

Robust Path-following Controller for Multirotor Vehicles in Uncertain Wind Conditions

John L. Bullock* and Anna C. Trujillo†
NASA Langley Research Center, Hampton, VA, 23681

Sheng Cheng‡ and Naira Hovakimyan§
University of Illinois Urbana-Champaign, Urbana, IL, 61801

This paper presents simulations of $\mathcal{L}_1$ adaptive controller on a multirotor vehicle for compensation of convective winds. There are increasing efforts to utilize unmanned vehicles for fighting wildfires, but such vehicles need to be able to compensate for convective updrafts and other uncertainties (e.g., wind shear and turbulence) caused by the wildfire. Other application domains, such as urban air mobility, also must account for convective or uncertain winds present in urban microclimates. Geometric controllers have great tracking performance properties in environments without uncertainties, but the tracking performance degrades when wind disturbances and external loads are present in the environment. Introducing $\mathcal{L}_1$ adaptive control into the controller architecture compensates for these wind disturbances and reduces the tracking error caused by such disturbances. The controller architecture is implemented on a small quadrotor. The simulations demonstrate the advantages of the $\mathcal{L}_1$ adaptive augmentation for a sample figure-eight trajectory while encountering unknown variable wind conditions with an external load.

Nomenclature

A_s	=	a diagonal Hurwitz matrix
e_i	=	3-element vector with element i set to 1, and the other elements to 0
e_R	=	rotation matrix error
e_Ω	=	angular velocity error (rad/s)
f_b	=	3-element baseline force vector commanded by the geometric control (N)
g	=	acceleration due to gravity (m/s ²)
m	=	vehicle mass (kg)
$\mathbb{I}$	=	identity matrix
J	=	inertia matrix of the vehicle (kg · m ²)
M_b	=	moment vector commanded by the geometric control (N · m)
p	=	3-element position vector of the multirotor (m)
R	=	rotation matrix of the vehicle
R_d	=	desired rotation matrix of the vehicle
T_s	=	sample time for the discrete-time adaptive control (s)
u_b	=	the baseline control command from the geometric controller
u_{ad}	=	the adaptive control command from the $\mathcal{L}_1$ augmentation
v	=	velocity vector of the multirotor (m/s)
z	=	vector of partial states affected by uncertainties
$\hat{z}$	=	estimation of the partial states z
$\tilde{z}$	=	prediction error of the partial states
σ_m	=	the matched uncertainty
σ_{um}	=	the unmatched uncertainty

*Pathways Intern, Dynamic Systems and Control Branch, NASA Langley Research Center.

†Senior Research Engineer, Dynamic Systems and Control Branch, NASA Langley Research Center.

‡Postdoctoral Research Associate, Department of Mechanical Science and Engineering, University of Illinois Urbana-Champaign.

§Professor, Department of Mechanical Science and Engineering, University of Illinois Urbana-Champaign. AIAA Fellow.

Ω = angular velocity vector of the vehicle (rad/s)
 Ω_d = desired angular velocity of the vehicle (rad/s)

I. Introduction

One emerging application for multirotor vehicles is in the domain of wildland firefighting. NASA has recently started its Advanced Capabilities for Emergency Response Operations (ACERO) project, which is investigating the use of drones for wildfire suppression [1, 2]. Current aerial operations for fighting wildland fires are normally limited to Visual Meteorological Conditions (VMC) that typically last six to eight hours during the day. However, these hours may be further reduced due to smoke and smoke inversions, which result in Degraded Visual Environments (DVE). Drones, or Uncrewed Aerial Systems (UAS), offer a significant advantage by enabling aerial operations during DVE while ensuring pilot safety. This capability not only supports aerial suppression of wildfires but also expands operational possibilities for other fire-related missions, including search and rescue, logistical support to ground crews, and potentially the transportation of wildland firefighters.

However, wildfires pose a challenging environment for drone operations, in part because fires often generate their own weather through pyrocumulonimbus clouds and frequently occur in mountainous terrain. Pyrocumulonimbus clouds form when intense heat from the fire causes air to rise rapidly, forming a powerful updraft that carries smoke, ash, and water vapor high into the atmosphere, where it cools and condenses, forming massive clouds. This impacts flight stability over and around wildfires because of the convective updrafts and other wind disturbances (e.g., wind shear and turbulence) that the wildland fire can induce. These clouds can also interact with the traditional mountainous terrain winds that produce mountain winds, valley winds, and mountain waves. Mountain winds typically occur at night when it is cooler, with the cooler and heavier air blowing down the mountain. Valley winds exhibit the opposite behavior, typically occurring during the day with the wind blowing up mountains. Mountain waves are vertical oscillations downwind of high ground, and these strong winds can contribute to unexpected wildfire behavior, growth, and spread. In order to effectively counter these unknown and turbulent conditions, drones should be equipped with a robust controller for trajectory tracking, which can compensate for these wind uncertainties that are present in wildfires.

Furthermore, wildland fire aerial operations often involve retardant or water drops to help suppress the fire. These drops often use a sling load such as a Bambi[®]* bucket [3]. Sling loads further complicate the dynamics that the aerial asset must manage when following a trajectory, especially in the highly dynamic environment described above. These loads may be released at various rates, such as immediately if the aerial asset is delivering provisions to firefighters on the ground or at a fairly steady rate when spreading water or fire retardant at or near the fire line. Once again, the controller must be able to handle varying load release rates and the disturbances associated with them.

We consider a geometric controller with an $\mathcal{L}_1$ adaptive controller on a multirotor to compensate for these wind and load uncertainties. The initial vehicle was a NASA Langley Research Center quadrotor [4]. Once the geometric controller and $\mathcal{L}_1$ adaptive controller are validated in simulations using both internal and sling loads, the system must be tested in flight. After this, a spiral development is planned that involves moving on to larger drones used by wildland firefighters, such as Freefly's[®] Alta X, with the ultimate goal of development and flight testing on a crewed wildland fire aerial asset. Besides demonstrating the efficacy of these algorithms for wildland firefighting missions, an end goal of the ACERO project and an objective of this research is to develop a process and a vehicle system control architecture that will enable this control system to be adapted to many wildland firefighting aerial assets.

