# Mimicking Real Forces on a Drone Through a Haptic Suit to Enable Cost-Effective Validation

Carl Hildebrandt[1], Wen Ying[1], Seongkook Heo[1], and Sebastian Elbaum[1]

***Abstract*— Robots operate under certain forces that affect their behavior. Consider, a drone meant to deliver packages must hold its pose as long as it operates under its weight and wind limits. Validating that such a drone handles external forces correctly is key to ensuring its safety. Nevertheless, validating the system's behavior under the effect of such forces can be difficult and costly. For example, checking the effects of different wind magnitudes may require waiting for the matching outdoor conditions or requiring wind tunnels. Checking the effects of different package sizes and shapes may require many slow and laborious iterations, and validating the combinations of wind gusts and package configurations is often hard to replicate. This work introduces a framework to overcome such challenges by mimicking external forces exercised on a drone with limited cost, setup, and space. The framework consists of a haptic suit device with directional propellers that can be mounted onto a drone, a synthesizer to transform intended forces into setpoints for the suit's directional propellers, and a controller for the suit to meet those setpoints. We conduct a study to assess the framework's capabilities under multiple scenarios involving various forces. Our findings show that the haptic suit framework can recreate real-world forces on the drone with acceptable precision.**

## I. Introduction

Drones are incredibly versatile, allowing them to complete a multitude of missions in a variety of environments. They can carry sensors and objects [1], [2], balance inverted pendulums [3], [4], juggle balls [5], [6], grasp or push objects [7], [8], or take off from underwater [9]. Throughout all these missions, external forces are exerted on the drone. Carrying weights adds a downwards force, hanging pendulums introduce a swinging force, juggling balls create an intermittent impact force, grasping objects adds a downwards and rotational force, and taking off from underwater adds a resistive force that is removed once in the air.

*Validating that drones can adequately behave under those external forces is critical to ensure their correct and safe operation.* Such validation takes multiple forms, from model simulation, which offers the opportunity to quickly test many variants of the system and approximations of the forces in the environment, to field tests that exercise the entire system in a limited number of real environments with real physical forces. In this work, we focus on the challenge of conducting full system tests carried out in both a lab and in the field.

Validating the effect of external forces on drones brings significant challenges. Consider a drone that must hold its pose while delivering cargo. First, precisely validating the effects of wind would require specialized mechanisms like wind tunnels. Second, validating the space of behaviors would require validating various cargo configurations (e.g., weight, shape). This can be laborious as testers must often set each scenario manually. Third, some forces are extremely difficult to replicate, such as those caused by the subtle combination of gusts of wind and cargo configuration. These challenges make drone behavior validation under external forces costly, difficult, and often infeasible.

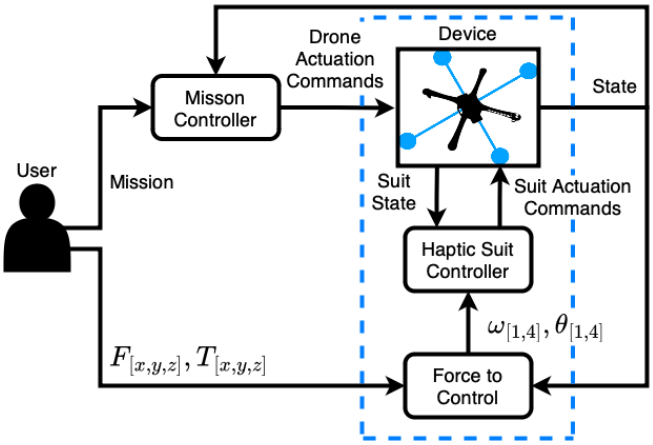



Fig. 1: Overview of the Haptic Suit Framework

In this work, we introduce a framework to address these challenges. Inspired by haptic feedback in interactive systems that simulate forces to provide immersive experiences to users [10], [11], this framework recreates forces on the drone to replicate real-world scenarios. Figure 1 provides an overview of the framework and its three key components: 1) an electro-mechanical device that attaches to the drone under test to provide directional propellers that can generate a rich space of forces and torques; 2) a synthesizer that transforms user-specified force $F_{[x,y,z]}$ and torque $T_{[x,y,z]}$ functions into a set of target motor speeds $\omega_i$ and angles $\theta_i$ for each motor $i$, using the inverse kinematic equations of the suit; and 3) controllers that read the state of the suit and the drone and use the $\omega_i$ and $\theta_i$ as setpoints to actuate the suit propellers to match those targets.

The contributions of this work are:

- An innovative concept connecting drone validation and haptic feedback to induce external controlled forces.
- A framework that enables the exertion of a rich set of programmable forces for the validation of drones.
- A study showing that the haptic suit framework can successfully replicate forces from 5 distinct real scenarios.

