Haptic/kinesthetic feedback in Remotely Piloted Aircraft Systems (RPAS) human-machine interaction: final report

Lambert, Jason Michel

For the publisher's version, please access the DOI link below./ Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

https://doi.org/10.4224/40001215

NRC·CNRC

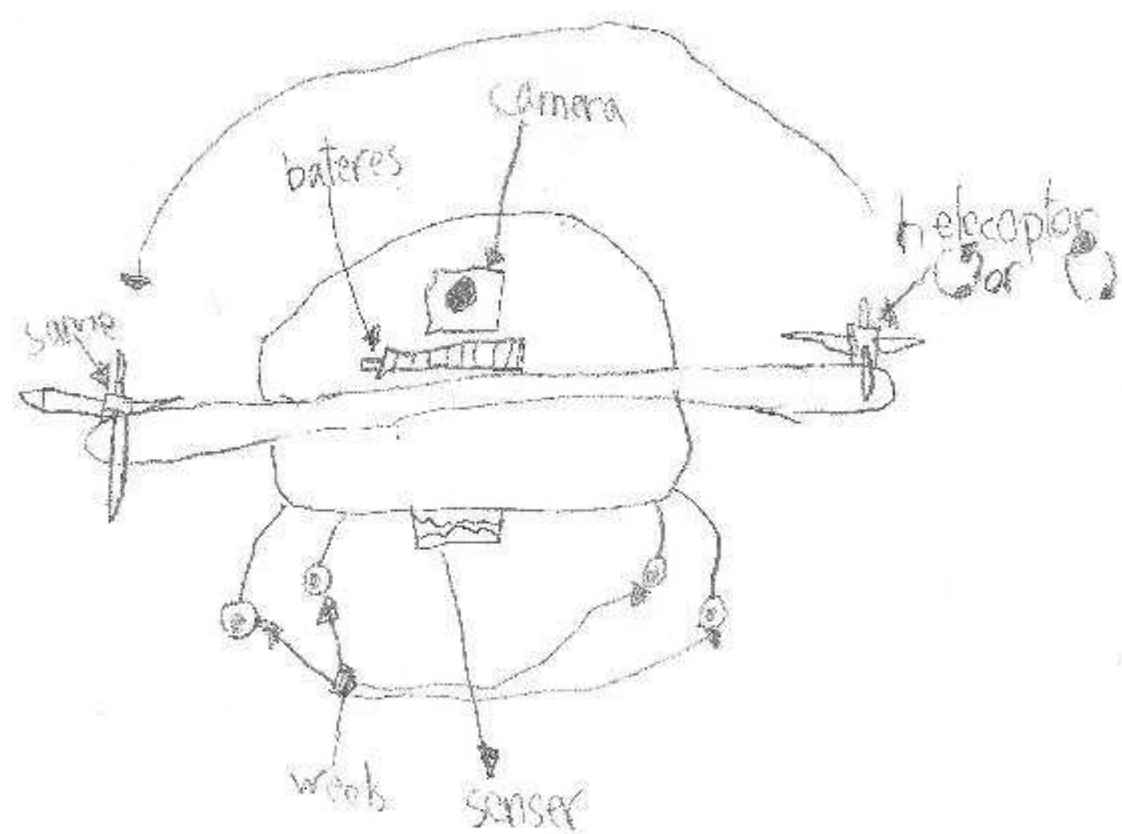
Enhanced Situation Awareness for RPAS Final Report

Date: April18, 2019
Authors: Jason Michel Lambert
Research Officer
Robotics and Autonomy

Aerospace Research Center

National Research Council Canada | Conseil national de recherches Canada

Canada


Camera
bateres
helecaptor
or
same
week
senser

National Research Council Canada

Haptic/Kinesthetic Feedback in Remotely Piloted Aircraft Systems (RPAS) Human-Machine Interaction: Final Report

Jason Michel Lambert

Research Officer
Robotics and Autonomy
Aerospace Research Center

April 18, 2019

Contents

1 Introduction

Over the past decade, Remotely Piloted Aircraft Systems (RPAS), which encompasses fixed-wing and rotary-wing aircraft of any size and payload-carrying capacity, have taken the technology landscape by storm. Partly enabled by affordable components and compact, powerful electronics, drones of various sizes and configurations became commercially available and the number of applications for which they are used skyrocketed at a pace that outran systematic Research and Development and Regulations. While some of their uses have minimal impact on public safety and does not require technological sophistication, more critical activities e.g. inspection of utility infrastructures, search and rescue, or activities in urban areas imply considerable risk and hence must be properly regulated, which implies technological compliance in order to guarantee performance and safety.

Activities requiring Beyond Visual Line Of Sight (BVLOS) operation certainly falls under that category due to the structure of the system itself. BVLOS activities both introduce physical remoteness from the site and communication delays, both of which have a profound impact on system configuration and behaviour. Failure of properly addressing these issues can endanger both the platform and the environment in which it evolves so these aspect must be well-known and regulated. Many have tried to automate RPAS navigation and mission execution however, this has proven to limit applicability due to time consuming planning and the inability of the system to perform tasks to their full extent automatically, such that an autonomous platform still requires human supervision and intervention. Automation and its younger, more adaptable brother Autonomy, may at first glance suggest a simplification and reduction of workload on behalf of the operator as it is after all one of its primary goals. However an automatic or autonomous system that does not take user interaction into account starting from its early design phases may actually have the adverse effect, bringing in complexity, added supervisory workload and difficulty in intervening online when human action is necessary. These factors, identified in a classic paper, have been found to be still relevant today [1].

Haptic feedback is considered by many to be an essential component of human-machine interaction and has been part of teleoperated systems since its early days, whether for performing remote manipulation of navigation. Moreover, haptic feedback has also be demonstrated to be a suitable communication medium for implementing shared automation, where humans and robots actively exchange ownership of tasks and control inputs [2], [3]. This report details the work performed in investigating the effects of using kinesthetic or haptic feedback for RPAS in BVLOS situations, and its role in improving system performance, stability and safety. First, a survey on the current research work being carried-out in haptic feedback applied to RPAS and teleoperation in general is provided. Next, effects of communication latency in remote operation of dynamic systems are are investigated, with and without haptic feedback. Next, a hardware/software environment for the study and prototyping of RPAS BVLOS tasks is introduced. Finally, conclusions and recommendations are provided.

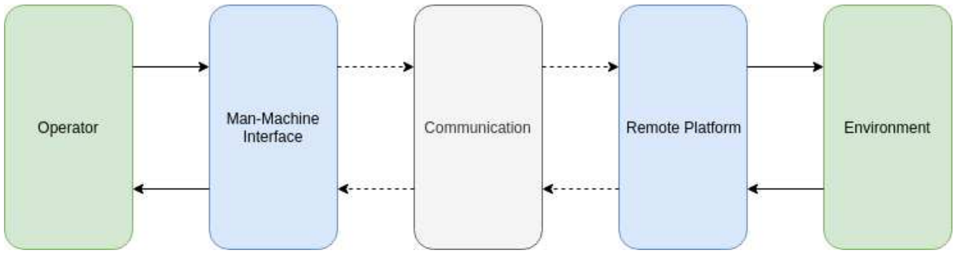


Figure 1: RPAS Dynamics General Architecture

2 RPAS General Architecture

Any remotely operated system can be represented as having 5 basic components arranged as shown in Figure 1. They are described as follows:

- **Operator:** The operator is in charge of sending commands to the platform so that is successfully interacts with the remote environment, and to make decisions based on feedback provided by the system. Due to the operator being remotely localized from the site, any feedback comes from the system itself; that is, the operator is never in direct contact with the remote site such that information goes directly from the site to him/her. Information fed back in a BVLOS context with always be produced by the system itself. It is thus evident that the quality of the decisions and resulting actions are directly related to the ability of the system in transmitting information efficiently.

- **Man-Machine Interface:** This is the interface through which the operator sends commands to the remote platform and through which feedback is experienced. The Man-Machine Interface typically consists of an input device through which motion and possibly force commands are sent, and a feedback component for informing the user of remote parameters. Feedback is most generally visual in nature, but can also include information conveyed back to the operator through the input device if the latter has actuation capabilities.

- **Communication:** Communication between the local and remote sites introduces delays between commands and resulting action. in RPAS in a BVLOS context, these communication delays can vary greatly from one configuration to the other, and for a given duration variability can be experienced depending on the communication medium. Along with contact at the remote site, communication delays are a primary source of instability in remotely operated dynamic systems. The stabilization in presence of communication delays typically introduces new dynamics that affects the overall behaviour of the system and may degrade performance.

- **Remote Platform:** The remote platform comprises of the payload, actuator system, sensors and accompanying electronics. Remote operation necessarily imparts a certain level of autonomy on behalf of the platform, whether for navigation, interaction and sensor processing. Once again, remote operation implies that the current state of the platform can only be assessed through system-generated information and must be provided to the user in one form or another.

- **Environment:** The environment includes everything external interacting with the platform, or being interacted upon at the remote site. These include fixed and moving objects, people, environmental factors (light, snow, wind, etc.) and such. All these elements can potentially destabilize the system, and must be addressed when planning a new mission or defining regulations. This poses a challenge as the potential applications of BVLOS RPAS are quite diverse, with the environment ranging from being well-known and structured to completely unknown, changing and unstructured.

3 Operator Considerations

Being part of the control loop, the state of the operator will have a direct impact on system behaviour. More specifically, the ability of the operator to perceive, understand, project and decide/act upon the system given the available information is crucial to performance and safety. Situation awareness is the general concept underlying these considerations, and will be influenced both by the operator state and the link between human and machine.

3.1 Situation Awareness

Situation Awareness(SA) can be viewed as the perception of environmental elements and events with respect to time and space, the comprehension of their meaning, and the projection of their future status [4]. Three levels are of Situation Awareness are distinguished, namely

- **Level 1: Perception**: Level 1 SA, the most basic level of SA, involves the processes of monitoring, cue detection, and simple recognition, which lead to an awareness of multiple situational elements (objects, events, people, systems, environmental factors) and their current states (locations, conditions, modes, actions).

- **Level 2: Comprehension**: The next step in SA formation involves a synthesis of disjointed Level 1 SA elements through the processes of pattern recognition, interpretation, and evaluation. Level 2 SA requires integrating this information to understand how it will impact upon the individual's goals and objectives. This includes developing a comprehensive picture of the world, or of that portion of the world of concern to the individual.

- **Level 3: Projection**: The third and highest level of SA involves the ability to project the future actions of the elements in the environment. Level 3 SA is achieved through knowledge of the status and dynamics of the elements and comprehension of the situation (Levels 1 and 2 SA), and then extrapolating this information forward in time to determine how it will affect future states of the operational environment.

The Situation Awareness Global Assessment Technique (SAGAT) is a tool to assess SA based on operator SA requirements. Using SAGAT, a simulation of the system of interest is frozen at selected times and operators are queried about their perception of the situation. SAGAT is designed to measure all 3 levels of situation awareness: Level 1 (perception data), Level 2 (comprehension of meaning) and Level 3 (projection of the near future), both for the system under operation and its environment [5]. The most important element to address to implement this approach is the formulation of queries in line with SA requirements analysis. Closely related is the necessity to formulate these queries in as simple and direct a form as possible. Requirements are formulated as a set of goals and sub goals that get associate with a decision resulting from the 3 levels of SA listed above. SA requirements should be automation-independent as much as possible that is, the SA requirements should stay the same regardless of the level of automation present in the system. Then, automation can be evaluated from an SA point of view. Focus should be put only on dynamic situational information affecting operators, and not on the correct application of pre-established rules. It is to be noted that the establishment of these requirements can take a considerable amount of energy, of the order of 1 person-year. SAGAT queries are then derived from those requirements. Properly made queries can help pinpoint situation awareness improvements that would otherwise require additional training to truly display the improvement. Although SAGAT has been used extensively in the aerospace domain, its uses also have been demonstrated in more general automation applications as well, such as nuclear plant monitoring. Studies in automated driving using SAGAT has shown loss in SA associated with high levels of automation. Recommendations pertaining to the implementation and administration of SAGAT are as follows:

1. **Training:** Participants should be briefed on the procedure, as well as go through 3-5 runs of practice responding to the queries before doing the actual runs.

2. **Test design:** No particular restrictions are present at the test design phase, other that plan some SAGAT-free test runs.

3. **Procedures:** Subjects should attend their tasks as they normally would; queries should be answered without additional visual aids or other. Best guesses are encouraged.

4. **Query selection:** In the presence of time constraints, queries may be randomly selected at each session. It is important that all selected queries

be answered because their relative importance is contextual, and an item judged not important may be resulting from lowered SA for this aspect. Shifts of SA can be misinterpreted as an SA increase in that domain so it is important to cover all aspects of the task.

5. **Timing and amount of data:** Timing of the simulation freeze for SAGAT administration can be randomly determined and should be unpredictable to operators. Attention should be paid to freeze simulation during all stages e.g. critical tasks and routine operations. Multiple stops may be accomplished at each trial. Freeze time should be constant regardless of the number of questions. More than one freeze may be implemented per run. Freeze times of up to 6 minuted have been tested without memory decay, and freeze times as long as 2 minutes have been implemented without subsequent performance degradation.