The remainder of this paper is organized as follows: Section II provides the technical details of the multirotor dynamics, the implemented geometric controller, and the $\mathcal{L}_1$ adaptive controller. Section III introduces the architecture of safely integrating the geometric and adaptive controllers with the current vehicle's control system. Section IV presents results from simulations that include carrying a load, such as retardant or supplies. Lastly, Section V details the conclusions about the effectiveness of using a geometric controller with an $\mathcal{L}_1$ adaptive controller with regard to the path following capabilities of a multirotor in uncertain wind conditions with carrying and dropping a load.

II. Technical Details

In this section, the general dynamics of the multirotor are briefly described. Then, the geometric and $\mathcal{L}_1$ adaptive controllers used in the simulations are presented.

*The use of trademarks or names of manufacturers in this report is for accurate reporting and does not constitute an official endorsement, either expressed or implied, of such products or manufacturers by the National Aeronautics and Space Administration.

A. Dynamics

The drone is modeled with the following dynamic system typically seen for multirotor vehicles:

$$\dot{p} = v, \quad p(0) = p_0, \quad (1a)$$

$$\dot{v} = ge_3 - \frac{f_b}{m} Re_3, \quad v(0) = v_0, \quad (1b)$$

$$\dot{R} = R\Omega^\wedge, \quad R(0) = R_0, \quad (1c)$$

$$\dot{\Omega} = J^{-1}(M_b - \Omega \times J\Omega), \quad \Omega(0) = \Omega_0, \quad (1d)$$

with $\wedge : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ denoting the function mapping to the space of skew-symmetric matrices such that $(x^\wedge)y = x \times y$ for all $x, y \in \mathbb{R}^3$ [5].

B. Geometric Controller

This section presents the design of the baseline path-following controller. There are a variety of design methodologies for controlling nonlinear systems like multirotors, and this work considers a geometric controller. Geometric controllers have the desired properties of being globally exponentially stable for multirotor attitude errors less than 90° , and being almost globally exponentially attractive for attitude errors less than 180° [6]. Such properties will ensure that the system can quickly converge to a desired trajectory in the absence of wind disturbances or other uncertainties in the multirotor system. For this section, the desired outputs of the controller are the generalized force, f_b , and moment, M_b , generated by the multirotor. Control allocation is not considered for the purposes of this work.

The geometric controller established in [6] provides the requisite equations for computing the baseline force f_b and moment M_b . The equations for the commanded moment and force, respectively, are

$$M_b = -K_R e_R - K_\Omega e_\Omega + \Omega \times J\Omega - J(\Omega^\wedge R^T R_d \Omega_d - R^T R_d \dot{\Omega}), \quad (2)$$

$$f_b = (K_p e_x + K_v e_v + mg - \ddot{p}_d) Re_3. \quad (3)$$

The error terms used in (Eq. 2) and (Eq. 3) are computed as:

$$e_R = \frac{1}{2} (R_d^T R - R^T R_d)^\vee, \quad (4a)$$

$$e_\Omega = \omega - R^T R_d \Omega_d, \quad (4b)$$

$$e_x = p - p_d, \quad (4c)$$

$$e_v = v - v_d, \quad (4d)$$

with $\vee : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$ denoting the inverse mapping of the $\wedge$ operator, as defined in [5] and [6].

C. $\mathcal{L}_1$ Adaptive Control

Now that a baseline controller has been established, the baseline commands f_b and M_b are augmented with adaptive control signals to compensate for any disturbances in the system. The $\mathcal{L}_1$ adaptive control architecture is used for this application because of its capability for fast adaptation in the presence of rapidly changing disturbances [7], such as wind gusts. Figure 1 provides a diagram of the $\mathcal{L}_1$ architecture and how it integrates into the baseline geometric controller design. The architecture of $\mathcal{L}_1$ is composed of three main components, which are the state predictor, the adaptation law, and the low-pass filter.

1. State Predictor

The formulation of $\mathcal{L}_1$ adaptive control for this problem mirrors the derivation in [8]. First, we note that the only dynamics in (Eq. 1a)–(Eq. 1d) that are affected by uncertainties are (Eq. 1b) and (Eq. 1d). Hence, for the state predictor only these equations need to be considered.

To represent these uncertain dynamics concisely, a partial state can be defined as $z = [v, \Omega]$, which considers the state variables directly subject to the uncertainties. The baseline controls are also lumped into one vector $u_b = [f_b, M_b]^T$ for conciseness. With the uncertainties, the dynamics of the partial state can then be written as

$$\dot{z}(t) = f(z(t)) + B(R(t))(u_b(t) + u_{ad}(t) + \sigma_m(t, x(t))) + B^\perp(R(t))\sigma_{um}(t, x(t)), \quad z(0) = z_0, \quad (5)$$

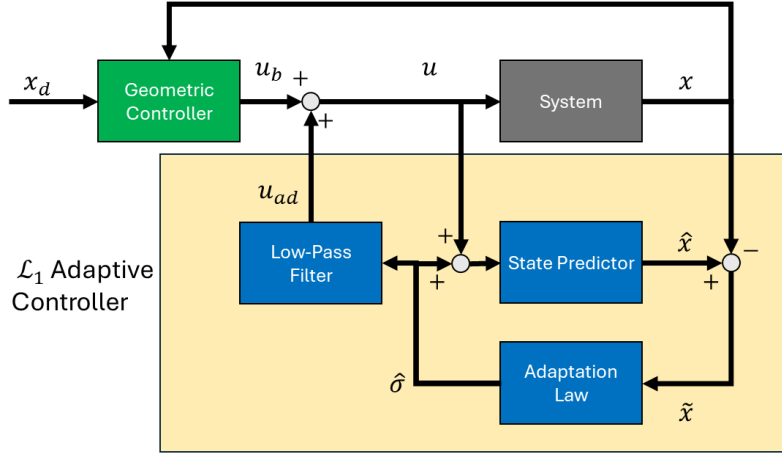


Fig. 1 Block Diagram of the Controller Architecture

where the matched uncertainty is denoted as σ_m , and the unmatched uncertainty is denoted as σ_{um} . From (Eq. 5), we can see that the matched uncertainty acts in the same channels as the control input, and the unmatched uncertainty acts in the null space of $B(R(t))$. Using the equations (Eq. 1b) and (Eq. 1d), we can define the f and B terms in (Eq. 5) as:

$$f(z) = \begin{bmatrix} ge_3 \\ -J^{-1}(\Omega \times J\Omega) \end{bmatrix},$$

$$B(R) = \begin{bmatrix} -\frac{1}{m}Re_3 & 0_{3 \times 3} \\ 0_{3 \times 1} & J^{-1} \end{bmatrix},$$

$$B^\perp(R) = \begin{bmatrix} \frac{1}{m}Re_1 & \frac{1}{m}Re_2 \\ 0_{3 \times 1} & 0_{3 \times 1} \end{bmatrix}.$$