[1]University of Virginia, USA {*hildebrandt.carl, wy7yv, seongkook, selbaum*}*@virginia.edu*

This work was funded in part through NSF grants #1924777 and AFOSR grant #FA9550-21-1-0164.

## II. Related Work

Several approaches exist to exercise and manipulate specific external forces on drones. For example, to mimic wind forces, various forms of fans [12], and wind tunnels [13], [14], [15] have been proposed, offering trade-offs between cost, flexibility, and accuracy in the forces being exercised. Similarly, to mimic payloads, it is relatively common to build devices that enable the attachment of different masses with varying volumes [16], [17]. In this case, the level of control can be accurate but often requires a manual setup [18]. Beyond simple forces, we find that general mechanisms for applying forces to a drone are lacking.

In this paper, we aim to accurately mimic more complex external forces without needing expensive setups and laborious procedures. Although not pursuing the same goals, the closest line of work comes from the variety of testbeds meant to support drone development [19], [20], [21], [22], [23]. This work aims to secure the drone such that it can not crash while allowing for some form of movement. For example, Ding et al. proposed using a humanoid arm to hold a drone while it is flying. The goal was to secure the drone while minimizing the effect of this connection [23]. While the purpose of this work was not to exert forces on the drone, one could imagine that with adjustments to the controller function, as well as the development of a new set of kinematics, one could recreate larger forces with precision. Still, these devices are complex and expensive, and for most of the forces we intend to mimic, it would be bounded by the arm's reach, degrees of freedom, and velocities, while our haptic suit does not have those limitations.

Another related line of work that in part inspired this work is in the area of haptics. Researchers have shown that propeller-based propulsion can be used to produce high-fidelity directional force feedback and provide realistic force feedback to its users [10], [11]. However, these methods are designed for handheld use, and their use for robot testing has not been investigated. The closest line of work mimics real sensor values on drones to provide a mix of real world and simulated sensor data [24], [25], [26]. However, this work lacks the ability to exert actual forces onto the drone.

## III. Framework

The framework aims to mimic the forces a drone may experience during flight, enabling engineers to validate their drones under various scenarios more efficiently, quickly, and easily. We now describe its key components in more detail.

### A. Haptic Suit Device

The following requirements guided the haptic suit design. First, the device needed to generate forces with a wide range of magnitudes and directions to support many scenarios (including the ones we later studied). Second, the device needed to mount on the host drone without any point of contact with other external entities in order to reduce flying constraints or interference. Third, the device needed to minimize the disturbance to the drone's normal behavior to avoid failures that the introduction of the suit may cause.

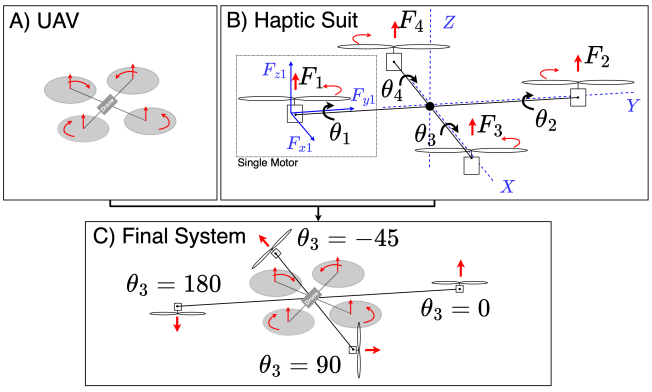



Fig. 2: Quadrotor, Haptic-Suit, and Haptic-Suit integrated with Quadrotor (grey circles represent quadrotor propellers).

The suit design follows from those requirements. Conceptually, the suit is elegant in its simplicity in that it adopts the host-drone structural design. Without loss of generality and to facilitate the explanation, we exemplify the integration of the haptic suit to a quadrotor as shown in Figure 2. For a quadrotor, shown in Figure 2-A, the suit device consists of four arms, each one with a rotor at its end, as shown in Figure 2-B, reflecting the design of the drone. In the suit, each of the $i$ ($i = 4$ for quadrotor) rotor's thrust $F_i$ and torque values $T_i$ can be independently controlled by varying the motor speed $\omega_i$. However, unlike a typical drone where $F_i$ is always perpendicular to the drone, the suit can independently rotate each arm by $\theta_i$ through additional motors at the base of each arm. To allow for precise control, each arm is fitted with an IMU reporting its attitude. This allows us to vary the magnitude of $F_i$ and $T_i$ by varying $\omega_i$, as well as vary the direction by varying $\theta_i$, to create a wide range of forces on the drone regardless of the drone's current pose.

