Automation Platform for Aerodynamic Drone Characterization

Marcus Cemes, Andrea Vicari, Julius Wanner, Dario Floreano

***Abstract*—The design and robust control of morphing Unmanned Aerial Vehicles (UAVs) require high-fidelity aerodynamic models that are difficult to obtain experimentally. The high-dimensional and coupled flight dynamics of these systems make traditional characterisation methods, such as manual wind tunnel testing, prohibitively slow and unable to capture unsteady effects. This work details the development and application of the CARAC platform, a Comprehensive Automated Robotic Aerodynamic Characterisation system, integrating a 6-DOF robotic arm, high-frequency force/torque sensors, and an open-jet wind tunnel. We first validate the platform's precision and then use it to demonstrate the limitations of an analytical model when predicting body forces during dynamic motion. A comparative study on data collection strategies for training neural network models reveals that a time-efficient, free-flight data collection method yields models with superior generalisation across a wide range of flight conditions compared to systematic axis sweeps. Furthermore, we show that predictive performance is currently constrained not by model capacity, but by the richness of the dynamic events within the training data, providing a clear direction for future work in this domain.**

***Index Terms*—aerodynamics, automation, robotics, UAV**

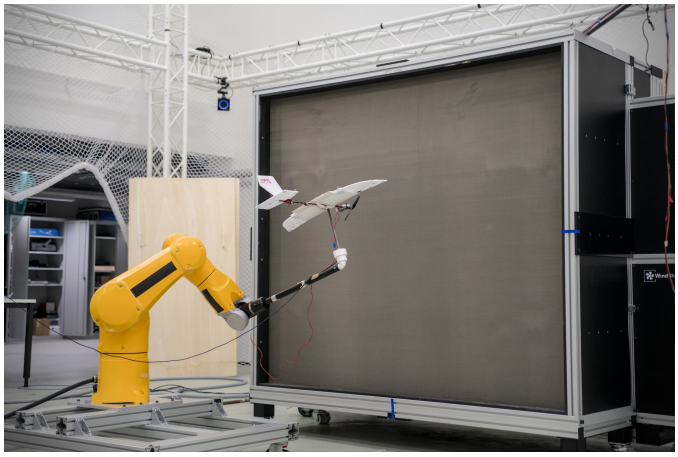

Fig. 1: The CARAC experimental apparatus, showing the Stäubli TX2-90 robotic arm (yellow), WindShape tunnel (black), the morphing drone (white) mounted via the custom adaptor (white cylinder) and a motion capture camera in the background (blue ring).

I. Introduction

HIGH-FIDELITY aerodynamic models are essential for the design and robust control of morphing Unmanned Aerial Vehicles (UAVs), yet their characterisation is complicated by high-dimensional, coupled flight dynamics. Morphing wings offer the potential to dynamically optimise an aircraft's shape for different flight conditions, improving overall performance across the flight envelope [1]. However, to bridge the simulation-to-reality gap for these complex systems, a comprehensive experimental dataset is required that captures both static and, critically, dynamic aerodynamic responses. This work details the development and capabilities of a platform designed to acquire this data with high efficiency and repeatability.

Aerodynamic analysis traditionally relies on a trade-off between computational and experimental methods. While low-order simulation tools are fast, they often fail to accurately predict drag and other viscous effects; conversely, high-order Computational Fluid Dynamics (CFD) is too computationally intensive for exploring the vast parameter space of a morphing vehicle [2]. As noted in a comprehensive review by Li et al., experimental wind tunnel testing is therefore a crucial step for validating models and advancing morphing concepts beyond the proof-of-concept stage [1]. However, traditional manual testing is prohibitively slow [3], and extracting clean coefficients from noisy free-flight data remains a significant challenge, motivating the development of automated experimental platforms.

The use of such automated system is an active area of research. For instance, Baris et al. [4] utilised a Stewart platform for dynamic testing but noted that the motion rates achieved were insufficient to fully resolve the unsteady aerodynamic effects present in rapid manoeuvres. Work by Jeong et al. highlights how comprehensive datasets from static tests can yield high-fidelity models for aerodynamically coupled systems like quadrotors, but this approach does not extend to dynamic characterisation [5]. Furthermore, recent studies in model-based navigation underscore the importance of dynamic data, identifying angular rates as having a significant impact on a vehicle's aerodynamic model [6].

To address these challenges, the CARAC (Comprehensive Automated Robotic Aerodynamic Characterisation) platform was developed to perform aerodynamic load measurements for coupled dynamic and static rotations of a drone at any combination of actuator configurations, with high precision and repeatability (Fig. 1). The platform is an evolution of a foundational automated system by Avoni, which demonstrated a nearly seven-fold increase in data acquisition speed for *static* measurements but did not possess the capability for dynamic testing [7]. By integrating a high-precision 6-DOF robotic arm with a sensitive, lightweight load cell under a custom experiment orchestrator, CARAC can systematically map static characteristics and execute high-rate dynamic motions (up

The authors are with the Laboratory of Intelligent Systems, Ecole Polytechnique Federale de Lausanne (EPFL), CH1015 Lausanne, Switzerland.

to $2\,\mathrm{rad\,s^{-1}}$ in this study) to capture unsteady effects. This provides the rich dataset necessary for creating and validating both analytical and data-driven aerodynamic models. While the project's initial scope included a real-time adaptive loop, the focus was refined to first address the more fundamental and prerequisite question of determining the most effective data collection strategy for training generalisable models.

This paper first details the CARAC platform's hardware and software architecture, followed by the experimental methodology. We then present system validation results, an evaluation of a baseline analytical model, and finally, a comparative study on data collection strategies for training robust neural network models.

II. Method

To enable efficient, high-degree-of-freedom aerodynamic characterisation, the CARAC (Comprehensive Automated Robotic Aerodynamic Characterisation) platform was developed. This section details the platform's implementation, comprising the experimental hardware apparatus, a custom control and data acquisition architecture, and the defined experimental and data processing protocols.

A. Experimental Platform

The physical testbed integrates several key hardware components interconnected via Ethernet to a central control laptop.

1) Robotic Manipulator: The core positioning system is a *Stäubli TX2-90* 6-DOF industrial arm. Its primary attributes for this application are its high joint speeds (up to $760\,^\circ\,\mathrm{s^{-1}}$) enabling dynamic testing, a large workspace, and positional repeatability of $0.02\,\mathrm{mm}$ [8]. This high level of repeatability is a critical enabler for the data compensation methods used in our experimental protocols, allowing for non-sequential calibration runs.