6. **Data collection:** Queries should be addressed one by one, in a manner that is straightforward to the user. Ideally, these should be implemented as a computerized survey. Scoring can be attributed based on an accurate response within predetermined tolerance.

As it is the case with manned aircraft, the majority of incidents with UAVs occur at the takeoff/landing phases. SA in RPAS requires detailed information on the environment as well as information about the platform health and status. Challenges for SA in remote operations stem from high workload and poorly designed interfaces reducing SA. These problems are detailed as such

- Poor sensory data, intermittent data and time lags
- Difficulties in unmanned vehicle localization. Vehicle orientation is particularly important to providing good SA
- Demanding tasks in complex environments. Visual scenes can be hard to interpret due to clutter, low lighting, and the poor quality of visual imagery that is provided through on-board cameras and sensors, often due to limited bandwidth restrictions
- Low-level data overload and interface design
- Multitasking in RPAS operations
- Lack of support for multi-person operations
- Increased autonomy for RPAS Operations. Autonomy that leaves the operator out of the loop contributes to decreasing situation awareness and can be detrimental to system safety whenever a critical situation requiring contextually relevant action occurs.

Design recommendations associated with implementing automation and supporting multiple-person team operations will be particularly important for the design of these systems. Although it is easy to create more problems than solutions by introducing automation, studies show that careful selection of tasks for automation and implementation of the automation can be beneficial in many types of unmanned vehicle systems. Other feedback channels can also be exploited in order to enhance situation awareness. Although auditory feedback is present in many teleoperated systems, there is not much research addressing its role in enhancing situation awareness, and how it can be used best [6]. Much like augmented reality, audio feedback can consist of simply reproducing sounds at the remote site, using specially designed audio (or audio processing on the environment) to provide additional information, or a combination of both. While outside of the scope of this project, it is believed audio feedback could significantly contribute to increasing the sense of being present at the remote site and thus, improve situation awareness.

3.2 Automation

Automation has been a driving force in systems design in various domains. While one of the primary goals of automation is to simplify an operator's task, a lack of consideration for user interaction at the design stages can have the exact opposite effect. The additional complexity brought about by the automated system can both confuse and cause an operator to disconnect from the system, resulting in the so-called out-of-the-loop syndrome [7]. In addition to a lack of understanding about the inner workings of the automated system, out of the loop syndrome can be triggered by a loss of vigilance associated with the passive nature of the operator's task. An operator with reduced active involvement with the system is likely to lose concentration and awareness such that important decisions or actions can be delayed or misled. Passive processing of information is done so in a much less tangible manner than active processing, much like a car passenger is less likely to remember the path took in getting to a particular place than the person driving. Rendering of information through the appropriate channel also greatly contributes to enhanced SA. Pilots using early fly-by-wire systems reported a sense of disorientation due to the lack of cues through the input stick related to aircraft dynamics, even if the data was present visually. Haptic cues are now an essential part of fly by wire systems. Too much reliance on one communication medium increases workload and reduce salience of monitoring quantities, creating a blur of de-prioritized information which numbs the operator. Another source of degraded SA when dealing with automated systems stems from the lack of a proper mental model of the system and its operating modes. Much research has pointed out the confusion associated with not knowing in what control mode the system is in. It seems like the operator needs a critical minimum understanding of the system's inner workings in order to "track" its behaviour and interact with it. New approaches to automation are being explored to address these issues. Adaptive Automation (AA)[7] seeks to systematically allow certain modes to be controlled by the human, either at certain intervals or driven by the following occurrences:

- Occurrence of critical events
- Detection of human performance below a certain level
- Psycho physical monitoring e.g. for detecting loss of awareness
- Phases during the automated process where human action is optimal.

Ongoing topics or research on AA include establishing methods for activating AA and determining optimal allocation strategies. AA requires enhancement of any interface in order to minimize potential SA challenges. The particular mode in which the system is of crucial importance to the operator interacting in AA fashion. Most of the AA research to date has been conducted under laboratory conditions. An alternate approach uses different levels of automation to keep the operator involved on an ongoing basis rather than subjecting her to periods of passive control. It has been found that involving automation at the levels of generation of options and decision selection tends to increase workload, whereas automation working at the information gathering and display level is more efficient. The following are guidelines for designing user-centric automated systems[7]

- **Automate only if necessary:** functions to automate should be defined from a user perspective, staring with an interface that supports enhanced SA and decision-making
- **Use automation for assistance in carrying out routine actions rather than higher level cognitive tasks**
- **Provide SA support rather than decisions:** A system that takes into account the broadest picture of the situation and informs the operator will be better at reducing workload than a system presenting only an "optimal" solution.
- **Keep the operator in control and in the loop:** Loss of mental engagement due to passive automation can have catastrophic effects in case of a failure. Intermediate and lower levels of automation are better at keeping SA high than higher levels of automation. The operator should be kept active in an ongoing fashion with high levels of task involvement.
- **Avoid the proliferation of automation modes:** The use of automation modes increases system complexity, and thus decreases the ability of operators to develop a good mental model of how the system works in all of its possible modes. More automation modes also make it harder to keep up with which mode the automation is in at the present time in order to develop correct expectations of what it will do next. A flexible tool that allows user customization of system functioning to fit the current circumstances is probably better than one that simply proliferates modes.

- **Enforce automation consistency:** The use of a consistent set of principles for system design that is followed across modes and displays can greatly minimize the errors that can come from automation use.
- **Avoid advanced queuing of tasks:** Automated systems that allow the operator to set up in advance a number of different tasks for the automation to perform are most likely to leave that operator slow to realize there is a problem that needs intervention. Interestingly, current FMSs in cockpits and batch processing systems in manufacturing work under just such an approach. Approaches that maintain operator involvement in the decisions for each task execution should be considered.
- **Avoid the use of information cueing:** Information cueing helps direct operator attention to those areas or pieces of information the system thinks are most important. Unless there is a very low probability of error, this type of cueing should be avoided in favour of approaches that allow people to use their own senses more effectively.
- **Use methods of decision support that create human/system symbiosis:** Other ways of combining people and computers to create more effective synergy between the two have been insufficiently studied (in favour of traditional decision support systems that supply the human with advice). Alternate approaches for decision support include
  - Supporting "what-if" analysis, encouraging people to consider multiple possibilities and performing contingency planning that can help people formulate Level 3 SA
  - Systems that help people consider alternate interpretations of data, helping to avoid representational errors
  - Systems that directly support SA through calculations of Level 2 SA requirements and Level 3 SA projections
- **Provide automation transparency:** Given that automation will probably be employed in many systems, providing transparency of its actions (current and future) will greatly reduce the problems leading to automation-induced accidents. While this may seem like an obvious thing to convey, in most automated systems the operator is left guessing as to why the system is behaving the way it is. Approaches such as these that make system behaviour clear to the user will greatly reduce many current problems with automation.

3.3 Operator Dynamics

A haptic shared control system allows a human and a machine's controller to share control of a dynamical process through a common interface. The haptic space is shared by both human and machine such that a human can accept proposed commands or overrule them in a continuous fashion, by tuning his own dynamic response. The relationship between force and motion in dynamical systems can either be described by impedance, which is the ability of the system to oppose motion by producing force, or its reciprocal admittance, the ability of the system to produce motion in response to force inputs. While haptic feedback is found in many remotely operated system, systematic procedure for tuning this feedback are scarce. Rather than trying to adapt the feedback to operator changes in real-time, [8] proposes a procedure based on one given "relaxed" dynamic response. Improperly tuned haptic feedback results in user discomfort and workload increase, which defeats the purpose. While a heuristics-based approach has been the de facto way of tuning haptic feedback e.g. for obstacle avoidance, it usually results in strong haptic guidance, which creates a bias toward automation, and makes overriding commands in a shared control difficult, creates discomfort and increases workload. As the human neuromuscular system is a main component of the feedback, its effect need to be taken into account in determining the final interaction impedance. Three levels of stiffness are distinguished for the feedback.

1. Intrinsic stiffness, where the feedback matches the operator's relaxed admittance
2. Lower than intrinsic stiffness, where feedback is below the operator's relaxed admittance, hence inducing a bias towards human control
3. Higher than intrinsic stiffness, where feedback is above the operator's relaxed admittance, hence inducing a bias towards UAS control

Operator relaxed admittance was identified using a method that suppresses the effect of reflexes in changing the endpoint admittance [9], [10], [8]. In tuning the haptic feedback, a UAV dynamic model was used representing an easy to fly closed-loop helicopter, mapping forward and sideways stick input to forward velocity and yaw rate, respectively. The environment consisted in virtually represented buildings, with the PRF collision avoidance field implemented [11]. Tuning was implemented directionally, as admittance varies spatially. Subjects were asked to fly in a simulated urban environment through waypoints while avoiding obstacles. Metrics used were similar to [12], with the addition of haptic activity and agreement between moments from haptic feedback and operator inputs. The following observations were made from the experiment and results:

- Inclusion of the neuromuscular response in the design process is essential in order to get the right level of feedback
- Overtuning the feedback response creates more workload than without any assistance. This suggest a strong correlation between proper tuning and assistance efficiency
- Tuning with satisfactory results occurs over a broad range of intrinsic stiffness and lower than intrinsic stiffness values, Disagreement can then be expressed by stiffening the arm.

- Safety (number of collisions) and performance (average velocity) were poor metrics for evaluating haptic feedback.
- Haptic feedback is particularly useful in lateral directions, where there is limited visual information available. There is greater chance of disagreement between the operator and controller when the visual content is good, so cues in this direction should be designed accordingly.

4 Communication and telepresence

Also referred to as teleoperation, the remote operation of robotic systems, whether for mobile manipulation, navigation or a combination of both, necessarily introduces dynamic elements additional to the platform's own dynamics that must be taken into consideration when assessing overall system stability and performance. In order to ensure proper tracking of the operator commands by the slave system in the presence of communication delays, noise and unaccounted for disturbances, remote controllers are often introduced. In addition, path tracking and forces experienced at the remote site are typically rendered back to the user through the master input device, such that operator commands presumably take these elements into consideration. The introduction of local and remote controllers and the presence of communication delays influences what is felt by the operator and his or her sense of telepresence (the feeling of being physically present at the remote location) and can thus affect stability and performance. A study on the effects of latency on situation awareness shows a direct link between communication delays and human performance, and outlines the necessity of having proper metrics to characterize this effect [13]. Furthermore, the presence of control loops couples the dynamics of operator, remote platform and environment such that they also become factors that must be taken into consideration for performance and stability assessment. The study and analysis of teleoperated systems is often carried-out using 2-port formalism, which provides means of modelling and designing teleoperated systems using a linear control approach. A general teleoperated system architecture has been formulated and the concept of transmitted impedance introduced in [14]. Transmitted impedance and its reciprocal, transmitted admittance, refers to the dynamics experienced at the master device interaction port and encompasses any behaviour resulting from platform dynamics, controllers, delays and such. From there, stability and performance indices may be defined and used in system design, and effects of operator dynamics can be taken into account[15]. A detailed literature survey on teleoperation and the influence of human operator dynamics can be found in [16]. The particular case of RPAS teleoperation brings in elements such as limited computing power at the remote site and variable-length communication delays that can directly influence overall system dynamics to the point where stability can be compromised and hence must be taken into consideration when designing or analyzing such systems [17], [18].

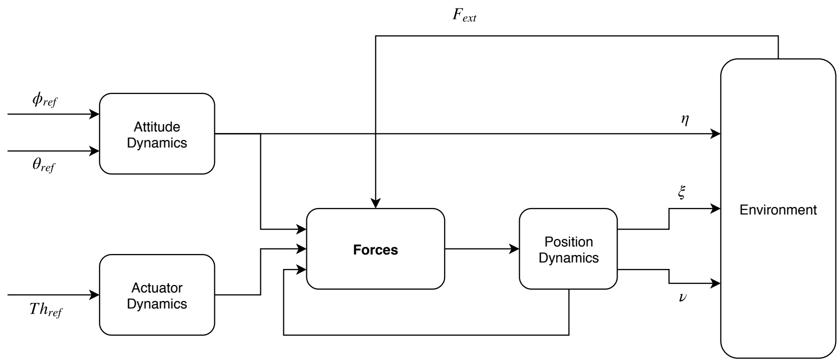


Figure 2: RPAS Dynamics Signal Flow

5 Platform Dynamics and Control

Of the many factors that must be monitored during RPAS operation, platform stability is a primary element. The ability of the platform to maintain its commanded configuration in the presence of uncertainties and external disturbances has a direct impact on the success of the task being performed. For example, platform stability may affect visual sensors' outputs, which in turn can affect localization, visual data recorded and inspection tasks. An unstable platform may be unable to dock or perform manipulative tasks. Instability can lead to loss of control resulting in crashing and/or damage to the environment. It is thus essential to be able to monitor platform stability. Notwithstanding control at the servo level, RPAS can be thought of as having two control layers, the attitude controller and the trajectory controller [19], [20]. The attitude controller is in charge of setting and maintaining the platform's orientation which is physically regulated by differential i.e. non-symmetric thrust delivered by the actuators (in the case of multirotor RPAS). The trajectory controller is in charge of maintaining Cartesian position with respect to a given reference frame. As the two (commanded position and attitude) are not in general decoupled, the two controllers must be cascaded i.e. the output of one becoming the input to the other. This dual nature is also present in the determination of the platform dynamics, where the attitude dynamics combined with actuator dynamics determine the forces acting on the platform, which in turn drive the position dynamics, as represented by Figure 2.