The integration of the haptic suit and drone is shown in Figure 2-C. The haptic suit is mounted onto the drone with a shift of 45 degrees. This integration meets our requirements in the following ways. First, as each arm has a motor that can independently rotate, it can generate a wide range of forces, allowing for the simulation of a wide range of scenarios. Second, the suit is not attached to anything but the drone, thus not limiting the drone to any specific environment. Third, the suit is symmetrical, lightweight, and has arms that protrude out further than the drone's arms. This minimizes the changes in both the total and center of mass of the drone while also minimizing interference among airflow through propellers. These design choices reduce the chance that the suit will affect the normal behavior of the drone.

The conceptual design simplicity offers a rich space of trade-offs that will determine the magnitude of the forces that the suit can generate. For example, more powerful motors can generate greater forces. However, their weight, the weight of the arms required to support them, and the battery requirements may undermine some of the forces they can generate and the preciseness of the overall force manipulation. The arms' length and rotational speed can also affect the type and

magnitude of forces that can be applied. This also indirectly affects the airflow of the other drone's rotors. The profile of all these elements can affect the dynamics of the drone, from adding weight to changing the drag coefficients. We will discuss the particular choices we made for one instantiation of the framework under Section III-F.

### B. Force to Control Set Points

Given a haptic suit device, the framework must convert a set of user-defined forces $F_{[x,y,z]}$ and torques $T_{[x,y,z]}$ into a set of motor speeds $\omega_{[1,4]}$ and rotations $\theta_{[1,4]}$. For example, if a user wants to simulate carrying a package, which is equivalent to applying negative force in the $Z$ direction $(-F_z)$, we would need all motors to spin at equal speeds $(\omega_1 = \omega_2 = ...)$ such that they generate a force of the correct magnitude $(F_z)$. Furthermore, each of the motors would also need to be pointing directly downwards $(\theta_{[1,4]} = 180)$, so that the direction of the force is in the same direction as gravity $(-F_z)$ to exert the same force as a package. Similarly, if a user wants to mimic a horizontal wind with a force $F_y$, whose magnitude is the same as that of the first example, then motors 2 and 3 would need to create forces in the $F_y$ plane by rotating $(\theta_{[3,4]} = 90)$, and each would need to operate at double the speed of the first example to compensate for the lack of motors 1 and 2.

To perform such conversion computations, we define a set of kinematic equations for the haptic suit. Consider $\text{motor}_1$ in Figure 2-B, with target speed $\omega_1$ and rotation $\theta_1$. The force $F_1$ and torque $T_1$ generated by that motor are $F_1 = k_f \times \omega_1$ and $T_1 = k_m \times \omega_1$ where $k_f$ and $k_m$ are the proportionality constants for thrust and moments respectively [27]. Since the capability to rotate the motor affects the forces and torques direction, we must decompose $F_1$ and $T_1$ into $(F_{x1}, F_{y1}, F_{z1})$ and $(T_{x1}, T_{y1}, T_{z1})$. Equations 1 showcase the decomposition of $F$, with similar equations holding for $T$.

$$F_{x1} = -F_1 \times sin(\theta_1);\ F_{y1} = 0;\ F_{z1} = F_1 \times cos(\theta_1) \quad (1)$$

We can now generalize these equations to a full suit as per Equations 2.

$$\begin{aligned} F_{xi} &= -F_i \times sin(\theta_i); F_{yi} = 0; F_{zi} = F_i \times cos(\theta_i) : i \in [1,2] \\ F_{xi} &= 0; F_{yi} = -F_i \times sin(\theta_i); F_{zi} = F_i \times cos(\theta_i) : i \in [3,4] \end{aligned} \quad (2)$$

These equations are derived from Equation 1 and thus $F_{yi} = 0$ for motors 1 and 2, while $F_{xi} = 0$ for motors 3 and 4, as these motors are incapable of rotating in the $Y$ and $X$ planes respectively. To compute torque we use a similar derivation except that since motors on opposite sides of an arm rotate opposite each other, their torque cancels out, so the terms for $T_2$ and $T_4$ are negated.

The final step is to compute the total force $F_x, F_y, F_z$ as per Equation 3. Computing the torques $T_x, T_y, T_z$ must take into consideration the additional torque placed on the system by any unbalanced forces in any given directions. For example, a torque around the $X$ axis can be created by an imbalance of the forces in the $Z$ direction between motors 1 and 2.

$$F_{[x,y,z]} = \sum_{i=1}^{4} F_{[x,y,z]i} \quad (3)$$

To obtain the speed and rotations needed to generate a given force and torque, we compute the inverse kinematics. Given the complexity of the equations and the fact that the number of degrees of freedom is different from the number of variables, we approximate the inverse using numerical methods described in more detail in the implementation.

$$\begin{aligned} T_x &= \sum_{i=1}^{4} T_{xi} + (F_{z1} - F_{z2}) \\ T_y &= \sum_{i=1}^{4} T_{yi} + (F_{z3} - F_{z4}) \\ T_z &= \sum_{i=1}^{4} T_{zi} + ((F_{x1} - F_{x2}) + (F_{y3} - F_{y4})) \end{aligned} \quad (4)$$

### C. Controller

The final step is to actuate each of the suit's motors and arms into the correct configuration. The framework uses two traditional PID closed-loop controllers per arm. The first controls the motor speed, using the current motor speed as feedback, while the second uses the IMU attitude information as feedback. Note that having an IMU on each arm allows the framework to control the haptic suit independently of the drone's pose and behavior, which allows users to define forces in the world frame while ignoring the drone's pose. The set points are given by $\omega_{[1,4]}$ and $\theta_{[1,4]}$ produced by the previous framework component.

