

# Intercontinental Haptic Teleoperation of a Flying Vehicle: a Step Towards Real-time Applications

Abeje Y. Mersha\*, Xiaolei Hou\*\*, Robert Mahony\*\*, Stefano Stramigioli\*, Peter Corke# and Raffaella Carloni\*

**Abstract**—This paper describes the theory and practice for a stable haptic teleoperation of a flying vehicle. It extends passivity-based control framework for haptic teleoperation of aerial vehicles in the longest intercontinental setting that presents great challenges. The practicality of the control architecture has been shown in maneuvering and obstacle-avoidance tasks over the internet with the presence of significant time-varying delays and packet losses. Experimental results are presented for teleoperation of a slave quadrotor in Australia from a master station in the Netherlands. The results show that the remote operator is able to safely maneuver the flying vehicle through a structure using haptic feedback of the state of the slave and the perceived obstacles.

## I. INTRODUCTION

Recent advancements in the field of aerial robots have greatly expanded application areas where flying vehicles can be considered. Some of them include sample picking, inspection by contact, manipulation, and exploration of unstructured and uninhabited areas, where non-destructive (active but safe) interaction is required [1]. These applications often need human-in-the-loop primarily for superior intelligence and safety. As a result, considerable attention has been given to haptic teleoperation of flying vehicles [2], [3], [4].

Most works in the field have so far used haptic feedback to indicate approaching obstacles based on optical flow [2] or telemetric sensory measurements [5], [6]. In [7], [8], the force feedback is used as a cue about the state of the flying vehicle. All the aforementioned works employ an impedance control framework, i.e., the user applies a position/velocity to the master and is presented with a force. On the other hand, admittance control frameworks, where the user applies a force and is presented with a position/velocity, has been proposed in [9], [10].

Unique challenges in haptic teleoperation of flying vehicles include underactuatedness of the aerial vehicle, workspace incompatibility between the master and the slave, and continuous energy dissipation even near hovering. To

overcome the master workspace limitation, a car driving metaphor (i.e., rate control) has been adopted by several authors [6], [8]. A virtual slave is introduced in [11] to overcome the continuous energy dissipation by the action of gravity. This concept has been extended for multidimensional and underactuated flying vehicles in [8] and different variants of the concept have been implemented in [12]. In all these papers, an energy-based passivity approach has been followed to design teleoperation systems that maintain overall stability over a wide range of operating conditions. However, explicit treatment of network-induced imperfections, such as time-varying delays and packet losses, in a realistic real-time scenario is still missing or minimal, if any, in the literature.

In this paper, we aim to illustrate the practicality of a simple but effective tele-control scheme for real-time application in the presence of significant network-induced imperfections. The scheme is characterized by its clear distinction between the locally defined energetic circuits and the behavior-modifying signals exchanged between the master and the slave. We provide verifying experimental results of the longest intercontinental bilateral teleoperation of aerial vehicles ever. It has been conducted between the Netherlands and Australia to perform some useful tasks, such as navigation and obstacle avoidance while maneuvering through a structure. This experimental setting is chosen to mimic realistic scenarios and address challenges related to real-time deployment of flying vehicles for civilian applications.

Moreover, we present practical considerations and parameter tuning method that are necessary for effective utilization of the framework. System integration of various components and the implemented modular hardware and software frameworks are also described in detail.

The rest of the paper is organized as follows: In Sec. II, the teleoperation control architecture with methods of coping with network-induced imperfections are presented. Experimental results of the longest intercontinental teleoperation along with hardware and software architecture used for practical realization are detailed out in Sec. III. Finally, concluding remarks are given in Sec. IV.

## II. TELEOPERATION CONTROL STRUCTURE

### A. General Approach

We adopt the teleoperation algorithm based on the virtual slave concept [8], [11]. The fundamental principle to ensure passivity of the tele-control loop is to simply exchange

\*A.Y. Mersha, S. Stramigioli and R. Carloni are with the University of Twente, The Netherlands. e-mail: {a.y.mersha, s.stramigioli, r.carloni}@utwente.nl

\*\*X. Hou and R. Mahony are with the Australian National University, Australia. e-mail: {xiaolei.hou, robert.mahony}@anu.edu.au

#P. Corke is with the Queensland University of Technology, Australia. e-mail: peter.corke@qut.edu.au

This research is supported by the European Commission's Seventh Framework Program as a part of the project "Innovative Aerial Service Robots for Remote Inspection by Contact (AIRobots)" under grant number 248669 and the Australian Research Council through the Future Fellowship FT0991771 "Foundations of Vision Based Control of Robotic Vehicles".

energy-neutral<sup>1</sup> reference signals across the unreliable communication channel. These signals in turn, modulate the local energy tanks of the master and the slave, from which energy required to accomplish a desired task is consumed. As such, the desired behavior of the master and the slave is attained by the action of their respective controllers, provided that there is energy in the local tanks.