Here ϕ_{ref}, θ_{ref} and Th_{ref} are operator commands for desired roll, pitch and thrust, respectively. Also, η, ξ and ν are the platform's actual orientation, position and velocity vectors, respectively. Finally, F_{ext} represents external disturbances coming from the environment such as aerodynamic effects, meteorological disturbances (e.g. heavy rain, wind gusts, icing) or interaction forces. As an example, aerodynamic and blade flapping effects are studied in [21] and [22]. This dual nature of RPAS dynamics and control translates well into feedback to the user during operation, where Cartesian path tracking can be rendered back to the user in x, y and z directions (e.g. by forces displacing the hand), while attitude stability can be rendered in orientation space (e.g. by torques twisting the hand). Although there are a range of works using haptic feedback for obstacle avoidance, there does not appear to be work related to the rendering of vehicle dynamics via haptic feedback. Traditional approaches use input device position as inputs to the remote robot and generate forces according to data from the vehicle. In [23], input forces are used as motion commands, and joystick position is servoed to the velocity of the vehicle. Investigation of the transfer function for both approaches show that servoing position to vehicle velocity provides better low-frequency rendering than using the traditional approach. This may be desirable when haptic feedback for collision detection is used in conjunction with feedback for dynamics, because it helps decoupling both effects. Experiments were conducted where 6 subjects performed a positioning/stabilizing task, as well as a path following task in a cluttered environment. The following metrics were used

1. Rise time
2. Settling time
3. The NASA Task Load Index (TLX)[24]
4. Total flight time
5. Average velocity

Haptic feedback can be classified under two categories. Direct haptic feedback produces forces aimed at influencing an operator's behaviour e.g. by providing feedback on motion and/or force tracking. Indirect haptics, on the other hand, provides information to the operator but the choice of action is entirely left to the user e.g. vibrations to inform of a system failure of stiffening of the device in response to perturbations. The usefulness and applicability of both methods has been investigated in the context of wind gust effects rendering [25]. Wind gusts typically induce structural oscillations that are damped by the pilot. Indirect feedback was implemented by introducing a force resulting from the disturbance against the direction of motion, such that the pilot must oppose this force in order to restore the system to equilibrium. Direct feedback implemented force in the opposite direction, aiding the pilot. Proper tuning could have the device pilot the aircraft by itself, the presence of the operator increases robustness and performance. The dynamics of the input device was estimated using sine sweeps along with the Empirical Transfer Function Estimate function of the Mathworks' system id. toolbox. Indirect feedback is argued to be more sound from a neuromuscular control perspective, a force which must be opposed triggers the short-latency reflex thus reducing the delay that would be associated with a conscious action. Thus, initial response to the feedback is fast and reactive, with subsequent actions involving higher cognitive functions. The integral of the absolute error throughout the run was used as a performance measure. Results show that the Indirect method feels more "natural" and performs better in minimal training conditions. However, the direct method performs better and reduces workload after a period of training. In [26], the same study was performed in the context of a collision detection scenario. Direct haptic feedback, the method most commonly employed in the literature implements repulsive forces as a function of proximity to obstacles. Since attractive forces would drive the RPAS towards the object, indirect feedback was implemented just by shifting the neutral point of the restoring effect around, this way, motion of the platform due to this effect can be cancelled-out in a controlled manner. Three levels of visibility were implemented through fog simulation. It was found that under maximum degraded visibility, obstacle avoidance could not be carried-out without haptic feedback. Haptics can also be used to aid in path planning and kinesthetic boundary rendering. The input device is operated in admittance mode, where motion commands are derived from force inputs and servoes the device according to the platform's actual velocity. The planner generates goal points according to user inputs. Once a point has been set, a collision-free path is generated using a random search-based algorithm. Following creation of the path, virtual fixtures [27] are used to guide the operator along this path while allowing acceptable deviations. Proximity of obstacles will convert the virtual fixture feedback into a kinesthetic boundary, helping the operator to steer away from the obstacle. In [5], haptic feedback is used to render platform velocity. The effects of this feedback is evaluated for the most common modes of UAS operation. Haptic feedback was implemented as proportional to velocity and in opposite direction. As a means of comparison, haptic feedback as a function of distance to target was also implemented. Finally two operation modes, position and velocity inputs were tested. For the two operation modes, joystick displacements are mapped to roll, pitch and thrust. The experiment consisted of 4 waypoints that the platform had to pass through. It was found that the velocity mode is more intuitive and efficient overall, using metrics related to time required to complete the course and total distance travelled. These metrics were complemented with the NASA-TLX survey in showing the advantages of velocity input commands and velocity-driven haptic feedback.

In summary, several factors must be taken into account when designing and implementing haptic feedback for RPAS BVLOS operation. The following items are key elements to consider for such systems. Please note that these elements refer to the system itself (operator, input device, communication, platform) isolated from its environment. Factors related to the latter are listed in the following section.

1. **Choose an appropriate representation based on the nature of the feedback signal(s):** The term "haptic" is used with some liberties in this study to describe all forms of force feedback, from low-frequency kinesthetic feedback to high-frequency vibrations more closely related to the sense of touch. A choice along this spectrum should be made taking into consideration the kind of response expected on behalf of the user. For example, feedback on motion tracking is best conveyed by low-frequency feedback, whereas platform stability, a more indirect quantity could be

represented by vibrations with varying degrees of amplitude. Having an isolated bandwidth/power zone is also useful if feedback about two or more quantities must be provided.

2. **Assess signal quality and bandwidth:** In order to provide meaningful feedback without compromising system stability, candidate signals must fulfill requirements in terms of bandwidth and quality. Haptic feedback requires relatively high sample rates for rendering (500 Hz) and the presence of noise can severely degrade quality of the feedback. Some remote states may not be available for haptic feedback due to lack of bandwidth and/or noise. In these cases, strategies like locally generating the feedback signals based on remote telemetry, filtering and such should be considered.

3. **Design taking into account human neuromuscular dynamics:** Humans learn and physically interact with systems differently from one individual to the next, and depending on the task at hand. Stabilizing a task requires stiff dynamics, while becoming proficient at a task usually translates into more relaxed dynamics, but not necessarily so. These varying conditions influence how haptic feedback is perceived, and their interplay can to a certain extent be a source of instability. Implementation and tuning of haptic feedback should take both objective(e.g. through the use of dynamic neuromuscular models) and subjective (e.g. NASA-TLX, SAGAT surveys) assessments of human characteristics in order to maximize safety and usefulness of the feedback.

4. **Address stability issues first, performance later:** As mentioned repeatedly, providing haptic feedback can potentially destabilize the system, like any other closed-loop system. This is even critical when subject to communication delays such as is the case in BVLOS operations. The feedback parameters must be selected like any other control system, and stability is paramount. Techniques associated with linear control and robust control approaches lend themselves well for this, as many of the system characteristics (latency, operator dynamics) can vary during operation, albeit within bounds that can be pre-established.

5. **Determine the influence of feedback on remote perception:** Closing the loop and introducing controllers at various stages in the system will necessarily alter overall dynamics and will influence what is perceived at the interaction port (i.e. the input device), perhaps enhancing certain aspects and/or diminishing others. Transparency(the exact rendering of remote states unhindered by intermediate, system-dependent dynamics) is an idealized level of telepresence, however it is hardly attainable in practice. Assessment of the transmitted dynamics should be made and shaping the latter for optimal telepresence should be considered.

6 Feedback From the Environment

The lack of sensory information associated with teleoperation of remote platforms creates a need for status information to be conveyed to the operator. This is usually accomplished through a visual interface. In order to alleviate the information load displayed visually, haptic feedback can be used to render information about the remote site to the user. One particular area where haptic feedback appears to be efficient is that of obstacle detection and collision avoidance [11]. Haptic information can be generated from an artificial force field that maps environment constraints (real or artificially created) to repulsive forces. Known problems resulting from the use of artificial force fields are the existence of local minima and the difficulty of negotiating narrow passages. One additional challenge is the necessity of implementing haptic feedback without compromising platform stability and degradation of the control loop. The role of artificial force fields(AFF) or potential fields is to generate attractive or repulsive forces as a function of location. A repulsive force from an obstacle is generated by taking the gradient of the potential field. One of the first use of potential fields for robot manipulation was developed by Khatib [28] , later adapted for navigation purposes [29] and further enhanced in [30] for haptic feedback during teleoperation of Uninhabited Aerial Systems(UAS). The general principles behind these fields is as follows. Early versions were performing collision avoidance based on the distance between the robot and the object, the force being calculated from the field's gradient. For navigation, the robot's velocity component is used instead of the distance to object, resulting in the so-called Generalized Potential Field (GPF). Use of the robot's motion information allows the definition of the field based on the time required to decelerate (or accelerate) the robot to avoid the obstacle. Obviously, formulation of this potential field requires some knowledge about the environment, and its efficiency is influenced by the amount of knowledge. In a BVLOS context ,the environment is often partially structured or completely unknown, such that generation of the potential field is dependent on a combination of prior existing knowledge and the ability of the system to map its environment and localize itself within it through the use of onboard sensors. Limitations in the use of the GPF with hardware for haptic feedback lead to the creation of the Basic Risk Field (BRF), which a finite maximum value. The size of the field can be increased or shrunk, resulting in greater or less sensitivity to obstacles. In addition, an additional distance-related term is added to the field to take into account objects located on to the side of the platform, something not present in the original GPF. Parametrization of the size and shape of the field leads to the Parametric Risk Field. The BRF and PRF also simplifies the final avoidance force vector generating process by eliminating the need to use a gradient-based approach. Instead, the magnitude of the components is uniquely defined, with direction either chosen to be radial or tangential to the platform. Simulation-driven experiments involving obstacle avoidance in various configurations were carried-out and yielded good results for the BRF and PRF, the latter having slightly better performance and being less dependent on the choice of radial or tangential directions. Cases are as follows:

1. Change of direction due to the obstacle, with a sharp 90° turn. This can cause unstable motion near the wall.

2. Narrow, shrinking corridor. This typically induces oscillatory behaviour on behalf of the platform.

3. Stopping in front of a long, flat wall

4. Corridor ending in a wall

5. Passing between two close obstacles

6. Corridor with a bottom wall

Despite sophistication of the algorithm, its efficiency still relies a lot on careful tuning of the parameters, for which there does not seem to have a systematic procedure established. This dependency is further increased if one includes operator-related considerations e.g. endpoint impedance, workload and such. The main limitation concerning the results presented resides in the fact that the operator dynamics are not taken into consideration; haptic feedback will be felt differently depending on operator stiffness, his attention, the current manoeuvre she's performing, etc. Haptic information has to be conveyed in a meaningful and useful manner to the operator. More work also needs to be performed using humans in the loop in order to investigate when and how to render the information, and what kind of shared control can be implemented between man and machine. Finally, as with any teleoperated system, the effect of communication delays and sensor dynamics need to be taken into consideration. Providing kinesthetic feedback of environment constraints in a meaningful way is a challenge in part due to the differing workspaces between the robot and the input device. In [12], environment constraints are mapped onto the master input device workspace and a policy based on modifying the input velocity by a bounded velocity profile. The approach was validated against two other approaches, the PKF [11] and the time to impact (TTI) [31] algorithm. For the experiment, subjects were asked to steer a virtual UAS through an obstacle course in a simulated environment, while feedback was provided through an admittance-controlled input device. Results show good performance of the approach, based on the following criteria:

1. Total time from start to finish

2. Average velocity

3. Number of collisions

4. Mechanical work performed by the operator on the input device

5. The NASA Task Load Index (TLX), a global evaluation of cognitive load [24]

6. User questionnaire

One of the advantages of the proposed approach is the fact that the haptic cue for environment perception can be differentiated from the robot dynamics, as the former is implemented as a position constraint (a virtual wall) whereas the latter is rendered via dynamically changing forces. Implementation of the algorithm is also relatively easy. In addition to the considerations mentioned in the previous section, the following should also be considered in order to provide feedback about the environment and how the platform interacts with it.

1. **Assess the level of knowledge about the environment:** Differing environments have different levels of structure and prior knowledge about them. An RPAS operating in a manufacturing environment will benefit from a well-known, relatively structured and unchanging conditions, thus making it feasible to provide feedback based on a local, spatial representation of this environment. On the other hand, an RPAS operating in an unknown, unstructured environment such as the ones encountered in search and rescue activities will not benefit from such a pre-established representation of its environment. The more knowledge there is about the environment, the less tedious feedback implementation will be.

