Preprints of the 23rd IFAC World Congress
Busan, Republic of Korea, August 23-28, 2026

The *E-Rocket*: Low-cost Testbed for TVC Rocket GNC Validation ⋆

Pedro Santos * **André Fonte** ** **Pedro Martins** **
Paulo Oliveira *

* *IDMEC and ISR, IST, University of Lisbon, 1049-001 Portugal (e-mail: {pedrodossantos31, paulo.j.oliveira}@tecnico.ulisboa.pt)*
** *IST, University of Lisbon, 1049-001 Portugal (e-mail: {andrevfonte, pedromcamartins}@tecnico.ulisboa.pt)*

Abstract: This paper presents the *E-Rocket*, an electric-powered, low-cost rocket prototype for validation of Guidance, Navigation & Control (GNC) algorithms based on Thrust Vector Control (TVC). Relying on commercially available components and 3D printed parts, a pair of contra-rotating DC brushless motors is assembled on a servo-actuated gimbal mechanism that provides thrust vectoring capability. A custom avionics hardware and software stack is developed considering a dual computer setup which leverages the capabilities of the PX4 autopilot and the modularity of ROS 2 to accommodate for tailored GNC algorithms. The platform is validated in an indoor motion-capture arena using a baseline PID-based trajectory tracking controller. Results demonstrate accurate trajectory tracking and confirm the suitability of the *E-Rocket* as a versatile testbed for rocket GNC algorithms.

Keywords: Aerial robotics, avionics and onboard equipments, GNC, thrust vector control.

1. INTRODUCTION

The space industry is undergoing a paradigm shift toward reusable launch systems for their cost-effectiveness (Ragab and Cheatwood, 2015) and higher launch cadence. Innovating in this domain is recognized as an expensive, high-risk, and elitist endeavour. In this context, small-scale aerial vehicles emerge as accessible and safe platforms for advanced Guidance, Navigation, and Control (GNC) development and test, appealing to both industry and academia (Jacinto et al., 2024). Commercial off-the-shelf (COTS) prototypes, such as multirotors, have been widely adopted in GNC research for their affordability and ease of use. However, these lack the specific dynamics and challenges of rocket flight. The herein proposed electrically-powered rocket, *E-Rocket* [1], fills this gap by providing a replicable testbed to explore novel control strategies tailored to the unique aspects of rocketry under controlled conditions.

Few scientific works address the niche of electric rockets under thrust vector control (TVC). Notably, two works from Switzerland have been published with roots in model predictive control (MPC) and extended Kalman filter (EKF) based state estimation (Spannagl et al., 2021; Linsen et al., 2022). Both solutions were validated in simulation and experimentally, demonstrating the agile capabilities of the platform. Broadening the scope reveals a richer variety of strategies (Hua et al., 2013). Nonlinear layered architectures have been applied to sounding rockets yet lack an accessible testbed for ultimate flight validations (Santos and Oliveira, 2025).

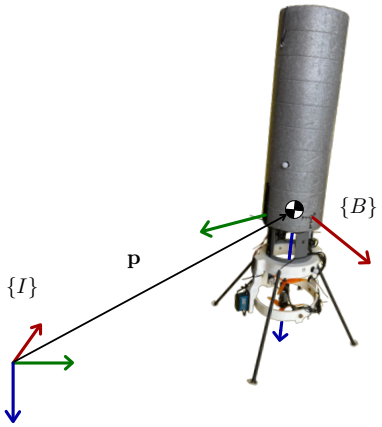


Fig. 1. The *E-Rocket* (red, green, and blue correspond to the x, y, and z axes, respectively).

Contributions of this work span across various domains. The cost-effective mechanical design, which relies on readily available components, facilitates the replication of the platform in academic settings. The avionics stack leverages a dual computer setup to offload low-level sensing and actuation to the well-established PX4 autopilot (Meier et al., 2015) while accommodating for custom GNC algorithms (which may have authority up to the actuators) inside a novel, open-source application running a ROS 2 network (Macenski et al., 2022). An original model is introduced, together with a PID-based control strategy for trajectory tracking, to serve as baseline for this class

⋆ The authors acknowledge Fundação para a Ciência e Tecnologia (FCT) for its financial support via LAETA (DOI: 10.54499/UID/50022/2025). Pedro Santos holds a Ph.D. scholarship from FCT (2023.00268.BD).

[1] All developed software and CAD files of the vehicle are publicly available at: https://github.com/The-E-Rocket-Project.

©2026 the authors. Accepted by IFAC for publication under a Creative Commons Licence CC-BY-NC-ND

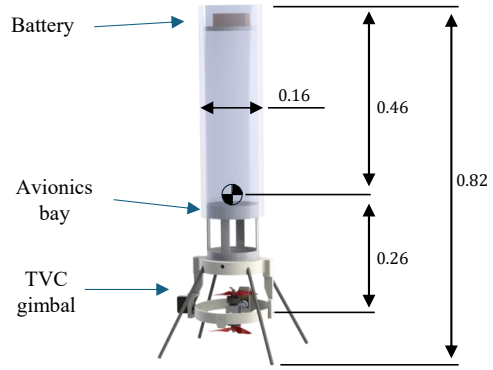


Fig. 2. *E-Rocket* schematics. All dimensions in meters.

of vehicles. Finally, the platform is validated in an indoor flight arena with the proposed baseline control strategy.

