldsymbol{w}_{op} = (f_{op,x}, f_{op,y}, f_{op,z}, \tau_{op,\gamma})^T \in F^H \tag{1}$$

$$w_{op,i} = K_{p,i}(x_i - x_{d,i}) - K_{d,i}\dot{x}_i \ , i \in \{x, y, z, \gamma\}, \tag{2}$$

where $\boldsymbol{K}_p \in \mathbb{R}^4$ and $\boldsymbol{K}_d \in \mathbb{R}^4$ are the proportional and derivate gains of the impedance controller.

Since the operator can apply (unintentionally) large physical deflections on the haptic interface, the operational wrench $w_{op,i}$ is clipped $w_{op,i}^{max}$

$$w_{op,i} = \min\{\max\{w_{op,i}, -w_{op,i}^{max}\},\ w_{op,i}^{max}\},\ i \in \{x, y, z, \gamma\}. \tag{3}$$

Equation (3) ensures that despite of large physical deflections a maximal applicable operational wrench is guaranteed, which also sets a limit to the maximum command input.

At the event-barrier, the signal of $\boldsymbol{w}_{op}$ is permanently present, i.e. refreshed with the same frequency as the impedance controller. The moment a new event-loop is permitted, one value of $\boldsymbol{w}_{op}$ can pass through the event-barrier as a new command input to the flying robot. Once the new command is triggered, an exponential mapping (see Fig. 6) according to (4) is applied on the signal, resulting in the desired vehicle velocity $\boldsymbol{\nu}_d^n$

$$\nu_{d,i}\left(w_{op,i}\right) = \nu_{d,i}^{\max}\ \operatorname{sign}(w_{op,i})\ \frac{\exp\left(\frac{|w_{op,i}|}{w_{op,i}^{\max}}\sigma\right) - 1}{\exp\left(\sigma\right) - 1}, \tag{4}$$

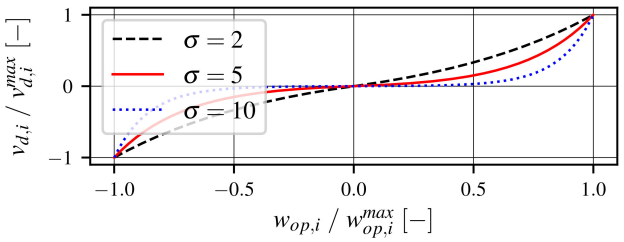

Fig. 6: Exponential mapping of operational wrench to desired twist.

where σ denotes the slope of the transformation function.

Although a linear mapping, such as in common drone controllers, from the operational wrench $\boldsymbol{w}_{op}$ to desired twist $\boldsymbol{\nu}_d^n$ is possible, the exponential mapping results in a higher dynamic range of control inputs. This turned out to allow for a more precise control of small robot velocities, however, still preserving the capability to operate at full speeds, as with a regular linear mapping.

Subsequently, for practical reasons, a deadzone handles an allowed tolerance as well as noise in velocity commands such that only operator interaction wrenches exceeding a threshold result in a distinct command input. The threshold $\theta_i \in \mathbb{R}$ is applied individually on each of the wrench elements, in order to differentiate individually between desired maximum translational and rotational velocities, e.g. during different operation goals

$$\nu_{d,i}^n = \begin{cases} \nu_{d,i}^n, & |\nu_{d,i}^n| \geq \theta_i \\ 0, & |\nu_{d,i}^n| < \theta_i \end{cases}, i \in \{x, y, z, \gamma\}. \tag{5}$$

The command is dispatched into the communication channel and arrives after a delay Δt_1 at the follower – the flying robot with aerodynamic interaction.

Important to note is that although the operator can move the haptic interface while an event loop is already running, no new commands will be sampled until feedback is received and the current event loop closed. Based on the desired movement command $\boldsymbol{\nu}_d^n$ and the current robots state, the drone controller calculates the necessary motor torques for the robot to reach the desired velocity while simultaneously being subject to external interaction wrenches with the environment and wind. The estimated external wrench $\hat{\boldsymbol{w}}_{\text{ext}}$ and impacting wind velocity $\hat{\boldsymbol{\nu}}_{\text{wind}}$ are – corresponding to the leader side – permanently present at the event-barrier, i.e. refreshed with the same frequency as the drone controller.

Again, corresponding to the leader side, the arrival of a new command triggers the event-barrier on follower side, sending the current estimated external wrench $\hat{\boldsymbol{w}}_{\text{ext}}$ and impacting wind velocity $\hat{\boldsymbol{\nu}}_{\text{wind}}$ as feedback through the event-barrier into the communication channel towards the leader.