The state predictor provides an estimate of what the partial states should be, given the control inputs and the estimated uncertainties that the system is subject to. The state predictor makes its estimates, $\hat{z}(t)$, according to the following equation:

$$\dot{\hat{z}} = f(z(t)) + B(R(t))(u_b(t) + u_{ad}(t) + \hat{\sigma}_m(t)) + B^\perp(R(t))\hat{\sigma}_{um}(t) + A_s \tilde{z}(t), \quad \hat{z}(0) = \hat{z}_0. \quad (6)$$

In (Eq. 6), the estimations of the matched and unmatched uncertainties are denoted by $\hat{\sigma}_m$ and $\hat{\sigma}_{um}$, respectively. The state prediction error is represented by $\tilde{z} = \hat{z} - z$, and A_s is a Hurwitz matrix.

2. Adaptation Law

The adaptation law is used to update the uncertainty estimates $\hat{\sigma}_m$ and $\hat{\sigma}_{um}$, which depend in part on the prediction error term $\tilde{z}$. The piecewise-constant adaptation law used in [8] is given as

$$\hat{\sigma}(t) = -\bar{B}(iT_s)^{-1}\Phi^{-1}\mu(iT_s), \quad \forall t \in [iT_s, (i+1)T_s], \quad (7)$$

where

$$\hat{\sigma}(t) = [\hat{\sigma}_m(t), \hat{\sigma}_{um}(t)]^T, \quad (8a)$$

$$\bar{B}(t) = [B(R(t)), B^\perp(R(t))], \quad (8b)$$

$$\Phi = A_s(\exp(A_s T_s) - \mathbb{I}), \quad (8c)$$

$$\mu(iT_s) = \exp(A_s T_s)\tilde{z}(iT_s), \quad (8d)$$

for time index i . Assuming that the sampling time T_s is small enough, this adaptation law is shown in [8] to reduce $\|\tilde{z}\|$, meaning that the estimated lumped uncertainties are providing a good approximation of the true uncertainties.

3. Control Law

The last component of the $\mathcal{L}_1$ architecture is the control law. The control law is implemented with a low-pass filter so that the adaptive command only compensates for the uncertainties within the bandwidth of the filter.

In the frequency domain, the adaptive command can be computed as

$$u_{ad}(s) = -C(s)\hat{\sigma}_m(s), \quad (9)$$

where $C(s)$ is a low-pass filter.

III. Architecture

The geometric + $\mathcal{L}_1$ controller will be a component inside the Flight Envelope Protection System (FEPS) for the vehicle, which was developed at NASA Langley Research Center as part of the ACERO project. The FEPS will allow for the use of the geometric + $\mathcal{L}_1$ controller in conjunction with a commercial off-the-shelf (COTS) controller for the vehicle. The FEPS will provide the safety assurances of the COTS controller while the performance of the geometric + $\mathcal{L}_1$ controller is being experimentally verified. This ability of being able to revert back to the COTS controller is similar in principle to the NASA AirSTAR flight test infrastructure [9, 10], which was experimentally validated through flight testing.

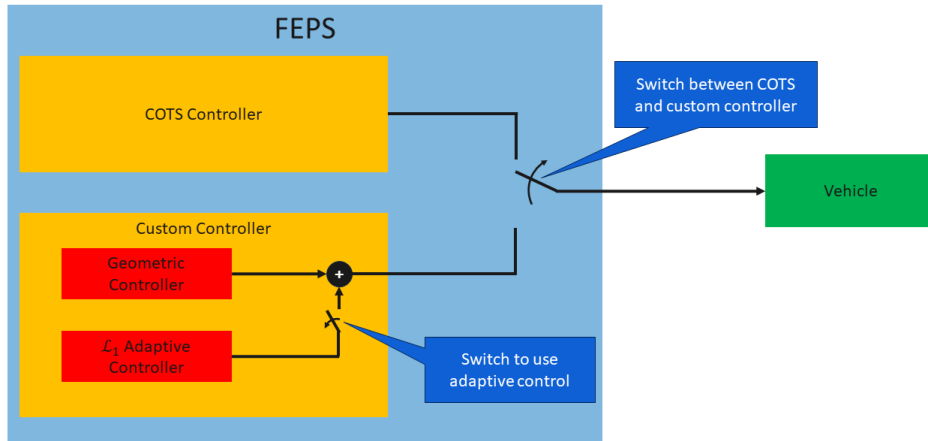


Fig. 2 Structure of Flight Envelope Protection System

IV. Simulations and Simulation Results

This controller architecture was first validated in a quadrotor simulation that accounts for details such as dynamic thrust, air resistance, and blade flapping [11]. The specific quadrotor simulated is the IMPACT vehicle [4]. The top level Simulink® diagram maintained the structure of [4] (Fig. 3) with an added “Disturbances & Dynamic Events” subsystem (yellow box in Fig. 3).

A. Geometric and $\mathcal{L}_1$ Controller

The geometric and $\mathcal{L}_1$ controller are under “Multirotor Controller” (orange box in Fig. 3). The geometric controller has vehicle position and velocities along with desired dynamics—in particular, desired position, velocity, acceleration, and yaw—as inputs. The outputs of the geometric controller are force and torque commands (Section II.B). These commands are eventually translated into RPM commands for the motors.

The $\mathcal{L}_1$ adaptive controller has inputs of vehicle position and velocities, previously commanded states, and uncertainties. The outputs are the state predictions, adaptive commands, and an updated estimate of the uncertainties (Section II.C).