2. SYSTEM OVERVIEW

2.1 Structural Design

The *E-Rocket* design prioritized material cost efficiency and availability, emphasizing the use of in-house 3D-printed components and readily available low-cost parts. Figure 2 presents a schematics of the *E-Rocket* with some important dimensions and components identified, while Table 1 displays the system mass breakdown.

A gimbal mechanism with two rotational degrees of freedom, represented in Fig. 3, is the fundamental component, which allows for Thrust Vector Control (TVC). Thrust is generated by a pair of contra-rotating brushless electric motors with 5 inch propellers. The motor pair is attached to an iron rod whose rotation is controlled by a servomotor, providing the first actuation degree of freedom. This innermost actuation mechanism is attached to a 3D printed outer ring which can rotate around an axis orthogonal to the inner one. A second servomotor controls rotation around this axis, providing the second actuation degree of freedom and completing the TVC gimbal. Each brushless motor can be controlled independently, allowing for the generation of a differential torque on the axis aligned with the thrust vector. This additional actuation input acts mainly on the yaw channel to prevent spinning motion or to actively control the yaw angle.

The gimbal mechanism is attached via two arms to a 3D printed disk that serves as core structural element of the vehicle. Four carbon fiber tubes attached to it work as legs, which allow the vehicle to take-off and land vertically on the ground. Additionally, a PVC tube, which houses the

Table 1. System mass properties.

Component	Value (kg)	% Total
Gimbal mechanism and motors	0.650	40.6
PVC tube	0.136	8.5
EPS tube	0.152	9.5
Carbon fiber legs	0.033	2.1
Battery	0.294	18.4
Avionics	0.338	21.1
Total mass	1.6	100

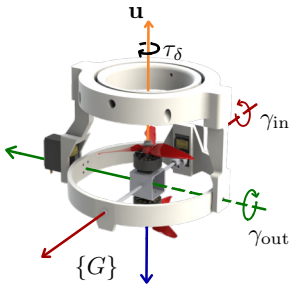


Fig. 3. Gimbal mechanism representation.

avionics bay, is mounted on top of this disk and secured in place with bolts and nuts. Ventilation ducts are present such that airflow to the motors is permitted.

Serving as the *E-Rocket*'s fuselage, the last structural component is a 0.5 m long Expanded Polyestirene (EPS) tube. The battery that powers all electronic components is placed on the top part of the EPS tube. Positioning the battery at this location moves the center of mass towards the top of the vehicle, increasing the TVC lever arm and, consequently, control authority.

2.2 Avionics architecture

The avionics architecture follows a five-layer organization integrating sensing, computation, actuation, communication, and supervision:

Hardware layer - Sensors (IMU, barometer, GNSS), actuators (servomotors and electric motors), and computing units interface with the physical system.

Flight-control layer - A Pixhawk flight computer runs the PX4 Autopilot to perform real-time tasks such as sensor fusion, attitude estimation, actuator control, and safety monitoring.

Application layer - Running on a Raspberry Pi, ROS 2 nodes execute GNC algorithms, mission logic, and PX4 flight-mode management.

Middleware bridge - Two UART links interconnect the Pixhawk and Raspberry Pi, one enabling deterministic data transfer between PX4 and ROS 2, and the other accommodating for MAVLink telemetry.

Ground Control Station (GCS) - The ground segment includes a Linux-based workstation running PX4 Ground Control Station (QGroundControl) and ROS 2 visualization tools. It interfaces via Wi-Fi for telemetry and mission supervision, supporting real-time monitoring, data logging, remote control through a terminal user interface (TUI), and simultaneous physics simulations.

This layered design ensures deterministic control through PX4 while retaining high-level modularity through ROS 2. The hardware configuration implements the layered architecture through COTS components, as summarized in Table 2. The electronics were selected for their proven reliability, availability, and software support. The *E-Rocket* employs a dual-computer setup, partitioned as depicted in Fig. 4, tailored to the early-stage testing of control algorithms for tracking of offline trajectories. That said, flexibility is considered a key requirement and other configurations are trivially implemented within the network.