### D. Generality

As defined, the framework can be directly instantiated to support multiple drone configurations under two conditions: 1) the device placement can coincide with the drone's center of mass, and 2) a symmetrical distribution of suit motors is a good fit for the drone structure and the intended forces and torques. As part of the framework presentation we introduced a 4-motor-suit configuration that fits, for example, common quadcopters and octocopters, in the following study we use a 2-motor-suit configuration that suffices to explore the target scenarios on a quadcopter, and extensions to more motors should be trivial to instantiate reusing a similar suit structure (with different motors and arms' length), body of kinematic equations, and controllers.

### E. Limitations and Trade-offs

The haptic framework's range of forces is constrained by two factors. The first results from the suit design, which mimics a typical drone. Drones control yaw by varying the speeds of motors that spin in opposite directions. A speed imbalance will produce a non-zero overall torque, resulting in a yaw. Three things occur when our suit rotates an arm, as seen in Equation 4. First, as the arm rotates, the force component in the $F_z$ direction will decrease, resulting in a roll or pitch. Second, the arms rotation will induce a $Fx$, or $F_y$ component, resulting in additional yaw. Third,

Fig. 3: Two-arm haptic suit prototype (marked using dashed lines) mounted on a DJI Flamewheel F450 drone [28].

the torque generated by the motor's spin will also become imbalanced, adding additional roll, pitch, or yaw depending on the rotation. Therefore while we can generate a force in any direction, some forces may result in rolls, pitches, or yaws that can be modeled and should be monitored.

The second results from mounting the suit onto the drone. The drone's design and capabilities (e.g., weight, footprint, lift, controller) and its tolerance for the unintended perturbations caused by the suit (e.g., interference with airflow, additional weight) directly affect the design of the suit (e.g., the motors it can carry, the propellers dimension, power supply, the arms' length) and thus the magnitude of the forces our haptic suit can produce.

### F. Implementation

Our implementation, which is publicly available[1], uses a DJI F450 [28] with a Pixhawk controller [29] governed by the Freyja mission controller [30]. The experiments only required operation in the $XY$-plane and thus only required a suit with 2 arms. Since the drone's arms are 25cm, to minimize airflow interaction, we used 5mm-diameter and 29cm length carbon fiber tubes. Each arm connects to a DC motor for rotation driven by a dual-motor driver (Adafruit TB6612). Each arm also has an IMU (SparkFun BNO080) to determine its current rotation angle. 3D-printed components were used to firmly attach the suit to the drone.

The Flamewheel can carry up to 1kg of payload with limited noticeable behavioral changes, which drove the selection of components. We used two tri-blade propellers (length: 5 inch, pitch: 4 inch) and two brushless motors (GARTT ML2204S 2300KV) that are driven by two electronic speed controllers (HGLRC 30A) to generate thrusts. The brushless motors are connected to two UXCELL GA12-N20 DC motors with reduction gears (6V, 70 RPM). Each motor was connected to the end of the carbon fiber tube. The DC and brushless motors were controlled using an Adafruit Feather M0, to which the user could send commands over WiFi. We equipped the suit with its own battery to avoid affecting the drones' behavior by sharing a power source. The suit, including battery, weighed under 800 grams to minimize the effects on the drone's behavior.

The remaining components are implemented on top of the Robotic Operating System (ROS) [31] as nodes developed in Python. The suit PID controllers are implemented on the Adafruit Feather M0. The numerical solver used to compute the inverse kinematics in Matlab [32] was the *vpasolver*.

[1] https://github.com/hildebrandt-carl/HapticSuit

TABLE I: The force and torques used by each scenario.

| Scenario | Force and Torque (Newtons) |
|---|---|
| Weight (Indoor) | $F_z = -3$ |
| Steady Wind (Indoor) | $F_y = 1$ |
| Gusting Wind (Indoor) | $F_y = f_1([0.5,1]), T_{[x,y,z]} = f_2([0.1,0.5])$ |
| Pendulum (Indoor) | $F_y = f_3(\rho), F_z = f_4(\rho)$ |
| Drop Weight (Outdoor) | $F_z = f(t) = \begin{cases} -5 & \text{if } 0 < t \leq 30 \\ 0 & \text{if } t > 30 \end{cases}$ |

## IV. Study

The study is meant to assess the extent to which the haptic suit framework could mimic real-world forces on a drone.