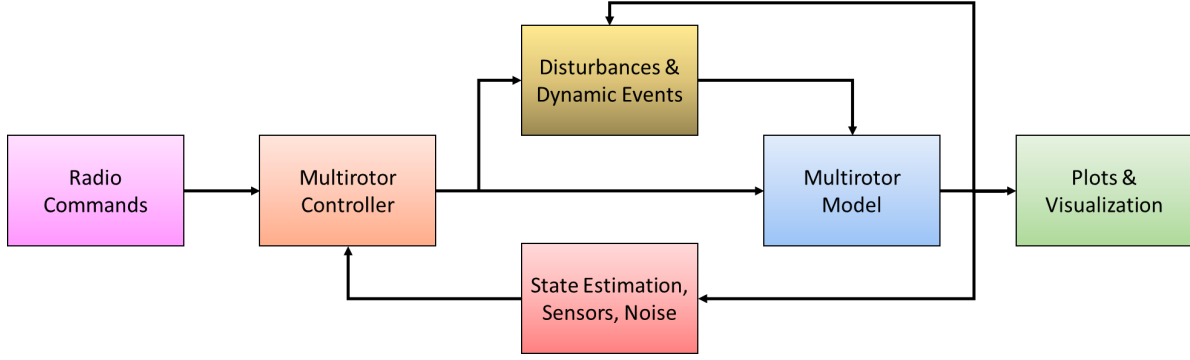


Fig. 3 Simulation Top Level

A figure-eight trajectory, shown in Fig. 4, was used to demonstrate the capability of this controller with unknown disturbances. The simulated vehicle is limited to track twice-differentiable trajectories, such as the figure-eight trajectory, due to the state information needed by the geometric controller.

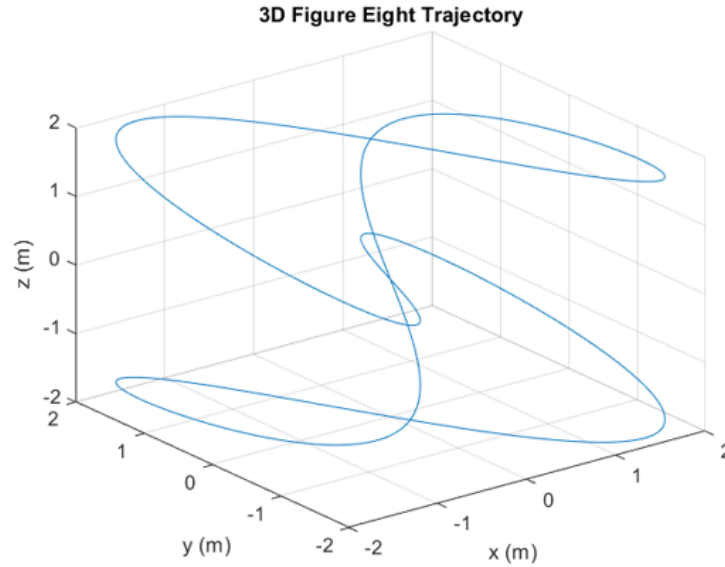


Fig. 4 Sample Figure-eight Trajectory

B. Winds

Winds are modelled using the “Environment” subsystem from the GUAM (Generic Urban Air Mobility) simulation [12]. The winds for this simulation were modelled as indicated in Table 1. The simulated wind disturbances had a horizontal and vertical wind component of 10 mph with the horizontal wind at 45° from north and a vertical updraft. These values are easily modified in a MATLAB® setup m-file.

The plots in Figs. 5–7 characterize the vehicle’s trajectory-tracking performance under the presence of wind. From Fig. 7, the geometric + $\mathcal{L}_1$ controller exhibits much smaller tracking errors than just the geometric controller alone. Table 2 shows that the z-direction exhibited a 93.6% reduction in the RMS z-error, which can also be seen in Fig. 5 and Fig. 7. When using the $\mathcal{L}_1$ augmentation, because the z-direction is in the span of the $B(R)$ matrix, the uncertainties acting in the z-direction are called matched uncertainties and can be compensated for with the adaptive commands. Also of note, the $\mathcal{L}_1$ plot in Fig. 6 shows that it can compensate for the wind disturbances with smooth command signals, which leads to smooth transient performance in the presence of the wind.

Table 1 Wind Parameters

Parameter	Amount
Vehicle altitude	50 ft MSL [*]
Horizontal wind speed	10 mph
Horizontal wind direction	45 deg
Vertical wind speed	-10 mph [†]
Turbulence level	1 for first 4 sec then 3 [‡]

^{*} Mean Sea Level

[†] - = up and + = down

[‡] 1 = none, 2 = low, 3 = moderate, 4 = severe

Table 2 RMS Error with Wind Disturbance

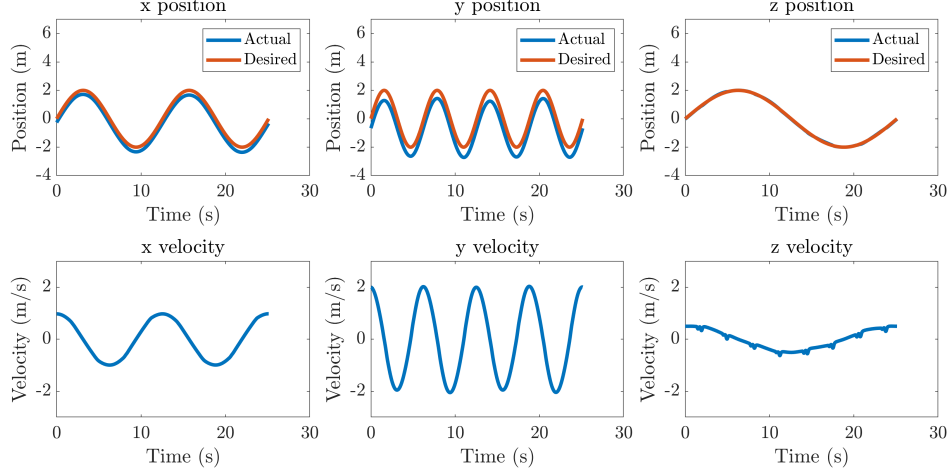
	No $\mathcal{L}_1$		$\mathcal{L}_1$		RMS Error Reduction with $\mathcal{L}_1$
	RMS Error	Max Error	RMS Error	Max Error	
X-position	0.200 m	0.379 m	0.196 m	0.369 m	2.0 %
Y-position	0.640 m	0.806 m	0.621 m	0.796 m	3.0 %
Z-position	0.469 m	0.661 m	0.030 m	0.044 m	93.6 %
Yaw	12.410°	20.525°	12.300°	20.556°	0.9 %

