Optimal Thrust Vector Control of an Electric Small-Scale Rocket Prototype

Raphaël Linsen[†], Petr Listov[†], Albéric de Lajarte[†], Roland Schwan and Colin N. Jones

Abstract— **Recent advances in Model Predictive Control (MPC) algorithms and methodologies, combined with the surge of computational power of available embedded platforms, allows the use of real-time optimization-based control of fast mechatronic systems. This paper presents an implementation of an optimal guidance, navigation and control (GNC) system for the motion control of a small-scale electric prototype of a thrust-vectored rocket. The aim of this prototype is to provide an inexpensive platform to explore GNC algorithms for automatic landing of sounding rockets. The guidance and trajectory tracking are formulated as continuous-time optimal control problems and are solved in real-time on embedded hardware using the PolyMPC library. An Extended Kalman Filter (EKF) is designed to estimate external disturbances and actuators offsets. Finally, indoor and outdoor flight experiments are performed to validate the architecture.**

I. Introduction

Thrust Vector Control (TVC) is a key technology enabling rockets to perform complex autonomous missions, such as active stabilization, orbit insertion, or propulsive landing. This is achieved by independently controlling the thrust direction and magnitude of each of its engines. Compared to aerodynamic control such as fins or canard, it guarantees a high control authority even in the absence of an atmosphere, i.e. during high altitude launches or exploration of other planetary bodies.

It is common in the aerospace industry to develop low-cost and low-risk vehicles to experiment on GNC [1]. Particularly, Masten Space Systems, an aerospace company, that has worked on various test vehicles since 2009, mainly to support NASA in developing autonomous landing technologies [2]. The company deployed a real-time version of the G-FOLD landing algorithm [3] developed by the Jet Propulsion Laboratory (JPL) of NASA. The algorithm is used to find the optimal trajectory to the landing site by minimizing fuel consumption. An adapted version of G-FOLD was integrated into the Xombie vehicle for the Autonomous Ascent and Descent PoweredFlight Testbed (ADAPT) program [4]. NASA itself has worked on many test vehicles since its Lunar Landing Research Vehicle (LLRV) developed for the Apollo program. Their Mighty Eagle vehicle was used to test the final phase of landing using classic cascaded PID controllers to stabilize the vehicle orientation and track the landing position. [5]. A more recent SPLICE program (Safe and Precise Landing – Integrated Capabilities Evolution) explored new technologies to improve the landing precision up to tens of meters [6] employing a six-degree-of-freedom (DoF) flying vehicle model for optimal guidance problems [7] [8].

The academic research in this domain is limited due to high manufacturing costs and exploitation risks of such flying vehicles. In order to foster the research and explore different GNC technologies, we developed a small-scale electric prototype that serves as a test platform for deployment and in-flight validation of GNC algorithms. Our vehicle relies on inexpensive materials and standard drone components, making it simple for development and modification at a relatively low cost of around 500 US dollars. Compared to traditional test vehicles integrating a rocket engine for its propulsion, we use electrically powered propellers as the source of thrust, which significantly reduces the cost and logistics associated with the operation of a rocket engine, while still maintaining a high degree of similarity to the dynamics of a rocket. This allows early and diverse flight tests and fast design iterations of the GNC system.

A similar approach was proposed by Spannagle et al. who developed an electric low-cost Vertical Take-Off and Landing (VTOL) vehicle to test a reusable launcher GNC system [9]. Albeit sharing some similarities, the project presented in this paper features a different mechanical design of the flying vehicle and optimal guidance and trajectory stabilization formulations. Contrary to their approach with a separate thrust allocation algorithm and cascaded PID loops for orientation control, we use a nonlinear MPC algorithm to simultaneously track the optimal reference trajectory and stabilize the attitude of the vehicle. Importantly, the software presented in this paper is solely based on open-source components and is publicly available through the following link: https://github.com/EPFLRocketTeam/tvc_drone.

This paper is structured as follows. First, the hardware and software design of the vehicle are presented in sections II and III. Then a complete mathematical model is derived and identified in section IV. This model is then used in section V to formulate optimal guidance and control problems, and in section VI to build an extended state observer for disturbances estimations. Finally, section VII presents indoor and outdoor flights results.

II. Hardware design

The vehicle is 60 [cm] tall, and has a diameter of 26 [cm], totaling a weight of 1.7 [kg]. It has a maximum thrust of about 2.3 [kg]. The center of mass is located at a distance of 21.5 [cm] from the propellers. The batteries are mounted

This work was partially supported by the Swiss National Science Foundation under the NCCR Automation project, grant agreement 51NF40_180545.

[†]The authors contributed equally.