Using this strategy, the communication channel is made energetically neutral. This approach provides additional merits, which include: difficulty or even failure of convergence of energy levels of both tanks, unavailability of the energy trapped in the communication channel, and additional dynamics that may be required to passivise the channel with respect to the states used in the mapping strategies.

Note that since the workspace limitation of the master device is primarily inherent to the translational domain, only teleoperation in this domain is considered in our formulation. Should there be a need to tele-control the rotational degree of freedom (DoF) of the slave, the proposed algorithm can be extended without fundamental modification.

### B. The Master System

The master system consists of a haptic device, its controller and a local energy tank. Since the haptic interface is a passive mechanical system, ensuring passivity of its controller implies passivity of the master system.

If  $x_m$  and  $p_m$  denote the 3–dimensional position and translational momenta of the tip of a generic haptic interface that has gravity compensation, its dynamics in port-Hamiltonian systems framework is given by

$$\begin{cases} \begin{pmatrix} \dot{x}_m \\ \dot{p}_m \end{pmatrix} = \begin{pmatrix} 0 & I \\ -I & -R_m \end{pmatrix} \begin{pmatrix} \frac{\partial H(x_m, p_m)}{\partial x_m} \\ \frac{\partial H(x_m, p_m)}{\partial p_m} \end{pmatrix} + \begin{pmatrix} 0 \\ I \end{pmatrix} F_m \\ \dot{x}_m = \begin{pmatrix} 0 & I \end{pmatrix} \begin{pmatrix} \frac{\partial H(x_m, p_m)}{\partial x_m} \\ \frac{\partial H(x_m, p_m)}{\partial p_m} \end{pmatrix} \end{cases} \quad (1)$$

where  $H(x_m, p_m) = \frac{1}{2} p_m^T M_m^{-1}(x_m) p_m$  is its Hamiltonian;  $M_m$  and  $R_m$  are  $3 \times 3$  positive definite inertial and damping matrices, and  $(\dot{x}_m, F_m)$  are velocity and force port variables.

To overcome the master workspace limitation, a rate control strategy is adopted, i.e., the position of the tip of the haptic interface,  $x_m$ , is mapped to a scaled velocity reference of the slave,  $\dot{x}_s^*$ . The mapping function  $\phi$  is

$$\begin{cases} \dot{x}_s^*(t) = \phi(x_m(t - \tau_f)) \\ x_m^*(t) = \phi^{-1}(\dot{x}_s^*(t - \tau_b)) \end{cases} \quad (2)$$

where  $\tau_f$  and  $\tau_b$  denote the variable time delay in the forward and backward traveling signals. The superscript \* indicates the reference signals that are received over the communication channel from the other side of the teleoperation loop.

The master controller uses  $x_m^*$  as reference to generate force feedback to provide the operator with a haptic cue about the state and the environment of the flying vehicle. A combination of a repulsive and dissipative forces can be used as haptic feedback.

<sup>1</sup>The signals are energy-neutral in the sense that no energy exchange information is inferred from them as in other algorithms [13], [14].

$$\begin{cases} \dot{x}_{mc} = \dot{x}_m \\ F_{mc} = F_m = \frac{\dot{x}_{mc}}{H(x_{mc}, \tilde{x}_m^*)} - d_m \dot{x}_{mc} \end{cases} \quad (3)$$

where  $H(x_{mc}, \tilde{x}_m^*) = \frac{1}{2}(\tilde{x}_m^* - x_{mc})^T k_m(\tilde{x}_m^* - x_{mc})$ ;  $k_m$  and  $d_m$  are  $3 \times 3$  positive definite stiffness and damping matrices;  $(\dot{x}_{mc}, F_{mc})$  are port variables of the master controller.

In Eq. 3,  $\tilde{x}_m^*$  is the modified version of  $x_m^*$  to ensure passivity of the master controller. The passivity of the controller is guaranteed by monitoring the content of the local energy tank. This tank controls the energy injected/withdrawn by the operator. Based on its content, it modifies the desired controller action. The power conserving interaction dynamics between the tank and controller is given by

$$\begin{pmatrix} \dot{\lambda} \\ \dot{\tilde{x}}_m^* \end{pmatrix} = \begin{pmatrix} 0 & \delta \\ -\delta & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial H(\lambda)}{\partial \lambda} \\ \frac{\partial H(x_{mc}, \tilde{x}_m^*)}{\partial \tilde{x}_m^*} \end{pmatrix} \quad (4)$$

where  $\lambda$  and  $H(\lambda)$  are the state and the Hamiltonian of the master tank.  $\delta$  is a transformation matrix that modulates the power flow between the tank and the controller.

In the presence of enough energy in the tank, Eq. 4 gives  $\tilde{x}_m^* = x_m^*$ . Note that the local energy tank of the master should be saturated from above to limit possible indefinite build up of energy, which otherwise may result even in loss of passivity in the desired operating conditions.