2) Force/Torque Sensor: Force and moment data is acquired using an *ATI NANO25-E* F/T sensor, selected for its high acquisition rate ($7\,\mathrm{kHz}$ over UDP), low latency, and lightweight design which minimises inertial artifacts. The sensitivity of this sensor necessitated several practical solutions to ensure data integrity. Preliminary tests showed that direct airflow caused temperature fluctuations, resulting in a measurement drift of $0.5\,\mathrm{N}$ to $1.0\,\mathrm{N}$ over a minute. This was mitigated by the custom shield described in Sec. II-A3 and by allowing a 30-minute thermal stabilisation period for the sensor after power-on [9]. Further noise reduction was achieved by establishing a common ground between the load cell's dedicated terminal and the large metal chassis of the robotic arm.

3) Drone Prototype and Mounting Assembly: The drone prototype is a custom-built, $120\,\mathrm{g}$ morphing-wing drone with 2-DOF wing sweep, developed for ongoing PhD research into state-transition modelling and agile flight control. Morphing surfaces are commanded via a custom servo controller integrated into the orchestrator over WiFi.

A custom adaptor, 3D-printed with hard PLA, serves as the critical mechanical interface. It orients the drone at $90°$ to the arm to minimise aerodynamic interference and to leverage the arm's sixth joint for fast, simple pitching motions. The drone is attached to the load cell tool interface via a $30\,\mathrm{cm}$ carbon rod, a design choice that further aids in reducing flow disturbances. Integrated into this adaptor is a two-piece environmental shield that fully encloses the load cell, maintaining a $4\,\mathrm{mm}$ air gap to thermally isolate the transducer from airflow while remaining mechanically decoupled (Fig. 10).

4) Airflow Generation: Airflow is generated by a *WindShape* open-jet wind tunnel with 1008 counter-rotating fans and a honeycomb flow straightener.

5) Motion Capture System: An *OptiTrack* motion capture system (up to $240\,\mathrm{Hz}$) is used purely for post-facto validation, confirming the drone's rigid body pose and the position of its morphing surfaces.

B. Software and Control Architecture

The platform is orchestrated by a custom software solution designed for cross-platform compatibility and minimal dependencies.

1) The CARAC Orchestrator: The orchestrator is an asynchronous application built in Rust using the Tokio runtime. This choice provides deterministic performance without a garbage collector and enables efficient, low-level manipulation of network protocols and binary data. Initial evaluations found that the ROS2 framework could not reliably handle the required data throughput (see Appendix E).

The orchestrator implements a central `DataSink` for data collection. Each hardware agent registers one or more *streams* (a collection of synchronous channels) and receives a `StreamWriter` handle. This allows agents to append `u32`-timestamped (relative to experiment start time) `f32` data to memory buffers[1]. Each experiment *run* is saved as an uninterrupted recording into a custom, space-efficient binary file with minimal metadata, preserving the raw data for non-destructive post-processing.

2) Low-Level Control and Communication: A custom VAL3 program running on the Stäubli CS9 controller exposes a rich command interface over UDP. The protocol supports a handshake mechanism and instructions for motion (direct, linear, circular), speed profile updates, and dynamic tool offset changes. This architecture leverages the CS9's robust real-time motion planning while providing full remote control. The controller reports the end-effector pose based on its internal PID setpoint at a frequency of $100\,\mathrm{Hz}$, which provides a less noisy state estimate than the external motion capture system. All custom UDP protocols use a standardised format (magic header, command ID, payload) with big-endian byte order.

C. Data Acquisition and Processing Pipeline

The conversion of raw sensor signals into analysis-ready data follows a multi-stage pipeline designed for high-throughput and repeatability (Fig. 2) The entire process is handled by a custom Python application, enabling batch processing of experimental sessions.

[1] `u32` and `f32` correspond to Rust's type naming scheme for 32-bit unsigned integers and IEEE 754 single-precision floating-point numbers.

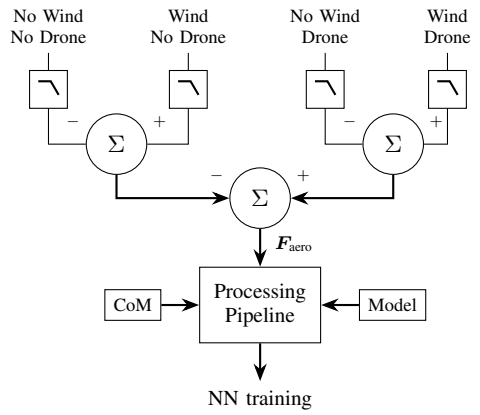

Fig. 2: Experimental workflow with summation blocks showing wind and load effect calculations, integrated with centre of mass data and analytical modelling.

1) **Data Isolation and Calibration:** Aerodynamic data is isolated using a differential calibration method. For each experiment, four runs are performed: one with wind (*wind-on*), another following the identical motion profile without wind (*wind-off*), and then both again without the drone mounted. The high repeatability of the robotic arm allows these runs to be executed non-sequentially within a larger test matrix. In post-processing, both *wind-off* runs, which capture gravitational and inertial forces, are subtracted from their respective *wind-on* runs, to produce net aerodynamic influence and the aerodynamic influence of the mount. The latter is then subtracted from the former to isolate aerodynamic effect purely acting on the drone itself. The load cell is re-biased before each experiment to correct for short-term drift, adding a $\sim 20\,\mathrm{s}$ overhead per run.
2) **Payload Characterisation:** As part of the initial setup, an automated routine determines the payload's mass and centre of mass (CoM). This information is required for transforming measured moments from the Load Frame to the Body Frame during post-processing. The procedure and its validation are detailed in Appendix D.
3) **Data Synchronisation:** Raw time-series data for each run is loaded into memory. The data from all asynchronous streams is then synchronised by linearly interpolating samples onto a common, artificial time base with a user-specified number of time slices (Fig. 3).
4) **Filtering and Augmentation:** A configurable 3rd-order zero-phase Butterworth digital filter, which matches the behaviour of MATLAB's `filtfilt` algorithm [10], is applied to the load cell data to remove sensor noise and structural oscillations. Subsequently, the synchronised data is augmented with angular rates (PQR) and drone velocities (UVW), which are derived from the robot's reported attitude using a Savitzky-Golay filter.
5) **Export:** The final, processed data is exported into Apache Parquet files for efficient storage and subsequent analysis.

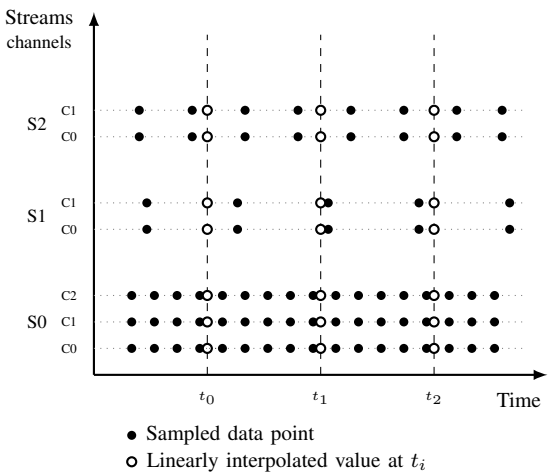

Fig. 3: Data synchronisation process. Raw samples (solid circles) from asynchronous streams (S0, S1, S2) are linearly interpolated to generate synchronised values (open circles) at discrete time slices (t_0, t_1, t_2).