All authors are with Laboratoire d'Automatique, EPFL, Switzerland. {petr.listov, roland.schwan, colin.jones}@epfl.ch

at the top of the vehicle to achieve similarity with rocket dynamics by moving the center of gravity (CoG) further from the propellers. The 31 [Wh] battery allows for a maximum flight time of 130 [s].

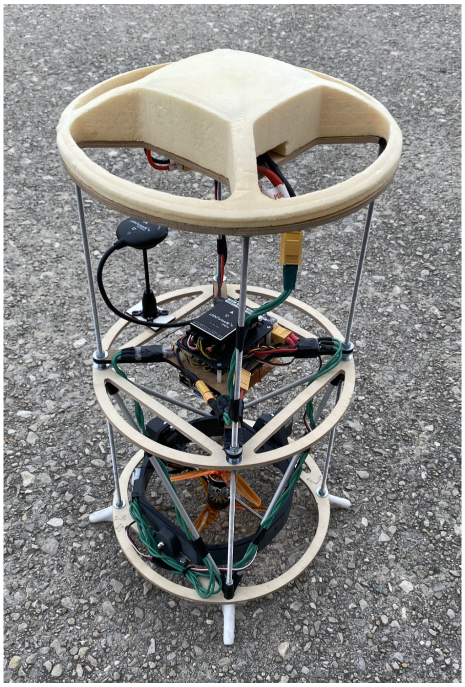

Fig. 1: TVC Prototype.

The thrust is generated by two counter-rotating propellers mounted on a two-axis gimbal, which is controlled by two servomotors, as shown in Figure 2.

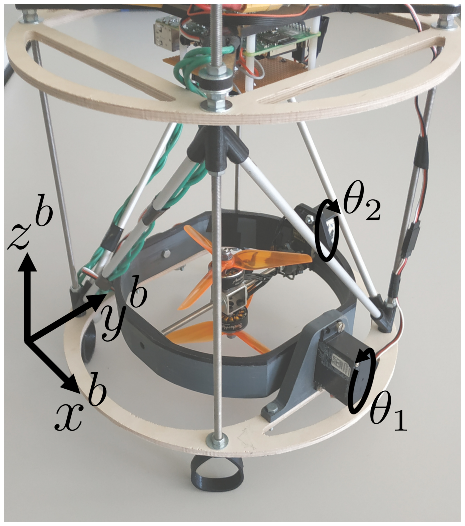

Fig. 2: Two-axis gimbal with mounted propellers.

The structure is built in a modular fashion to allow for fast iterative improvements.

A Pixhawk 4 mini is used for state estimation and servo control. It contains an inertial measurement unit (IMU) with a magnetometer, as well as an external GPS antenna. The state estimates are accessed through the MAVROS protocol. For indoor flights, the motion capture system Optitrack replaces the GPS measurements.

A Raspberry Pi 4 model B, equipped with a quad-core Cortex-A72 processor and 4 GB of RAM, plays the role of the onboard computer and hosts the guidance, control and disturbance estimation algorithms. Each of the aforementioned algorithms is assigned to a separate core to enhance the performance.

III. Software design

The robot operating system (ROS) [10] running on an embedded Ubuntu Server provides the interprogram communication interface between the software components. The GNC architecture can be seen in Figure 3. The current state estimate is provided by the Kalman filter on Pixhawk. The guidance algorithm computes an optimal trajectory linking the current and the target states. This trajectory is then tracked by the nonlinear model predictive controller (NMPC).

IV. Model

A. Notations

Throughout the paper, the vectors without superscript are assumed to be given in the fixed inertial reference frame (IRF), and the vectors with superscript b are given in the body reference frame (BRF). The position vector is denoted by $p = \begin{bmatrix} x & y & z \end{bmatrix}$, the velocity $v = \begin{bmatrix} v_x & v_y & v_z \end{bmatrix}$, and the orientation is defined by the quaternion $q = \begin{bmatrix} q_x & q_y & q_z & q_w \end{bmatrix}$, while the angular speed in BRF is given by $\omega^b = \begin{bmatrix} \omega_x^b & \omega_y^b & \omega_z^b \end{bmatrix}$. The rotation matrix from BRF to IRF derived from the quaternion q is denoted $R(q)$.