2. **Provide means of mapping the platform and localizing the environment:** Following from the previous element, in many cases haptic feedback about the environment will need to rely on *in situ* mapping and localization of the platform in this environment. A simulation environment would be useful in this case to validate the resulting environment and the capacity to generate meaningful feedback from it, which can be a function of sensors, signal quality and communication bandwidth. A 2-stage process (1- Mapping and 2-Operation with feedback) could also be considered

3. **Design with merging signals in mind:** A RPAS system providing haptic feedback will most likely prioritize platform-related states such as tracking, stability and command limits. Consequently, any feedback about the environment will need to coexist with platform-centric feedback. As with any other information channel, the haptic channel can also quickly overload so each type of feedback requires its own "space" e.g. low-frequency for tracking, high-frequency for stability, hard limits for environment constraints. When considering the environment, complementarity of visual, haptic and other types of feedback should be taken into consideration. A concrete example reported above is the use of haptic feedback to represent only constraints outside the operator's remote field of view, such as objects to the side or back of the platform, thus alleviating the haptic workload and minimizing redundancy of information.

7 Effects of Latency

As with any teleoperated robot, communication latency plays a fundamental role in determining operability of the system. Delays in communication in-

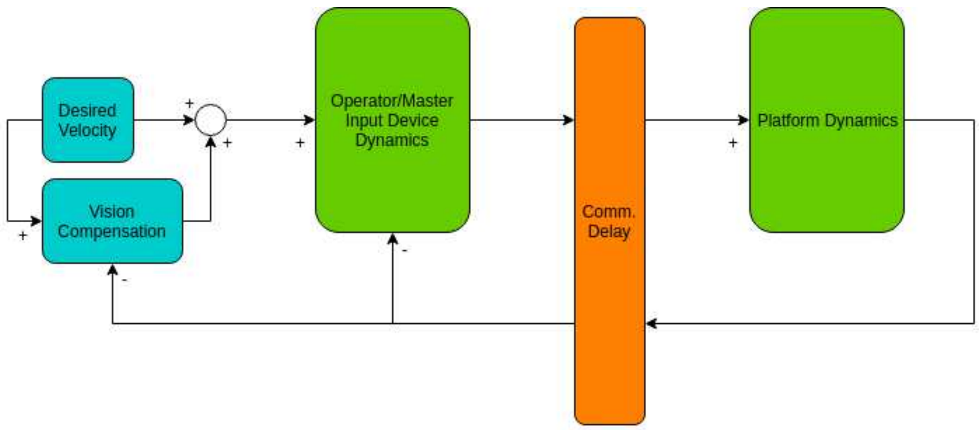


Figure 3: Model Schematic

formation from the remote site to the local operator station can have effects ranging from degraded situation awareness and operability to completely destabilizing the remote platform and compromising overall security of the operation. In order to concretely illustrate the effects on latency on a remotely operated dynamic system, a simulation was realized using very basic components so as to clearly outline these effects. The model is described and results presented in the following sections.

7.1 Model Components

The model schematic is shown in Figure 3. Components were kept very simple in order to isolate effects of communication delays. All dynamic components are passive such that they cannot be a source of instability in themselves. Two model configurations were implemented, corresponding to two operating use cases.

1. **No feedback to the input device:** In this configuration, no haptic feedback to the user is present. The only means of compensating for errors is through visual feedback of the delayed signals which generate an adjustment in desired velocity.

2. **Haptic feedback via the input device:** In this configuration, feedback is provided to the input device such that the device assists the operator in tracking the platform. Forces resulting from the action of the input device are felt by the operator, thus providing haptic feedback.

Components are detailed below.

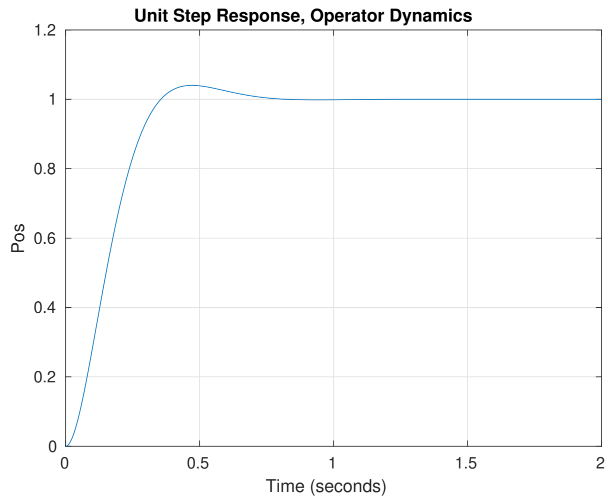


Figure 4: Operator Unit Step Response

7.1.1 Operator/Master Input Device Dynamics and Vision Compensation

In this model, desired motion commands are sent to a dynamic model of the upper limb, which outputs motion commands to the input device filtered by the inertial, damping and stiffness effects of the arm. The dynamics are implemented as a linear, time-invariant second-order model

$$Z_o = \frac{1}{m_o s + b_o + \frac{k_o}{s}} \tag{1}$$

where Z_o is the operator impedance, m_o is the mass of the arm, b_o and k_o are the arm's neuromuscular damping and stiffness characteristics, respectively. The isolated operator response to a unit position step input is shown in figure 4.

Compensation using visual feedback is implemented as a proportional-integral controller on delayed platform actual motion acting on the desired motion command

$$v_{cmd} = v_d(t) + (k_{Pv} + \frac{k_{Iv}}{s})(v_d(t) - v_a(t - \delta)) \tag{2}$$

where v_{cmd} is the commanded velocity to the operator dynamics, $v_d(t)$ is the desired velocity at the current time, $v_a(t - \delta)$ is the platform actual velocity at delayed time, and k_{Pv} and k_{Iv} are the proportional and integral gains associated with visual feedback. We assume the operator is trained enough that his compensation has both stiffness and damping effects. For the case where haptic feedback is present, an additional compensation term is introduced as another

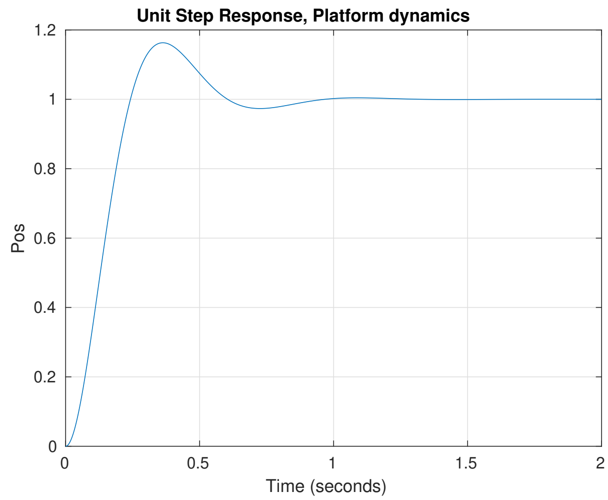


Figure 5: Platform Unit Step Response

proportional-integral controller, this time providing a driving force output at the operator/input device interface

$$f_m = (k_{Pm} + \frac{k_{Im}}{s})(v_d(t) - v_a(t-\delta)) \tag{3}$$

Such that the resulting commanded velocity to the platform becomes

$$v_{d_{pl}} = \int \frac{1}{m_o}(f_m + b_m v_{cmd} + k_m \int v_{cmd}) \tag{4}$$

7.1.2 Platform Dynamics

Here again, platform dynamics were implemented as a simple, passive second-order linear time-invariant system, similar to the operator dynamics

$$Z_{pl} = \frac{1}{m_{pl}s + b_{pl} + \frac{k_{pl}}{s}} \tag{5}$$

where Z_{pl} is the platform impedance, m_{pl} is the mass of the platform, b_{pl} and k_{pl} are the platform's closed-loop damping and stiffness characteristics, respectively. The isolated platform response to a unit position step input is shown in figure 5.

7.1.3 Communications Delay

Latency was implemented as a time delay on both incoming and outgoing signals such that at any instant t, velocity command to the platform is delayed by δ seconds and so is the actual velocity fed back to the operator and input device

Description	Variable	Value
Operator mass	m_o	$2\,kg$
Operator damping	b_o	$30\,\frac{Ns}{m}$
Operator stiffness	k_o	$200\,\frac{N}{m}$
Platform mass	m_{pl}	$1\,kg$
Platform damping	b_{pl}	$10\,\frac{Ns}{m}$
Platform stiffness	k_{pl}	$100\,\frac{N}{m}$

Table 1: Operator and Platform Parameters

$$v_{d_{plr}}(t) = v_{d_{pl}}(t-\delta), \quad v_{a_l}(t) = v_a(t-\delta) \tag{6}$$

where $v_{d_{plr}}$ is the velocity command to the platform at the remote site and v_{a_l} is the actual velocity feedback at the local site. A simulation was implemented to investigate system response with and without delays. Both cases described above were implemented in Simulink, using a fixed step size of $0.001s$. Parameters for the operator and platform dynamics were kept constant for all cases and are listed in Table 1. It is to be noted that the particular values for these parameters are not optimal. Rather they are used as baseline dynamics to compare undelayed and delayed responses.

7.2 System Response Without Direct Haptic Feedback

The first case looks at system response without dynamic feedback. In addition to the above-listed parameters, the following parameter values were used for vision-based compensation $k_{Pv} = 15$ and $k_{Iv} = 50$. Gains were selected in order to give the platform a critically damped response, although they are not optimized for performance. The system was first simulated without any communication delays, and the response to a unit step input shown in Figure 6.

Results show that the operator constantly adjusts his commands in an oscillatory fashion in order to efficiently regulate the platform's motion. which has a critically damped response. In contrast, response of the same system subject to the same inputs but with a communication delay of $\delta = 0.05s$ show unstable behaviour quickly settling in.

7.3 System Response With Direct Haptic Feedback

The second case was simulated under the same conditions, with the addition of a control loop at the input device level. Physically, this requires the input device to be backdrivable, and to possess actuation capabilities so that both input device and platform can track each other, thus synchronizing their motion.

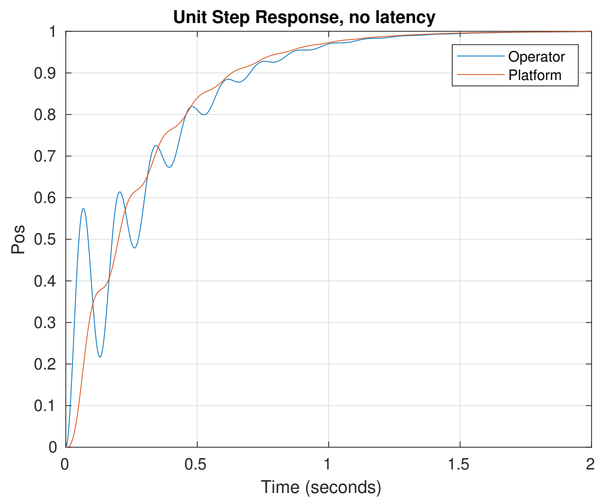


Figure 6: System Unit Step Response Without Haptic Feedback, No Latency

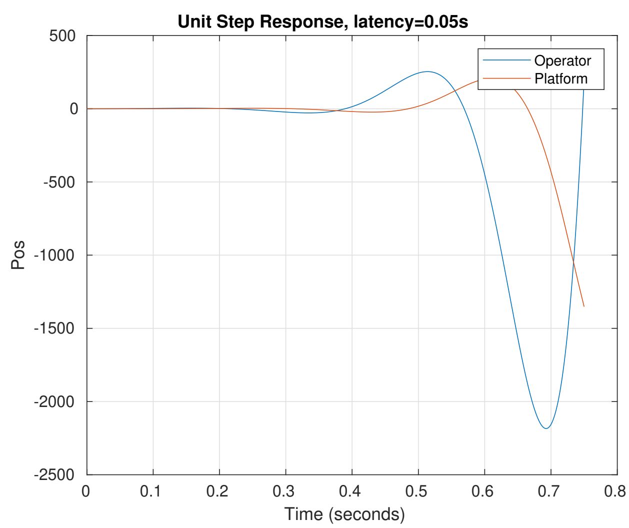


Figure 7: System Unit Step Response Without Haptic Feedback, 0.05s Latency

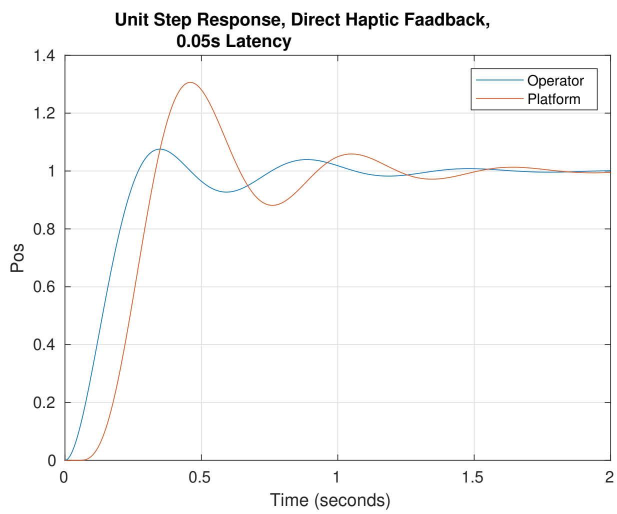


Figure 8: System Unit Step Response With Direct Haptic Feedback, 0.05s Latency

Actuation is felt at the operator/input device interface as a force, thus providing haptic feedback as a function of system tracking, again a basic, traditional use of feedback in teleoperated systems. The following control parameter values were used for this simulation: $k_{Pv} = 0.05$, $k_{Iv} = 0.2$, $k_{Pm} = 5$ and $k_{Im} = 25$. It is interesting to note that the gains for the visual compensation had to be lowered in the presence of haptic feedback. This underlies the fact that haptic feedback reduces dependency on visual feedback in remote, BVLOS situations. Response of the system to a unit step input in the presence of a $0.05s$ time delay is shown in Figure 8. Results show that the system now behaves in a stable manner, where instability was evident when haptic feedback was absent. As a more dramatic demonstration of the effect of feedback, response of the system was simulated with a time delay of $0.5s$, an order of magnitude greater than the previous case, without changing the control gains. While the response takes longer to settle and has a more oscillatory behaviour, it remains nevertheless stable, as shown in Figure 9.