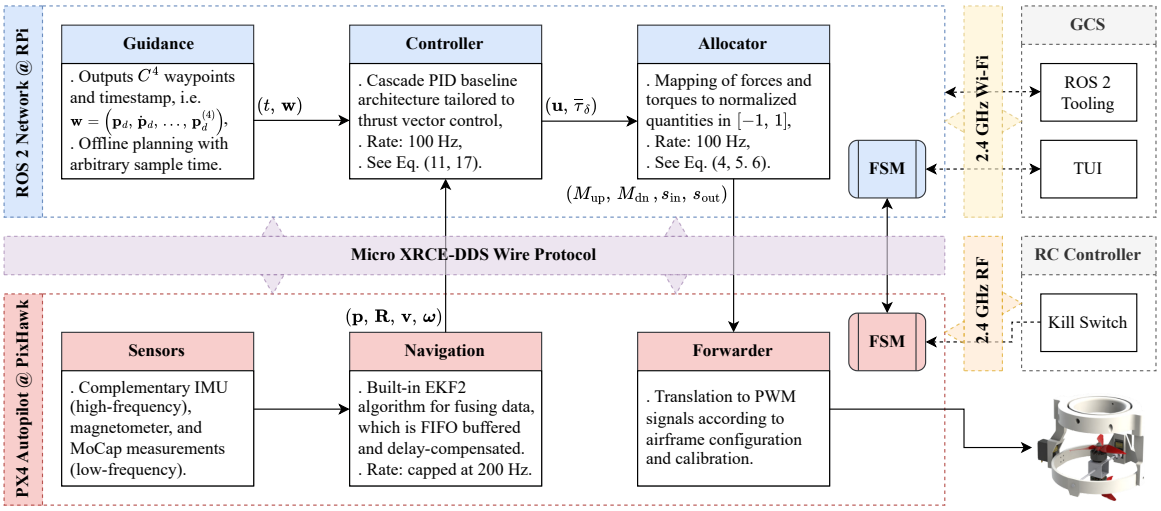


Fig. 4. Proposed software partition for control and offline planning experimentation.

In our proposal, a *Pixhawk 6C Mini* flight computer executes the PX4 Autopilot (Meier et al., 2015) for sensor fusion, estimation, and actuator commanding, and is connected to a *Holybro M9N GPS* that provides GNSS and magnetometer data. A *Raspberry Pi 5 Model B* (8 GB RAM) serves as companion computer, running Ubuntu and Robot Operating System 2 (ROS 2) (Macenski et al., 2022) for control, logging, and communication purposes. Further details are provided in Section 3. Each electric motor is driven by an Electronic Speed Controller (ESC) with an integrated 5 V Battery Eliminator Circuit. Paired with 5-inch propellers the motors are able to provide a combined thrust of 3 kgf, sufficient for hovering at approximately 60% throttle. Metal-gear servomotors were selected for their torque capacity and durability. A power module regulates and monitors energy distribution, delivering 5 V at 3 A to avionics and measuring current and voltage for PX4 telemetry. The system is powered by a 6-cell 1800 mAh LiPo battery able to provide approximately three minutes of continuous hover with sufficient reserve for failsafe maneuvers. For safety-critical override, an RC transmitter and receiver pair provide an independent 2.4 GHz radio link connected directly to the Pixhawk S.BUS input, supporting manual intervention and arming/disarming functionality. The architecture also supports integration with a motion-capture system that provides high-accuracy position and attitude measurements to improve state estimation accuracy. This functionality is essential for indoor testing, where GNSS is unavailable.

Table 2. Avionics hardware components.

Component Type	Model
Flight Computer	Holybro Pixhawk 6C Mini
Companion Computer	Raspberry Pi 5 Model B (8GB)
Brushless Motors	T-Motor F60 PRO V LV KV2020
Propellers	Hurricane MCK 51466 V2
Electronic Speed Controllers	T-Motor AT55A
Servomotors	Power HD 30KG
RC Transmitter	FrSky Taranis X9D Plus
RC Receiver	FrSky X8R
GNSS Module	Holybro M9N GPS
Power Module	Holybro PM06 V2
Lipo Battery	Tattu R-Line 1800mAh 6SP 150C

3. SOFTWARE STRATEGY

The PX4 autopilot supports most high-level UAV interactions but lacks native compatibility with thrust-vectored rockets. Taking advantage of the modularity of the PX4 architecture, its internal GNC pipeline was overridden by custom processes on a companion computer. Multiple customization levels are possible; in this work, guidance, control, and allocation were replaced with ROS 2–based components. PX4 thus functions primarily as a hardware abstraction layer, managing sensor acquisition, state estimation, and actuation via standard message interfaces, while the companion computer runs ROS 2 to implement GNC algorithms, mission logic, and communications.

Despite the flexibility of this modular setup, we adopt the software partition shown in Fig. 4 for early-stage control experiments with custom trajectory tracking. Reference trajectories are processed by the control cascade (Section 4) alongside state estimates from the onboard EKF. The control variables are translated to PWM signals through a sequence of allocation combining ROS 2 and PX4 modules, later provided to the onboard actuators.

Communication within ROS 2 relies on the DDS standard (Pardo-Castellote, 2003). A Micro XRCE-DDS agent on the companion computer bridges PX4 and ROS 2 by exposing selected PX4 publish–subscribe interfaces as ROS 2 topics and services, while also ensuring time synchronization and handling race conditions transparently.