After the delay Δt_2 the feedback signal arrives at the leader side and is processed. The measured external wrench is scaled by a dimensionless constant tuning factor c_{ext} $[-]$

$$\hat{w}_{f\text{ext},i} = c_{\text{ext}}\ \hat{w}_{\text{ext},i},\ i \in \{x, y, z\} \tag{6}$$

such that the feedback range matches the displayable range of the haptic interface. The wind feedback is scaled similarly

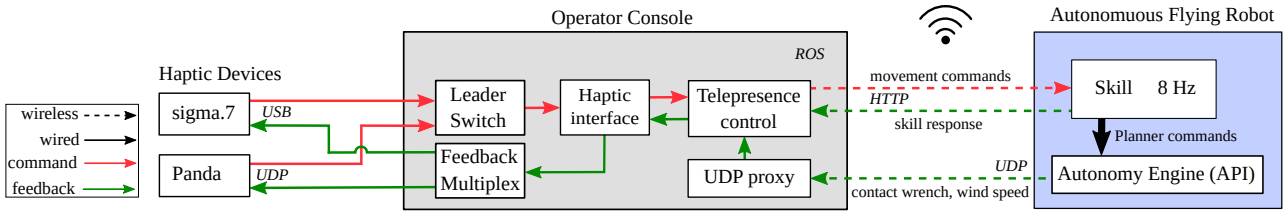

Fig. 7: Overview of the implemented software structure. While both haptic devices can simultaneously receive and render feedback, only one at a time can be the leader and send commands to the flying robot.

by a constant factor c_{wind} $[\frac{\text{N s}}{\text{m}}]$

$$\hat{w}_{f\text{wind},i} = c_{\text{wind}}\ \hat{\nu}_{\text{wind},i},\ \ i \in \{x, y, z\}. \tag{7}$$

Since there are two feedback sources, the total feedback wrench $\boldsymbol{w}_f^n$ can e.g. be composed of both the feedback from external wrenches on the vehicle $\hat{\boldsymbol{w}}_{\text{ext}}$ and the wind feedback $\hat{\boldsymbol{\nu}}_{\text{wind}}$ by superposition:

$$\hat{\boldsymbol{w}}_f = \hat{\boldsymbol{w}}_{f\text{ext}} + \hat{\boldsymbol{w}}_{f wind}. \tag{8}$$

However, (8) may also be easily adapted online to only render feedback from one of the sources. To not exceed the displayable force by the haptic interface, the composed feedback signal $\hat{\boldsymbol{w}}_f$ is clipped. Subsequently, the feedback signal triggers the event-barrier to emit a new command $\boldsymbol{\nu}_d^{n+1}$ into the communication channel. This marks the completion of the event loop n.

The feedback loop is slow in comparison to the haptic control loop. Although the feedback is rendered in the haptic control loop, its changes during one event would be applied at once in the next haptic control step. Therefore, the feedback is smoothened by a lowpass filter in order to distribute its changes over multiple haptic control steps.

IV. Experimental Setup and Software Structure

We implemented our telepresence system for experimental validation using the setup shown in Fig. 2. The software framework is shown in Fig. 7. ROS [18] is used for local interprocess communication and logging. The central component is a teleoperation ROS node written in Python that exchanges messages with the flying robot and the haptic interface. The latter is an abstraction written in C++ to interact with the sigma.7 haptic device and the Panda robot over UDP. The sigma.7 device is controlled via its robotic SDK. A PD-controller (2) was implemented to map the end effector deflection to a restoring wrench and set it, superposed with the feedback, as the desired wrench for the haptic force controller. The Panda robot is running an impedance controller as well, however the setpoint, input range and damping differ from the sigma.7. The desired resting pose of the Panda end effector was set at 40 cm above and in front of the base. The stiffness for both devices was set to allow a maxium wrench $(10\ \text{N}, 0.4\ \text{Nm})$ at the preferred input range of the end effector, while the damping ratio was set to 1 for the sigma.7 and to $\frac{\sqrt{2}}{2}$ for the Panda.

The teleoperation node communicates directly with the drone via HTTP over WiFi to send commands to the active skill and receive responses. For fast data retrieval directly from the autonomy engine, the drone proxy is a ROS node implemented in Python that serves to establish an UDP link with the Skydio drone. Received data is published via ROS and linked to skill responses by the teleoperation node. The feedback is transmitted to the active leader input device, but can also be multiplexed to both sigma.7 and Panda arm simultaneously. Experiments are performed indoor for maximum control over environmental conditions.