B. Equations of motion

The state vector is denoted $\boldsymbol{x}$, and contains the position vector, the velocity, the orientation q, and the angular speed in the body frame ω^b.

$$\boldsymbol{x} = \begin{bmatrix} p & v & q & \omega^b \end{bmatrix}^T \tag{1}$$

The control vector is denoted $\boldsymbol{u}$. The prototype is controlled through the command servo angles θ_1 and θ_2, as well as the speed of the bottom and top propellers, P_B and P_T respectively (in % for the rest of the paper). It is more convenient, however, to consider the propellers' average command speed as an input $\overline{P} = \frac{P_B + P_T}{2}$ and the command speed difference $P_\triangle = P_T - P_B$:

$$\boldsymbol{u} = \begin{bmatrix} \theta_1 & \theta_2 & \overline{P} & P_\triangle \end{bmatrix}^T \tag{2}$$

The state equations are given by the generic 6 DoF solid body dynamics. Omitting the atmosphere interaction, the forces acting on the vehicle are: gravity mg, thrust F_T^b and total torque M^b in BRF. M^b comprises the torque due to the thrust vector F_T^b and the torque M_P^b caused by the speed difference between the two propellers.

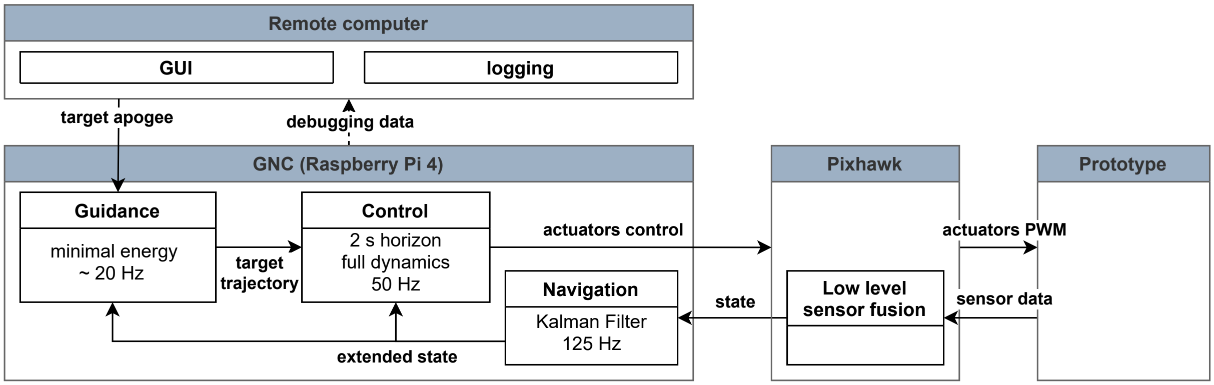

Fig. 3: Software architecture.

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix}\dot{p}\\ \dot{v}\\ \dot{q}\\ \dot{\omega}^b\end{bmatrix} = \begin{bmatrix} v \\ \dfrac{R(q)F_T^b}{m} + g \\ \frac{1}{2} q \circ \omega^b \\ I^{-1}(M^b - \omega^b \times (I\omega^b))\end{bmatrix} \tag{3}$$

$$M^b = r \times F_T^b + M_P^b$$

with I the inertia matrix of the drone, and r the position of the thrust F_T^b from the center of mass.
Both the thrust F_T^b and the torque M_P^b vectors are determined by the gimbal angles:

$$\frac{F_T^b}{\left\|F_T^b\right\|} = \frac{M_P^b}{\left\|M_P^b\right\|} = \begin{bmatrix}\sin\theta_2 \\ -\sin\theta_1\cos\theta_2 \\ \cos\theta_1\cos\theta_2\end{bmatrix} \tag{4}$$

The relations between absolute values of F_T^b and M_P^b and control parameters $\overline{P}, P_\triangle$ are difficult to establish from the physical principles. Therefore, these relations are identified experimentally, as will be shown in the next section:

$$\begin{aligned}\left\|F_T^b\right\| &= f_1(\overline{P}, P_\triangle)\\ \left\|M_P^b\right\| &= f_2(\overline{P}, P_\triangle)\end{aligned} \tag{5}$$

C. Propellers thrust and torque

In order to identify thrust and torque models, the prototype was attached to a load cell while allowing rotation around the z-axis.