The companion computer runs a customized ROS 2 network composed of three main nodes: (i) a Flight-Mode node that manages PX4 state transitions via validated service calls and maintains synchronization between systems; (ii) a Mission node that publishes reference trajectories and triggers mode changes according to mission phases; and (iii) a Controller node that generates actuation commands from state estimates and tracking references. For early testing, trajectories are planned offline, though the architecture can support real-time guidance methods.

Mission profiles undergo validation prior to execution by the Flight-Mode node, which governs transitions through a finite-state machine (FSM). Each state represents an

Table 3. Propositional symbols and meanings.

Sym	Type	Description
p_1	PX4 sync	Offboard request acknowledged by PX4
p_2	PX4 sync	Arm request acknowledged by PX4
p_3	PX4 sync	Disarm request acknowledged by PX4
c_0	Command	Abort command validated
c_1	Command	Takeoff command validated
c_2	Command	Trajectory tracking command validated
c_3	Command	Landing command validated
t_1	Timeout	Timeout landing condition reached

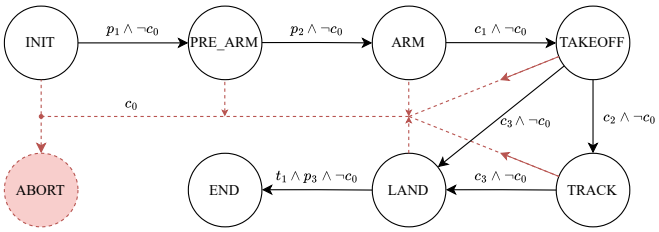


Fig. 5. State machine governing flight testing phases.

abstract action, with transitions triggered by validated events or internal logic. Hierarchical fail-safes can override transitions, including heartbeat monitoring and a direct radio-controlled kill switch on PX4. The state workflow (Fig. 5, Table 3) operates alongside PX4's internal modes, with synchronization handled via discrete events. During flight, PX4 remains in offboard mode to bypass its native controllers. Every experimental workflow begins with a nominal deployment sequence. After the PX4 firmware initializes in manual mode, its integration within the ROS 2 environment and external communication links must be verified. Once both computing units are successfully armed, a takeoff command can be issued. The prototype is designed to perform a smooth ascent to a user-defined altitude, initiating the tracking mission once commanded to. The system is to execute a landing sequence upon mission completion, after which the state machine transitions to its ending state and requests PX4 disarming.

4. MODELING AND CONTROL

In order to design flight control laws for this platform, it is essential to derive a physical model that captures its dynamics and kinematics. Figures 1 and 3 depict the three reference frames in use: an inertial frame $\{I\}$, a body frame $\{B\}$ attached to the center of mass of the vehicle, and a gimbal frame $\{G\}$ attached to the gimbal mechanism. The system has four control inputs: the inner and outer gimbal angles, $\{\gamma_{\text{in}}, \gamma_{\text{out}}\} \in \mathbb{R}$, respectively, the combined thrust magnitude, $T \in \mathbb{R}^+$, and the magnitude of the torque caused by differential thrusting, $\tau_\delta \in \mathbb{R}$.

The gimbal angles permit expressing the thrust vector $\mathbf{u} \coloneqq [\,u_1 \;\; u_2 \;\; u_3\,]^\mathsf{T}$ in $\{B\}$ by applying two consecutive rotations, $\mathbf{R}_x(\gamma_{\text{in}}) \to \mathbf{R}_y(\gamma_{\text{out}}) \in \mathrm{SO}(3)$ to its representation in $\{G\}$:

$$\begin{aligned}\mathbf{u} &\coloneqq -T\,\mathbf{R}_y(\gamma_{\text{out}})\,\mathbf{R}_x(\gamma_{\text{in}})\,\mathbf{e}_3 \\ &= -T\,[\,\sin\gamma_{\text{out}}\cos\gamma_{\text{in}} \;\; -\sin\gamma_{\text{in}} \;\; \cos\gamma_{\text{out}}\cos\gamma_{\text{in}}\,]^\mathsf{T} \quad (1)\end{aligned}$$

where $\mathbf{e}_i$ with $i \in \{1, 2, 3\}$ denotes the standard orthonormal basis of $\mathbb{R}^3$. The torque input, $\boldsymbol{\tau} \in \mathbb{R}^3$, results from the thrust vectoring and differential thrusting components and is defined as

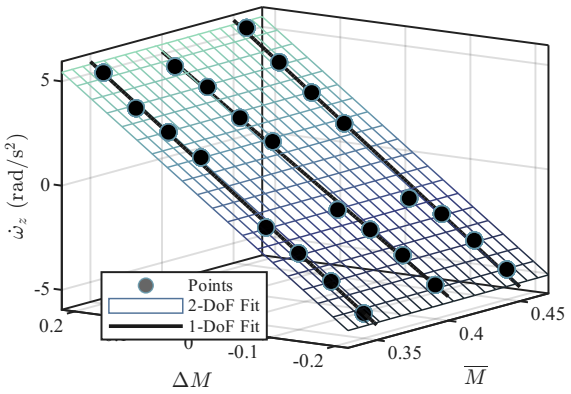


Fig. 6. Linear identification of longitudinal dynamics.

$$\boldsymbol{\tau} \coloneqq L\,\mathbf{S}(\mathbf{e}_3)\,\mathbf{u} + \tau_\delta\,\frac{\mathbf{u}}{\|\mathbf{u}\|}\,, \quad (2)$$

where L is the distance between the gimbal point and the center of mass of the vehicle and $\mathbf{S}(\cdot) : \mathbb{R}^3 \mapsto \mathbb{R}^{3\times3}$ is the skew operator, used for cross products.