### A. Setup

*1) Suit:* To assess the haptic suit framework capabilities, we used the prototype described in Section III-F with the suit shown in Figure 3. Through a set of empirical calibrations, we set the $k_f$ and $k_m$ of the kinematic model and the controller coefficients introduced in Section III. The synthesizer that transforms forces and torches to motor velocities and rotation angles was executed offline, while the controller was executed online as the scenarios were running.

*2) Scenarios:* We flew the drone under a series of scenarios, listed in Table I, designed to expose different types of forces. The weight scenario produces a constant force that points in the $Z$ direction. The wind scenarios produce either a constant force in the $Y$ direction, or in the case of the gusting wind a force in the $Y$ direction and torques in the $X, Y$, and $Z$ direction sampled from a set interval. The pendulum scenario produces a force in the $Z$ and $Y$ direction that varies based on the position of the drone $\rho$. The scenarios were performed indoors using a motion capture system, Vicon [33], that is capable of measuring the vehicle's pose at 200Hz with a 5mm accuracy. The outdoor drop-weight experiment follows a piecewise function based on the time $t$, and measurements were taken based on the commands generated by Ardupilot [34].

For each scenario, we compare the drone's behavior under the real external forces versus the forces induced by the haptic suit framework. The scenarios with real forces used physical weights, wind generated by fans, and attached pendulums to create a baseline. For the weight scenario, we selected a 100g, 200g, and 300g cargo. For the wind experiments, we used an industrial fan capable of generating gusts of winds up to 4m/s. To create the pendulum scenario, we attached a hinge allowing a unidirectional motion to a carbon fiber rod that carried weights of 100g, 200g, and 300g. For the outdoor drop-weight scenarios, we used 500g cargo that was dropped after a set time interval using a rope connected to a pin which, when pulled, released the weight.

### B. Results for Weight Scenario

In this scenario, the drone was set to hover at 1m while carrying a set weight. Each of the experiments was run 5

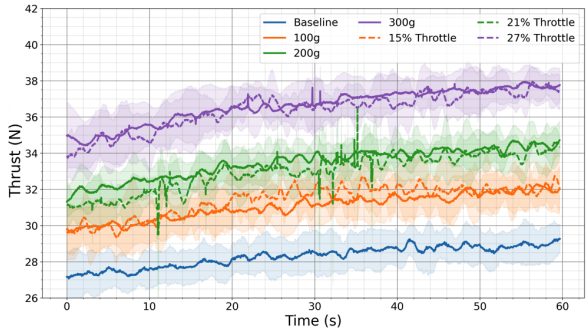



Fig. 4: Average thrust at different real weights and weights induced by the haptic suit.

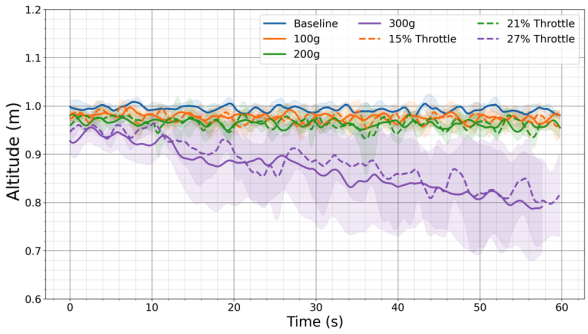



Fig. 5: Average altitude induced using different real weights and weights induced by the haptic suit.

times for 1 minute. We assess the drone's behavior in terms of the thrust commands sent by the controller Freyja [30] and the drone's altitude while carrying different weights.

Figure 4 shows the thrust commands sent to the drone by the controller in newtons. Solid lines represent thrust readings using real weights, while dashed lines represent thrust using the haptic suit. As expected, the more weight that is added, the higher the thrust command sent to the drone to maintain altitude. More interestingly, the thrust command's average and standard deviation are almost identical when using the haptic suit to replicate the real weights. This suggests that, from the drone's perspective, the suit is exercising almost the same external forces. Figure 5 shows the altitude of the drone as measured by Vicon. The average and standard deviations on altitude are almost identical when comparing the real and the replicated scenarios using the haptic suit. Even the slow decline in altitude at 300g observed in reality, likely caused by an erroneous drone configuration parameter, is replicated when using the haptic suit. Last, the validation with the haptic suit was significantly faster and less error-prone than the real payload, which required frequent stops to remove a weight and secure a new one.