The thrust is assumed to be independent of $P_\triangle$, as $P_\triangle$ is constrained to stay small with respect to $\overline{P}$. During the identification experiment for the thrust, stair speed commands were applied to both propellers, and the steady-state thrust was measured for each control value. The results can be seen in Figure 4. A second order polynomial model was fit to the points:

$$\left\|F_T^b\right\| = f_1(\overline{P}) = a\,\overline{P}^2 + b\,\overline{P} + c \tag{6}$$

The torque model was identified through its effect on rotational motion by estimating angular acceleration:

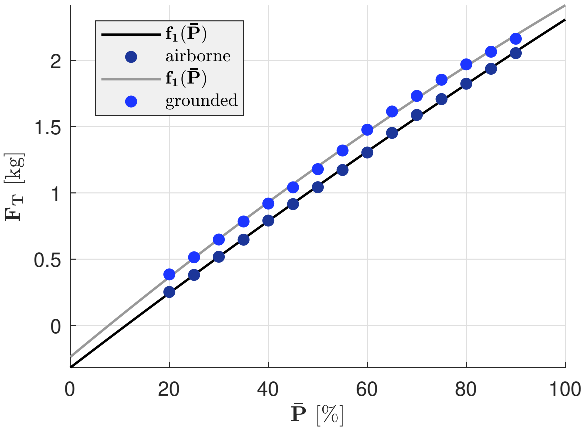

Fig. 4: Experimental identification of the thrust curve.

$$\left\|M_P^b\right\| = I_{zz}\frac{d\omega_z^b}{dt} \tag{7}$$

We carried out an experiment where various piece-wise constant velocity commands were sent to the propellers, in order to cover the possible $[\overline{P}, P_\triangle]$ combinations in given intervals. $P_\triangle$ ranged from -18% to 18% with 4% increments, while $\overline{P}$ ranged from 35% to 80% with 5% increments.

Furthermore, a simple proportional controller was implemented, in order to slow down the rotation between $[\overline{P}, P_\triangle]$ samples. The result can be seen in Figure 5.

It can be observed from the data that the torque is independent of the average propeller speed $\overline{P}$. We conclude that a linear approximation is sufficient:

$$\left\|M_P^b\right\| = f_2(P_\triangle) = c\;I_{zz}P_\triangle \tag{8}$$

V. Guidance and Control

In this section, we describe the design of the optimal guidance and tracking algorithms. A free terminal time optimal control problem minimizing the energy consumption is solved to find a trajectory linking the vehicle with a target position. This trajectory is consequently tracked by an NMPC algorithm in real-time. The details of the formulations

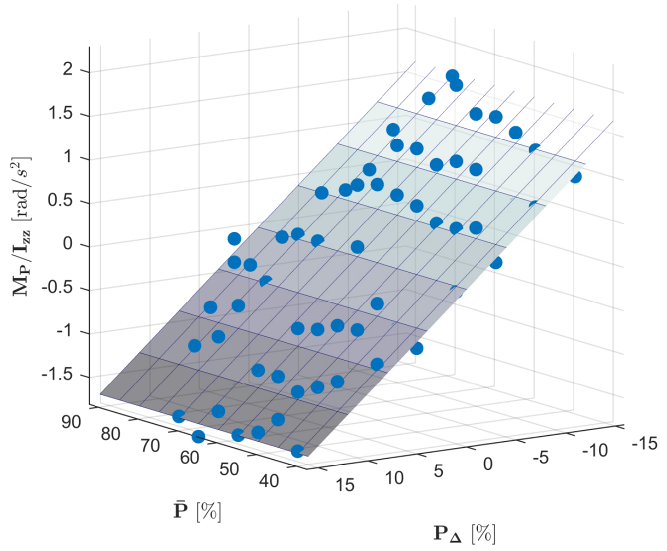

Fig. 5: Experimental torque curve identification.

and numerical implementation are discussed in the following subsections.

A. Guidance

Similar to [3] and [11], the guidance algorithm is based on a point mass model and uses a minimal energy formulation where terminal time t_f is an optimization variable. The final position $p(t_f)$ is constrained to the small neighborhood of the target position p_t with zero-velocity.

$$\begin{aligned} &\min_{F_T(t),\phi(t),\psi(t),t_f} \int_{t_0}^{t_f} F_T^2(t)dt + \rho\eta \\ &s.t. \quad \dot{p}(t) = v(t), \ \dot{v}(t) = \frac{F_T(t)}{m} \begin{bmatrix} \sin\phi\sin\psi \\ -\cos\phi\sin\psi \\ \cos\psi \end{bmatrix} + g \\ &\qquad v_{min} \leq v(t) \leq v_{max} \\ &\qquad F_{min} \leq F_T(t) \leq F_{max} \\ &\qquad -\psi_{max} \leq \psi(t) \leq \psi_{max} \\ &\qquad p(t_0) = p_0 \\ &\qquad v(t_0) = v_0 \\ &\qquad z(t_f) = z_t \\ &\qquad (x(t_f) - x_t)^2 + (y(t_f) - y_t)^2 \leq \eta \\ &\qquad v(t_f) = 0 \end{aligned} \tag{9}$$

The propulsion vector of the rocket is defined in spherical coordinates, where F_T is an absolute value of thrust, azimuth angle ϕ and polar angle ψ. Symmetric constraints on the polar angle define the aperture of the reachable cone. Since the attitude of the vehicle is not explicitly considered in the guidance problem, the polar angle is usually related to the tilt angle of the rocket, thus should not be too large.