D. Experimental Campaigns

To comprehensively characterise the drone's aerodynamics and evaluate different modelling approaches, three primary experimental campaigns were designed and executed.

- **Systematic Axis Sweeps (AX Dataset):** To establish a performance baseline and systematically map the flight envelope, a test matrix comprising single- and coupled-axis rotations was executed. These sweeps provide structured data ideal for validating the analytical model and understanding its behaviour under predictable conditions.
- **High Angle of Attack (AoA) Sweeps:** To observe the aerodynamic effect of roll and yaw, a specialised test campaign was conducted that would change the drone's profile relative to the wind. The drone was subjected to systematic roll and sideslip sweeps at a fixed pitch angle $\phi = 30°$.
- **Free-Flight Trajectories (FF) & Extended Free Flight (EFF) Datasets):** To test the hypothesis that a more time-efficient data collection strategy could yield robust models, continuous motion experiments were performed. The drone followed complex, time-varying trajectories while its actuators were commanded with random inputs every $5\,\mathrm{s}$. This method was designed to capture a richer, more continuous distribution of dynamic states compared to the discrete nature of axis sweeps (Fig. 5). The FF dataset comprises three unique, shorter trajectories for initial model training. Due to programming complexity arising from robot kinematic singularities, only one long, continuous trajectory was feasible to execute, forming the EFF dataset.

The properties of the Axis (AX), Free Flight (FF) and Extended Free Flight (EFF) datasets collected by the CARAC platform are summarised in Table I.

TABLE I: Collected Dataset Comparison

	AX	FF	EFF
Run Duration [s]	1 to 10	42	333
Time Divisions	1 000	10 000	100 000
Angular Velocities [$\mathrm{rad\,s^{-1}}$]	0.1, 2	0 to 2.1	0 to 2.9
Wind Velocities [$\mathrm{m\,s^{-1}}$]	4.1, 5.8	2.3, 5.2, 6.5	5.8
No. of Trajectories	10	3	1
No. of Actuations	27	9	67
Total Samples	360 000	90 000	300 000
Runtime	∼4 hours	∼20 minutes	∼10 minutes

TABLE II: Neural Network Training Datasets

Model	Training Dataset(s)
M1	Axis
M2	Free-Flight (FF)
M3	Free Flight + Extended Free Flight

TABLE III: Experimentally Determined Centre of Mass

Component	Mass [g]	CoM z offset [mm]
Mount only	67.90 ± 0.07	34.70 ± 0.05
Mount & drone	208.68 ± 0.05	185.21 ± 0.04

E. Modelling and Evaluation Framework

To quantify the performance of different data collection strategies, a comparative framework was established, using the existing analytical model as a baseline and training two types of neural networks on the collected datasets.

1) Analytical Model: An existing analytical physics-based model, which computes aerodynamic forces and moments was used as a performance baseline against which all data-driven models were compared.

2) Data-Driven Models: Two neural network architectures were implemented to compare stateless and stateful learning approaches for capturing unsteady dynamics.

- **Multi-Layer Perceptron (MLP):** A feed-forward network with three hidden layers of size [128, 64, 64] was implemented to model the instantaneous relationship between state and aerodynamic response.
- **Long Short-Term Memory (LSTM):** A recurrent neural network, configured with two layers, a hidden size of 32 and sequence length of 5 ($\sim$30 ms), was implemented to capture potential temporal correlations and unsteady aerodynamic effects.

Both models have approximately 15 000 trainable parameters, a size chosen to be small enough to mitigate the risk of overfitting while being sufficient to investigate the fundamental differences in learning capabilities.

The analytical and trained models take 15 inputs (the attitude quaternion $\boldsymbol{q}$, angular velocity vector $\boldsymbol{v}_{\mathrm{PQR}}$, translation velocity vector $\boldsymbol{v}_{\mathrm{UVW}}$ and drone state/inputs $\boldsymbol{x}_d, \boldsymbol{u}_d$) and predict resultant body force and moment ($\boldsymbol{F}_b, \boldsymbol{M}_b$),

3) Evaluation Strategy: The central goal of the study was to assess the efficacy of different data collection strategies. To achieve this, three experimental models (M1, M2, M3) were defined, each trained on a different dataset or combination thereof, as summarised in Table II. The generalisation capability of each trained model was then evaluated by measuring its performance across all three dataset types (AX, FF, and EFF). The primary metric for comparison is the Root Mean Square Error (RMSE) of the predicted lift component. Lift was chosen as the key indicator due to observed inconsistencies in thrust generation that affected the reliability of the drag calculations.

III. Results & Discussion

The CARAC platform was used to conduct several experimental campaigns to characterise the aerodynamic properties of the morphing drone. This section presents the results from the initial system validation tests, followed by a detailed analysis of the aerodynamic forces measured during systematic axis rotations. Finally, it details a comparative study using this data to develop and evaluate data-driven aerodynamic models based on neural networks.

A. System Validation

Before evaluating aerodynamic models, it is essential to validate the fundamental measurement capabilities of the CARAC platform. This validation comprises two key aspects: the precision of static payload characterisation and the repeatability of dynamic motion trajectories.

A key aspect of this validation is the precise determination of the payload's centre of mass (CoM), which was accomplished using the automated characterisation routine detailed in Appendix D. The results, summarised in Table III, show a high degree of precision. The mean and the standard error of the mean (SEM) indicate low variability across five repeated measurements. This level of precision is fundamental for accurately transforming measured forces and moments from the load cell frame to the drone's CoM in post-processing.

Another critical requirement for the experimental methodology is the ability to repeat motion trajectories with high fidelity, as this allows for the effective isolation of aerodynamic forces through the subtraction of calibration runs. To quantify this capability, the measured forces from two independent, identical free-flight experiments were compared. A cross-correlation analysis of the signals, shown in Fig. 4, reveals a time lag of only 9.98 ms between the two runs. This result confirms the high temporal and spatial repeatability of the robotic system, validating the differential measurement approach used throughout this study.

B. Aerodynamic Characterisation and Analytical Model Evaluation

With the platform's measurement fidelity validated, a series of experiments were conducted to perform a baseline aerodynamic characterisation and to evaluate the performance of the existing analytical model.