### C. Result: Wind Scenarios

We first wanted to explore how the drone would behave under a constant wind force. Testing this often requires an elaborate environment to produce steady wind flows. It is not difficult to conjecture, however, how the drone should behave under a steady wind. We expect that the drone should lean against the wind to maintain its position. Thus, for this

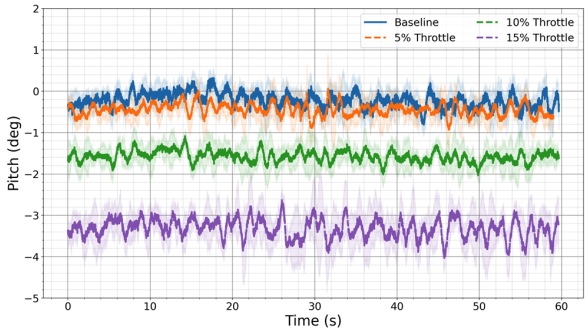



Fig. 6: Average pitch reported at different constant wind velocities induced by the haptic suit

first wind scenario, rather than testing our drone against a real constant wind, we validated our hypothesis using the haptic suit to replicate the drone operating under steady winds. The test involved increasing the magnitude of the force in the $F_y$ direction using 3 constant values. These correlated with suit throttle values ($\omega$) of 5%, 10%, and 15%. Each of these configurations was run 5 times, and the average and standard deviation of the drone's pitch is shown in Figure 6. The pitch adjustments of the drone match our conjecture. When the haptic suit's motors were set to 5%, the pitch was not significantly different from the baseline. However, as the horizontal force placed on the drone grew, the drone made more drastic pitch adjustments to counteract this force. At 15% throttle, which generates a wind equivalent of 1N of force, the drone must almost continuously perform corrections of up to five degrees.

For the varied wind scenario, we rely on industrial fans to mimic wind forces on drones. The airflow in the area where the drone is to hover was recorded between 2.9m/s and 4.1m/s, reflecting the noisy airflow produced. To recreate this scenario using the haptic suit, we randomly varied both $F_y$ and $T_{[x,y,z]}$ inside set intervals. Each random sample was selected from a normal distribution to mimic the real wind gusts. The pitch of the drone was then recorded over 5 runs when exposed to the fans and 5 runs when using the haptic suit forces. As shown in Figure 7, the recorded pitch values show a greater amplitude than with the steady wind, as the drone must continuously adjust to different forces to maintain its position. We find that the behaviors are similar when comparing the real wind gusts and the haptic suit-induced ones. However, the haptic suit, on average, produces slightly more pitch than the wind gusts. This is possibly due to slight variations in the drone's altitude, which would take it in and out of the airflow produced by the fans. This difference points to the limitations of employing such current devices compared with the haptic suit.

### D. Results: Oscillating Pendulum Scenario

For this scenario, the drone flies at an altitude of 1m along the x-axis starting at -1m and traversing to 1m before aggressively turning back until it gets to the start point, where it would wait 20 seconds before repeating the process.

To define the forces induced by the pendulum that would

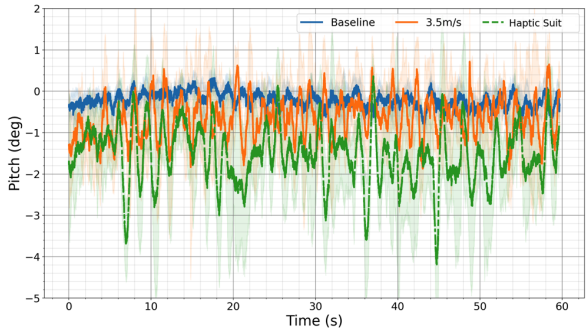



Fig. 7: Average pitch induced by real fans and haptic suit.

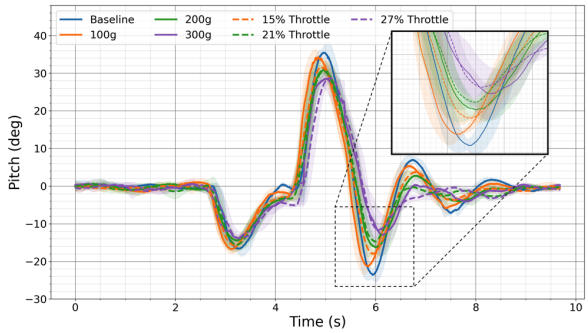


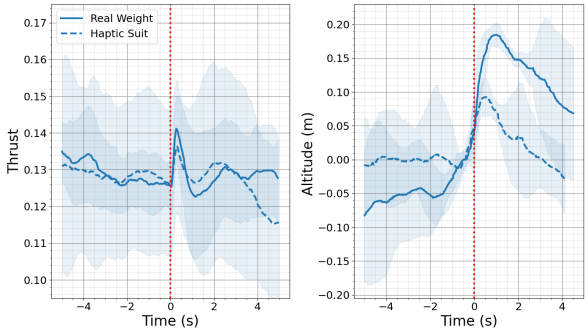

Fig. 8: Average pitch induced by a pendulum with different weights and haptic suit.

serve as inputs to the haptic suit, we modified OpenAI's cartpole simulation [35] to match our scenario (our pendulum is facing down, has a range from 0-180 degrees). We modified the simulation to include friction and a PID controller that moved the base of the pendulum to a given setpoint $\rho$ corresponding to the pose of the drone. Thus, the simulation pendulum angle would be updated as the drone moved. The magnitude and direction of the transformed force of the simulator were then sent to the force to control module, which forwarded the correct $\omega_{[1,2]}$ and $\theta_{[1,2]}$ to the haptic suit.

