Dynamic UAV Flight Simulator utilizing a Stewart Platform

Anurag Mukherjee*†, Irfan Khan†, William Turner‡, Peter Sullivan‡,
Chris Mahony‡, Oliver Gilbert‡ and John Economou†

†Cranfield University, Defence Academy of the UK, Shrivenham, United Kingdom

‡Richmond Defence Systems Ltd, North Lopham, United Kingdom
Email: *anurag.mukherjee@cranfield.ac.uk

***Abstract*—This paper focuses on replicating the motion profile of a UAV flying in different conditions on a Stewart platform testbed.**

The resulting system provides a safe platform to operate indoors reproducing the same signals to those of a UAV flying outdoors. This approach results in more frequent use of all flying abilities while retaining high safety standards. To illustrate the effectiveness of the testbed, experimental results are presented.

Six linear actuators are used in the testbed to achieve the desired orientation and position. The desired length of each actuator is calculated via an inverse kinematics algorithm. To validate the algorithm, Inertial Measurement Unit (IMU) data from three aerial platforms of different configurations/size are used. The IMU data captured from flying/hovering the aerial platforms indoors, with and without disturbance, is pre-processed using Fast Fourier Transform (FFT) and used as an input to the Stewart platform.

***Index Terms*—Digital Signal Processing, Modelling and Simulation, UAVs**

I. Introduction

The use of unmanned aerial vehicles has increased significantly in the civil and military sectors. A common use case is a platform for sensing devices such as cameras, lidars and radars. On the military side, systems for surveillance, target tracking, and directed beam weapons are used in times of conflict [1]. The challenge is that many of these systems require a high standard of stability to achieve the precision needed.

There are drone/motion simulators that can create scenarios, test out various algorithms and conduct drone-based research within in a virtual environment. Two of the most accurate physics-based simulators for UAVs include Airsim [2] and Gazebo [3]. AirSim offers high-fidelity physics and real-time hardware-in-the-loop simulations using Unreal Engine 4, making it ideal for accurately simulating drone motion and external disturbances. It is modular and supports advanced algorithm testing, but requires significant computational resources and supports limited UAV models. Gazebo [4] is versatile and integrates well with ROS (Robot Operating System), supporting various physics engines and UAV models. It is feature-rich with a large user community, but has less advanced rendering compared to AirSim and may need additional setup for specific protocol support.

In terms of stability, complex gimbal systems are used to compensate for the disturbances introduced by wind gusts. However, the development of complex gimbal systems cannot rely solely on computer simulations. While simulations can provide valuable insights, they often rely on approximations, linearizations, and other simplifications to replicate real-world motions and disturbances. These simplifications fail to capture all the non-linearities of an actual system, such as friction, stiction, inertia, etc. Consequently, a trade-off must be made between the level of detail, model complexity, and computational resource requirements [5].

Therefore, real flight tests and verification of acquired data become crucial to accurately determine the performance of these systems. However, conducting flight tests presents challenges, especially when the payload is confidential or when obtaining flight test permissions, is difficult due to factors like location, time of day, proximity to public areas, drone weight, and the availability of standard safety procedures [6]. These constraints complicate the execution of regular and repeated flight tests during the research and development phase of a product. In such cases, the use of a physical simulator to replicate expected or real drone movements offers an effective alternative to actual flight tests. Physical simulations provide a repeatable and safe testing environment, eliminating the need for spatial and temporal permissions required for drone flights.

A Stewart platform is a parallel manipulator consisting of a fixed base and a movable platform connected by six adjustable-length legs (actuators). Each leg has a universal or ball joint at both ends, allowing motion in six degrees of freedom: (X, Y, Z, pitch, roll, yaw). The platform's position and orientation are controlled by varying the lengths of the legs, making it suitable to produce high precision movements.

Recent advancements in the design and control of parallel manipulators have significantly expanded their applications. Notable researches include [7] that introduces a novel analytical geometry method for accurately determining the kinematic and dynamic parameters of Stewart platforms. This approach provides precise calculations without relying on iterative approximations. Another study [8] discusses an isotropic Stewart

Funded by UKRI (UK Research and Innovation)

platform configuration, detailing the design and construction of an experimental prototype. This research includes verification experiments to test the platform's isotropic characteristics and provides a comprehensive analysis of its dynamic behaviour. [9] investigates the dynamics and control mechanisms of a six-axis vibration isolator using a flexible Stewart platform. Further, [10] presents a new dynamic model for Stewart platforms featuring flexible hinges. This model leverages the complex Kane equation and the principle of virtual forces to enhance the understanding of the platform's dynamics.