Let $\mathbf{p} \coloneqq [\,x \;\; y \;\; z\,]^\mathsf{T} \in \mathbb{R}^3$ and $\mathbf{v} \coloneqq [\,u \;\; v \;\; w\,]^\mathsf{T} \in \mathbb{R}^3$ be the inertial position and velocity of the vehicle, respectively, $m \in \mathbb{R}^+$ its mass, $\mathbf{J} \in \mathbb{R}^{3\times3}$ its inertia matrix, $g \in \mathbb{R}^+$ the gravitational acceleration, $\mathbf{R} \in \mathrm{SO}(3)$ the rotation matrix from $\{B\}$ to $\{I\}$, and $\boldsymbol{\omega} \in \mathbb{R}^3$ the angular velocity of $\{B\}$ with respect to $\{I\}$ expressed in $\{B\}$. Then, using Newton-Euler equations for rigid body motion, the vehicle dynamics and kinematics can be expressed by the system

$$\dot{\mathbf{p}} = \mathbf{v} \quad (3a)$$

$$\dot{\mathbf{v}} = g\,\mathbf{e}_3 + \frac{u_3}{m}\,\mathbf{R}\,\mathbf{e}_3 - \frac{1}{m\,L}\,\mathbf{R}\,\mathbf{S}(\mathbf{e}_3)\left(\boldsymbol{\tau} - \tau_\delta\,\frac{\mathbf{u}}{\|\mathbf{u}\|}\right) \quad (3b)$$

$$\dot{\mathbf{R}} = \mathbf{R}\,\mathbf{S}(\boldsymbol{\omega}) \quad (3c)$$

$$\mathbf{J}\,\dot{\boldsymbol{\omega}} = -\mathbf{S}(\boldsymbol{\omega})\,\mathbf{J}\,\boldsymbol{\omega} + \boldsymbol{\tau} \quad (3d)$$

when neglecting aerodynamic effects and actuator dynamics. The body is idealized axially symmetric, meaning diagonal tensor $\mathbf{J}$ with equal radial components, i.e. $j_\perp \coloneqq j_{xx} = j_{yy}$. Besides the underactuated nature, it is important to note the force-torque coupling appearing in the position dynamics in (3b), which can lead to undesired non-minimum phase behaviour (Hauser et al., 1992).

The system inputs must be mapped to the PX4 signals that command the actuators, requiring an identification procedure. The gimbal angles have a direct 1:1 mapping to the servo angles, which are limited to $\pm30°$, leading to

$$\gamma_\ell = (\pi/6)\,s_\ell \;\; [\text{rad}]\,, \quad \ell \in \{\text{in, out}\}\,, \quad (4)$$

where $s_\ell \in [-1, 1]$ are the PX4 servo command signals. As for the combined thrust magnitude, data from a static test lead to the linear approximation

$$T = 26.45\,\overline{M} - 0.3821 \;\; [\text{N}]\,, \quad (5)$$

with $\overline{M} \coloneqq 0.5\,(M_{\text{up}} + M_{\text{dn}})$, where $M_{\text{up}}, M_{\text{dn}} \in [0, 1]$ are the motor command signals. Finally, the torque produced by differential thrusting was identified by measuring the angular acceleration around the z-axis for different values of the differential, $\Delta M \coloneqq M_{\text{up}} - M_{\text{dn}}$, and average, $\overline{M}$, commands. Experimental results shown in Fig. 6 lead to the approximation on $\overline{\tau}_\delta \coloneqq \tau_\delta / j_{zz}$, written

$$\overline{\tau}_\delta = 23.99\,\Delta M + 0.4031\,\overline{M} + 0.02432 \;\; [\text{rad/s}^2]\,, \quad (6)$$

where j_{zz} is the inertia value around the z-axis. For this purpose, the obtained data was constrained to the sub-hover domain for safety. That said, both Eq. (6)—the coefficients of $\overline{M}$ and ΔM are orders of magnitude apart—and related literature results (Linsen et al., 2022) depict the invariance of the generated torque with the common propulsion mode. As such, it's valid to extrapolate these results to wider domains of thrust.