Systematic rotational sweeps around the drone's principal axes provide a direct comparison between the analytical model's predictions and the experimental data. During high-rate dynamic pitching tests ($2\,\mathrm{rad\,s^{-1}}$), a clear hysteresis loop

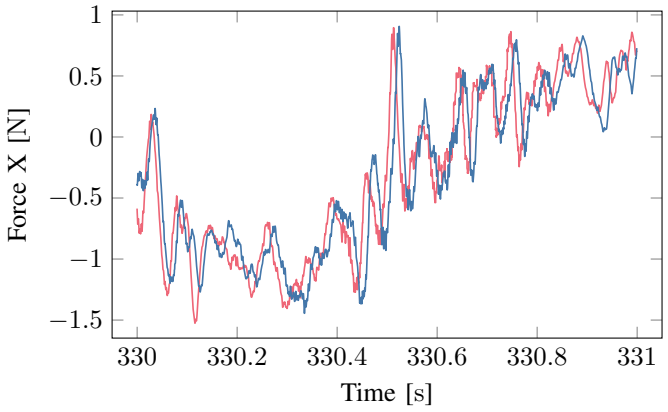

Fig. 4: Force measurements from two independent, extended free-flight experiments. The signals were conditioned with a zero-phase Butterworth low-pass filter ($f_c = 100\,\mathrm{Hz}$). Cross-correlation analysis reveals a time lag of $9.98\,\mathrm{ms}$ between the two measurements.

emerges between the forward (pitch-up) and reverse (pitch-down) motions, particularly in the lift profile (Fig. 6). This unsteady effect, which is not predicted by the analytical framework, results in a prediction discrepancy of $0.85\,\mathrm{N}$ (66 up to about $53\,\%$) at $\alpha = -30°$.

To further probe the model's limitations, its performance was evaluated in off-axis flight by subjecting the drone to roll and sideslip sweeps at a high, fixed angle of attack ($\alpha = 30°$), as shown in Fig. 6. The results reveal prominent quantitative errors. The analytical model incorrectly predicts that lift is dependent on both roll and sideslip angle, with a parabolic shape for roll and a fall-off for sideslip. In contrast, the experimental data shows that lift is largely insensitive to both inputs in this flight regime. This discrepancy results in a peak error of $0.2\,\mathrm{N}$ to $0.3\,\mathrm{N}$ ($18\,\%$ to $27\,\%$). Additionally, an unmodelled dynamic response was observed in the drag force, which decreased at higher roll and sideslip rates, and an even larger offset with respect to the static measurements, highlighting the model's failure to capture unsteady phenomena in this regime.

Collectively, these findings indicate that while the analytical model serves as a reasonable approximation for basic axis rotations, its quantitative accuracy degrades in dynamic or coupled off-axis flight regime. This motivates the use of a data-driven approach to capture these dynamic unsteady effects.

C. Data-Driven Modelling: Collection Strategy and Generalisation

To address the limitations of the analytical model, a comparative study was conducted to determine the most effective data collection strategy for training robust neural network models. The central research question was whether a time-efficient, free-form data collection method could produce models with strong generalisation capabilities.

1) Dataset Input Distributions: A fundamental difference between the collection strategies is the distribution of dynamic states they capture. The Free-Flight (FF and EFF) datasets provide a broad and continuous distribution of angular rates (PQR) and velocities (UVW) Fig. 5. In contrast, the Axis Sweep (AX) dataset captures a much narrower, discrete set of states, concentrated around specific operational points. This suggests that the free-flight data may be better suited for training models that can generalise across a wider range of flight conditions.

2) Quantitative Model Performance: The performance of the analytical model and the six trained neural network models was evaluated across all three datasets (Fig. 7). The results provide a clear quantitative comparison of their generalisation capabilities.

The analytical model, serving as a baseline, performs best on the structured AX dataset (RMSE of 0.229) but its accuracy degrades significantly on the more dynamic Free-Flight (FF) and Extended Free-Flight (EFF) datasets. This confirms that while it captures quasi-static relationships, it fails to represent the more complex, unsteady dynamics present in free-form trajectories.

The models trained exclusively on the Axis Sweep data (M1) demonstrate the limitations of a specialised training approach. While they are able to effectively learn their own dataset (AX), more than halving the error of the analytical model, they fail to generalise very well to the free-flight data. This indicates that while they have learned the specific patterns of the axis sweeps, they have not captured a more general representation of the drone's aerodynamics.

In contrast, the models trained on free-flight data show superior generalisation. The M2 models, trained on the short FF dataset, already show a strong ability to generalise to the much longer, unseen EFF dataset. The most significant result comes from the M3 models, trained on the combined FF+EFF data. The M3-LSTM model achieves the lowest overall error on the EFF test set (RMSE of 0.080). More importantly, these models demonstrate an improved ability to generalise back to the structurally different AX dataset. The M3-MLP model achieves an RMSE of 0.288 on the AX data, a substantial improvement over the M2-MLP's 0.429 and approaching the performance of the analytical model itself.

These results lead to a clear conclusion: the free-flight data collection strategy is not only more time-efficient but also produces models with better generalisation capabilities. The rich, continuous distribution of dynamic states within the free-flight data forces the models to learn a more fundamental representation of the system's aerodynamics, making them more robust and applicable to a wider range of flight conditions than models trained on systematic axis sweeps.

3) Qualitative Model Performance: To visually complement the quantitative metrics, the performance of all models was evaluated on an unseen segment of the Extended Free-Flight (EFF) validation dataset. As shown in Fig. 8, all models, including the analytical one, successfully capture the low-frequency evolution of the lift force, indicating that each has learned the primary quasi-static relationships.

However, critical differences emerge in their dynamic fidelity. The analytical model's prediction is noticeably rounded, failing to match the sharp peaks characteristic of rapid attitude changes. Similarly, the Axis-trained models (M1) consistently show the largest deviation from the ground truth. In contrast,