C. Load Weight

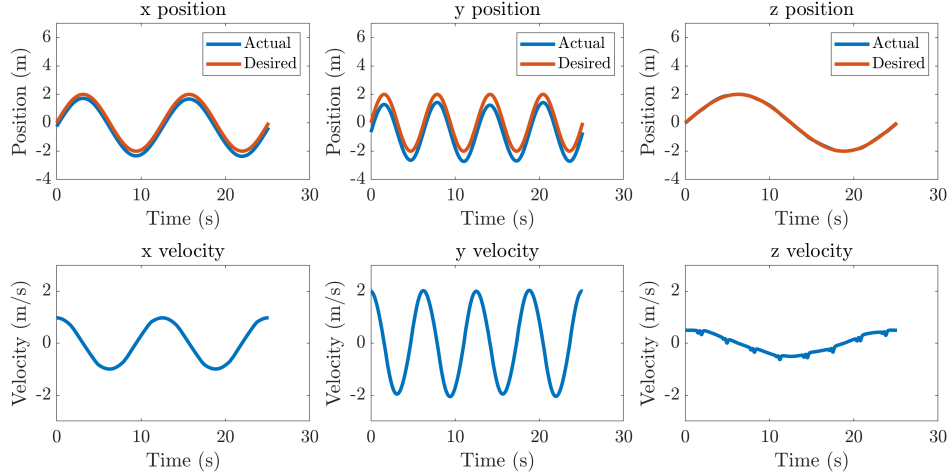
The load weight is modelled under the “Disturbances and Dynamic Events” subsystem (Fig. 3). Three types of load weights were simulated. The baseline condition was ‘no additional external load.’ The second type of external load mimicked the immediate dropping of the whole load. This simulated a provision delivery to wildland firefighters. For this simulation, the load weight was 0.2 kg with a container weight of 0.1 kg. The load plus container weights were selected based on the load capacity of the IMPACT drone. The last type of load simulated a fire retardant or a water drop with a steady decrease in weight. Once again, for this simulation, the load weight was 0.2 kg with a container weight of 0.1 kg, and the load was released over 5 s. As with the winds, these values are easily modified in a MATLAB[®] setup m-file.

In the simulation results shown below, an immediate load drop was simulated because this affects the dynamics of the aerial vehicle the most due to the immediate mass change. Therefore, a sudden 0.3 kg decrease in weight was simulated halfway through the flight, which represents a provision payload drop. Note that for this simulation, the load was considered to be close to the vehicle’s center of mass. Thus, the minimal change in the inertia matrix when the load was dropped was not factored in for this initial simulation. Figures 8–10 show how the load drop affects the vehicle, with and without $\mathcal{L}_1$ augmentation.

The addition of the $\mathcal{L}_1$ adaptive control significantly improves the tracking performance of the vehicle, reducing the RMS z-error by 96% (Table 3). This error reduction is especially evident in the state error plot shown in Fig. 10. The geometric controller by itself cannot compensate for the mass uncertainty caused by the load drop, hence it experiences large tracking errors, especially in the z-direction (Fig. 8a and Fig. 10a). However, the $\mathcal{L}_1$ controller is able to compensate for the matched disturbance in the z-direction caused by the mass change and have minimal error in tracking the z-coordinates (Fig. 8b and Fig. 10b). The uncertainties in the x- and y-directions are unmatched uncertainties, so the improvements of the $\mathcal{L}_1$ controller are less noticeable since we cannot directly compensate for disturbances in these directions. As with just the wind being present, the $\mathcal{L}_1$ plot in Fig. 9 shows that it can compensate for the wind and load disturbances with smooth command signals, which leads to smooth transient performance in the presence of the wind and the load drop.



(a) Geometric Controller Only



(b) Geometric + $\mathcal{L}_1$ Controller

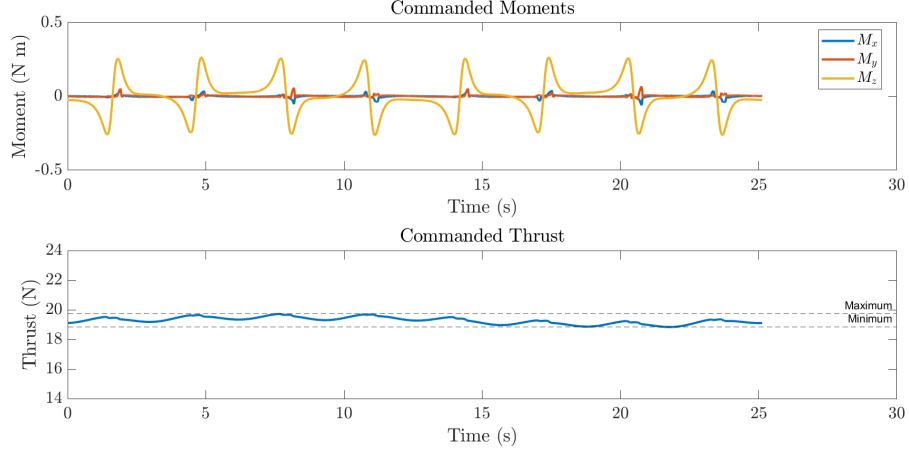
Fig. 5 Translational States with and without $\mathcal{L}_1$

Table 3 RMS Error with Wind Disturbance and Load Change

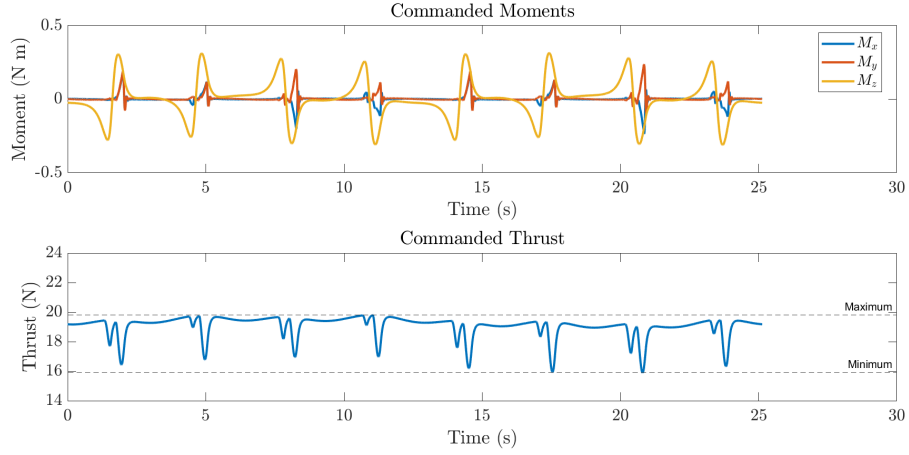
	No $\mathcal{L}_1$		$\mathcal{L}_1$		RMS Error Reduction with $\mathcal{L}_1$
	RMS Error	Max Error	RMS Error	Max Error	
X-position	0.202 m	0.405 m	0.180 m	0.344 m	10.9 %
Y-position	0.695 m	1.191 m	0.581 m	0.770 m	16.4 %
Z-position	3.093 m	4.582 m	0.113 m	0.134 m	96.3 %
Yaw	12.353°	20.513°	12.340°	20.861°	0.1 %

V. Conclusions

In the context of firefighting, it is important that a vehicle's controller is accurate and robust to avoid trees and other obstacles in the environment. The geometric controller augmented with $\mathcal{L}_1$ demonstrated the capability to accurately track a set trajectory, with the geometric controller providing almost global asymptotic stability under nominal flight conditions. The $\mathcal{L}_1$ controller specifically showed the ability to compensate for both wind disturbances and sudden load drops throughout the simulation, both of which are disturbances that can commonly occur in a firefighting setting.