Figure 8 shows the drone's pitch. We were specifically interested in the 6-second mark when the drone tries to become leveled but is affected by the swinging pendulum. We notice how well the haptic suit mimics the real-world forces, with a maximum average difference of 3 degrees at 6s. We conjecture the slight differences observed can be attributed to the simplistic simulation model we used (oversimplified friction approximation) or communication delays (the suit was controlled from a wireless station). Overall, we find it promising that this rather complex scenario can be modeled so accurately using the haptic suit.

### *E. Results: Drop-Weight*

In this outdoor scenario we manually flew the drone to roughly 2m above the ground while using Ardupilot's altitude hold mode to maintain the drone's current altitude without any pilot feedback. For the scenario we took off with a payload of 500g attached. After roughly 30 seconds, the payload is released. We expect that as the weight is released there is a sudden change in altitude accompanied by a change in thrust to recover the altitude.




(a) Average thrust (b) Average altitude

Fig. 9: Thrust and altitude when dropping real weights and using the haptic suit. Time 0s corresponds to time of weight being dropped.

Figure 9a showcases the average thrust (a unitless value reported by Ardupilot) when using the real weight and the haptic suit approximation, with time 0 corresponding to the weight being dropped. In both cases, when the weight is dropped, there is a spike in thrust. Figure 9b showcases the average altitude changes from the original altitude at the time of dropping weight. As before, when the weight is dropped, there is a sudden increase in altitude when using both the real weight and haptic suit. We notice that when using the real weight, there is a larger increase in altitude likely caused by the friction in the weight release mechanism that was not accounted for in the force functions. This point again highlights one of the advantages of the haptic suit to overcome the limitations of existing practices that would have required a setup that includes a more sophisticated release mechanism to validate the drone's stability to changes in weights while flying.

## V. Conclusions

We have introduced a haptic suit framework able to exert various forces on a drone as part of its validation. We found that the drone's thrust, attitude, pose, and behavior under real and haptic suit induced forces appear to be very similar. This provides evidence that the haptic suit can *recreate real-world forces on the drone*. We also found that surprising behaviors identified under real forces, such as losing altitude in the heavy-weight scenario, are also identified by the haptic suit, providing support for the haptic suit as an effective tool in complementing real-world testing. Last, we found that specific real forces, as simple as a constant wind or as complex as a nonlinear function that changes with the drone state, can be extremely difficult and costly to create in the lab but may be approximated through the suit with limited effort. In future work, we will explore scenarios that require richer functions, such as capturing a moving object in flight, we will create additional suits to fit other drone configurations, and we will extend the application of the suit to validate behaviors involving drone translation so that experiments that currently require large outdoor spaces can be done in the confines of the lab.

## References


[1] J.-P. Ore, S. Elbaum, A. Burgin, and C. Detweiler, "Autonomous aerial water sampling," *Journal of Field Robotics*, vol. 32, no. 8, pp. 1095–1113, 2015.

[2] E. Beachly, C. Detweiler, S. Elbaum, B. Duncan, C. Hildebrandt, D. Twidwell, and C. Allen, "Fire-aware planning of aerial trajectories and ignitions," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 685–692, IEEE, 2018.

[3] D. Brescianini, M. Hehn, and R. D'Andrea, "Quadrocopter pole acrobatics," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3472–3479, IEEE, 2013.

[4] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in *2011 IEEE International Conference on Robotics and Automation*, pp. 763–770, IEEE, 2011.

[5] W. Dong, G.-Y. Gu, Y. Ding, X. Zhu, and H. Ding, "Ball juggling with an under-actuated flying robot," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 68–73, IEEE, 2015.

[6] M. Müller, S. Lupashin, and R. D'Andrea, "Quadrocopter ball juggling," in *2011 IEEE/RSJ international conference on Intelligent Robots and Systems*, pp. 5113–5120, IEEE, 2011.

[7] M. Srikanth, A. Soto, A. Annaswamy, E. Lavretsky, and J.-J. Slotine, "Controlled manipulation with multiple quadrotors," in *AIAA Guidance, Navigation, and Control Conference*, p. 6547, 2011.