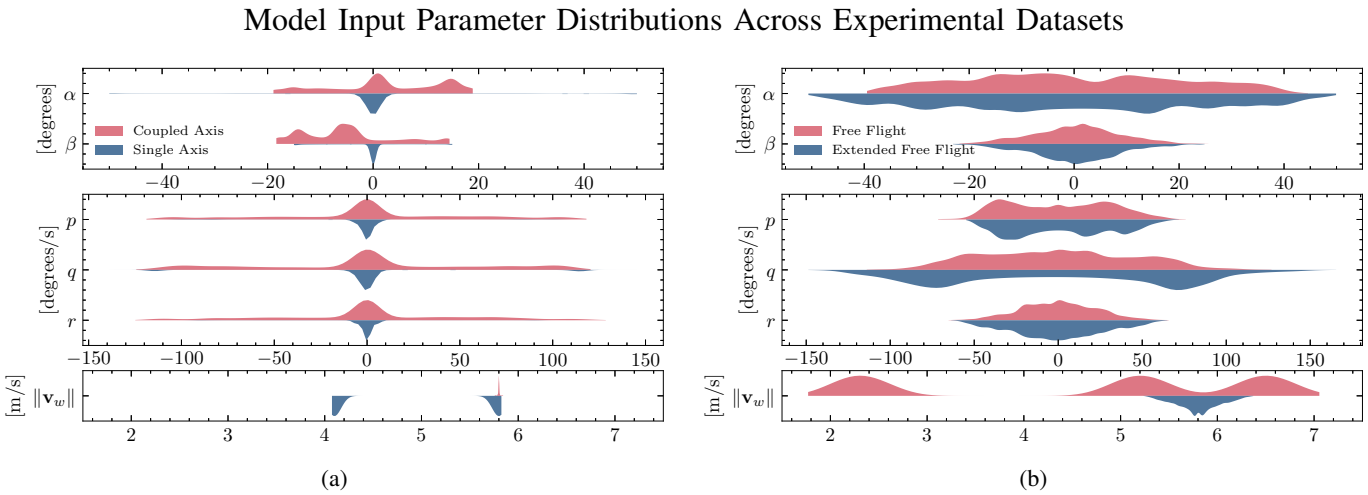

Fig. 5: Model input distributions for various experimental datasets. The inputs are flight angles α and β, angular velocities p, q, r, and the norm the the airspeed vector v (UVW). (a) distributions for the coupled (red) and single (blue) systematic axis sweep (AX) datasets. This panel shows a much narrower distribution, reflecting the highly systematic nature of the axis sweep data collection. The tail ends of the distribution are due to the acceleration at the beginning and end of the experiment. (b) distributions for the Free Flight (red) and Extended Free Flight (blue) datasets. These distributions are notably wider, capturing a greater range of dynamically continuous moving parameters due to the nature of continuous flight.

TABLE IV: MLP Model Complexity Evaluation

Model	Hidden layers	RMSE
A	1	0.577
B	8	0.216
C	2-2	0.652
D	16-8	0.219
E	32-16	0.228
F	128-128-64	0.244
G	256-128-64-16	0.246

the models trained on free-flight data (M2 and M3) track the experimental signal with higher fidelity. The M3-LSTM model, in particular, excels at capturing the high-frequency transients.

D. Impact of Model Complexity on Learning Unsteady Dynamics

Having established the superiority of the free-flight training strategy, a final investigation was conducted to understand the relationship between model capacity and performance. This study aimed to determine whether increasing model complexity would lead to a corresponding increase in performance, or if learning was constrained by the richness of the training data itself.

A range of MLP architectures, from a single neuron to a large, four-layer network, were trained on the combined FF+EFF dataset and evaluated on an unseen free-flight validation trajectory. The results, summarised in Table IV, reveal a non-monotonic and highly insightful relationship between model size and predictive accuracy.

Contrary to the common assumption that larger models yield better performance, the best result (RMSE of 0.216) was achieved by a remarkably simple architecture: a single hidden layer with only eight neurons (Model B). Increasing the model's depth and complexity did not lead to further improvement. Notably, the two-layer, two-neuron model (C) performed exceptionally poorly, indicating that a certain minimum capacity per layer is necessary for effective learning.

This analysis strongly indicates that performance on this task is not limited by model capacity but by the nature of the training data and the training technique. While the free-flight dataset is rich enough for a simple MLP to learn the dominant dynamic relationships, capturing even more subtle, high-frequency unsteady effects would likely require a data collection strategy explicitly designed to excite those modes, such as incorporating rapid step inputs or high-frequency oscillations into the test trajectories.

IV. Conclusion

This work detailed the development, validation, and application of the CARAC platform, an automated system designed to address the significant challenge of characterising the complex aerodynamics of morphing UAVs. By integrating a high-precision robotic arm, a high-frequency force/torque sensor, and a custom high-performance software orchestrator, CARAC enables the efficient acquisition of both static and, critically, high-rate dynamic aerodynamic data.

The platform's high spatial and temporal repeatability was validated, confirming the viability of its differential measurement technique for isolating purely aerodynamic forces. Using this capability, the quantitative limitations of a traditional, analytical model were demonstrated, particularly its inability to capture the hysteresis present in dynamic manoeuvres. This motivated a comparative study on data collection strategies for training robust data-driven models. The central finding of this work is that a time-efficient, free-flight data collection