(a) Geometric Controller Only



(b) Geometric + $\mathcal{L}_1$ Controller

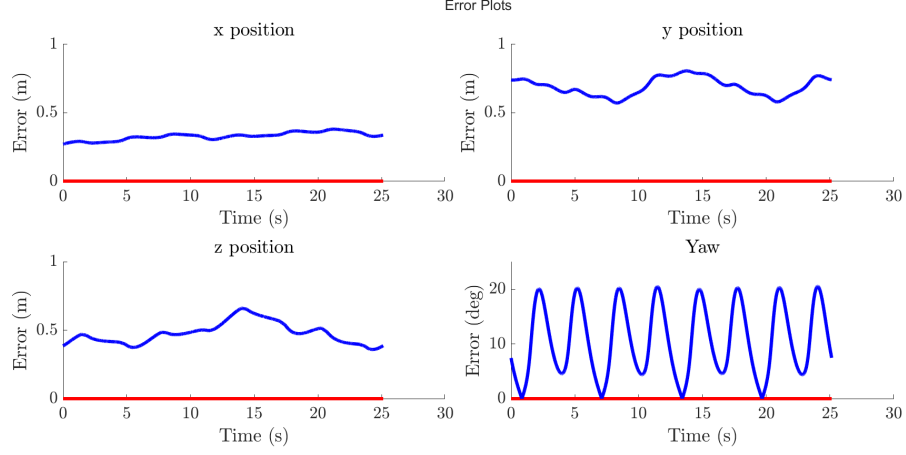
Fig. 6 Commanded Moments and Thrust with and without $\mathcal{L}_1$

The next step envisioned for this research is to implement a sling load modeled after [13] and then add sloshing effects, which may be significant during retardant and water drops in the harsh environment of wildland fires. During this addition, flight testing will be conducted using the IMPACT vehicle at various stages to validate the simulation. Once the simulation is validated by flight tests on the IMPACT vehicle, a geometric controller and $\mathcal{L}_1$ adaptive controller will be developed on a Freefly[®] Alta X, which is a resource of the US Forestry Service (USFS) [14], before moving on to piloted and remotely-piloted large aerial assets that are typically used in dropping water or fire retardant.

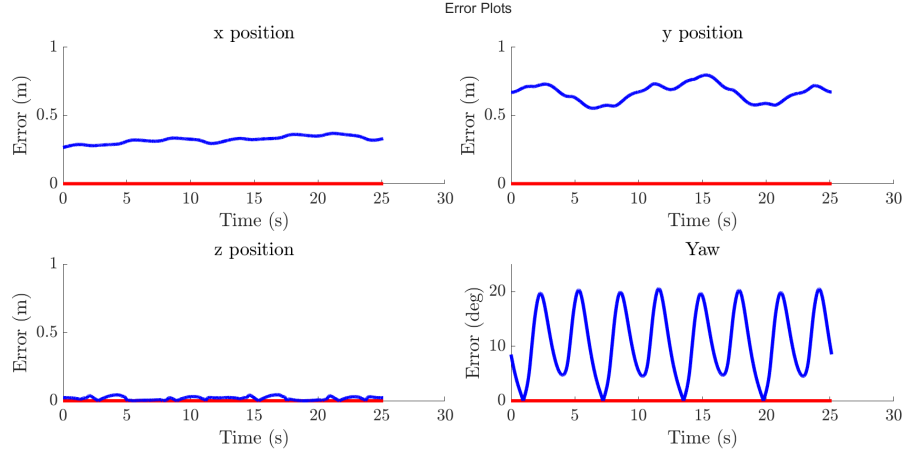
In conjunction with the simulation development and validation, the process for tuning the geometric controller and the $\mathcal{L}_1$ adaptive controller will be finalized so that those involved with the firefighting community providing aerial assets can implement these algorithms. Auto-tuning tools such as DiffTune [15, 16] have been shown to be effective in tuning geometric controllers, so auto-tuning methods can be applied to improve the tracking performance of firefighting vehicles. Concurrently, a system architecture is being developed as discussed in Section III that enables these controllers to “sit on top” of the COTS controllers. This architecture may allow quicker adaptation of these concepts since control can be reverted back to the COTS controller during development and validation tests.

Acknowledgments

This work is supported by the ACERO Project in NASA’s Airspace Operations and Safety Program. Additional input was provided by the Intelligent Contingency Management group that is support by the Transformational Tools and



(a) Geometric Controller Only



(b) Geometric + $\mathcal{L}_1$ Controller

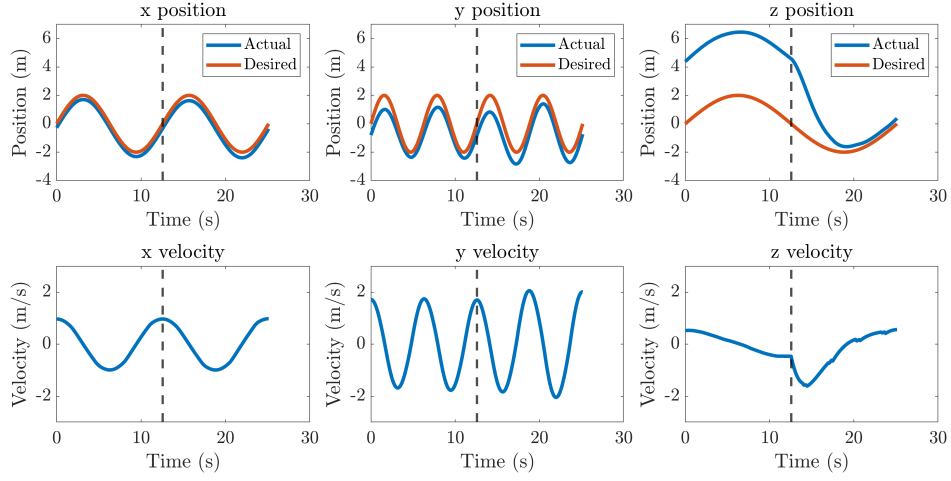
Fig. 7 Tracking Error with Wind

Technologies Sub-project under the Transformative Aeronautics Concepts Program.