[8] A. Shankar, S. Elbaum, and C. Detweiler, "In-air exchange of small payloads between multirotor aerial systems," in *International Symposium on Experimental Robotics*, pp. 511–523, Springer, 2018.

[9] H. Alzu'bi, I. Mansour, and O. Rawashdeh, "Loon copter: Implementation of a hybrid unmanned aquatic–aerial quadcopter with active buoyancy control," *Journal of field Robotics*, vol. 35, no. 5, pp. 764–778, 2018.

[10] S. Heo, C. Chung, G. Lee, and D. Wigdor, "Thor's hammer: An ungrounded force feedback device utilizing propeller-induced propulsive force," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2018.

[11] S. Je, M. J. Kim, W. Lee, B. Lee, X.-D. Yang, P. Lopes, and A. Bianchi, "Aero-plane: A handheld force-feedback device that renders weight motion illusion on a virtual 2d plane," in *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, pp. 763–775, 2019.

[12] J. Bannwarth, Z. Chen, K. Stol, and B. MacDonald, "Disturbance accomodation control for wind rejection of a quadcopter," in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 695–701, IEEE, 2016.

[13] H. Jeon, J. Song, H. Lee, and Y. Eun, "Modeling quadrotor dynamics in a wind field," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 3, pp. 1401–1411, 2020.

[14] J. González-Rocha, C. A. Woolsey, C. Sultan, and S. F. De Wekker, "Sensing wind from quadrotor motion," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 4, pp. 836–852, 2019.

[15] M. Jurij, E. M. Nor, A. S. M. Rafie, O. F. Marzuki, M. R. Saad, and S. Azrad, "Development and testing of open-jet wind tunnel for quadrotor flight testing," *Journal of Advanced Research in Fluid Mechanics and Thermal Sciences*, vol. 64, no. 2, pp. 304–315, 2019.

[16] J. A. Benito, G. Glez-de Rivera, J. Garrido, and R. Ponticelli, "Design considerations of a small uav platform carrying medium payloads," in *Design of Circuits and Integrated Systems*, pp. 1–6, IEEE, 2014.

[17] P. E. Pounds, D. R. Bersak, and A. M. Dollar, "Stability of small-scale uav helicopters and quadrotors with added payload mass under pid control," *Autonomous Robots*, vol. 33, no. 1, pp. 129–142, 2012.

[18] Windshape, "Services." `http://windshape.com/services/`, 2022.

[19] P. Daponte, L. De Vito, F. Lamonaca, F. Picariello, M. Riccio, S. Rapuano, L. Pompetti, and M. Pompetti, "Dronesbench: An innovative bench to test drones," *IEEE Instrumentation & Measurement Magazine*, vol. 20, no. 6, pp. 8–15, 2017.

[20] S. P. Khaligh, A. Martínez, F. Fahimi, and C. R. Koch, "A hil testbed for initial controller gain tuning of a small unmanned helicopter," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 289–308, 2014.

[21] N. Xuan-Mung and S.-K. Hong, "A multicopter ground testbed for the evaluation of attitude and position controller," *International Journal of Engineering & Technology*, vol. 7, no. 4.39, pp. 65–73, 2018.

[22] Y. Yu and X. Ding, "A quadrotor test bench for six degree of freedom flight," *Journal of Intelligent & Robotic Systems*, vol. 68, no. 3, pp. 323–338, 2012.

[23] C. Ding, L. Lu, C. Wang, and J. Li, "6-dof automated flight testing using a humanoid robot arm," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*, pp. 217–222, IEEE, 2018.

[24] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "Flightgoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6941–6948, IEEE, 2019.

[25] C. Hildebrandt and S. Elbaum, "World-in-the-loop simulation for autonomous systems validation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10912–10919, IEEE, 2021.

[26] T. Genevois, J.-B. Horel, A. Renzaglia, and C. Laugier, "Augmented reality on lidar data: Going beyond vehicle-in-the-loop for automotive software validation," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, pp. 971–976, IEEE, 2022.

[27] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles-5 Volume Set*. New York, NY, USA: Springer, 2014.

[28] DJI, "Flame wheel arf kit." `https://www.dji.com/flame-wheel-arf/download`, 2022.

[29] Pixhawk, "Pixhawk." `https://pixhawk.org/`, 2022.

[30] A. Shankar, S. Elbaum, and C. Detweiler, "Freyja: A full multirotor system for agile & precise outdoor flights," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 217–223, IEEE, 2021.

[31] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[32] MATLAB, *version 9.12.0 (R2020a)*. Natick, Massachusetts: The MathWorks Inc., 2020.

[33] V. M. S. Ltd, "Vicon motion capture." `https://www.vicon.com`, 2023.

[34] ardupilot, "ardupilot." `https://ardupilot.org`, 2022.

[35] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.