The slack variable η weighted by ρ is used to formulate a slack constraint on the target horizontal position, which may be violated when close to the target position.

Since the guidance OCP has free terminal time, the horizon is scaled to the interval [0, 1], $\tau \equiv \frac{t-t_0}{t_f-t_0}$, the dynamics become $\dot{x} = (t_f - t_0)f(x, u)$, and the horizon length $(t_f - t_0)$ then becomes a variable parameter in the OCP. An initial guess for this parameter is given to accelerate the solving of the OCP using a simple closed form solution.

B. NMPC Tracking Controller

The NMPC controller uses the full state dynamics (3), augmented with the disturbance estimation (see section VI-A). It has a prediction horizon of 2 seconds. Close to the target, the length of the horizon is scaled down to match the time to target provided by the guidance algorithm.

$$\begin{aligned} &\min_{\boldsymbol{u}(t)} \int_{t_0}^{t_f} l(\boldsymbol{x}, \boldsymbol{u}, t)dt + V_f(\boldsymbol{x}_f) \\ &s.t. \quad \dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) \\ &\qquad -\theta_{max} \leq \theta_1 \leq \theta_{max} \\ &\qquad -\theta_{max} \leq \theta_2 \leq \theta_{max} \\ &\qquad -\dot{\theta}_{max} \leq \dot{\theta}_1 \leq \dot{\theta}_{max} \\ &\qquad -\dot{\theta}_{max} \leq \dot{\theta}_2 \leq \dot{\theta}_{max} \\ &\qquad P_{min} \leq \overline{P} + P_\triangle/2 \leq P_{max} \\ &\qquad P_{min} \leq \overline{P} - P_\triangle/2 \leq P_{max} \\ &\qquad P_{\triangle_{min}} \leq P_\triangle \leq P_{\triangle_{max}} \\ &\qquad 0 \leq z \end{aligned} \tag{10}$$

The servo motors are constrained in a range of ±15°. We introduce derivative constraints to limit the maximum rate of inputs given by the controller and smoothen the open-loop trajectories.

The propeller speed models are directly included in the formulation, and along with the constraints on top and bottom propeller speeds P_T and P_B, provide a simple way to deal with the trade-off between roll control (through $P_\triangle$) and altitude control (through $\overline{P}$).

1) Stage Cost: The tracking residual at time t corresponds to the difference between the predicted $x(t)$ and the target guidance trajectory $x_G(t)$. The stage cost combines the squared tracking residual, penalty on the control input and penalty on the deviation from vertical orientation. The components $q_w q_x - q_y q_z$ and $q_w q_y + q_x q_z$ penalize deviations of pitch and yaw angles from zero [12]. The roll angle is not controlled directly, but rather the roll rate ω_z^b, as the final roll angle is not critical for the flight mission.

$$l(\boldsymbol{x}, \boldsymbol{u}, t) = e(\boldsymbol{x} - \boldsymbol{x}_G(t))^T Q \, e(\boldsymbol{x} - \boldsymbol{x}_G(t)) + \boldsymbol{u}^T R \boldsymbol{u} \tag{11}$$

$$e(\boldsymbol{x}) \equiv \begin{bmatrix} p & v & q_w q_x - q_y q_z & q_w q_y + q_x q_z & \omega^b \end{bmatrix}^T \tag{12}$$

2) Terminal Cost: In order to improve stability, a continuous-time linear quadratic regulator (LQR) based on a linearization around the zero-speed steady-state (13) is used as a terminal controller. The matrices Q and R for the LQR design are identical to the ones used in the stage cost.

$$A = \left.\frac{\partial f(x,u)}{\partial x}\right|_{x_s,u_s} \quad B = \left.\frac{\partial f(x,u)}{\partial u}\right|_{x_s,u_s} \tag{13}$$

Note that the states q_w and q_z are fixed in the linearization, as they are not controlled. The matrix of the terminal quadratic cost Q_f is then obtained by solving the continuous time algebraic Riccati equation (CARE).

The final cost is then:

$$V_f(\boldsymbol{x}_f) = e(\boldsymbol{x}_f)^T Q_f \; e(\boldsymbol{x}_f) \tag{14}$$

Compared to previously proposed methods, this formulation allows for simultaneous tracking of the optimal trajectory and vertical stabilization, and therefore improves the agility of the vehicle.