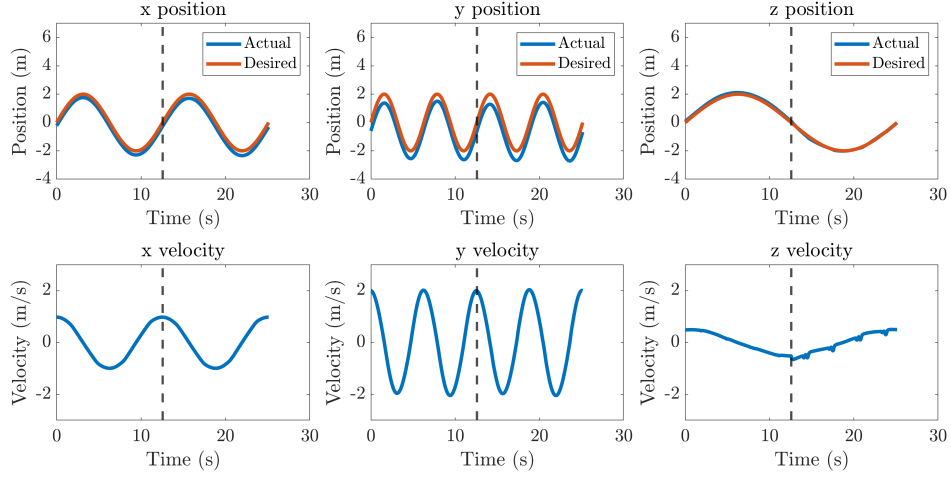
References

- [1] Smith, H., “Advanced Capabilities for Emergency Response Operations (ACERO),” NASA Web page, April 2023. URL <https://www.nasa.gov/centers-and-facilities/ames/advanced-capabilities-for-emergency-response-operations-acero/>.
- [2] Verma, K., “ACERO: Improving Wildland Firefighting Operations with Drones and Aviation Technologies,” *Interesting Engineering*, 2023. URL <https://interestingengineering.com/science/acero-improving-wildland-firefighting-operations>.
- [3] SEI Industries, “Bambi Bucket,” Web page, 2024. URL <https://www.sei-ind.com/products/bambi-bucket/>.
- [4] Asper, G. D., Simmons, B. M., Axten, R. M., Ackerman, K. A., and Corrigan, P. E., “Inexpensive Multirotor Platform for Advanced Controls Testing (IMPACT): Development, Integration, and Experimentation,” Tech. Rep. NASA/TM-20240000223, NASA Langley Research Center, Hampton, VA, March 2024. URL <https://ntrs.nasa.gov/api/citations/20240000223/downloads/NASA-TM-20240000223.pdf>.
- [5] Lee, T., Leok, M., and McClamroch, N. H., “Geometric Tracking Control of a Quadrotor UAV on SE(3),” *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425. <https://doi.org/10.1109/CDC.2010.5717652>.

- [6] Lee, T., Leok, M., and McClamroch, N. H., “Control of Complex Maneuvers for a Quadrotor UAV using Geometric Methods on $SE(3)$,” *ArXiv*, Vol. abs/1003.2005, 2011. URL <https://export.arxiv.org/abs/1003.2005v4>.
- [7] Hovakimyan, N., and Cao, C., $\mathcal{L}_1$ *Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*, Society for Industrial and Applied Mathematics, USA, 2010.
- [8] Wu, Z., Cheng, S., Ackerman, K. A., Gahlawat, A., Lakshmanan, A., Zhao, P., and Hovakimyan, N., “ $\mathcal{L}_1$ Adaptive Augmentation for Geometric Tracking Control of Quadrotors,” *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1329–1336. <https://doi.org/10.1109/ICRA46639.2022.9811946>.
- [9] Murch, A., “A Flight Control System Architecture for the NASA AirSTAR Flight Test Infrastructure,” *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2008-6990>.
- [10] Murch, A., Cox, D., and Cunningham, K., “Software Considerations for Subscale Flight Testing of Experimental Control Laws,” *AIAA Infotech@Aerospace Conference*, 2009. URL <https://arc.aiaa.org/doi/abs/10.2514/6.2009-2054>.
- [11] Vervoorst, J. W., “A Modular Simulation Environment for the Improved Dynamic Simulation of Multirotor Unmanned Aerial Vehicles,” Master’s thesis, University of Illinois Urbana-Champaign, 2016.
- [12] Acheson, M. J., *Generic-Urban-Air-Mobility-GUAM README*, NASA Langley Research Center, Hampton, VA, v1.1 ed., May 2024. URL <https://github.com/nasa/Generic-Urban-Air-Mobility-GUAM>.
- [13] Trachte, J. E., Toro, L. F. G., and McFadyen, A., “Multi-rotor with Suspended Load: System Dynamics and Control Toolbox,” *2015 IEEE Aerospace Conference*, 2015, pp. 1–9. <https://doi.org/10.1109/AERO.2015.7119210>.
- [14] Karpowicz, J., “Success and Scale with Drone Programs: How the US Forest Service Saves Taxpayer Dollars and Reduces Wildfire Response Risk,” *Commercial UAV News*, 2023. URL <https://www.commercialuavnews.com/public-safety/success-and-scale-with-drone-programs-how-the-u-s-forest-service-saves-taxpayer-dollars-and-reduces-wildfire-response-risk>.
- [15] Cheng, S., Kim, M., Song, L., Yang, C., Jin, Y., Wang, S., and Hovakimyan, N., “DiffTune: Autotuning Through Autodifferentiation,” *IEEE Transactions on Robotics*, Vol. 40, 2024, pp. 4085–4101. <https://doi.org/10.1109/TRO.2024.3429191>.
- [16] Cheng, S., Song, L., Kim, M., Wang, S., and Hovakimyan, N., “DiffTune⁺: Hyperparameter-Free Auto-Tuning using Auto-Differentiation,” *Proceedings of The 5th Annual Learning for Dynamics and Control Conference*, Proceedings of Machine Learning Research, Vol. 211, edited by N. Matni, M. Morari, and G. J. Pappas, PMLR, 2023, pp. 170–183. URL <https://proceedings.mlr.press/v211/cheng23b.html>.