### C. The Slave System

In classical teleoperation, the slave system contains the robot hardware that attempts to accomplish the desired task commanded by the operator with the aid of a local controller. Here, in addition, a virtual slave that is part of the slave's controller is implemented. It effectively decouples the energy required for maneuvering the real slave from the energy supplied to just maintain the flight [11]. Moreover, it closes the teleoperation loop, while the actual task execution commands of the real slave and its reaction to them are exchanged through coupling [8]. The principles of operation of the two slaves are briefly described next.

1) *The Virtual Slave*: It serves as the real slave's proxy. It consists of an inertial element, tracking controller, multi-state energy tank, a supervisor and a visco-elastic coupling with the real slave.

The inertial element in this work is a 3D point mass that operates in gravityless and frictionless ideal environment. The forces that act on this element are due to the tracking controller  $\bar{F}_s^t$  (couples it with the operator) and the visco-elastic coupling with the real slave  $\bar{F}_s^c$ . In this way, the virtual slave both commands the real slave and maps back its reaction to the operator. The dynamics of these couplings are given by

$$\begin{cases} \bar{F}_s^t = -d_t(\dot{\bar{x}}_s - \dot{x}_s^*) \\ \bar{F}_s^c = k_c(\bar{x}_s - x_s) + d_c(\dot{\bar{x}}_s - \dot{x}_s) \end{cases} \quad (5)$$

where  $\bar{x}_s$  and  $x_s$  denote the positions of the virtual and real slaves, respectively;  $d_t$ ,  $k_c$ , and  $d_c$  are control parameters of appropriate dimensions, see Fig. 1.

Similar to the master system, the virtual slave is also endowed with a local energy tank. Only variable coupling based on the availability of energy in the tank is allowed

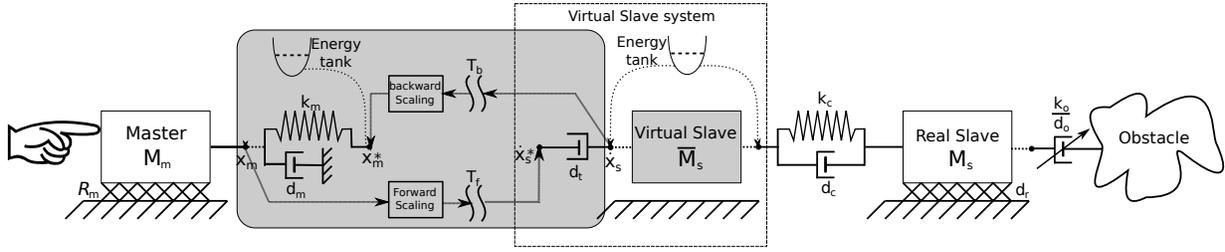


Fig. 1. The ideal physical model equivalent of the complete tele-control scheme. The master and the slave in the teleoperation loop exchange only setpoints through the unreliable communication channel. The broken lines connected to the energy tanks show that every control action generated to accomplish the desired task has energetic cost, which can be carried out only when the required amount of energy is available in the local energy tank.

between the tank and inertial element, see Fig. 1. The flow of energy between the two is realized by the aid of a passivity-enforcing supervisor that modulates the energy routing process. The coupled dynamics of the tank and the inertial element is given by

$$\begin{pmatrix} \dot{\bar{p}}_s \\ \dot{\chi} \end{pmatrix} = \begin{pmatrix} 0 & \xi \\ -\xi & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial H(\bar{p}_s)}{\partial \bar{p}_s} \\ \frac{\partial H(\chi)}{\partial \chi} \end{pmatrix} \quad (6)$$

where  $H(\bar{p}_s) = \frac{1}{2}\bar{p}_s^T \bar{M}_s^{-1} \bar{p}_s$  and  $H(\chi) = \frac{1}{2}\chi^T k_\chi \chi$  are the Hamiltonian functions of the inertial element and the tank, in which  $\bar{M}_s$  and  $k_\chi$  are positive definite matrices representing the inertia of the virtual slave and the control parameter of the local tank, respectively.  $\chi$  is the state of the tank and  $\bar{p}_s = \bar{M}_s \dot{\bar{x}}_s$  is the momenta. For passivity, the supervisor assigns the following values to the diagonal elements of  $\xi$ :

$$\xi(i, i) = \begin{cases} 0, & \text{if } \chi(i) < \gamma \text{ \& } (\bar{F}_s^t(i) + \bar{F}_s^c(i)) \dot{\bar{x}}_s(i) > 0 \\ \frac{\bar{F}_s^t(i) + \bar{F}_s^c(i)}{k_\chi(i, i) \chi(i)}, & \text{otherwise} \end{cases} \quad (7)$$

for  $i = 1, 2, 3$  and  $\gamma > 0$  is an arbitrary small number.

2) *The Real Slave*: The real flying vehicle and its controller are part of the real slave system. In this work, a Mikrokopter quadrotor is used as the flying vehicle. The low-level attitude control is based on the standard flight control that takes stick positions as input and regulate pitch, roll, yaw, and total thrust. The primary DoFs for the teleoperation are the  $(x, y)$  that require control of the pitch and roll dynamics due to the underactuated nature of the vehicle. These dynamics use the full structure of the virtual slave dynamics, while the height ( $z$ ) and yaw( $\theta$ ) dynamics are controlled using a more classical approach in this work.