C. Solver

The continuous-time guidance problem (9) and the tracking problem (10) are solved in real-time using the Chebyshev pseudospectral collocation method implemented in the open-source C++ package *PolyMPC* [13]. For the tracking problem, a Lagrange polynomial of order 6 on the Chebyshev-Gauss-Lobatto (CGL) grid is chosen to parametrize both state and control trajectories. The computation time on the Raspberry Pi 4 does not exceed 18 [ms] which allows the use of the NMPC algorithm for rocket stabilization directly. The guidance trajectory is interpolated by a two-segment spline, where in each segment a Lagrange polynomial of order 7 is collocated on the CGL grid. The guidance trajectory update is performed at least every 50 [ms]. In both cases, the parametrization is chosen based on the extensive simulation studies. The small number of required collocation nodes can be explained by the spectral convergence of the method for (10) and (9) [14].

The continuous solution representation allows for efficient sampling of the optimal control trajectory at any given time t using the Lagrange interpolation:

$$\boldsymbol{x}(t) = \sum_{k=0}^{N} \boldsymbol{x}_k \phi_k(t), \quad u(t) = \sum_{k=0}^{N} \boldsymbol{u}_k \phi_k(t) \tag{15}$$

where $x_k = x(t_k)$ and $u_k = u(t_k)$ are the state and controls evaluated at the so-called collocation nodes t_k, and ϕ_k is a Lagrange polynomial of order k. The solution of (10) can also be resampled if a faster low-level stabilization loop is required.

VI. Navigation

The state estimation is mostly performed on the Pixhawk which directly provides the state $\boldsymbol{x}$ from raw sensor data. To estimate the external disturbances as well as unmodeled dynamics, a continuous-time extended Kalman Filter (EKF) with discrete measurements [15] was implemented on the embedded computer.

A. Parameter estimation and offset-free tracking

The state equations (3) are extended to estimate parameters and disturbances: the external torque and forces in IRF M_{ext} and F_{ext}, the thrust and torque scaling factors α_1 and α_2 respectively. α_1 varies due to the aerodynamic ground effect as shown in Figure 4 and battery voltage decrease.

$$\boldsymbol{x}' = \begin{bmatrix}\boldsymbol{x} & \alpha_1 & \alpha_2 & F_{ext} & M_{ext}\end{bmatrix}^T \tag{16}$$

$$\begin{aligned} f'(\boldsymbol{x}', \boldsymbol{u}) &= \begin{bmatrix} v \\ \dfrac{R(q)\cdot F_T^b}{m} + g + F_{ext} \\ \frac{1}{2} q \circ \omega_b \\ I^{-1}(M - \omega^b \times (I\omega^b)) \end{bmatrix} \\ M &= r \times F_T^b + M_P^b + R(q)^{-1} M_{ext} \\ \left\|F_T^b\right\| &= \alpha_1 f_1(\overline{P}) \\ \left\|M_P^b\right\| &= \alpha_2 f_2(P_\triangle) \end{aligned} \tag{17}$$

The NMPC controller utilizes the equations (17) with the disturbances assumed constant. This approach allows estimating other model parameters, such as center of mass offset due to imprecise weight balancing. After stable flights were achieved in the absence of external disturbances indoors, the parameters were set to the estimated values to improve disturbance rejection.

VII. Results

This section presents indoor and outdoor flights experiments performed with the vehicle. The indoor experiment included tracking of a complex geometric pattern. For the outdoor flight, a simple launching and landing mission is chosen to demonstrate the combination of the optimal guidance and tracking algorithms.

A. Indoor experiment

For the GNC system validation, the controller had to track a complex "MPC"-shaped pattern with a constant speed of 0.35 [m/s]. The flight was performed indoors, where Optitrack was used to obtain the position and orientation data.

Figure 6 shows the resulting flight, along with the comparison with the simulation. In this project, we employ a real-time flight simulator developed in collaboration with EPFL Rocket Team, which can be found on the link: https://github.com/EPFLRocketTeam/real_time_simulator.

B. Disturbance rejection

To test the robustness of the prototype against external perturbations, an experiment was carried out using the WindShape fan array. The prototype starts from the ground and reaches a target located 1 m above its start while under the effect of strong wind coming from positive x, in this case with a speed of 3.1 [m/s].