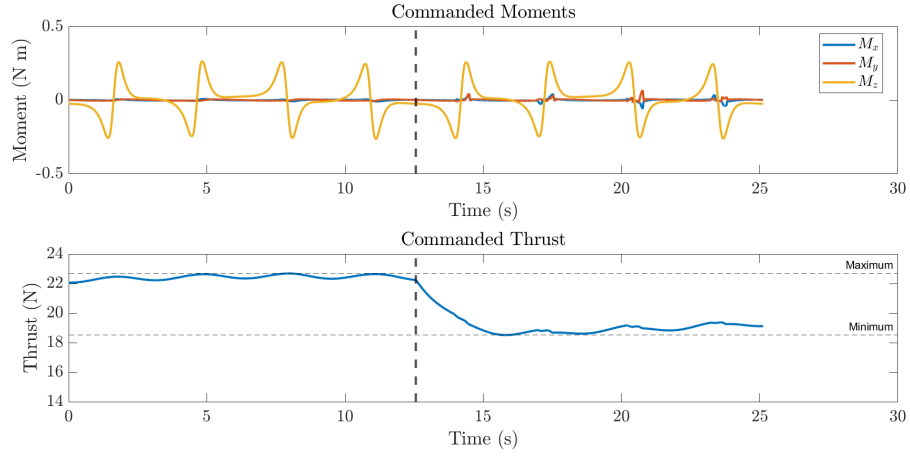


(a) Geometric Controller Only

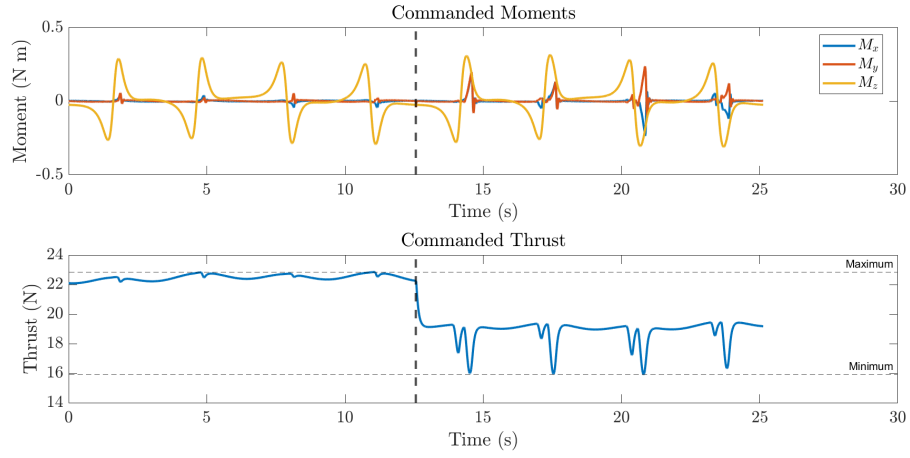


(b) Geometric + $\mathcal{L}_1$ Controller

Fig. 8 Translational States with Wind and Load Change with Load Drop Indicated by Dotted Line

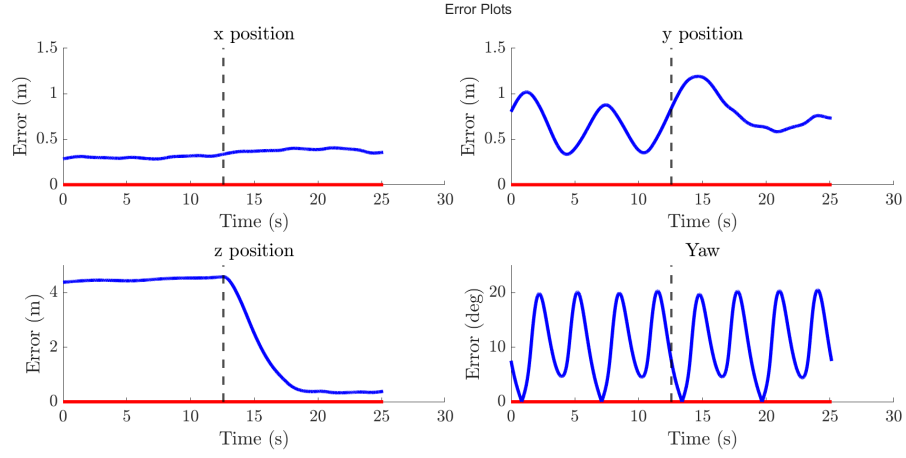


(a) Geometric Controller Only

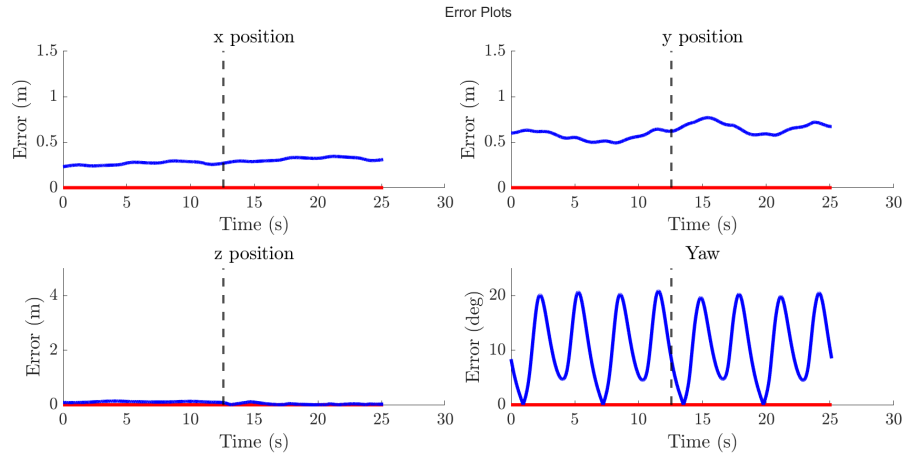


(b) Geometric + $\mathcal{L}_1$ Controller

Fig. 9 Commanded Moments and Thrust with Wind and Load Change



(a) Geometric Controller Only



(b) Geometric + $\mathcal{L}_1$ Controller

Fig. 10 Tracking Error with Wind and Load Change