V. Experimental Results and Discussion

In this section we report initial results and experience with the proposed flying robot telepresence system. The experiments are designed to assess the system fulfillment of the previously stated criteria: stability, efficiency and transparency. Specifically, we investigate the following behaviour in spite of varying time delays:

- How quickly are velocity commands from the operator executed by the flying robot if it is unobstructed by obstacles?
- How transparent can the flying robot interaction forces be experienced by the operator, despite the superimposed impedance behaviour on the leader side?

The first scenario evaluates the operational, i.e. unilateral, aspect of the telepresence, by analyzing fulfillment of operator commands on the flying robot with respect to event and time. Figure 8 shows the results from an experiment made with the sigma.7 (above) and the Panda arm (below) as input devices. The first 45 seconds show fast flight from a starting position to a position of about 10 m away and returning to the starting position, followed by a repetition, but with slow velocity at start, and finally at higher velocity and vice versa on the return flight. The second half showcases the system behaviour with higher frequency input: four periods of oscillatory input with about 0.5 Hz (at 45 s), eight repitions with about 1 Hz (at 58 s; some package loss on the Panda at 63 s). The last part (at 75 s) shows sudden release of the end effector after deflecting it from the resting position.

The results show that the vehicle can generally follow velocity commands from both the sigma.7 and Panda robot. Due to the dynamics of the vehicle, the actual velocity is always lagging the desired velocity. This can only be improved through higher maximum acceleration in combination with a more aggressive controller. The second half of the experiments show the system limitations, as the flying robot is not able to follow rapidly changing inputs (see 1 Hz oscillatory input). Figure 9 depicts the event duration or round-trip delay plotted against time as well as its distribution for the sigma.7

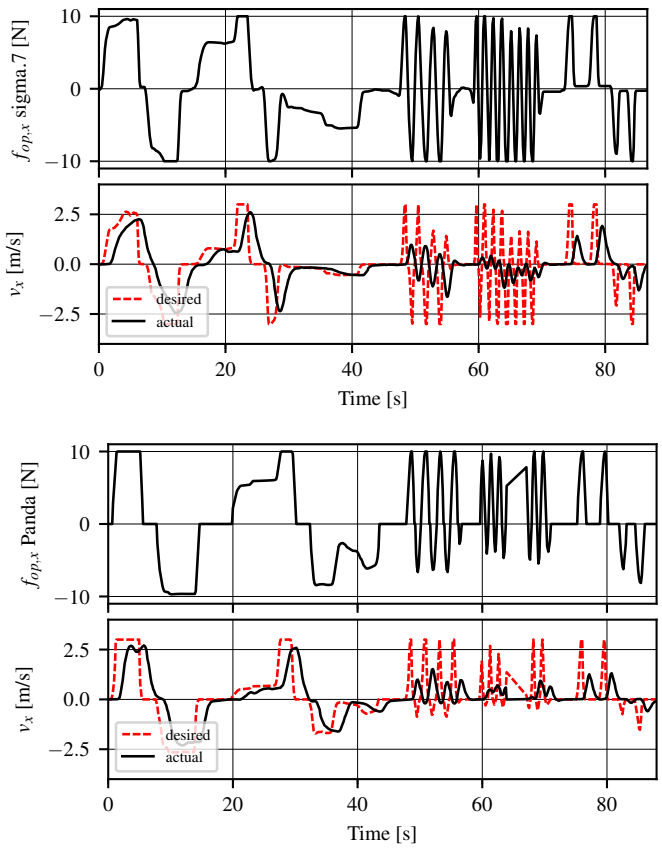

Fig. 8: Teleoperation experiment with sigma.7 and Panda as leader.

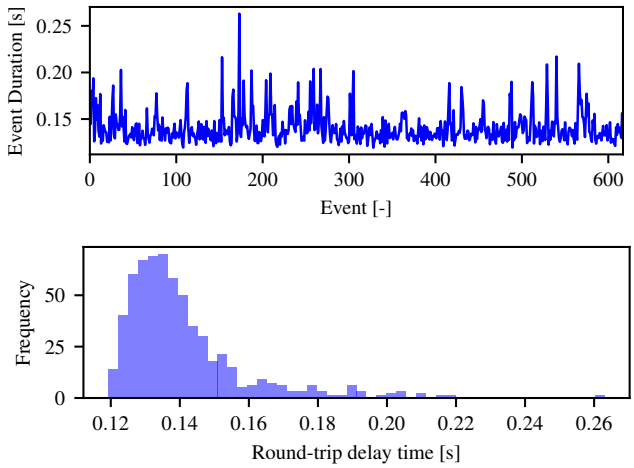

Fig. 9: Evaluation of time latency by event duration.