Of course the situation is more complex in reality; humans do not keep constant stiffness and damping in time, rather, these quantities change according to multiple factors during operation, and can themselves be a source of instability. Numerous additional delays come into the picture, coming from the neuromuscular system dynamics or delays attributed to limited situation awareness. Nevertheless, this simple model is enough to show the effects of adding a control loop at the local site.

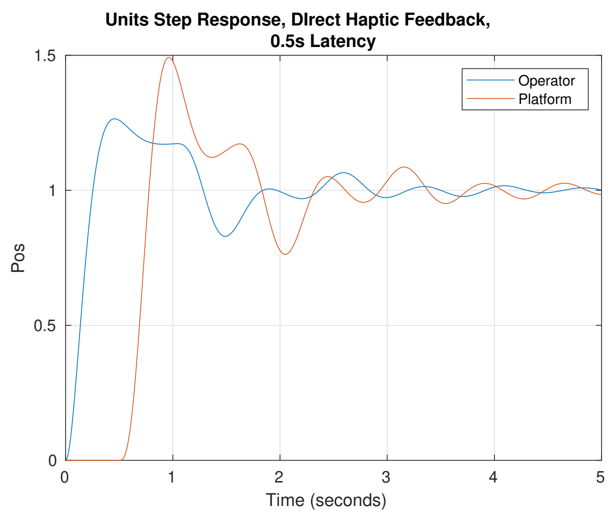


Figure 9: System Unit Step Response With Direct Haptic Feedback, 0.5s Latency

7.4 System Stability and Performance Metrics In the Presence of Latency

As discussed above, latency in RPAS can severely compromise safety and degrade overall performance. It is thus important to provide measures of the effects of latency and establish the safety envelope of the operator/input device/communication/platform/environment system. Below are provided metrics that can be used to assess system stability and establish the safety envelope inside which a RPAS can be safely operated. These measures are provided according to system configuration i.e. 1- with no feedback except visual compensation, 2- with motion feedback, and 3- with both motion and force feedback, for instances where the platform physically interacts with its environment.

7.4.1 Systems with no feedback

In configurations where no feedback is present, the system relies entirely on the ability of the operator to close the loop. Hence, stability will depend on many factors that are difficult to characterize such as situation awareness, passivity of the neuromuscular system, efficiency of visual compensation and such. These quantities, although bounded, vary from one operator to the other and, for a given operator, during operation. As such there is no analytical way to guarantee stability in the presence of arbitrary communication delays. Bounds must be imposed on system parameters so that the platform will always be operable within safety limits. The following quantities can be used to assess stability and define a safety envelope.

1. **Velocity limits:** Given the variability and difficulty in properly characterizing human behaviour, it would be unrealistic to provide quantitative human metrics. Instead, providing limits to the commands that can be sent to the platform is a more robust way of guaranteeing stability and providing a measure of safety. It is recommended to determine these limits empirically in a relevant application context, as they can vary depending on the platform used and the nature of communication delays.

2. **Platform response:** Another way of assessing the level of stability for a system with no feedback is to look at the overall response of the system, again in an application-relevant context. A system close to instability will have a tendency to have an underdamped, oscillatory response to remote commands. This can be used to determine system limits for safe operation.

3. **Comparing to baseline results:** A more empiric approach can be employed by comparing system behaviour with and without communication delays. Latency is relatively straightforward to simulate so that a given system configuration can be studied when subjected to various forms and levels of latency.

7.4.2 2-Channel Systems

The introduction of a feedback loop opens-up applicability of several analysis tools belonging to the domain of linear control, provided the system can be represented by linear, time-invariant components. While this can appear restrictive, this approach has been used in countless remotely-operated systems design and analysis, and can provide significant insight into their behaviour. Furthermore, the use of passive systems to represent operator and platform dynamics greatly reduces complexity of the analysis, as any passive system will remain passive when interacting with other passive systems. Consequently, given a passive operator and remote environment, the only necessary requirement for overall passivity is that the remaining elements also be passive. If the platform is not exchanging significant work with its environment, it may be sufficient to only transmit motion information between the local and remote sites. This configuration is known as 2-channel bilateral teleoperation. As these systems are single-input and single-output(SISO), they can be represented by transfer functions and analyzed in the frequency domain. Th following are guidelines and approaches to assessing stability of 2-channel RPAS.

1. **Teleoperator transfer function:** If we refer to the system described by Figure 3, and assume the operator and platform have passive dynamics, passive vision compensation and a combination of input device controller and communication delay that demonstrates passivity will guarantee overall system stability. In other words, the overall system will remain stable if all components can be shown as passive.

2. **Platform response:** Similarly to the previous case, platform response can give an idea of overall stability. An undamped, oscillatory behaviour

is an indication of potential instability, especially since it is expected that the platform's behaviour is stabilized by its flight controller. Here again, metrics can be established relative to a zero-latency baseline scenario.

3. **Establishing bounds on latency:** Robust control techniques can also be employed to make the system stable over a range of latencies. This is again application-dependent, as latency can vary to greater of lesser degrees depending on the communication medium, available bandwidth, etc.

4. **System Identification:** Lastly, in order to provide relevant analysis, relevant parameters must be provided. Quantities such as degree of latency, platform inertial components and closed-loop response must be properly identified, along with an appropriate model representation. Second-order linear systems are a good starting point for dynamic mechanical systems but their applicability must be assessed given the system at hand. Human dynamics typically vary over time, and their passive nature can be compromised by limited situation awareness and lack of attention. If analytical tools are to be developed for safety assessment, efforts should also be put into developing proper system identification methods as both go hand in hand.

7.4.3 Extension to 4-Channel Systems

While the use of a single-input, single-output 2-channel system representation is sufficient when only motion commands are involved, some instances may arise where interaction forces must also be remotely controlled e.g. when the remote platform performs manipulative tasks on the environment. for these cases, both motion and force signals are bilaterally exchanged at the local and remote sites, requiring the use of 4 communication channels. An extended formalism must be then used to describe the system's dynamics which has now 2 inputs and 2 outputs. Rather than using transfer functions, these so-called 2-port systems are described using immitance matrices which define the relationship between the effort(force) and flow(motion) variables involved. As an example, one such matrix is the admittance matrix which relates input/output motions to input/output forces

$$\begin{bmatrix} v_o \\ v_{pl} \end{bmatrix} = \begin{bmatrix} y_{o_{11}} & y_{o_{12}} \\ y_{o_{21}} & y_{o_{22}} \end{bmatrix} \begin{bmatrix} f_o \\ -f_{pl} \end{bmatrix} \tag{7}$$

where v_o and v_{pl} are the local and remote velocities, respectively, f_o and f_{pl} are the local and remote forces, respectively, and $y_{o_{OJ}}$ are the terms of the admittance matrix. The admittance matrix holds information about the communication delays and various control loops present in the system, and is typically quite complex and tedious to analyze. While numerous approaches have been devised to render such systems passive[16], [32], [33], assessment of its stability characteristics is best described by the so-called LLewellyn stability criteria[34]. This criteria distinguishes between passivity and stability, the former being a sufficient but not necessary condition to the latter. Consequently, the use of LLewellin's stability criteria imposes less restrictions on performance in order to guarantee stability. A more detailed treatment of 2-port systems is left out of this study and can be addressed when treating the specific case where 4-channel bilateral RPAS are required.

8 Simulation Environment for BVLOS Remote Operation

In order to study the effects of haptic feedback during remote operation of UAS with humans in the loop, a simulation environment incorporating hardware equipment was realized. The following considerations were taken into account in designing this environment.

1. **The ability to interface with external, haptic feedback-enabled input devices:** One of the main motivations for this design, the system needed to interact with human operators through an input device capable of haptic feedback. This required Real-Time, Hardware-In-the-Loop(HIL) capabilities for providing feedback to the user based on motion and/or force information coming from the remote site.

2. **An architecture representative of a typical BVLOS RPAS system:** The system needed to have all components that are found in a RPAS system: Master station, slave station. communication channels. Furthermore, the architecture was kept as modular as possible so as to incorporate hardware elements without major changes.

3. **The possibility of adding custom signal processing and control algorithms:** As this environment is to be used for development as well as testing, the ability to quickly prototype signal processing and control algorithms was required along with access to analysis tools.

4. **Being able to quickly prototype physics-based environments:** The virtual environment needed to be reconfigurable so as to quickly iterate on different scenarios, with physics-based (e.g. inertia, friction, collision) elements.

5. **Being able to quickly modify platform dynamics:**Similarly, the system needed to allow for quick reconfiguration of the platform's dynamics.

6. **The ability to interface with a UAS hardware controller in a Hardware-In-the-Loop configuration:** A flight controller is a complex device whose particular implementation can greatly influence overall system behavior. In order to maximise realism of the simulations, a real hardware controller was used as a basis for developing the system

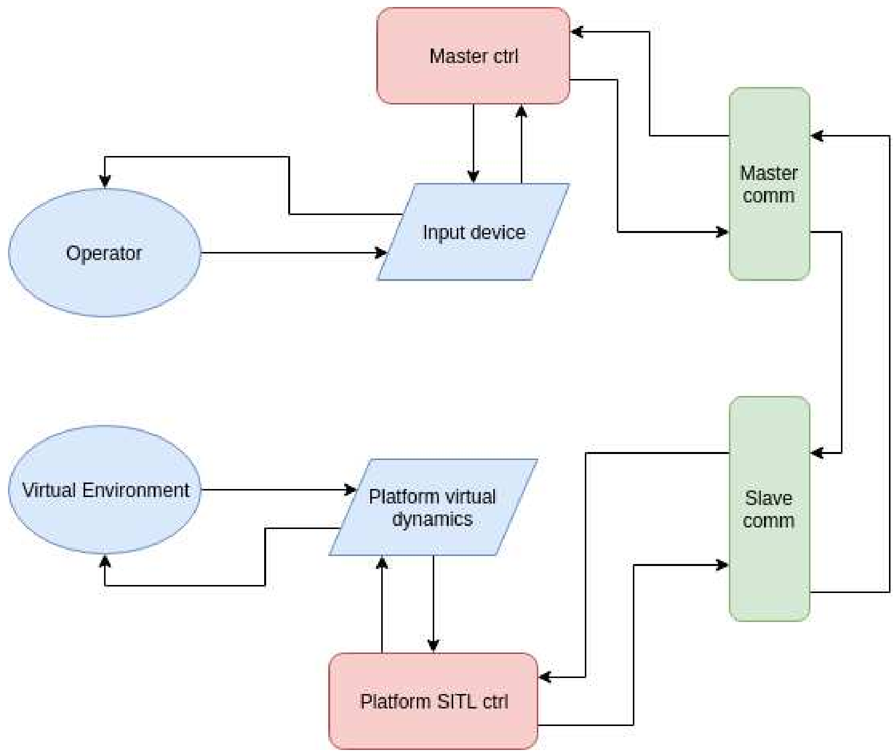


Figure 10: Simulation environment schematic, software configuration

7. **Online signal access for diagnostic and monitoring:** To simplify monitoring and analysis, an architecture providing online access to as many signals as possible was made a priority.

The simulation environment was developed with 2 configurations in mind, represented by figures 10 and 11. The software configuration runs a software emulation of the controller which controls a virtual platform evolving in a virtual environment. The Software-In-the-Loop(SITL) controller is an exact replica of the hardware controller, compiled for execution on a computer. The hardware configuration uses the real physical controller in the loop and communication channels that would be used in a BVLOS configuration. In order to make the system more convenient for development purposes, motion is provided by a robot manipulator on which the controller is mounted, thus providing a reference for location and velocity. Motion commands are sent to both the controller and the robot manipulator. Details of each configuration's implementation are provided below.