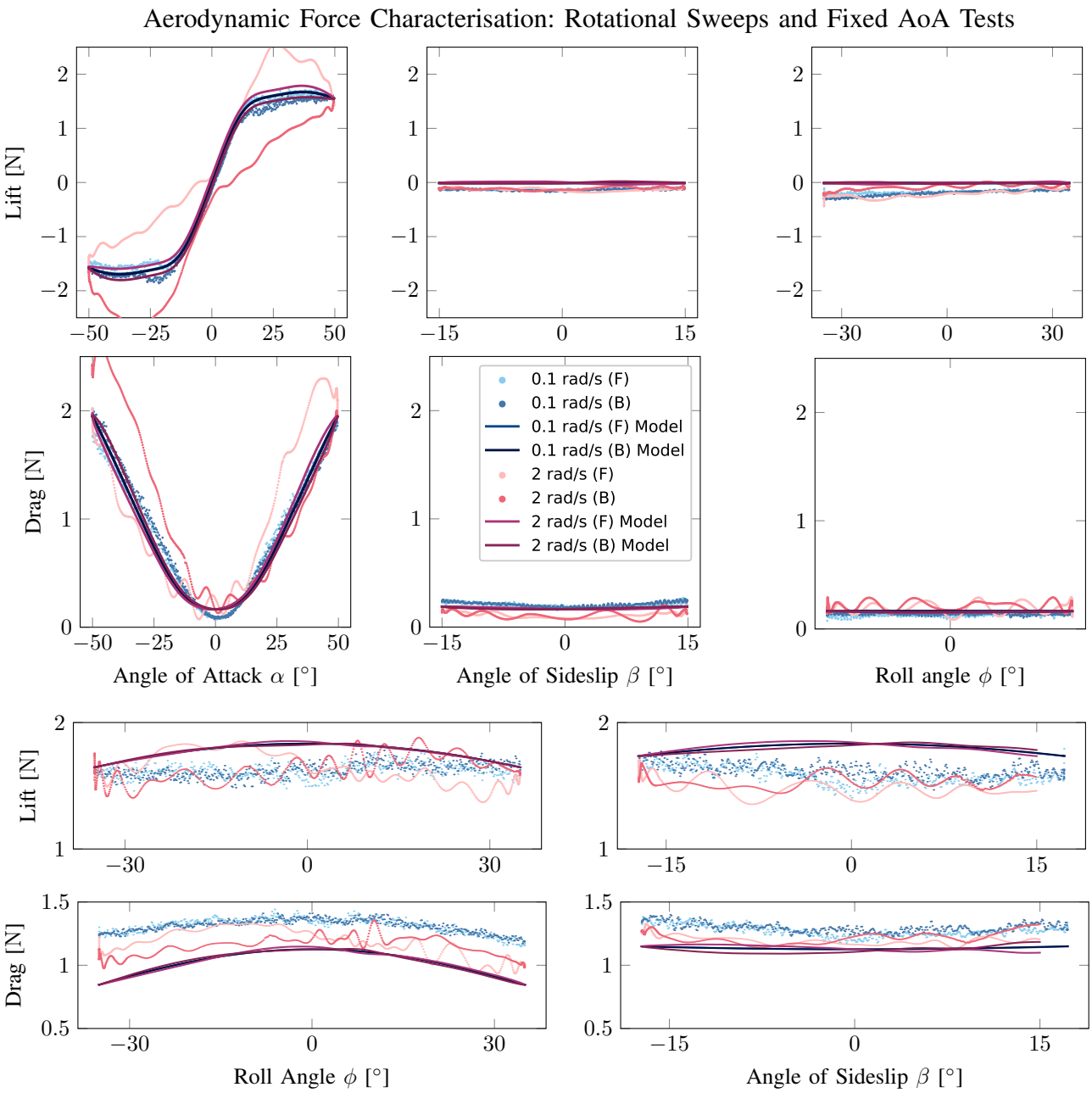

Fig. 6: Aerodynamic lift and drag forces (Wind Frame) from systematic rotational sweep experiments. The upper six plots display results from single-axis pitch (α), sideslip (β), and roll (ϕ) sweeps. The lower four plots show lift and drag as a function of roll and yaw angles at a fixed angle of attack $\alpha = 30°$. All plots are shown against the analytical model. Pitch-up (Forward) and pitch-down (Backward) passes are indicated by lighter and darker shades, respectively. A common legend applies to all subplots.

strategy produces neural network models with superior generalisation capabilities compared to those trained on systematic axis sweeps. Furthermore, an analysis of model complexity revealed that predictive performance is currently constrained not by model capacity, but by the richness of the dynamic events within the training data.

Despite these advancements, the platform has several limitations that inform future work. The differential measurement technique, while powerful for isolating airframe aerodynamics, inherently cancels out the effects such as propellor wash or the pumping of air by the wings. Additionally, motion planning is currently a manual process designed to avoid robot singularities, and system rigidity and induced oscillations remain a fundamental challenge.

Future work will proceed along several key avenues. A primary focus will be a comprehensive investigation into neural network architectures and data handling. Building on the findings regarding data richness, new data collection trajectories will be designed to explicitly excite high-frequency, unsteady aerodynamic modes. A significant next step will be

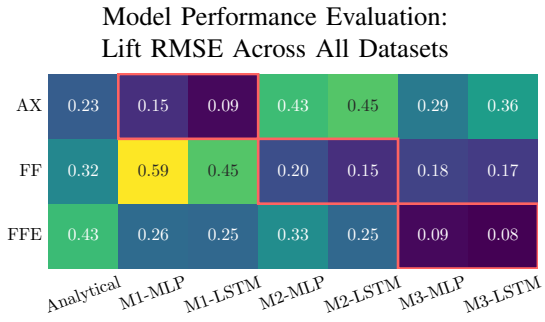

Fig. 7: Root Mean Square Error (RMSE) on predicted lift across all experimental datasets. Each cell represents a model (x-axis) evaluated on a specific dataset (y-axis). Cells where the model is evaluated on its training dataset are highlighted in red. Models M1, M2, and M3 were trained on the AX, FF, and FF+EFF datasets, respectively.

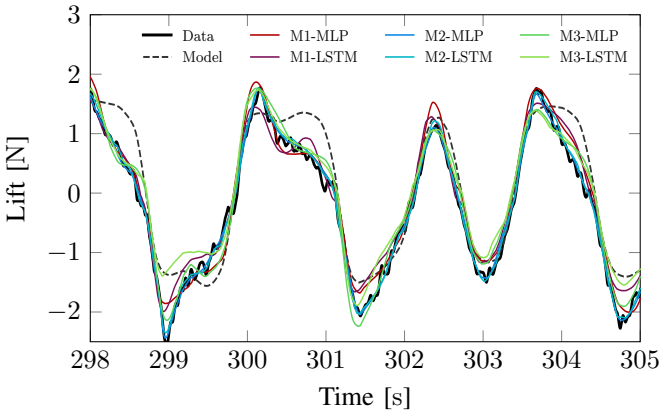

Fig. 8: Qualitative comparison of lift prediction for all models on an unseen segment of the Extended Free-Flight (EFF) validation dataset. While all models capture the general low-frequency trend of the experimental ground truth (black), their ability to track high-frequency dynamics and sharp peaks varies.

to develop a hybrid modelling approach that integrates the airframe aerodynamics measured by CARAC with a separate characterisation of the self-acting effects. Finally, the integration of a kinematic-aware motion planner will be pursued to fully automate the generation of safe and efficient test trajectories, removing the current manual design bottleneck and further enhancing the platform's capabilities. Such a planner would ensure that generated trajectories remain clear of robot singularities, thereby guaranteeing desired payload translation and angular velocities and preventing runtime errors on the CS9 controller. Furthermore, by actively planning paths within the robot's operational envelope, a kinematic-aware motion planner would significantly reduce the risk of collision between the payload and the WindShape or the robot itself, moving beyond current reliance on simple bounds-checking on movement amplitudes.

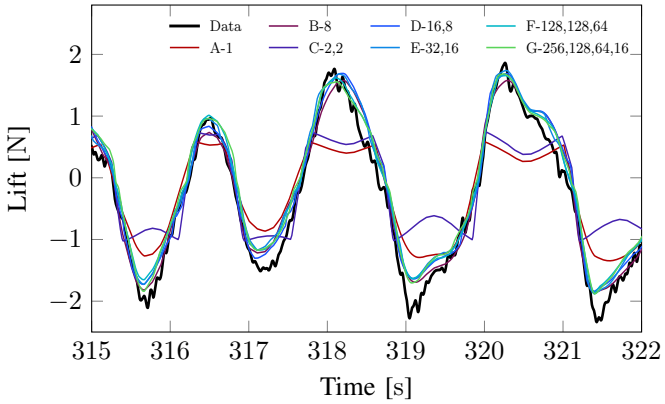

Fig. 9: Qualitative comparison of lift prediction for MLP models of varying complexity on an unseen free-flight validation trajectory. The experimental data (ground truth) is shown in black. Note that even a single neuron captures the basic trend, and performance improvements show diminishing returns as model size increases, suggesting the learning is constrained by the richness of the training data.

References

[1] D. Li, S. Zhao, A. Da Ronch, J. Xiang, J. Drofelnik, Y. Li, L. Zhang, Y. Wu, M. Kintscher, H. P. Monner, A. Rudenko, S. Guo, W. Yin, J. Kirn, S. Storm, and R. D. Breuker, "A review of modelling and analysis of morphing wings," *Progress in Aerospace Sciences*, vol. 100, pp. 46–62, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0376042117301835

[2] O. Dantsker and M. Vahora, "Comparison of aerodynamic characterization methods for design of unmanned aerial vehicles," in *2018 AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, 2018. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2018-0272

[3] J. B. Barlow, W. H. Rae, and A. Pope, *Low-speed wind tunnel testing*, 3rd ed. John Wiley & Sons, 1999.