The linear translational dynamics of the quadrotor are approximated based on the assumption of quasi-stationary flight. Ignoring aerodynamic induced lift and rotor flapping effects, this approximation leads to the thrust exactly compensating for the gravitational force, i.e.,  $T = m_s g$  (where  $M_s = m_s I_{3 \times 3}$  denotes the mass matrix of the real slave).

The  $x$  and  $y$ -axes translational dynamics are identified and approximated by a second order transfer function from a stick position input (pitch and roll) to a velocity output ( $x$  and  $y$  axes, respectively), i.e.,

$$H(s)_{x|y} = \frac{a_{(x,y)} \omega_{(x,y)}^2}{s^2 + 2\zeta_{(x,y)} \omega_{(x,y)} s + \omega_{(x,y)}^2}$$

where  $a_{( )}$  is a gain;  $\omega_{( )}$  and  $\zeta_{( )}$  are the resonance frequency and the relative damping ratio of the quadrotor translational dynamics along the  $x$  and  $y$ -axes.

Similarly, the transfer functions describing the translational dynamics along the  $z$ -axis and the  $yaw$  from position input to position output are given by

$$H(s)_{z,\theta} = \frac{a_{(z,\theta)}}{s^2 + b_{(z,\theta)} s}$$

where  $a_{( )}$  and  $b_{( )}$  are gains.

PID controllers on the desired velocities (in  $x$  and  $y$ ) and positions ( $z$  and  $yaw$ ) of the real slave, which are defined by the velocities and positions of the virtual slave, are realized based on the above dynamics. The fact that two different types of controllers, position and velocity, are used is to reaffirm that the teleoperation algorithm works in a plug-and-play fashion with any type of low-level controller without compromising the stability of the teleoperation loop [8].

The low-level translational dynamics control of the vehicle, particularly the  $x$  and  $y$ , are augmented by obstacle avoidance forces. This force is based on the concept of optical flow that compares estimates of time-to-contact and maximum stopping time to produce a force that becomes noticeable only when the vehicle approaches obstacles [10]. It is computed by

$$F^o = -\frac{k_o}{d_o} \dot{x}_s \quad \text{if } d_o < \underline{d} \quad (8)$$

where  $k_o$  is a constant;  $d_o$  is the displacement of the quadrotor from the obstacle, and  $\underline{d}$  is a threshold, below which the obstacle avoidance force is applied.

The effect of this force, in addition to the force felt due to the delays in the communication channel and in the dynamic response of the vehicle, is naturally felt by the operator through the haptic feedback. This is realized through the dynamic coupling between the real and virtual slave described in Sec.II-C.1. Fig. 1 shows the ideal physical model equivalent of the proposed teleoperation control scheme.

#### D. Controller Parameter Tuning

For the virtual slave to reliably convey the dynamic response of the real slave to the operator's and the environmental stimulus, its parameters should be tuned carefully. Let  $d_r$  collect all viscous damping effects applied on the real slave; with reference to Fig. 1, the dynamic map from  $\bar{F}_s^t$  and  $F^o$  to  $\dot{\bar{x}}_s$  and  $\dot{x}_s$  can be given by

$$\begin{pmatrix} \dot{\bar{x}}_s \\ \dot{x}_s \end{pmatrix} = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} \begin{pmatrix} \bar{F}_s^t \\ F^o \end{pmatrix} \quad (9)$$

where  $H_{ij}$ , for  $i, j = 1, 2$  are the transfer equations.

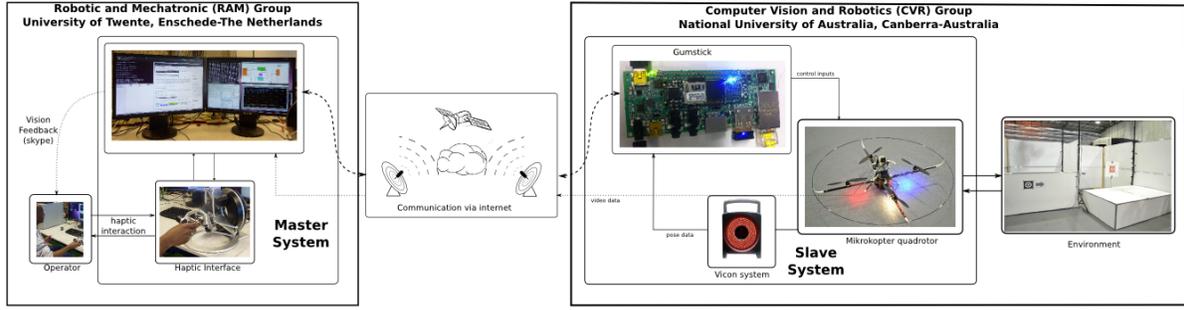


Fig. 2. The hardware structure of the experimental set-up used to realize the longest intercontinental haptic teleoperation of a flying vehicle. It is composed of the master system (the haptic interface and the base station) at the University of Twente (The Netherlands) and the slave system (the flying vehicle and the obstacle) at the Australian National University (Australia).