To validate the proposed testbed, a preliminary control strategy for trajectory tracking was employed. Leveraging the principles of hierarchical and thrust direction control, a typical inner-outer solution for quadrotor UAVs is used, serving as benchmark for future, more complex algorithms. Given the small expected magnitude of the gimbal angles, two assumptions are made that allow us to consider the vehicle as fully actuated in attitude and having independent force and torque inputs: i) the force-toque coupling appearing in the position dynamics is neglected, yielding

$$\dot{\mathbf{v}} = g\,\mathbf{e}_3 + \frac{u_3}{m}\,\mathbf{R}\,\mathbf{e}_3\,, \tag{7}$$

and ii) the torque input from differential thrusting is considered to act on the longitudinal axis only, i.e,

$$\boldsymbol{\tau} = L\,\mathbf{S}\left(\mathbf{e}_3\right)\,\mathbf{u} + \tau_\delta\,\mathbf{e}_3\,. \tag{8}$$

Consider now an inertial reference trajectory defined by $\mathbf{p}_d\left(t\right) \in \mathbb{R}^3$ of class C^4. We proceed by defining the tracking errors

$$\mathbf{e}_p \coloneqq \mathbf{p}_d - \mathbf{p}\,, \quad \mathbf{e}_v \coloneqq \dot{\mathbf{p}}_d - \mathbf{v}\,, \tag{9}$$

and a virtual acceleration input

$$\boldsymbol{\xi} \coloneqq \frac{u_3}{m}\,\mathbf{R}\,\mathbf{e}_3\,, \tag{10}$$

allocated through a PID control law with feedforward:

$$\boldsymbol{\xi} \coloneqq \mathbf{K}_p\,\mathbf{e}_p + \mathbf{K}_d\,\mathbf{e}_v + \mathbf{K}_i\int \mathbf{e}_p\,dt - g\,\mathbf{e}_3 + \ddot{\mathbf{p}}_d\,, \tag{11}$$

where $\{\mathbf{K}_p, \mathbf{K}_d, \mathbf{K}_i\} \in \mathbb{R}^{3\times3}$ are positive definite gain matrices. Considering the dynamics in (7), it is a known result from the literature (Hua et al., 2013) that setting $\boldsymbol{\xi}$ as in Eq. (11) renders the origin of the error system (9) globally asymptotically stable (GAS) in the presence of constant disturbances. However, $\boldsymbol{\xi}$ is a virtual input which hides the attitude dependency in the position dynamics. Hence, the third component of the thrust vector is set to

$$u_3 \coloneqq -m\,\left\|\boldsymbol{\xi}\right\|\,, \tag{12}$$

and, using (10) and (12), a desired attitude is defined as

$$\mathbf{R}_d\,\mathbf{e}_3 \coloneqq -\frac{\boldsymbol{\xi}}{\left\|\boldsymbol{\xi}\right\|}\,. \tag{13}$$

According to the ZYX Euler angle convention, with Euler angles $\boldsymbol{\lambda} \coloneqq \left[\,\phi\;\theta\;\psi\,\right]^\mathsf{T}$, we have that $\mathbf{R}_d \coloneqq \mathbf{R}_z(\psi_d)\mathbf{R}_y(\theta_d)\mathbf{R}_x(\phi_d)$, allowing us to write

$$\mathbf{r}_{3d} \coloneqq \mathbf{R}_y(\theta_d)\mathbf{R}_x(\phi_d)\mathbf{e}_3 = -\mathbf{R}^\mathsf{T}(\psi_d)\frac{\boldsymbol{\xi}}{\left\|\boldsymbol{\xi}\right\|}\,. \tag{14}$$

From Eq. (14), the desired the desired roll and pitch angles can be retrieved:

$$\phi_d = -\arcsin(\mathbf{e}_2^\mathsf{T}\mathbf{r}_{3d})\,, \quad \theta_d = \arctan\left(\frac{\mathbf{e}_1^\mathsf{T}\mathbf{r}_{3d}}{\mathbf{e}_3^\mathsf{T}\mathbf{r}_{3d}}\right)\,. \tag{15}$$

Note that the desired yaw angle, ψ_d, can be set independently. A control law is now needed to ensure attitude tracking. As a preliminary approach, a single-loop PID controller is used for each angular degree of freedom. By defining a compact attitude tracking error as

$$\mathbf{e}_{\boldsymbol{\lambda}} \coloneqq \boldsymbol{\lambda}_d - \boldsymbol{\lambda}\,, \tag{16}$$

the normalized torque input is set to

$$\overline{\boldsymbol{\tau}} \coloneqq \mathbf{J}^{-1}\,\boldsymbol{\tau} = \mathbf{K}_{p,\,\lambda}\,\mathbf{e}_{\boldsymbol{\lambda}} - \mathbf{K}_{d,\,\lambda}\,\boldsymbol{\omega} + \mathbf{K}_{i,\,\lambda}\int \mathbf{e}_{\boldsymbol{\lambda}}\,d\tau, \tag{17}$$