In this research paper, the use of a Stewart platform [11] is presented in the case of replicating the motions of different drones in different operating conditions such as hovering in an isolated environment or in the presence of wind disturbances and other factors. In [12], a hybrid six wheeled and legged robot is designed; the inclusion of the 6 legs in a Stewart platform configuration allows for higher flexibility. The robot can automatically manage the gait to adjust the height and wheel track, and the developed control system can maintain level attitude of the drone. [13] uses the Stewart platform for attitude compensation on a UGV to serve as a compliant and horizontal drone landing station. In [14], an earthquake simulator is developed utilizing a Stewart platform, which operates based on a ground motion movement profile consisting of position (x, y, z) and orientation (α, β, γ) based on real earthquake records. A Proportional-Derivative controller is used to control the linear actuators of the Stewart platform.

A lab-based emulation system of an underwater vehicle-manipulator using a KUKA IIWA14 robotic arm on a Stewart Platform is described in [15]. The setup simulates real-world underwater conditions, including water currents and dynamic coupling effects. It allows for rapid, repeated experiments, overcoming challenges of testing in actual water tanks. This setup aids in developing control techniques for underwater robotics. A low-cost flight simulator is designed in [16], which uses a 3-DOF Stewart platform combined with a computer system running Unity 3D based simulation software. A motion tracking system is designed in [17], that uses inexpensive displacement sensors and a 6-DOF Stewart platform using ball-and-socket joints. Kinematics involved in it use quasi-Newton approach to calculate position and orientation data of the top of the system.

In the majority of the research mentioned above, Stewart platforms and their variations are used mainly as vibration isolators and to create stabilized platforms for conducting experiments. A lot of work has been done to fine-tune the controller of the platform to produce the maximum vibration isolation and external disturbances. Although there have been research that use a Stewart platform to simulate motion to create flight or driving simulators [18],[19],[20], they are more geared to imprecise large movements. The research aims to simulate vibrations and small precise motions similar to those experienced by a VTOL drone in flight in different environmental conditions using a Stewart platform with low-cost actuators.

Data collected via an inertial measurement unit (IMU) stuck on a drone is pre-processed and fed as input to the Stewart platform, running a simple PID controller for each of its 6 linear actuators. Another IMU stuck on the Stewart platform is used to verify that the movement of the platform has the same characteristics and profile as that of the drone whose movements were fed as inputs. A detailed mathematical background and pre-processing algorithms as well as experimental procedures are present in subsequent sections of the paper.

II. Theoretical background

This section provides a concise overview of a Stewart platform, digital signal processing techniques, and corresponding mathematical expressions and equations.

A. Stewart platform

A Stewart platform is made up of six linear actuators, where the ends form two equilateral triangles offset $180°$ from each other. These triangles form the planes used for the base reference frame and the location of the platform. The translation vector $\vec{\mathbf{H}}$ is the lateral offset between the two planes and the 3x3 rotation matrix ${}^{P}R_{B}$ is how the Platform is rotated with respect to the Base. Where each of the six actuators intersect with the Base and Platform plane is given in TABLE I, defined as matrices $\mathbf{B}$ and $\mathbf{P}$ respectively.

To populate these matrices, the radius r of the circle that intersects all the points of a triangle is used along with the separation s between the end of two adjacent actuators. r' is the actual radius going through the attachment points. ϵ is the offset angle given by the separation between the two adjacent actuator ends. The relationship between planes is shown in figure 1 and the defined distances within each plane are shown in figure 2.

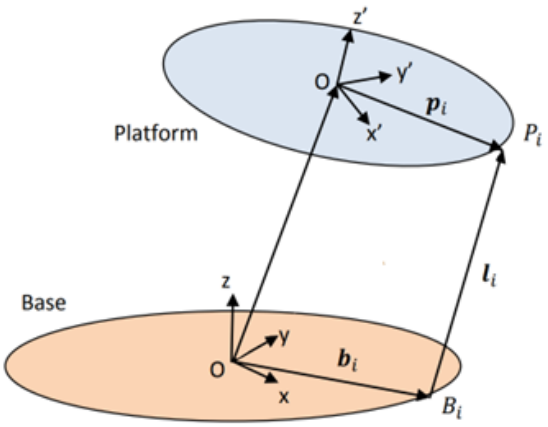

Figure 1: Stewart platform base and platform planes

The origin of plane P with respect to plane B is denoted by the 3D vector H. The location of the ends of the 6 actuators in the P plane and B plane are denoted by matrices B and P respectively. Both matrices are made up of 6 columns, each of which is a 3D column vector.

The offset angles ϵ in the 2 planes, ϵ_B and ϵ_P are defined as follows:

$$\epsilon_B = \arctan(\frac{s/2}{r_B}) \tag{1}$$

$$\epsilon_P = \arctan(\frac{s/2}{r_P}) \tag{2}$$

The actual radius r' (r'_P and r'_B) are defined as follows:

$$r'_B = \sqrt{r_B^2 + (\frac{s}{2})^2} \tag{3}$$

$$r'_B = \sqrt{r_P^2 + (\frac{s}{2})^2} \tag{4}$$

Given these definitions, the values for B and P can be populated using the equations in TABLE I. The z component can be either 0 or the offset between the connection plane and the desired reference plane.