$$H_{11} = \frac{M_s s^2 + (d_c + d_r)s + k_c}{d(s)}, \quad H_{12} = H_{21} = \frac{d_c s + k_c}{d(s)}$$

$$H_{22} = \frac{\bar{M}_s s^2 + d_c s + k_c}{d(s)}$$

where  $d(s) = \bar{M}_s M_s s^3 + \{(\bar{M}_s + M_s)d_c + \bar{M}_s d_r\}s^2 + \{(\bar{M}_s + M_s)k_c + d_c d_r\}s + k_c d_r$ .

Loosely speaking, based on the chosen mapping strategy, the correspondence between  $\hat{x}_s$  with  $\dot{x}_s$  can be considered as the reliability measure for the virtual slave. Thus, tuning the magnitudes of the virtual slave's parameters as in Eq. 10 guarantees  $H_{11} \approx H_{21}$  and  $H_{12} \approx H_{22} \Rightarrow \hat{x}_s \approx \dot{x}_s$ .

$$\bar{M}_s \ll M_s, d_c \gg d_r, d_c \gg M_s, k_c \gg d_r, k_c \gg M_s \quad (10)$$

In almost all applications, safety has higher precedence than obeying operator's command. Hence, to avoid collision with an obstacle in cases where the operator's command would make it collide, the obstacle avoidance control should be tuned as  $d_t < \frac{k_c}{d}$ . As such, the response from the obstacle avoidance is much faster than the virtual slave's controller action that tries to track the operator's command.

*Remark 1:* During real-time applications, the virtual and real slave can be coupled through a viscous damper  $d_c$  alone. However, the correspondence between the positions of two slaves can not be guaranteed. Besides, the position discrepancy can not be haptically felt by the operator.  $\triangleleft$

### III. EXPERIMENT

In this section, we describe the hardware and software architectures implemented to realize the longest haptic teleoperation of a flying vehicle. We also provide experimental results that demonstrate the task performance of the proposed algorithm. During the experiment, we had the master system in the Robotics and Mechantronics Lab of the University of Twente, Enschede (The Netherlands), and the slave system in the Computer Vision and Robotics Lab of the Australian National University, Canberra (Australia).

#### A. Hardware Architecture

Fig. 2 shows the hardware components and the overall architecture. The main hardware components are

- Haptic Interface:* The Omega6 haptic interface from Force Dimension [15] is used as a master device.
- Base Station PC:* A desktop PC running Ubuntu 10.04 (32 bit) operating system has been used as a ground

- station. It is used to compute the force rendered by the haptic interface and as a graphical user interface.
- Flying Vehicle:* A Mikrokopter quadrotor equipped with onboard camera and computational resources is used.
- Onboard Camera:* An analogue camera with  $110^\circ$  field of view is used. The images obtained from this camera only are displayed to the operator via Skype<sup>TM</sup>. This is to preserve the generality of the hardware framework as most remote environments, where the slave flies in, are not equipped with off-board cameras.
- Visual Tracking System:* Vicon [16] is used to provide the absolute position and attitude data of the flying vehicle with respect to the reference frame of its workspace at 200 Hz. A velocity observer is implemented to estimate the velocity of the slave.
- Flight Area:* A closed flight area of  $1.8 \times 4.8 \times 4.8 \text{ m}^3$ , in which obstacles are presented is used as a flying arena.

*Remark 2:* The coordinate frames of the flying vehicle and the haptic interface, with respect to which their motions are described, are adjusted to correspond with the image feedback from the onboard camera.  $\triangleleft$

#### B. Software Architecture

The implemented software architecture is shown in Fig. 3. The software environment is mainly based on the ROS system. The primary modules for the slave system are: the *Vicon* module, the *Velocity Observer and Controller* module, the *Virtual Slave* module, the *Obstacle Avoidance Force* module and the *Uplink* module. The *Velocity Observer and Controller* module is implemented to perform the local control tasks on the flying vehicle by receiving the position and attitude data streamed by the *Vicon* module. The *Virtual Slave* module implements a virtual slave that couples the master and the real slave system. The velocity and position set-points are generated by the *Virtual Slave* module and sent to the *Velocity Observer and Controller* module as a reference input. The *Obstacle Avoidance Force* module implements Eq. 8. All the control signals are sent to the *Uplink* module and then transmitted by a standard radio transmitter to control the quadrotor.