Figure 7 shows the resulting estimation and x-axis tracking. The parameter estimation algorithm quickly estimates the force and torque due to the wind along the x-axis,

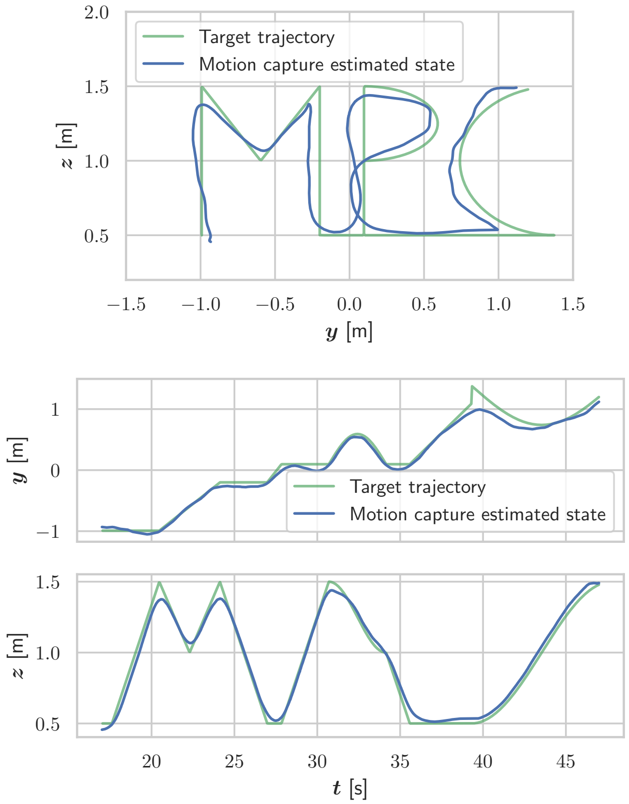

Fig. 6: NMPC Tracking performance in an indoor flight experiment.

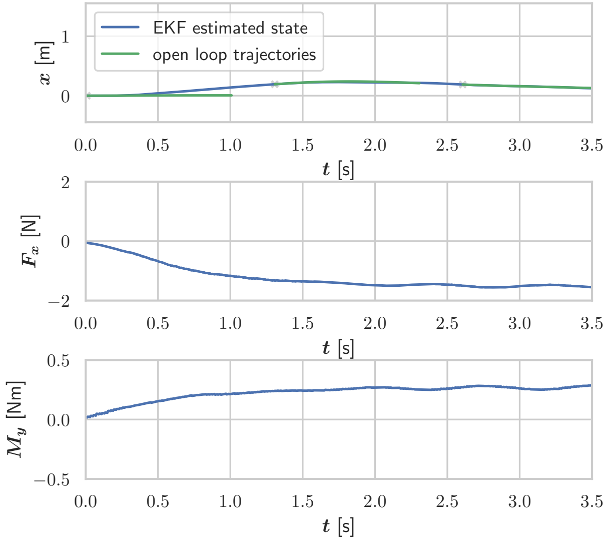

Fig. 7: Demonstration of the disturbance rejection. The constant wind disturbance is produced by the WindShape fan array during the indoor flight experiment.

cancelling the tracking offset in the negative x direction that would occur in the absence of estimation.

C. Outdoor apogee tracking and landing

In order to validate the overall architecture using the guidance and tracking, the outdoor experiment includes reaching a 3-meter apogee, followed by a controlled descent to a given landing point 2 meters away from the starting position.

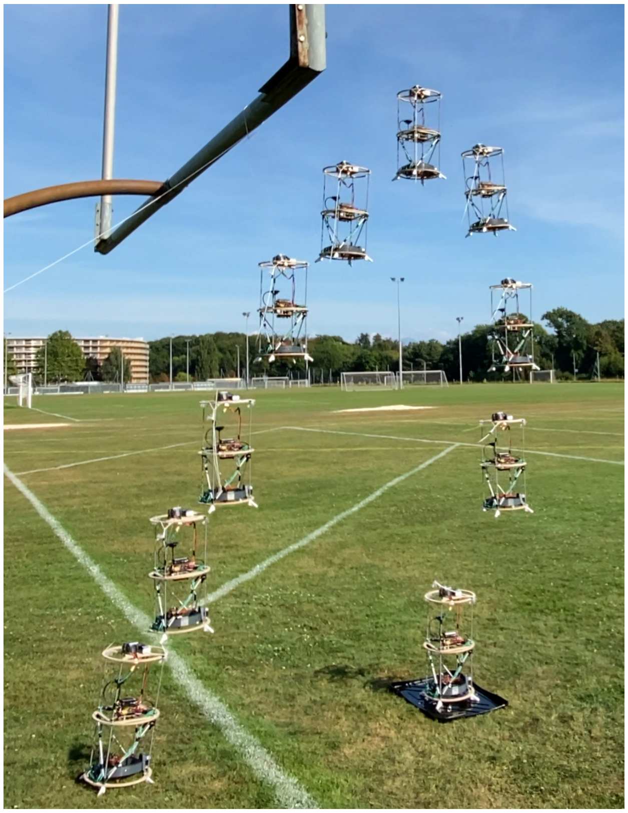

Fig. 8: Apogee tracking and controlled descent. It could be observed that the vehicle initially deviates from the vertical trajectory due to the wind disturbance, then recovers and at the altitude of 3 meters computes the descent trajectory.