Now for a given linear and rotational offset, each actuator's position vector can be calculated by:

$$L_i = H + {}^{P}R_B.P_i - B_i \tag{5}$$

From this, the actual length of the linear actuator can be calculated and then be used to command the actuator to the new position.

$$l_i = |L_i| = \sqrt{(L_{ix})^2 + (L_{iy})^2 = (L_{iz})^2} \tag{6}$$

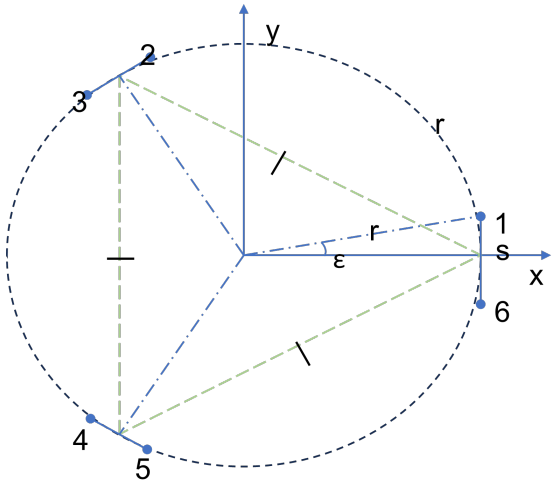

Figure 2: Stewart platform schematics of linear actuator ends on B and P planes

B. Fourier Transform and frequency domain analysis

The Fourier Transform [21] is a mathematical operation that transforms a time-domain signal into a frequency-domain representation. It decomposes a function into its constituent frequencies. An IMU attached to the chassis of the drone, centered from above and aligned with the front of the drone, was used to capture its angular velocities at a sample-rate of 400Hz. This time-domain velocity signal was then transformed into frequency-domain. Its power spectral density is calculated using Welch's method, which provides restriction of frequency resolution and filtering out of noise.

Welch's method [22] is a refinement of the periodogram method, which aims to reduce the variance of the frequency by averaging multiple periodograms obtained from overlapping segments of the signal. It divides a discrete time signal $x[n]$ into L overlapping segments each of length M and an overlap of D. A rectangular window function $w[n]$ is used to reduce spectral leakage. The l^{th} windowed segment is given by $x_l[n] = x[n+l.D].w[n]$. The DFT of each windowed segment $X_l[k]$ is then computed as:

$$X_l[k] = \sum_{n=0}^{L-1} x_l[n].e^{-j\frac{2\pi}{M}kn} \tag{7}$$

The periodogram $P_l[k] = \frac{|X_l[k]|^2}{M}$ for each segment is calculated and are averaged across all the windows to produce the final power spectral density given as

$$\hat{P}[k] = \frac{1}{L}\sum_{l=0}^{L-1} P_l[k] \tag{8}$$

The frequency associated with k is derived from M and the sample-rate R. An adapted form of Welch's method, implemented via Python's Scipy library, computes the power spectrum and its amplitude square-rooted to get the signal amplitude $\hat{A}[k]$. Phase data $\hat{S}[k]$ are derived separately from the DFTs complex mean argument.

$$\hat{F}[k] = \frac{R}{M}k \tag{9}$$

$$\hat{A}[k] = \sqrt{\hat{P}[k]} \tag{10}$$

$$\hat{S}[k] = \arg\sum_{l=0}^{L-1} X_l[k] \tag{11}$$

Due to hardware limits on the linear actuators, the max frequency h is capped at 4Hz and equates to $K = \lfloor \frac{M}{R}h \rfloor$ total sine curves. The original time-domain signal $\omega[n]$ is reconstructed by accumulating sine curves for timestep T and its accumulation $t_n = t_{n-1} + T$. This is repeated for the roll, pitch and yaw axes.

$$\omega[n] = \sum_{k=0}^{K-1} \hat{A}[k].\sin(2\pi.\hat{F}[k].t_n + \hat{S}[k]) \tag{12}$$

A quaternion-based kinematics model uses FFT-derived angular velocities for Stewart platform attitude propagation. Quaternions prevent gimbal lock and enable smooth interpolation between rotations. Given the angular velocities vector $\vec{\omega} = (\omega_x, \omega_y, \omega_z)$ for time t_n, the rotation angle $\theta = |\vec{\omega}|T$ is calculated. The quaternion for rotation δq is as follows:

$$\delta q = cos(\frac{\theta}{2}) + sin(\frac{\theta}{2})(\frac{\omega_x}{|\vec{\omega}|}i + \frac{\omega_y}{|\vec{\omega}|}j + \frac{\omega_z}{|\vec{\omega}|}k) \tag{13}$$