The main modules of the master are the *Simulink* module, the *Omega6* module and the *Telefly Master* module. The *Simulink* module links the master controller, which is im-

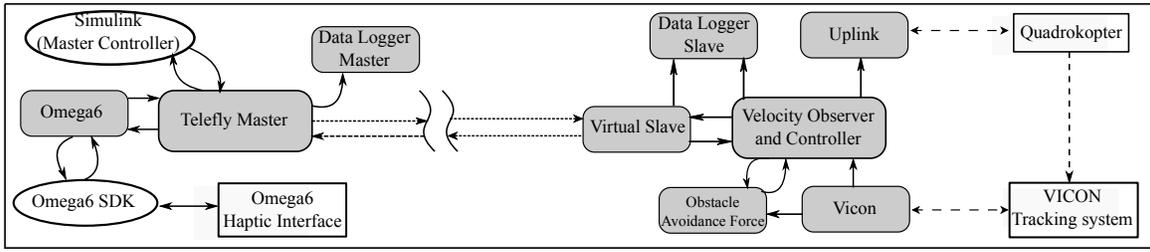


Fig. 3. The software architecture of the experimental set-up. The gray block are ROS nodes developed for the experiments. The unshaded rectangular and oval blocks represent hardware parts and additional software packages, respectively.

plemented in Simulink [17], and the *Telefly Master* module coordinates the task performed by the master system. The *Telefly Master* module exchanges relevant signals, such as the operator's command, the force feedback and the velocity of the slave with the master controller. The *Omega6* module is the interface between the Omega6 Software Development Kit (SDK) and the rest of the teleoperation control structure. It accepts the pose of the haptic interface and sends the desired force feedback to be displayed to the operator.

### C. Experimental Results

Two representative experimental tests that demonstrate the applicability and the effectiveness of the controller in the presence of variable time delays and packet losses are provided. In the experiments, the translational DoFs of the flying vehicle were haptically teleoperated. Though not haptically, the yaw of the flying vehicle was also controlled by the operator. The relevant parameters of the real slave and the teleoperation controller used in the experiments are given in Table. I. The control parameters are tuned according to the discussion in Sec. II-D.

*Remark 3:* During the experiment, a dead-zone around the neutral position of the haptic interface is implemented for ease of hovering command, which is difficult in rate control. Besides, filters are incorporated at the master and the slave sides to smooth out the data exchanged between them. It is important to note that the structure of the proposed controller ensures that somewhat arbitrary modifications of the control signals of this nature can be made without compromising the passivity of the overall system response. Hence, during practical implementation, it is easy to introduce signal conditioning that improves the performance of the system response without fear of destroying the system stability. ◁

TABLE I  
PARAMETERS {IN SI UNITS}

Quadrotor (real slave) parameters (scalar)	
$a_x = a_y = 0.445$	$a_z = 0.05, a_\theta = 0.38635$
$\omega_x = \omega_y = 1.187$	$b_z = -0.98, b_\theta = 10.44$
$\zeta_x = \zeta_y = 1.08$	$d = 0.1$
Quadrotor's PID controller ( $k_p(1 + \frac{1}{s} T_i + s T_d)$ ) parameters (scalar).	
$k_{p(x)} = k_{p(y)} = 24.5$	$k_{p(z)} = 98, k_{p(\theta)} = 580$
$T_{i(x)} = T_{i(y)} = 3.245,$	$T_{i(z)} = 11, T_{i(\theta)} = 1.52$
$T_{d(x)} = T_{d(y)} = 0.25$	$T_{d(z)} = 1, T_{d(\theta)} = 0$
Virtual slave and master parameters ( $I_{3 \times 3}$ is a $3 \times 3$ identity matrix.)	
$k_c = 35I_{3 \times 3}, d_c = 70I_{3 \times 3}$	$d_t = 17I_{3 \times 3}, \gamma = 0.2$
$M_s = 0.01I_{3 \times 3}, k_\chi = 0.5I_{3 \times 3}$	$k_m = 50I_{3 \times 3}, d_m = 10I_{3 \times 3}$

1) *Free Flight:* The results of this experiment show the effective tele-control of the slave during a free flight test in the presence of significant variable time delays and packet losses. For better comparison, the scaled motion of the master sent from the operator and received in the slave side, and the velocities of the virtual and the real slaves are shown Fig. 4. It can be seen from the figure that the operator's command is tracked by the virtual slave, and consequently, by the real slave due to the coupling between them. Closer look at Fig. 4 shows that the operator's command that are received by the slave side experience considerable variable time delays and packet losses. Fig. 5 shows the variable time delays registered in the forward and backward traveling signals with time-stamps. In this experiment, an average packet losses of 23% and 21% in the forward and backward traveling signals have been registered. The delays due to the dynamic response and the communication channel are clearly reflected in the haptic feedback displayed to the operator, see Fig. 6.