The resulting trajectory can be observed in Figure 8.

VIII. Conclusion

The paper presents a small-scale electric VTOL prototype whose goal is to foster academic research in the TVC domain. Along with the hardware design, we provide an experimentally validated and publicly available implementation of the rocket simulator and GNC system. The platform is suitable both for outdoor and indoor flights, capable of apogee tracking and landing as well as more agile maneuvers and robust to external disturbances.

The future research directions will include an adaptation of the proposed GNC framework for active attitude control and precise apogee tracking of a moderate-scale sounding rocket. Further, a 6DoF model is expected to improve the quality of the optimized trajectory, especially near the ground.

References

[1] W. J. Elke, J. Pei, R. J. Caverly, and D. Gebre-Egziabher, “A low-cost and low-risk testbed for control design of launch vehicles and landing systems,” *Optimal Control Applications and Methods*, pp. 1–12, 2021.

[2] R. V. Frampton, J. Ball, D. Masten, P. Ferguson, K. C. Oittinen, and C. Ake, *Planetary Lander Testbed for Technology Demonstration*, 2013. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2013-5337

[3] B. Acikmese and S. R. Ploen, “Convex programming approach to powered descent guidance for mars landing,” *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.

[4] D. P. Scharf, M. W. Regehr, G. M. Vaughan, J. Benito, H. Ansari, M. Aung, A. Johnson, J. Casoliva, S. Mohan, D. Dueri, B. Açikmeşe, D. Masten, and S. Nietfeld, “Adapt demonstrations of onboard large-divert guidance with a vtvl rocket,” in *2014 IEEE Aerospace Conference*, 2014, pp. 1–18.

[5] T. McGee, D. Artis, T. Cole, D. Eng, C. Reed, M. Hannan, D. Chavers, L. Kennedy, J. Moore, and C. Stemple, “Mighty eagle: The development and flight testing of an autonomous robotic lander test bed,” *Johns Hopkins Apl Technical Digest*, vol. 32, pp. 619–635, 12 2013.

[6] J. M. Carson, M. M. Munk, R. R. Sostaric, J. N. Estes, F. Amzajerdian, J. B. Blair, D. K. Rutishauser, C. I. Restrepo, A. M. Dwyer-Cianciolo, G. Chen, and T. Tse, *The SPLICE Project: Continuing NASA Development of GN&C Technologies for Safe and Precise Landing*, 2019. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2019-0660

[7] T. Reynolds, M. Szmuk, D. Malyuta, M. Mesbahi, B. Acikmese, and J. M. Carson, *A State-Triggered Line of Sight Constraint for 6-DoF Powered Descent Guidance Problems*, 2019. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2019-0924

[8] D. Malyuta, T. Reynolds, M. Szmuk, M. Mesbahi, B. Acikmese, and J. M. Carson, *Discretization Performance and Accuracy Analysis for the Rocket Powered Descent Guidance Problem*, 2019. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.2019-0925

[9] L. Spannagl, E. Hampp, A. Carron, J. Sieber, C. A. Pascucci, A. U. Zgraggen, A. Domahidi, and M. N. Zeilinger, “Design, optimal guidance and control of a low-cost re-usable electric model rocket,” *CoRR*, vol. abs/2103.04709, 2021. [Online]. Available: https://arxiv.org/abs/2103.04709

[10] M. Quigley, K. Conley, B. P. 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*, 2009.

[11] B. Açıkmeşe, J. M. Carson, and L. Blackmore, “Lossless convexification of nonconvex control bound and pointing constraints of the soft landing optimal control problem,” *IEEE Transactions on Control Systems Technology*, vol. 21, no. 6, pp. 2104–2113, 2013.

[12] B. Graf, “Quaternions and dynamics,” 2008.

[13] P. Listov and C. Jones, “PolyMPC: An efficient and extensible tool for real-time nonlinear model predictive tracking and path following for fast mechatronic systems,” *Optimal Control Applications and Methods*, vol. 41, no. 2, pp. 709–727, 2020.

[14] L. N. Trefethen, *Approximation Theory and Approximation Practice, Extended Edition*. Philadelphia, PA, USA: SIAM-Society for Industrial and Applied Mathematics, 2019.

[15] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. USA: Wiley-Interscience, 2006.