where $\{\mathbf{K}_{p,\,\lambda}, \mathbf{K}_{d,\,\lambda}, \mathbf{K}_{i,\,\lambda}\} \in \mathbb{R}^{3\times3}$ are positive definite diagonal gain matrices, and the angular derivatives were approximated as body rates, i.e. $\dot{\boldsymbol{\lambda}} \approx \boldsymbol{\omega}$, which is valid for the vertical orientation the *E-Rocket* maintains throughout the proposed indoor trajectory. Finally, using the already defined control laws, (12) and (17), the thrust vector input can be computed as

$$\mathbf{u} = -\frac{j_\perp}{L}\,\mathbf{S}\left(\mathbf{e}_3\right)\,\overline{\boldsymbol{\tau}} + u_3\,\mathbf{e}_3\,, \tag{18}$$

thus quantifying the four degrees of control of the system, described as

$$T = \left\|\mathbf{u}\right\|\,, \quad \overline{\tau}_\delta = \mathbf{e}_3{}^\mathsf{T}\,\overline{\boldsymbol{\tau}} \tag{19a}$$

$$\gamma_{\text{in}} = \arcsin\left(\frac{u_2}{T}\right), \gamma_{\text{out}} = -\arcsin\left(\frac{u_1}{\sqrt{T^2 - {u_2}^2}}\right) \tag{19b}$$

which are due the remapping to normalized quantities depicted in Fig. 4.

5. FLIGHT EXPERIMENTS AND RESULTS

For validation, the architecture was deployed onboard the vehicle and provided with waypoint references derived from a fixed-step discretization of a C^4-continuous polynomial trajectory. These were sampled at a constant rate into timestamped vectors $\mathbf{w} \in \mathbb{R}^{3\times5}$, containing position and its first four derivatives. The control laws were implemented in C++ within the Controller node. High-fidelity software-in-the-loop simulations, using Simulink-based nodes emulating the dynamics of (3), validated the system integration. Experimental flights were then conducted indoors [2] in a $7.8 \times 5.0 \times 2.5\,\text{m}$ arena equipped with an OptiTrack motion capture system, providing position and attitude measurements with accuracies of $\pm0.15\,\text{mm}$ and $\pm0.5°$, respectively. The system parameters were: $m = 1.6\,\text{kg}$, $j_\perp = 0.375\,\text{kg.m}^2$, $L = 0.26\,\text{m}$, $\mathbf{K}_p = \text{diag}(3,\,3,\,8.5)$, $\mathbf{K}_d = \text{diag}(2.5,\,2.5,\,3)$, $\mathbf{K}_i = \text{diag}(0.2,\,0.2,\,1)$, $\mathbf{K}_{p,\,\lambda} = \text{diag}(5,\,5,\,5)$, $\mathbf{K}_{d,\,\lambda} = \text{diag}(2,\,2,\,2)$, and $\mathbf{K}_{i,\,\lambda} = \text{diag}(0,\,0,\,0.5)$.

A 3D overview of the flight is provided in Fig. 7, where the inertial position is plotted against the reference. Fig. 7 also illustrates the *E-Rocket*'s position over time while highlighting the flight-mode transitions. The tracking results are qualitatively satisfactory, with only small-scale errors—position-wise, the root mean squared deviations remain centimetric. Despite the oscillations induced by the flight mode transitions, there is a clear tendency of error shrinkage. The attitude evolution confirms the expectedly tall posture and demonstrates the propelling system's ability to contain the yaw angle near its initial value.

Furthermore, the actuation signals evolve as expected and demonstrate the swiftness capabilities of the gimballed mechanism, despite evident room for optimizing the servo

[2] A video of the flight experiment here reported is available at https://www.youtube.com/watch?v=GIlzWH4nCHA.