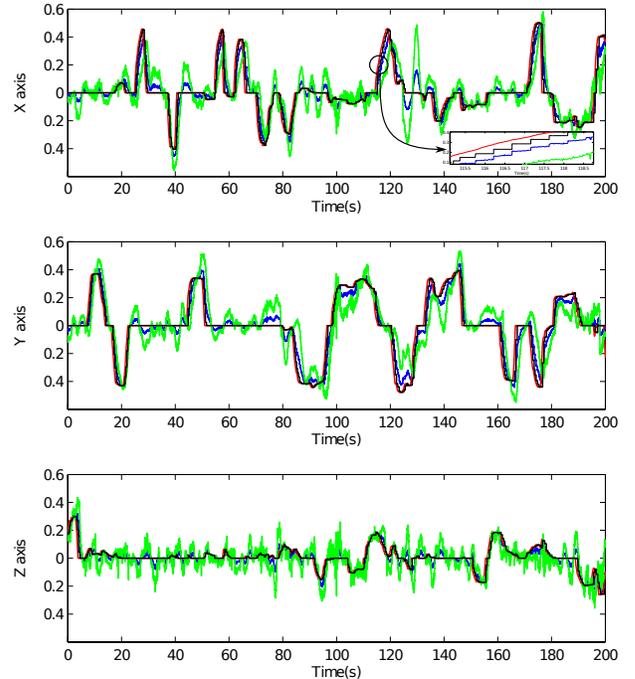


Fig. 4. Plots of the scaled command of the operator sent from the master (red) and received by the slave (black), and velocities of the virtual (blue) and real slaves (green). The positions and the velocities are in  $m$  and  $m/s$ .

2) *Maneuvering with obstacle avoidance:* The task performed in this experiment is flying through the structure shown in Fig. 2. The operator maneuvers the flying vehicle

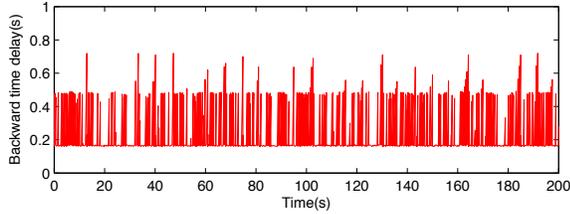
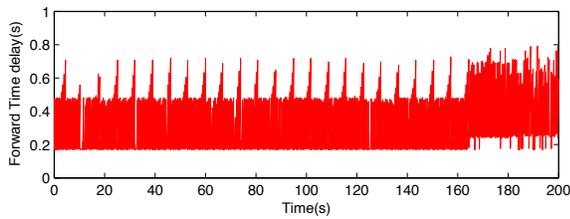


Fig. 5. Variable time delays in the forward and backward traveling signals.

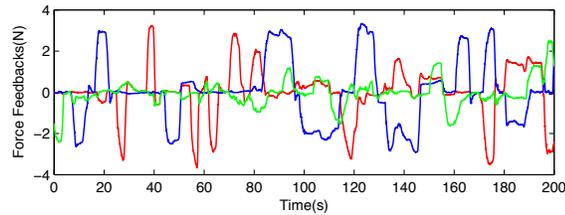


Fig. 6. Force feedbacks along  $x$  (in red),  $y$  in (blue) and  $z$  (in green).

through the structure and avoids collision with the walls and the supporting poles with the aid of the obstacle avoidance algorithm. The results of this experiment are depicted in Fig. 7-9. The command of the operator depicted is tracked by both slaves as shown in Fig. 7, until the obstacle avoidance force is applied on the flying vehicle. On the course of the experiment, the operator deliberately commanded the flying vehicle towards the walls and poles. However, the obstacle avoidance force slows the vehicle down once it gets closer than 10 cm. The environmental forces applied on the vehicle are also shown in Fig. 7 reduced by a scale of 10 for better comparison.

The deviation of the flying vehicle and the virtual vehicle from the command during the application of the obstacle avoidance force is reflected back to the operator through the force feedback, see Fig. 8. The variable time delays registered in this experiment are depicted in Fig. 9. Average packet losses of 19% and 22% have been registered in the forward and backward traveling signals.

#### IV. CONCLUSIONS

In this paper, we described the theory and application of a generic haptic teleoperation control algorithm for flying vehicles. We have described the theory for passive haptic teleoperation of aerial vehicles in the presence of time varying delays and considerable packet losses. We also discussed how to tune the parameters to better utilize the algorithm. System integration of both hardware and software frameworks, used to realize the longest intercontinental teleoperation of aerial vehicles, has been discussed. Verifying experimental results for teleoperation of a slave quadrotor located in Canberra (Australia) from a master station in

Enschede (Netherlands) have also been provided. It has been shown that the remote operator was able to safely maneuver the vehicle through a structure using haptic feedback of the state of the vehicle and the perceived obstacles.