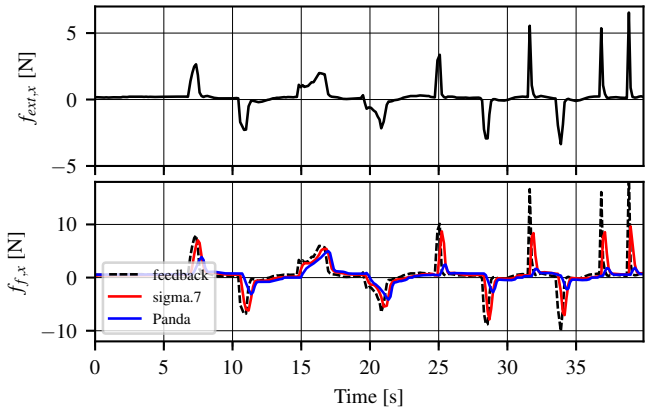

Fig. 10: External wrench feedback performance.

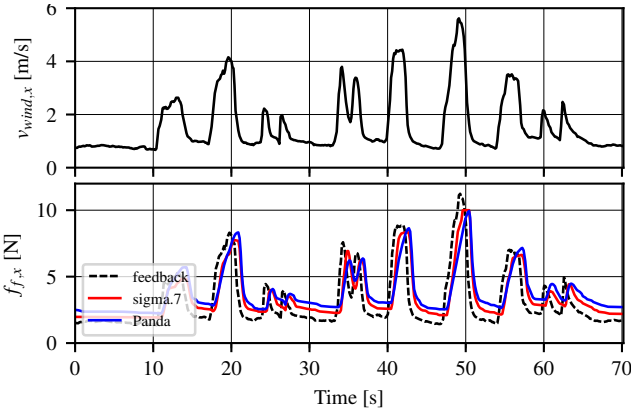

Fig. 11: Wind feedback performance.

experiment. As can be seen, the event duration is not constant and fluctuates between 0.12 s at best and 0.26 s in the worst case. Despite the fairly large delays, the flying robot felt responsive to the operator during the entire experiment. Due to higher mass and friction of the Panda arm compared to the sigma.7 device, the teleoperation felt less transparent to the operator.

In the second scenario we aim at evaluating the rendering of feedback forces to the operator despite closed loop impedance regulation control on leader side. Therefore, we assess the systems ability to display force signals of varying source and varying frequency. In order to keep human bias out of initial experimental results, we perform the experiments without operator and instead let feedback forces manifest as deflection on the haptic device end effector, essentially evaluating the unobstructed system behaviour, including the haptic device impedance behaviour. We show one experiment with feedback from external wrench estimation due to haptic contact forces and a second experiment with estimated wind velocity as feedback with a leafblower as wind source. The feedback is multiplexed to both the sigma.7 and the Panda robot. The flying robot was not moved by operator commands during both experiments.

In the first experiment (see Fig. 10) the flying robot was repeatedly pushed and pulled in its x-axis with varying durations of one second (at 7 s), two seconds (at 14 s), half a second (at 24 s), a quarter second (at 32 s) and poked two times (37 s and 39 s). The figures show, that both the sigma.7 and the Panda can track the lower frequency feedback during the first half of the experiment, but the performance decreases for the higher frequent contact during the second half. The behaviour of the system also is different for sigma.7 and Panda devices.

The second experiment (Fig. 11) was made with feedback from estimated wind velocity. A leafblower was used as a

wind source. It was pointed at the drone aligned with its x-axis repeatedly for durations of 1 s to 3 s of varying intensity. Compared to the first experiment, both devices are able to track the feedback more accurately.

Summarizing, the results show that the system is suitable to teleoperate the flying robot using either the sigma.7 or the Panda. Rendering feedback forces received from the robot is also possible on both devices while simultaneously operating.

However, there is one important aspect that needs to be considered carefully in future research. Both devices are force sensitive with a bandwidth of several Hertz, thus force rendering can be expected to be accuarate. The results showcase that the force transparency is obviously deteriorated by the local impedance behaviour, which becomes especially prevalent for higher frequent forces. This can be caused by several factors, including controller parameters, end effector mass and inertia. The difference between sigma.7 and Panda indicates, that the impedance behaviour and therefore force transparency can be shaped. Therefore, choosing the optimal local impedance behaviour such that the force transparency is maximized remains an open issue.