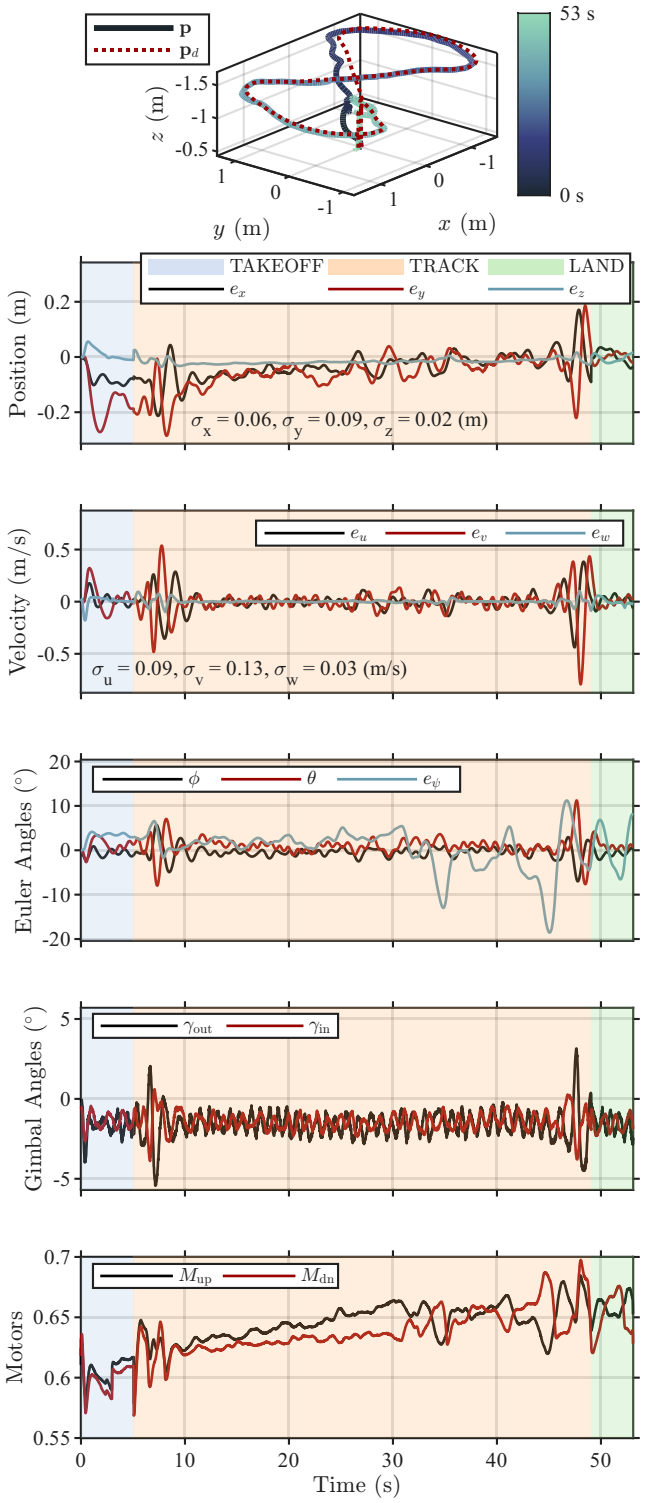


Fig. 7. Time evolution of states and allocation variables.

actuation. The steady-state deviation from the origin of the gimbal angles should result from some rough calibrations and an eventual misalignment of the centre of mass with respect to the longitudinal axis. Clear differences appear in the dynamics of γ_{in} and γ_{out} due to the asymmetric inner-outer structure, yet within the operating domains of the servomotors. The common mode of the propellers responds more slowly, primarily focusing in counterbalancing the weight as the arena trajectory involves minimal vertical motion. The positively-inclined evolution of the thrust command and corresponding PWM signals is a response to the voltage decay of the onboard batteries.

6. CONCLUSION

This paper presented the *E-Rocket*, an electric-powered, low-cost rocket prototype for validation of GNC algorithms based on TVC. The vehicle was designed with 3D printed parts and readily available components which contribute to the cost effectiveness of the platform and yield the prototype easily replicable in academic settings. A custom avionics hardware and software stack was developed considering a dual computer setup which leverages the capabilities of the PX4 autopilot and the modularity of ROS 2. The platform was validated in an indoor arena using a baseline PID-based trajectory tracking controller. Results demonstrated accurate trajectory tracking and confirm the suitability of the *E-Rocket* as a versatile testbed for rocket GNC algorithms. Future work includes outdoor testing, with the addition of an emergency parachute-based recovery system, and the design and validation of more advanced GNC algorithms, such as integrated nonlinear control strategies, onboard adaptive guidance, and custom navigation and state estimation methods.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Institute for Systems and Robotics (ISR - Lisboa) for providing access to their facilities, which were essential to the research.

REFERENCES

Hauser, J., Sastry, S., and Meyer, G. (1992). Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft. *Automatica*, 28(4), 665–679.

Hua, M. et al. (2013). Introduction to feedback control of underactuated VTOL vehicles: A review of basic control design ideas and principles. *IEEE Control Syst.*, 33(1), 61–75.

Jacinto, M. et al. (2024). Pegasus simulator: An Isaac Sim Framework for Multiple Aerial Vehicles Simulation. In *2024 ICUAS*.

Linsen, R. et al. (2022). Optimal Thrust Vector Control of an Electric Small-Scale Rocket Prototype. In *2022 ICRA*.

Macenski, S. et al. (2022). Robot operating system 2: Design, architecture, and uses in the wild. *Sci. Robot.*, 7(66).

Meier, L., Honegger, D., and Pollefeys, M. (2015). PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *2015 ICRA*.

Pardo-Castellote, G. (2003). OMG data-distribution service: architectural overview. In *23rd ICDCSW*.

Ragab, M. and Cheatwood, F.M. (2015). Launch Vehicle Recovery and Reuse. In *AIAA SPACE 2015*.

Santos, P. and Oliveira, P. (2025). Pitch Plane Trajectory Tracking Control for Sounding Rockets via Adaptive Feedback Linearization. In *2025 IEEE AeroConf*.

Spannagl, L. et al. (2021). Design, Optimal Guidance and Control of a Low-cost Re-usable Electric Model Rocket. In *2021 IEEE/RSJ IROS*.