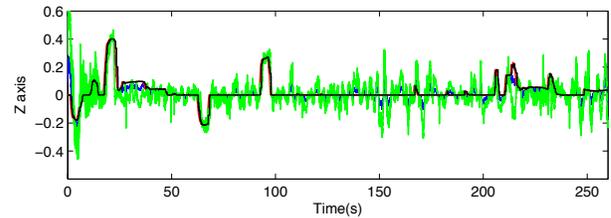
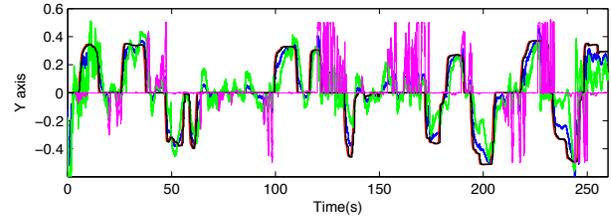
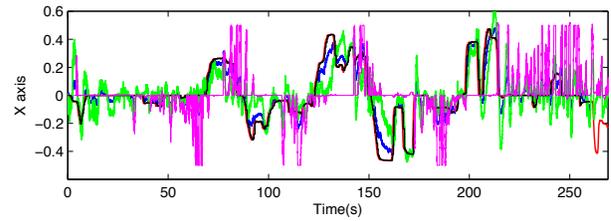


Fig. 7. Plots of the scaled command of the operator sent from the master (red) and received (black), and velocities of the virtual (blue) and real slaves (green). The positions and the velocities are in  $m$  and  $m/s$ . The obstacle avoidance force (in N) is indicated in violate.

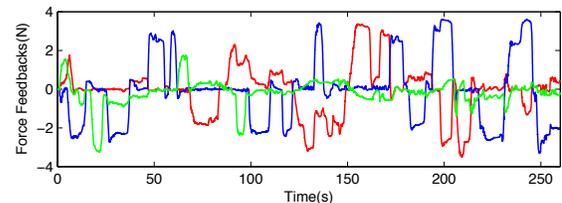


Fig. 8. Haptic Feedbacks

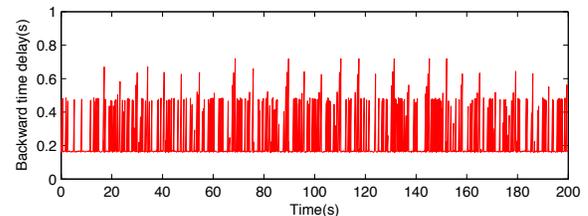
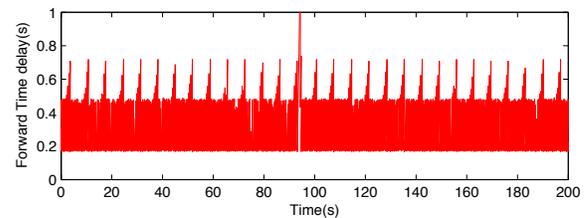


Fig. 9. Time delays in the forward and backward traveling signals.

#### REFERENCES

- [1] "Airobots." <http://www.airobots.eu>.

- [2] R. Mahony, F. Schill, P. Corke, and Y. Oh, "A new framework for force feedback teleoperation of robotic vehicles based on optical flow," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009.
- [3] T. Lam, H. Boschloo, M. Mulder, and 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.
- [4] A. Y. Mersha, A. Rüesch, S. Stramigioli, and R. Carloni, "A Contribution to Haptic Teleoperation of Aerial Vehicles," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2012.
- [5] T. Lam, M. Mulder, and M. van Paassen, "Haptic feedback for UAV tele-operation - force offset and spring load modification," in *Proceedings of the International Conference on Systems, Man and Cybernetics*, 2006.
- [6] H. Rifaï, M. Duc Hua, T. Hamel, and P. Morin, "Haptic-based bilateral teleoperation of underactuated unmanned aerial vehicles," in *Proceedings of the IFAC World Congress*, 2011.
- [7] A. Y. Mersha, R. Carloni, and S. Stramigioli, "Modeling and control of underactuated aerial vehicles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2011.
- [8] A. Y. Mersha, S. Stramigioli, and R. Carloni, "Bilateral teleoperation of underactuated aerial vehicles: the virtual slave concept," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2012.
- [9] F. Schill, R. Mahony, P. Corke, and L. Cole, "Visual force feedback teleoperation of the insectbot using optic flow," in *Proceedings of the Australian Conference on Robotics and Automation*, 2009.
- [10] X. Hou, R. Mahony, and F. Schill, "Representation of Vehicle Dynamics in Haptic Teleoperation of Aerial Robots," in *Proceedings of the Australasian Conference on Robotics and Automation*, 2013.
- [11] S. Stramigioli, R. Mahony, and P. Corke, "A novel approach to haptic tele-operation of aerial robot vehicles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2010.
- [12] A. Y. Mersha, S. Stramigioli, and R. Carloni, "Switching-based Mapping and Control for Haptic Teleoperation of Aerial Robots," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2012.
- [13] B. Hannaford and J. Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.
- [14] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, "Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency," *IEEE Transactions on Robotics*, vol. 27, pp. 741–756, aug. 2011.
- [15] Force Dimension, "Omega.6." <http://www.forcedimension.com>.
- [16] Vicon, "Motion Capture Systems." <http://www.vicon.com>.
- [17] MathWorks, "MATLAB/Simulink." <http://www.mathworks.com/>.