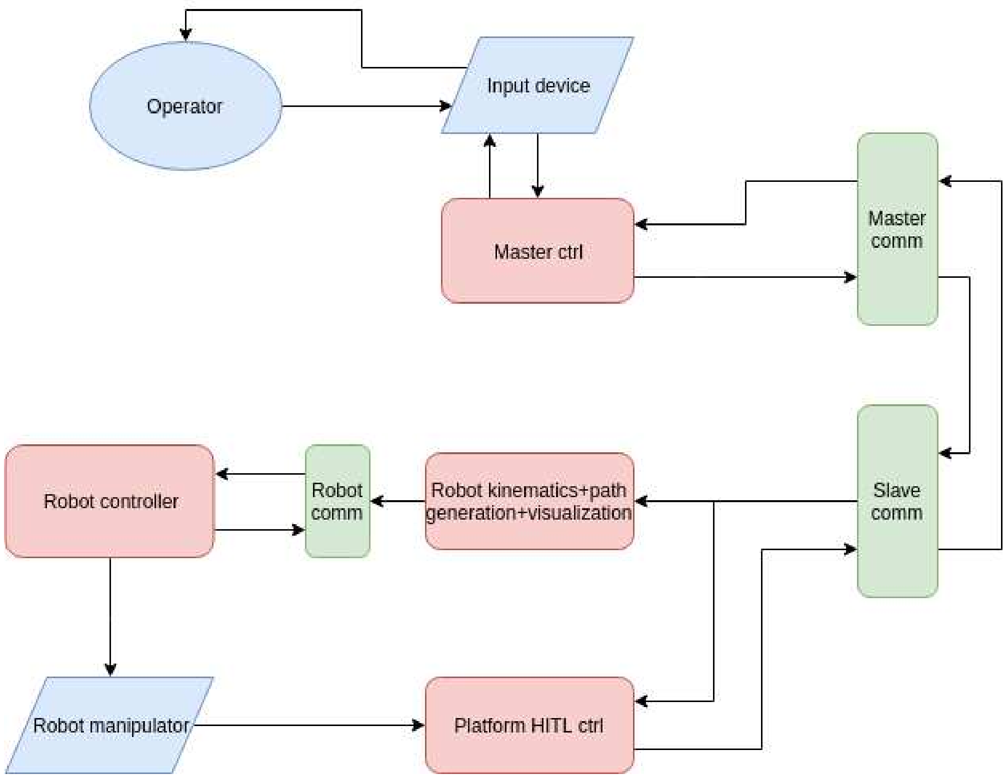


Figure 11: Simulation environment schematic, hardware configuration

8.1 Software-In-the-Loop Configuration

Implementation of the SITL configuration is shown in figure 12. The system runs on 2 separate computers, one associated with the local site and one associated with the remote site. The local computer (see figure 13) runs a QuaRC real-time systems design software (https://www.quanser.com/products/quarc-real-time-control-software/) which allows the real-time implementation of models created using Mathworks' Simulink. This real-time system is controlling a haptic input device (https://www.quanser.com/products/hd2-high-definition-haptic-device/) through which an operator sends motion commands and receives feedback. The real-time controller can also be used to run local models aiding in dealing with latency issues. Communication between the local and remote stations is accomplished via UDP-IP. While this is not representative of a real-life scenario which would be using Radio Frequency(RF) channels, characteristics of RF communication can be emulated on the local or remote station. The local and remote stations exchange motion data (desired and actual) bidirectionally. The remote station (see figure 14) runs the virtual environment (http://gazebosim.org/), the SITL controller and a robotics application development framework (http://www.ros.org/) which handles communication between the SITL controller and the rest of the system using the MAVLINK protocol.

The whole simulation environment runs at a base rate of $1ms$ however the various modules exchange data at different rates, which can introduce limitations in the system. It was found that the telemetry being output from the SITL controller occurs at a rate which could compromise stable feedback. Figure 15 shows the original motion feedback, in yellow, being output at a rate of $12Hz$. Filtering the signal using a Butterworth Low-Pass filter smoothens the curve considerably (in blue), but introduces a slight delay. Errors between the commanded and filtered actual velocities are shown in figure 16.

In a simulation environment involving multidomain components, it is important that a common timebase is maintained so that synchronisation between the various modules is acheived. This sychronization must be strictly enforced when communicating with hardware i.e. closing the loop with the haptic input device in our case, but can accept some jitter for purely software elements. In order to verify that the whole environment runs in a synchronous manner, time from the hardware real-time clock was compared to a software clock generated from the simulation environment.

$$t_{diff} = \lfloor t_{wall} \rfloor - t_{sim} \tag{8}$$

Where $\lfloor t_{wall} \rfloor$ is the wall time coming from the real-time clock, rounded-down to an integer and t_{sim} is the simulation time and t_{diff} is the time difference. Figure 17 shows that the time offset stays constant throughout the simulation, which implies that everything is (soft real-time) synchronized and no component is drifting with respect to the wall clock. Jumps between to two constant values are consequence of the roundoff, an indication of jitter in the synchronisation. Indeed, looking at the communication delay shows strong jitter, as displayed in figure 18. The trace represents the time required for a value

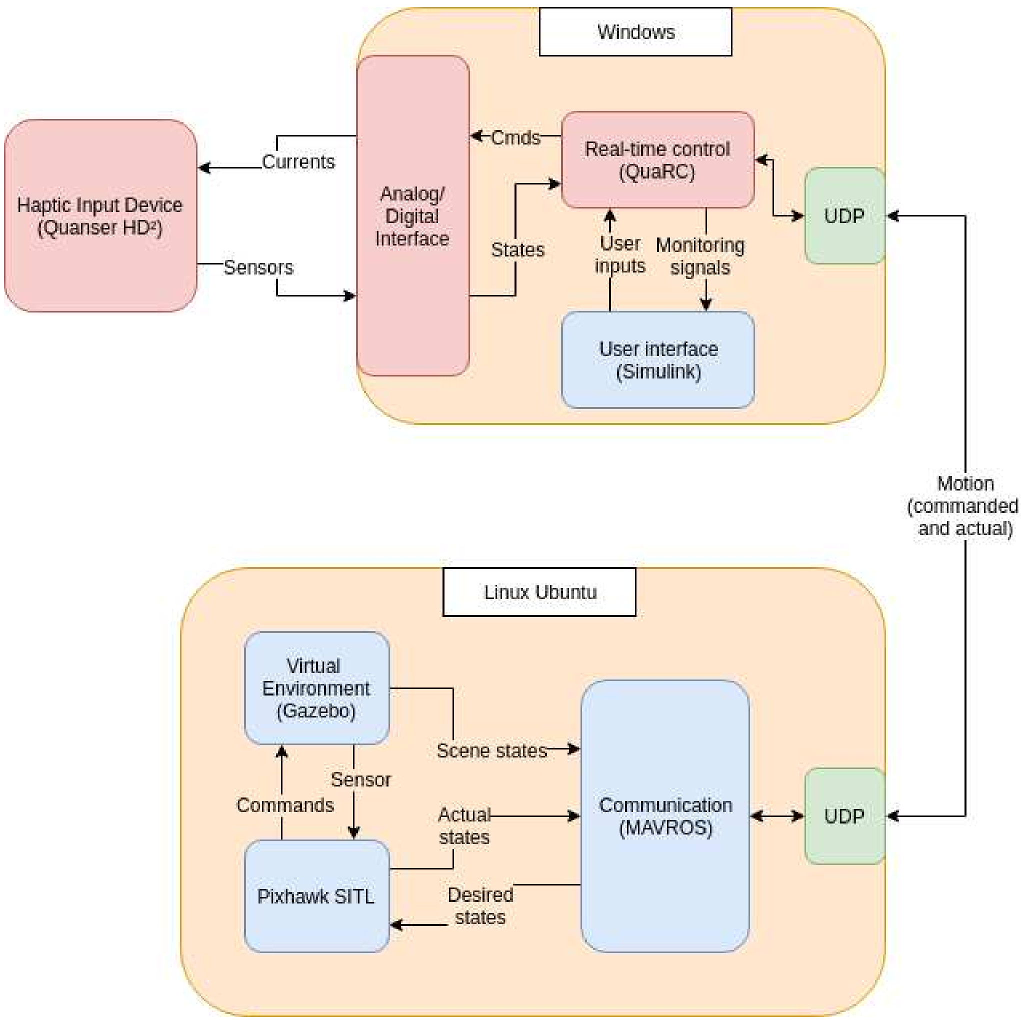


Figure 12: Simulation environment schematic, software configuration

Figure 13: Local station and input device

to do a round trip from the local station to the remote station and back. There is considerable delay ($0.3s$) along with jitter (50%). The latter is acceptable due to the operating system not being real-time however the delay times are quite high. A good portion of this delay is associated with subscribing to and accessing data from the ROS environment as implemented in Simulink, regardless if the simulation is run from Simulink or from an automatically generated ROS node. The same roud-trip delay was generated without the model subscribing to any ROS topic and is considerably lower($0.08s$), as shown in figure 19.

8.2 Hardware-In-the-Loop Configuration

As depicted in figure 11, the hardware-in-the-loop configuration of the simulation environment uses a real flight controller to accept commands and generate feedback. While controller behavior is expected to be identical to the STIL version, this configuration has the advantage of using communication channels which are representative of a real, physical scenario. In order to provide control over motion parameters and to easily locate the controller, the latter is being mounted on a robot manipulator which receives the same motion commands as the controller.

Implementation of this HIL environment uses the same local station configuration. The remote station is still based on the same architecture as previously described, with the addition of the robot manipulator's trajectory generation

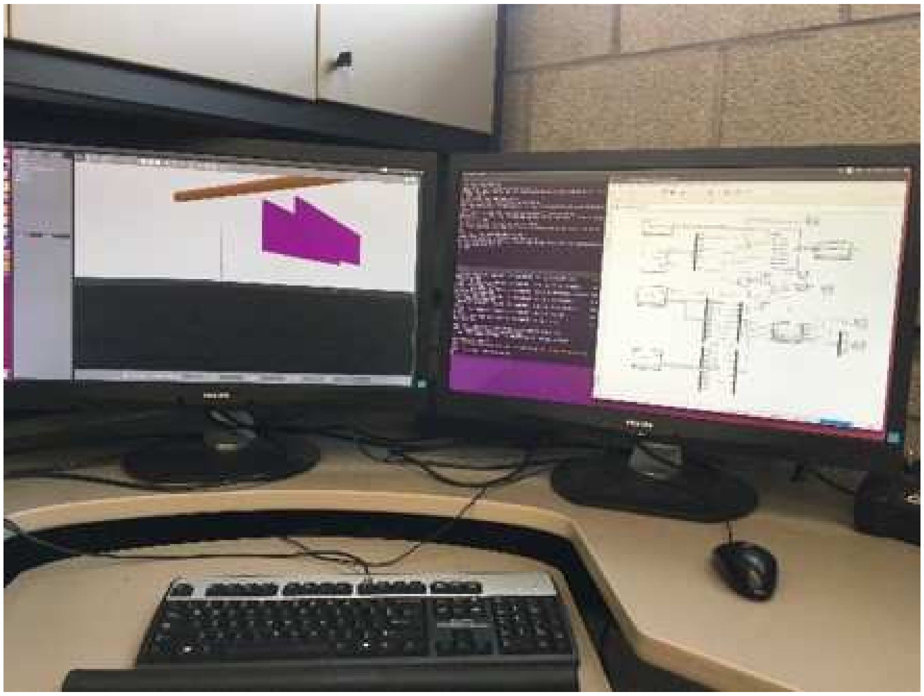


Figure 14: Remote station

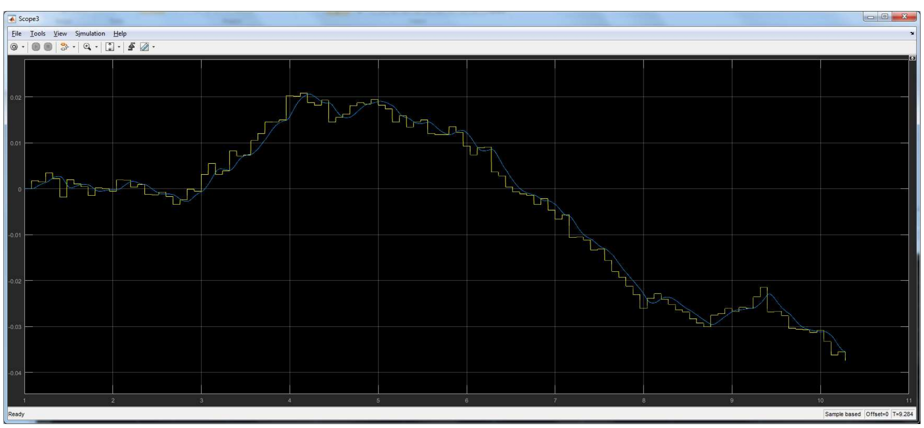


Figure 15: Actual velocity from the SITL controller, raw(yellow) and filtered(blue)