[4] E. Baris, C. P. Britcher, and G. Altamirano, "Wind tunnel testing of static and dynamic aerodynamic characteristics of a quadcopter," in *AIAA Aviation 2019 Forum*. American Institute of Aeronautics and Astronautics, 2019. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2019-2973

[5] H. Jeong, J. Suk, S. Kim, Y.-G. Lee, T. Cho, and J. Jeong, "Aerodynamic modeling and verification of quadrotor UAV using wind-tunnel test," *International Journal of Aeronautical and Space Sciences*, vol. 25, no. 3, pp. 809–835, 2024. [Online]. Available: https://doi.org/10.1007/s42405-023-00698-x

[6] P. Longobardi and J. Skaloud, "Aerodynamic modeling of a delta-wing UAV for model-based navigation," *CEAS Aeronautical Journal*, vol. 15, no. 2, pp. 283–301, 2024. [Online]. Available: https://link.springer.com/10.1007/s13272-024-00727-9

[7] L. Avoni, "Automated aerodynamic characterization of a morphing VTOL," Master's thesis, EPFL, 2023.

[8] Stäubli. (2025) TX2-90 datasheet. [Online]. Available: https://www.staubli.com/content/dam/robotics/products/robots/tx2/TX2-90-6-axis-product-data-sheet-EN.pdf

[9] ATI. (2020) FAQ - force/torque sensors. [Online]. Available: https://www.ati-ia.com/library/documents/FT_FAQ.pdf

[10] MathWorks. (2025) filtfilt. [Online]. Available: https://www.mathworks.com/help/signal/ref/filtfilt.html

APPENDIX A
COORDINATE FRAMES & CONVENTIONS

This section defines the coordinate frames, transformations, and conventions used for motion control and data analysis. All coordinate frames are right-handed Cartesian systems.

A. Coordinate Frames

- **World Frame (WF):** A fixed inertial frame whose origin is at the base of the robotic arm. The X-axis points forward, the Y-axis points left, and the Z-axis points up.
- **Tool Frame (TF):** A frame rigidly attached to the mounting flange of the robot's end-effector. Its pose (position and orientation) is defined relative to the WF by the robot's forward kinematics.
- **Load Frame (LF):** A frame originating at the geometric center of the ATI F/T sensor where forces and moments are measured. It is rigidly attached to the TF. The LF is defined with an orientation rotated 90° about the TF's z-axis, which aligns the LF's x-axis with the nominal wind direction.
- **Body Frame (BF):** A body-fixed frame with its origin at the payload's Center of Mass (CoM). The BF is oriented with its x-axis pointing toward the drone's nose, its y-axis to the left wing, and its z-axis up. The analytical model's parameters (such as incoming wind direction) are defined with respect to this frame.
- **Wind Frame:** A frame oriented with the incoming airflow, such that its x-axis is aligned with the Body Frame drag direction and its z axis is aligned with the Body Frame lift direction.

B. Aerodynamic Angles

The orientation of the Wind Frame relative to the Body Frame is defined by two angles, the Angle of Attack α and the Angle of Sideslip β. They are defined as:

$$\alpha = \arctan\left(\frac{-w}{u}\right) \quad \beta = \arcsin\left(\frac{v}{|\boldsymbol{v}_{\text{UVW}}|}\right) \tag{1}$$

where $\boldsymbol{v}_{\text{UVW}}$ is the wind vector (wind velocity + world translation velocity) oriented into the wind.

C. Transformations

- **LF-to-BF:** To analyze measured forces as if they were acting at the drone's CoM, the moments measured in the LF ($\boldsymbol{M}_{\text{LF}}$) must be corrected for the moment induced by the forces ($\boldsymbol{F}_{\text{LF}}$) acting at a distance ($\boldsymbol{t}_{\text{CoM}}$). The transformed moment in the Body Frame ($\boldsymbol{M}_{\text{BF}}$) is calculated as:

$$\boldsymbol{M}_{\text{BF}} = \boldsymbol{M}_{\text{LF}} - \boldsymbol{t}_{\text{CoM}} \times \boldsymbol{F}_{\text{LF}} \tag{2}$$

  The transformation from the LF to the BF consists of a single translation vector $\boldsymbol{t}_{\text{CoM}}$, determined experimentally by the payload characterisation procedure described in Appendix D.

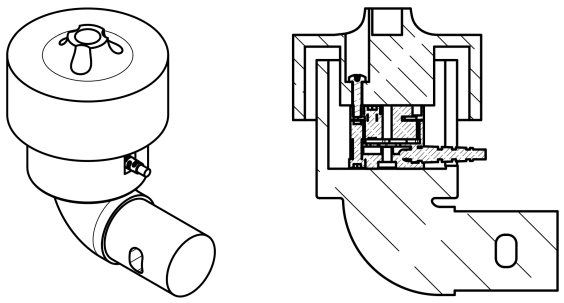

Fig. 10: Custom 3D-printed load cell adaptor and environmental shield. (a) Perspective view of the two-piece enclosure. (b) Section view detailing the 4 mm air gap isolating the load cell body.

D. Orientation Convention

- **Tait-Bryan Angles:** The orientation of the Body Frame (BF) relative to the World Frame (WF) is described using a set of *Tait-Bryan* angles (ϕ, θ, ψ). The convention used is a sequence of *extrinsic* rotations applied in the order XYZ.

E. Payload Definition

- **Payload:** The *payload* refers to all components whose weight is measured by the F/T sensor. This includes the drone itself, the carbon mounting rod, and the 3D-printed adapter that attaches to the sensor and the load cell transducer head.

APPENDIX B
DATA PROCESSING AND STATE ESTIMATION

This section details the numerical methods used in the post-processing pipeline to convert raw, synchronised data into the final analysis-ready data frames.

A. Filtering

To remove high-frequency structural oscillations and sensor noise from the force/torque data, a 3rd-order digital Butterworth filter was applied. A zero-phase implementation (equivalent to MATLAB's `filtfilt`) was used to prevent introducing phase lag into the signal, which is critical for analysing dynamic events. The cutoff frequency (f_c) was typically set between 5 Hz and 10 Hz, based on FFT analysis of the raw signal which identified the primary structural mode at approximately 16 Hz.

B. Derivation of Velocity and Angular Rates

The CARAC platform uses the Stäubli CS9 controller's setpoint as the primary source for the drone's pose, as it is less noisy than the external motion capture system. The controller provides the end-effector's position and orientation (as Tait-Bryan angles) at a constant rate of 100 Hz.

To derive the drone's linear velocities in the body frame (UVW) and angular rates (PQR), a Savitzky-Golay filter was applied to the time series of position and orientation data. This

filter performs a local polynomial regression to smooth the data and compute its derivatives, which is more robust to noise than simple finite differencing. For this study, a filter window size of 99 points (corresponding to $\sim 36\,\mathrm{ms}$) and a polynomial order of 3 were used to calculate the first derivatives.