VI. Conclusion

In this paper, we presented a novel telepresence scheme that provides haptic feedback directly from the *true forces*, in contrast to artificial forces, acting on a flying robot. An event-based scheme is employed to synchronize the flying robot and haptic device under highly time-varying communication channels by event instead of time. This results in an elegant scheme to operate even complex systems without the need for low-level control access on the robot. Effectiveness of the scheme has been validated through experiments using both a Force Dimension sigma.7 7-DoF haptic device and a Franka Emika Panda teleoperating a Skydio R1 autonomous drone. By responding to the forces acting on the robot, the human operator is an essential part of the overall closed loop system and can fundamentally influence the overall behavior of the robot, for example by changing the stiffness of the arm operating the haptic device. In future work we aim to investigate how to choose the local leader impedance behaviour such that maximal transparency is achieved. Transparency depends not only on the system but also on characteristics and requirements of the human operator, such that these have to be included in the evaluation, e.g. within a user study. Furthermore, although the system appears to be stable overall, its stability has to be further investigated. As the event-based approach is conservative, we also aim to compare it with other well established approaches that have stricter limits on the communication channel.

Acknowledgements

We thank the Vodafone Group plc for their support and for funding of the haptic devices and robots used in this study.

We also thank L. Johannsmeier for his support in the Panda robot software framework.

References

[1] P. F. Hokayem and M. W. Spong, "Bilateral teleoperation: An historical survey," *Automatica*, vol. 42, no. 12, pp. 2035–2057, dec 2006.

[2] T. M. Lam, H. W. Boschloo, M. Mulder, and M. M. Van Paassen, "Artificial force field for haptic feedback in uav teleoperation," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 6, pp. 1316–1330, 2009.

[3] R. Mahony, F. Schill, P. Corke, and Y. S. Oh, "A new framework for force feedback teleoperation of robotic vehicles based on optical flow," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 1079–1085.

[4] H. Rifaï, M.-D. Hua, T. Hamel, and P. Morin, "Haptic-based bilateral teleoperation of underactuated unmanned aerial vehicles," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 13 782–13 788, 2011.

[5] X. Hou, R. Mahony, and F. Schill, "Comparative study of haptic interfaces for bilateral teleoperation of vtol aerial robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 10, pp. 1352–1363, 2015.

[6] S. Omari, M.-D. Hua, G. Ducard, and T. Hamel, "Bilateral haptic teleoperation of vtol uavs," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 2393–2399.

[7] S. Stramigioli, R. Mahony, and P. Corke, "A novel approach to haptic tele-operation of aerial robot vehicles," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 5302–5308.

[8] A. Y. Mersha, S. Stramigioli, and R. Carloni, "Bilateral teleoperation of underactuated unmanned aerial vehicles: The virtual slave concept," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 4614–4620.

[9] T. Tomić, P. Lutz, K. Schmid, A. Mathers, and S. Haddadin, "Simultaneous contact and aerodynamic force estimation for aerial robots," *International Journal of Robotics Reseach (IJRR)*, 2018.

[10] T. Tomić, C. Ott, and S. Haddadin, "External wrench estimation, collision detection, and reflex reaction for flying robots," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1467–1482, Dec 2017.

[11] T. Tomić and S. Haddadin, "Towards interaction, disturbance and fault aware flying robot swarms," in *Robotics Research. Springer Proceedings in Advanced Robotics*. Springer, 2020, pp. 183–198.

[12] ——, "A unified framework for external wrench estimation, interaction control and collision reflexes for flying robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4197–4204.

[13] ——, "Simultaneous estimation of aerodynamic and contact forces in flying robots: Applications to metric wind estimation and collision detection," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5290–5296.

[14] N. Xi and T. J. Tarn, "Stability analysis of non-time referenced internet-based telerobotic systems," *Robotics and Autonomous Systems*, vol. 32, no. 2-3, pp. 173–178, 2000.

[15] ——, "Action synchronization and control of internet based telerobotic systems," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, vol. 1. IEEE, 1999, pp. 219–224.

[16] I. Elhajj, N. Xi, and Y.-h. Liu, "Real-time control of internet based teleoperation with force reflection," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, vol. 4. IEEE, 2000, pp. 3284–3289.

[17] I. Elhajj, N. Xi, W. K. Fung, Y. H. Liu, W. J. Li, T. Kaga, and T. Fukuda, "Haptic information in internet-based teleoperation," *IEEE/ASME Transactions on mechatronics*, vol. 6, no. 3, pp. 295–304, 2001.

[18] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, Japan, 2009, p. 5.