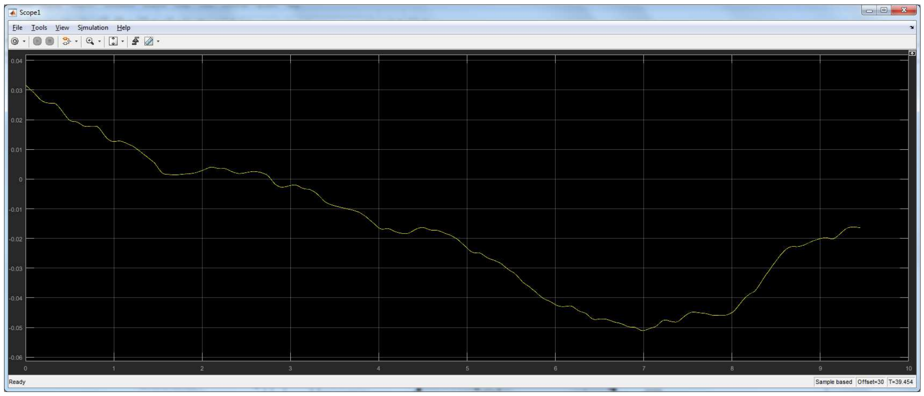


Figure 16: Error betweeen commanded and actual velocity

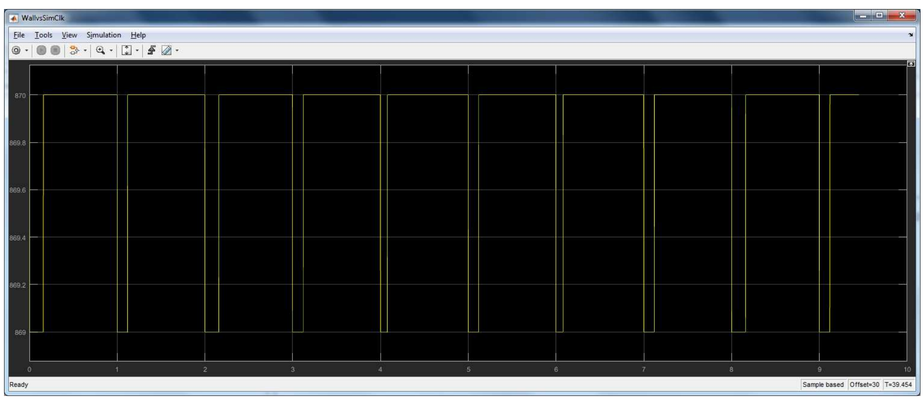


Figure 17: Comparison between wall clock and simulation clock

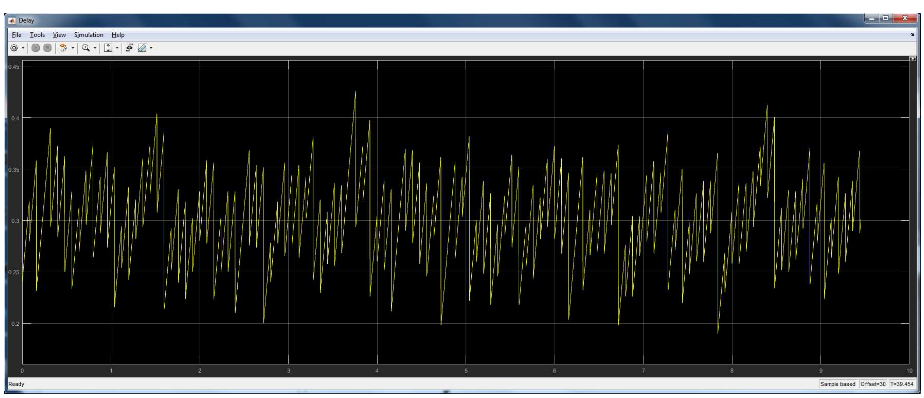


Figure 18: Communication delay and jitter with ROS node subscription

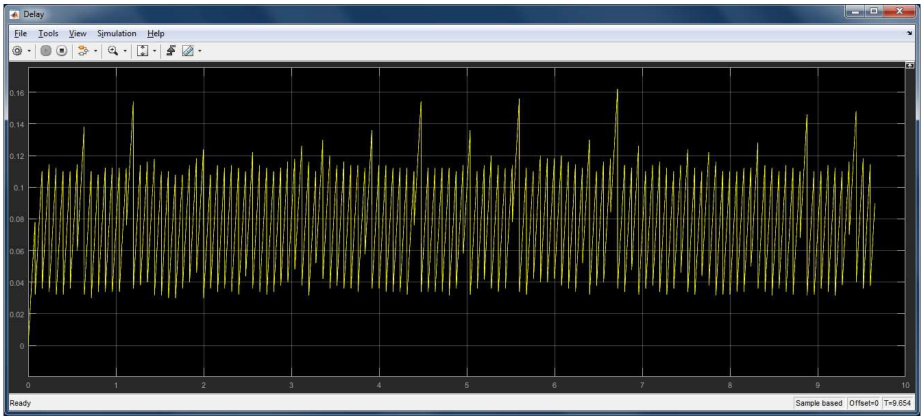


Figure 19: Communication delay and jitter, no ROS node subscription

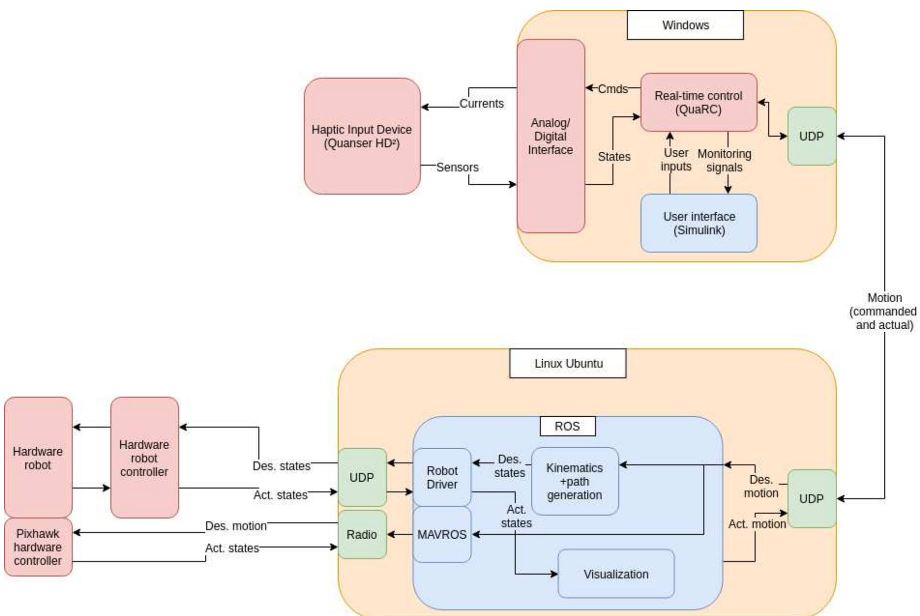


Figure 20: Simulation environment, Hardware-In-the-Loop configuration

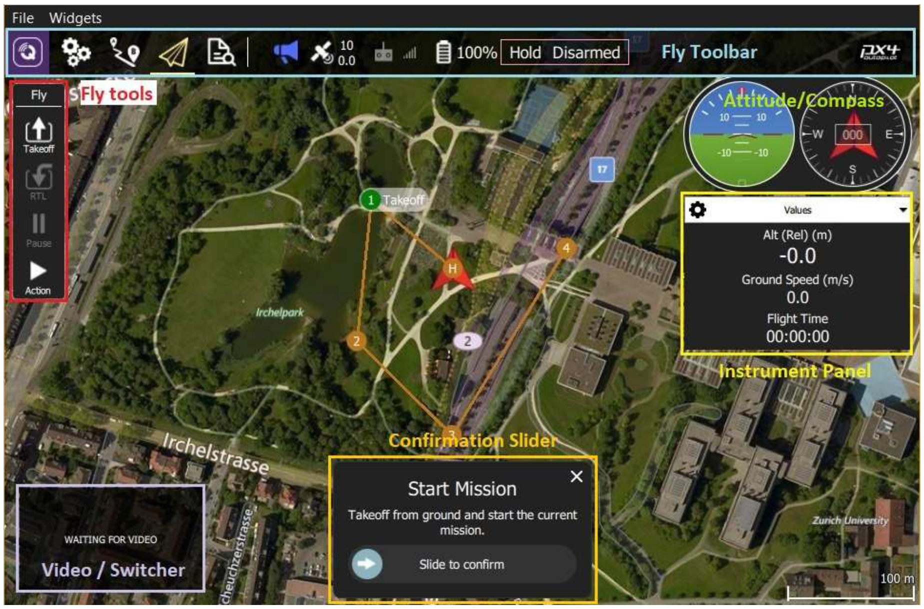


Figure 21: QGroundControl Interface, Fly Mode

modules and communication driver, all of them using the same robot application development framework as in the software case, so the HIL configuration does not introduce major architecture changes.

9 Visual Interfaces for BVLOS Operations

The majority of commercially available visual interfaces were developed for line-of-sight operations, where visual contact is essential. Consequently, these interfaces were not focused on maximizing the efficiency of information display. Most visual interfaces are part of so-called Ground Control Stations(GCS) and focus on the planning aspects of a mission, while providing online updates and an on-board camera video feed, whenever available. Figure 21 displays one such interface set in "fly mode" as opposed to "planning mode". The interface shows the updated planned path along with some basic localization properties (heading and altitude) as well as a small video feed.

Below is a table of commercially available GCS software. They are usually developed by and associated with autopilot manufacturers, except for open source software, which tends to be more general.

The great majority of GCS hardware configurations are designed to be portable, since proximity of the stations to the platform is less demanding in terms of wireless communication for control and telemetry. Remote stations are more commonly found in high-end applications such as the ones used for military operations. While many of stations are dedicated to a specific plat-

Name	Manufacturer	Associated Autopilots
QGroundControl	Open Source	Pixhawk PX4, Ardupilot, Autoquad
Piccolo Command Center	Cloud Cap Technology	Piccolo
Horizon	MicroPilot	MicroPilot
DJI GS Pro	DJI	A2, Ace, Wookong, Naza
IRIS UAS	Kongsberg Geospatial	N.A.
UAV CRAFT	Presagis	N.A.

Table 2: Commercial GCS software

form/autopilot, some GCSs are designed to be configurable and adaptable to various systems. Recent trends in military-compliant UASs increase modularity and interoperability between system components. This tendency is bound to migrate to civil applications as well. Below are some examples of Ground Control Stations with various levels of portability:

With regards to BVLOS operation, none of the above-reported software is adressing this particular configuration except the IRIS UAS, which explicitely lists a BVLOS setup for their GCS, shown in figure 25

While these interfaces could be suitable for unconstrained flights in a BVLOS context, additional considerations come into play when looking at applications that require a stronger sense of telepresence, including haptic feedback. In a VLOS context, these applications do not require prior path planning and piloting is mostly manual, such as structures inspection and maintenance or navigation in an unknown environment. When treated in a BVLOS context, these applications require a dedicated interface that is not yet available commercially. Below are some conceptual elements that would be required in situation where haptic feedback is also used to increase telepresence.

- **POV display:** Given that the main source of visual information about the remote scene in BVLOS operations would be the platform-mounted camera, the visual interface should be designed based on a Point-Of-View scene representation as a starting point. This can then be modified if extensive mapping is performed.

- **Mapping capabilities and mixed-reality representation:** BVLOS operation will most likely involve partially known or unknown remote environments. In order to provide a meaningful visual interface, it is likely that the system will require mapping and localization capabilities, along with an interface capable of displaying mixed-sensor and mixed-reality information, known elements co-existing with a locally mapped scene.

Figure 22: UAS Europe Portable Ground Control Station

Figure 23: UAV Factory Portable ground Control System

Figure 24: AAI Textron Configurable Universal Ground Control Station

Figure 25: IRIS UAS BVLOS configuration

- **Redundancy w.r.t. haptic feedback:** It is highly unlikely that quantities rendered by haptic feedback will allow the simplification of a visual interface by removing elements. Consider the case where haptic feedback is used to render platform oscillations. A visual interface that does not display these oscillations will not improve Situation Awareness and will, in fact, probably do the opposite. Rather, overall SA might be improved if haptic feedback is also visually rendered without overloading the interface, in a way that complements haptic feedback

- **AR for Telepresence Augmentation:** A natural consequence of the above-mentioned elements, the augmentation of a camera-rendered scene with synthetic overlays greatly contributes to maximizing information content in an interface, while minimizing distractions that would arise from consulting multiple displays. This concept is already applied to the commercial visual interfaces described earlier. Extending it to BVLOS operations where haptic feedback is present, augmented reality provides means of enhancing remote scene representation with additional geo-referenced information e.g. about collision awareness, motion constraints, platform stability, application-dependent goals and instructions and such.