APPENDIX C
NEURAL NETWORK TRAINING DETAILS

- **Loss Function:** The Mean Squared Error (MSE) between the predicted and true aerodynamic forces and moments was used as the loss function.
- **Optimizer:** The Adam optimizer was used for all training, with an initial learning rate of $1 \cdot 10^{-3}$.
- **Batch Size:** A batch size of 64 was used for all experiments.
- **Epochs:** Models were trained for a maximum of 50 epochs, with an early stopping criterion that terminated training if the validation loss did not improve for 5 consecutive epochs. The model weights from the epoch with the lowest validation loss were saved and used for the final evaluation.

APPENDIX D
EXPERIMENTAL PAYLOAD CHARACTERIZATION

Precise knowledge of the CoM is fundamental for correctly interpreting raw sensor data. Forces and moments must be transformed from the Load Frame to the Body Frame, which is centred at the payload's CoM. While force vectors are invariant under this translation, moments must take into account the lever arm between the sensor origin and the CoM.

An automated algebraic method was developed to simultaneously solve for the payload's mass, its CoM offset vector, and the load cell's intrinsic measurement bias. The procedure takes static force and moment readings at three distinct, mutually orthogonal orientations of the payload with respect to gravity.

By changing the orientation, the direction of the gravity force vector changes within the Load Frame, while the CoM and sensor bias remain constant. This creates a solvable system of linear equations. This approach is distinct from the numerical minimization method used in prior work [7], as it is a direct, non-iterative algebraic solution. The technique, along with a $2\,\mathrm{s}$ stabilisation period in each position, takes about $10\,\mathrm{s}$. Each position was recorded for $1\,\mathrm{s}$, with low-pass filtering at $5\,\mathrm{Hz}$, followed by averaging collected samples to obtain a single reading.

A notable practical challenge of this method was the physical execution of the required orthogonal poses. The 90° rotations, combined with the drone's position on a long mounting strut, introduced a significant collision risk with the robotic arm's own structure. Safe execution required careful supervision and application of robot angular velocity constraints.

For the primary analysis of aerodynamic lift and drag forces detailed in the Method, explicit CoM characterisation was not strictly necessary, as the approach relied on a differential technique to cancel out gravity/inertia contributions and isolate aerodynamic components. However, the CoM characterisation capability was fully developed and retained in case it is used for future analysis of the high-resolution moments, which were also recorded for more advanced studies.

Framework/Protocol	**QoS/Method**	**Loss** [kHz]	**Limit** [kHz]
ROS2 (`rclpy`)	Reliable		2.5
ROS2 (`rclpy`)	Best Effort	1.3	2.5
ROS2 (`rclcpp`)	Reliable		2.7
ROS2 (`rclcpp`)	Best Effort	4.8	6.8
UDP (Python)	Loopback		33
UDP (Rust)	Loopback		950
Tokio MPSC (Rust)	In-process		11 300

TABLE V: Message Throughput Comparison

APPENDIX E
PERFORMANCE ANALYSIS AND IMPLEMENTATION RATIONALE

This appendix provides a quantitative basis for key architectural decisions made during the development of the CARAC platform, particularly the choice of a custom Rust-based orchestrator over ROS2 and the design of the high-throughput data acquisition sink.

A. Performance Evaluation of ROS2 vs. Custom Solution

A primary requirement for the orchestrator was its ability to handle high-frequency, low-latency data streams from multiple hardware agents simultaneously. To validate the suitability of existing frameworks, a performance benchmark was conducted on a single machine (localhost) to measure the maximum reliable message frequency of ROS2.

1) Methodology: A C++ publisher node was created to publish a monotonic sequence of `Int64` messages at a gradually increasing frequency. C++ and Python subscriber nodes received the messages and validated the sequence number. Two Quality of Service (QoS) profiles were tested: *Best Effort* and *Reliable*, both with *Volatile* durability and a *History Depth* of 10. The performance of these configurations was compared against a simple Python UDP loopback test and an in-process Tokio MPSC (Multi-Producer, Single-Consumer) channel, which represents the internal communication primitive used in the final Rust orchestrator.

2) Results and Justification: The performance results, summarized in Table V, revealed significant limitations in ROS2's throughput. The receiver process would attain a core utilisation of $100\,\%$ before the reception frequency started to degrade. The *Keep All* history setting caused unbounded memory allocation, eventually raising a C++ exception. In contrast, the custom Rust and Python UDP implementations demonstrated orders of magnitude higher throughput.

This performance gap, combined with the project's limited need for ROS2's broader features like distributed discovery, solidified the decision to develop a custom orchestrator. While further ROS2 tuning (e.g., different DDS implementations, message batching) was possible, the superior performance and development flexibility of the custom Rust solution were compelling. The orchestrator uses $2.5\,\%$ CPU core utilisation, receiving $5.5\,\mathrm{Mbit\,s^{-1}}$ (or $7360\,\mathrm{UDP\,packets/s}$).

B. Data Sink Architecture Rationale

The `DataSink` is the core component for data collection, designed to handle concurrent, high-frequency writes from multiple asynchronous hardware agents without introducing contention bottlenecks or data loss. Its architecture is based on a two-level locking strategy and efficient buffer management.

1) Concurrent Writing Strategy: Upon registration, each hardware agent is provided with a lightweight `StreamWriter` handle. This handle contains the index to a dedicated buffer within a shared vector, allowing each agent to write its data without interfering with the buffers of other agents. The shared state is protected by a two-level locking mechanism:

1) An outer ‘RwLock‘ guards the shared state containing the recording status. During data acquisition, hardware agents only require a read-lock to check if recording is active. Read-locks can be held concurrently, so multiple agents can proceed without blocking one another. An exclusive write-lock is only taken for the brief moments when starting or stopping a recording run.
2) An inner ‘Mutex‘ protects the data buffers. This lock is acquired only after the cheap read-lock on the outer state confirms that a sample should be recorded. The critical section is extremely short: it only involves indexing the vector and appending a sample to the agent’s dedicated buffer.

This design ensures that contention is minimized during the high-throughput recording phase.

2) Buffer Management and Data Persistence: Instead of using a growable/pre-allocated `Vec<u8>`, each stream’s buffer is implemented with the `chunked_bytes::ChunkedBytes` crate. This data structure allocates memory in larger, non-contiguous chunks, avoiding the potentially costly reallocation of a simple growable vector during a long recording session.

A core principle of the platform is the non-destructive preservation of raw data. At the end of a ”run,” the `DataSink` locks the buffers and drains their binary contents into a custom file format. This raw file contains all recorded samples, including noise. All filtering and analysis is performed in a separate post-processing step. This ensures that experiments never need to be re-run to test a different filter parameter; the raw data is always available for reprocessing, which is a critical aspect for ensuring experimental repeatability and exploring different analysis techniques.