10 Conclusions and recommendations

A Remotely Piloted Aircraft System is a multidisciplinary collection of many elements interacting together in order for an operator to be able to make appropriate decisions and actions occurring at a remote site, a complex task in itself. Not being physically located on site contributes to reducing situation awareness and man-machine interface design must take this factor into consideration, as the user relies entirely on information generated and provided by the system. Beyond Visual Line of Sight Regulations must be such that remote operation does not introduce elements that may compromise public safety. This report surveyed the work done in remote piloting of drones from the point of view of situation awareness enhancement through haptic feedback. Focus on illustrating the effects of latency on RPAS along with strategies to mitigate these effects was then put, with support from simulation-based results. Finally, a prototype software/hardware simulation environment fro evaluating human and system performance for BVLOS operations was developed. It has been found that many aspects underlying a system with haptic feedback must be considered in order not only to increase situation awareness, but to make such a system safe and properly regulated. The following are a summary of key findings and recommendations made throughout this report.

1. **Haptic feedback contributes to increasing situation awareness if properly selected and well-tuned:** Several papers reviewed have reported that the use of haptic feedback in teleoperating UAS can increase situation awareness. This has been observed in situations where haptic feedback is used to render wind gust effects[25], collision awareness[26],

[12], and velocity rendering[5]. However, the manner in which rendering is implemented, along with the selection of parameters must be carefully selected as a function of the task at hand, environment and operator considerations, as haptic feedback in itself cannot guarantee improved situation awareness. Whether it will useful or detrimental depends on many factors.

2. **Haptic feedback can significantly improve safety in the presence of latency:** The use of feedback in order to close the loop of a remotely operated system is essential in providing robustness against disturbances and forms the basis of the design of teleoperated robotic systems, of which RPAS are one example. Latency is one such disturbance which can potentially compromise system safety. While visual feedback can accomodate for these disturbances it cannot be relied on for safety due to the its dependency on operator proficiency, thus making it non-deterministic. A feedback loop on an actuated, backdrivable input device can both guarantee stability under bounded environmental(operator and remote site)conditions and provide haptic feedback to the user.

3. **Haptic feedback could reduce dependency on visual feedback:** Simulations realized using basic, simple operator and platform representations have shown that visual feedback can account for increased tracking performance, but is sensitive to latency. The addition of feedback at the input device stage not only improves robustness against latency but also lowers dependency on visual feedback, shown by reduced feedback gains when haptic feedback is present. While this may appear obvious, the two feedback loops were acting on different signals of the system (trajectory and command to the input device) and they could have coexisited without requiring adaptation of one to the presence of the other. Of course, the fact that this was demonstrated using a very simple model calls for more thorough investigations, but it also outlines the importance of this aspect.

4. **Human dynamics should be taken into consideration when tuning haptic feedback, both offline and online:** The dynamics of the human neuromuscular system is complex, has great variability(although bounded) and adapts with time. While inherently stable, its coupling to external dynamical systems can lead to instability due to improper training, delayed decisions and reduced situation awareness in general. While passive, time-invariant representation of humans have been used extensively in teleoperator system design, it is very conservative and addresses reflex-type behavior. Tuning of haptic feedback calls for a more elaborate human dynamics model that incorporates non-passive elements associated with signal transmission delays in the body, noise and cognitive effects. Furthermore, human dynamic characteristics adapt over time so some form of adaptation or robustness to evolving parameters should also be part of feedback design and tuning.

5. **BVLOS RPAS will most likely require localization and mapping capabilities for safe interaction with the environment:** Only in cases where the environment is well-known can one rely on a 3-D virtual model of the remote scene to implement safe RPAS-environment interaction e.g. for inspection, tasks involving manipulation, and such. This particularily the case as applications involving interaction with the environment typically involves complex motions that may require manual control. As many BVLOS applications take place in partially known, unstructured environments, means of mapping this environment and localizing the platform in it will be required. Again, this is very application dependent, and should not be seen as a blanket requirement.

6. **System stability and performance will require human in the loop characterization:** The two main tools to evaluate Situation Awareness, SAGAT and the NASA TLX, are qualitative in nature and are used to provide a global holistic interpretation of SA. While Linear-Time-Invariant, passive operator models are used in the design of remotely operated dynamic systems for convenience, their applicability is quite limited, as a human operator can potentially destabilize the system by introducing active elements. In cases where safety and/or performance is critical, characterization of human dynamics may be required for stability assessment. This is reinforced by the fact that these dynamics are changing online as operation takes place, making stability and performance time-dependent, requiring robustness and/or adaptability as a function of human dynamics.

7. **A software/hardware simulation environment can help in establishing regulations for operations in known and partially known/unknown environments:** BVLOS operations will enable new applications as well as new ways to utilize RPAS. The diversity of these applications make the establishment of regulations a real puzzle, especially when the platform has to interact with its environment which can be partially-known or completely unkown. Interaction does not need to be physical; a platform doing close inspection of e.g. communication towers will require safety regulations that a free-flight applications e.g. mapping do not need to be concerned with at the same degree. Consequently, it may envisionable that application-specific regulations will need to be enabled along with a core set of regulations that apply to any situation, given a specific case. As applications cannot be studied on a case by case basis due to the amount of work involved, a set of basis, elementary tasks would require to be studied and regulations deduced from them. These strawman tasks would be best studied in a laboratory environment using a combination of simulation and hardware components. For example, a laboratory setup may include a physical mock-up of a remote site, with elements pertaining to the actual real site (latency, environmental perturbations, visual degradation...) simulated to increase the level of realism. As a general rule, the more elements that can be captured using hardware elements, the more realistic results will be, while simulation can be used to study

particular aspects related to the application that are tedious to implement in a laboratory environment using hardware components.

8. **Consider a baseline, Turing-test-like approach in evaluating autonomous systems:** As an extension to the previous observation, the remote aspects of BVLOS operation will most likely foster the development of RPAS with varying degrees of autonomy, making human action more or less critical to the task. This will add an additional layer of complexity to the establishment of regulations due to the great variety in the way autonomy is present in a system. The lack of preestablished metrics and non-determinism aspects of autonomy make it difficult to characterize such systems. Rather than addressing system characteristics in defining regulations, it may be worth considering comparing system performance to a baseline set of performances for specific activities performed by a well-trained pilot. One can envision a laboratory setup similar to an obstacle course with specially designed "stations" where the system needs to do a specific task or manoeuvre. Much like the Turing test for Artificial Intelligence, if performance of an autonomous system is indisinguishable from the performance of a system under manual operation, that autonomous system would then qualify for operation. This represents a shift from regulating "what the system is" to "what the system does", which could be a more feasible way of qualifying (and quantifying) autonomous systems.

Better human-machine interface is an essential step towards system autonomy, as human intervention should be progressively be removed from the critical aspects of tasks to be performed, all the while making sure situation awareness does not degrade beyond the point of not being able to intervene in emergency situations, a challenging task in itself. Haptic feedback contributes to creating a more efficient link between human and machine. By doing so, it increases both operator and platform's capabilities, where the remote platform can be seen as an extension of the operator. It is by working on optimizing this link that we can safely engage on the path to automation and make technologically, socially and ethically meaningful steps along this path.

References

[1] G. Baxter, J. Rooksby, Y. Wang, and A. Khajeh-Hosseini, "The ironies of automation: still going strong at 30?," in *Proceedings of the 30th European Conference on Cognitive Ergonomics*, pp. 65–71, ACM, 2012.

[2] J. Smisek, E. Sunil, M. M. van Paassen, D. A. Abbink, and M. Mulder, "Neuromuscular-system-based tuning of a haptic shared control interface for UAV teleoperation," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 449–461, 2017.

[3] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?," *Cognition, Technology & Work*, vol. 14, no. 1, pp. 19–28, 2012.

[4] Wikipedia contributors, "Situation awareness — Wikipedia, the free encyclopedia," 2019. [Online; accessed 13-March-2019].

[5] M. R. Endsley, "Direct measurement of situation awareness: Validity and use of SAGAT," in *Situational Awareness*, pp. 129–156, Routledge, 2017.

[6] A. Scharine, T. Letowski, and J. B. Sampson, "Auditory situation awareness in urban operations," *Journal of Military and Strategic Studies*, vol. 11, no. 4, 2009.

[7] M. R. Endsley, *Designing for situation awareness: An approach to user-centered design*. CRC press, 2016.

[8] J. Smisek, M. M. Van Paassen, M. Mulder, and D. A. Abbink, "Neuromuscular analysis based tuning of haptic shared control assistance for UAV collision avoidance," in *World Haptics Conference (WHC), 2013*, pp. 389–394, IEEE, 2013.

[9] T. Tsuji, P. G. Morasso, K. Goto, and K. Ito, "Human hand impedance characteristics during maintained posture," *Biological cybernetics*, vol. 72, no. 6, pp. 475–485, 1995.

[10] W. Mugge, D. A. Abbink, and F. C. van der Helm, "Reduced power method: how to evoke low-bandwidth behaviour while estimating full-bandwidth dynamics," in *Rehabilitation Robotics, 2007. ICORR 2007. IEEE 10th International Conference on*, pp. 575–581, IEEE, 2007.

[11] 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.

[12] X. Hou and R. Mahony, "Dynamic kinesthetic boundary for haptic teleoperation of VTOL aerial robots in complex environments," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 5, pp. 694–705, 2016.

[13] L. Allender, "A cognitive systems engineering approach for human-robot interaction: Lessons from an examination of temporal latency," *Human-Robot Interactions in Future Military Operations*, pp. 315–316, 2010.

[14] D. A. Lawrence, "Stability and Transparency in Bilateral Teleoperation," *IEEE Transactions on Robotics and Automation*, vol. 9, pp. 624–637, Oct. 1993.

[15] S. Alaimo, L. Pollini, and H. Bülthoff, "Admittance-based bilateral teleoperation with time delay for an unmanned aerial vehicle involved in an obstacle avoidance task," in *AIAA Modeling and Simulation Technologies Conference*, p. 6243, 2011.

[16] J. M. Lambert, "Bio-mimetic and human-based performance approach to the design and control of teleoperation systems - a literature review," Technical Report LTR-SMM-2017-0168, 2017.

[17] L. R. Salinas, E. Slawiñski, and V. A. Mut, "Complete bilateral teleoperation system for a rotorcraft UAV with time-varying delay," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[18] E. Slawiñski, D. Santiago, and V. Mut, "Control for delayed bilateral teleoperation of a quadrotor," *ISA transactions*, vol. 71, pp. 415–425, 2017.

[19] D. Ruivo, *Formation of unmanned vehicles with collision avoidance capabilities.* PhD thesis, Instituto Suprior Técnico, Universidade de Lisboa, 2015.

[20] T. Luukkonen, "Modelling and control of quadcopter," *Independent research project in applied mathematics, Espoo*, vol. 22, 2011.

[21] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 3277–3282, IEEE, 2009.

[22] G. Hoffmann, H. Huang, S. Waslander, and C. Tomlin, "Quadrotor helicopter flight dynamics and control: Theory and experiment," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, p. 6461, 2007.

[23] X. Hou, R. Mahony, and F. Schill, "Representation of vehicle dynamics in haptic teleoperation of aerial robots," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1485–1491, IEEE, 2013.

[24] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research," in *Advances in psychology*, vol. 52, pp. 139–183, Elsevier, 1988.

[25] S. M. Alaimo, L. Pollini, J. P. Bresciani, and H. H. Bülthoff, "A comparison of direct and indirect haptic aiding for remotely piloted vehicles," in *RO-MAN, 2010 IEEE*, pp. 506–512, IEEE, 2010.

[26] S. M. Alaimo, L. Pollini, J. P. Bresciani, and H. H. Bülthoff, "Evaluation of direct and indirect haptic aiding in an obstacle avoidance task for teleoperated systems," in *Proceedings of the IFAC World Congress, Milano, Italy*, vol. 28, Citeseer, 2011.

[27] J. J. Abbott and A. M. Okamura, "Pseudo-admittance bilateral telemanipulation with guidance virtual fixtures," *The International Journal of Robotics Research*, vol. 26, no. 8, pp. 865–884, 2007.

[28] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 500–505, IEEE, 1985.

[29] J. Minguez, L. Montano, and O. Khatib, "Reactive collision avoidance for navigation with dynamic constraints," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 1, pp. 588–594, IEEE, 2002.

[30] H. W. Boschloo, T. M. Lam, M. Mulder, and M. M. Van Paassen, "Collision avoidance for a remotely-operated helicopter using haptic feedback," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 1, pp. 229–235, IEEE, 2004.

[31] A. M. Brandt and M. B. Colton, "Haptic collision avoidance for a remotely operated quadrotor uav in indoor environments," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, pp. 2724–2731, IEEE, 2010.

[32] G. Niemeyer and J.-J. E. Slotine, "Stable Adaptive Teleoperation," *IEEE Journal of Oceanic Engineering*, vol. 16, pp. 152–162, Jan. 1991.

[33] D. Lee and M. Spong, "Passive bilateral teleoperation with constant time delay," *Robotics, IEEE Transactions on*, vol. 22, no. 2, pp. 269–281, 2006.

[34] W. H. Zhu and S. Salcudean, "Stability Guaranteed Teleoperation: An Adaptive Motion/Force Control Approach," *IEEE Trans. on Automatic Control*, vol. 45, pp. 1951–1969, Nov. 2000.