The new quaternion at time t_n is given by $q_{t_n} = slerp(q_{t_0}, \delta q.q_{t_{n-1}}, 0.9^T)$. To prevent dead reckoning errors

TABLE I: Equations of length of actuators given Stewart platform attitude and position

Actuator No	B_x	B_y	P_x	P_y
1	$r'_B.\cos(\epsilon_B)$	$r'_B.\sin(\epsilon_B)$	$r'_P.\cos(\frac{\pi}{3}-\epsilon_P)$	$r'_P.\sin(\frac{\pi}{3}-\epsilon_P)$
2	$r'_B.\cos(\frac{\pi}{3}+\epsilon_B)$	$r'_B.\sin(\frac{\pi}{3}+\epsilon_B)$	$r'_P.\cos(\frac{\pi}{3}+\epsilon_P)$	$r'_P.\sin(\frac{\pi}{3}+\epsilon_P)$
3	$r'_B.\cos(\frac{2\pi}{3}+\epsilon_B)$	$r'_B.\sin(\frac{2\pi}{3}+\epsilon_B)$	$r'_P.\cos(\frac{3\pi}{3}-\epsilon_P)$	$r'_P.\sin(\frac{3\pi}{3}-\epsilon_P)$
4	$r'_B.\cos(\frac{4\pi}{3}-\epsilon_B)$	$r'_B.\sin(\frac{4\pi}{3}-\epsilon_B)$	$r'_P.\cos(\frac{3\pi}{3}+\epsilon_P)$	$r'_P.\sin(\frac{3\pi}{3}+\epsilon_P)$
5	$r'_B.\cos(\frac{4\pi}{3}+\epsilon_B)$	$r'_B.\sin(\frac{4\pi}{3}+\epsilon_B)$	$r'_P.\cos(\frac{5\pi}{3}-\epsilon_P)$	$r'_P.\sin(\frac{5\pi}{3}-\epsilon_P)$
6	$r'_B.\cos(-\epsilon_B)$	$r'_B.\sin(-\epsilon_B)$	$r'_P.\cos(\frac{5\pi}{3}+\epsilon_P)$	$r'_P.\sin(\frac{5\pi}{3}+\epsilon_P)$

due to integration, a 10% dampener over time is added. This is done with spherical linear interpolation ('slerp' for short) between the new rotation and the initial rotation. This is then converted to matrix representation for the Stewart platform input.

III. Experimental Setup

'x-IMU3', a commercially available MEMS-based real-time wireless IMU sensor, collected inertial data at 400Hz from different DJI drones (Phantom 4, Matrice 100, Matrice 600 Pro). These were flown indoors for 1 minute with and without simulated and controlled wind disturbances (generated by a fan kept close to the drone while it hovered). Each experiment had 3 runs to ensure data collected was uniform and unbiased and were aimed to collect drone dynamics data under varying conditions.

Data from drone attitude experiments were processed and fed as velocity signals into an Arduino Due. These signals mimicked drone motion profiles using sine waves derived from FFT-analyzed data, crucial for controlling the Stewart platform. The IMU sensor is then attached to the Stewart platform to repeat the same frequency analysis as before.

The system employs Pololu LACT6P linear actuators and a Jrk 21v3 motor controller, compatible with multiple interfaces and operating on voltage ranging from 5V to 28V. It provides up to 3A continuous (5A peak) current output, supports analog voltage and tachometer feedback for closed-loop servo systems, and includes a user-friendly Windows configuration utility for calibration and setup.

IV. Results and Discussion

This section compares the results obtained after simulating on the Stewart platform several flight conditions from different drones.

For the sake of comparison, roll, pitch and yaw are combined to get an overall angular speed. Phase is excluded from the comparison as it's not guaranteed to correlate without a predefined start point. This reduces the motion profile to a motion signature represented by a sequence of K amplitudes mapped to a common sequence of frequencies. For each drone scenario, six motion signatures are obtained that correspond

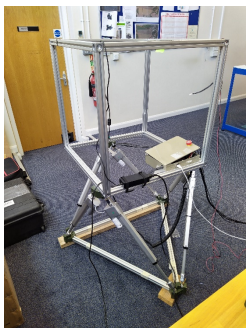

Figure 3: Stewart platform.

to three original flights and three respective Stewart platform simulations. We define a function D that calculates the total standard deviation within a set of signatures. It takes a matrix M as input, with N signatures as rows and each of their K amplitudes as columns:

$$D(M) = \sqrt{\frac{1}{NK}\sum_{n=1}^{N}\sum_{i=1}^{K}\left(M_n[i] - \frac{1}{K}\sum_{j=1}^{K}M_n[j]\right)^2} \quad (14)$$

In the case of the original flight signature O_i, $i \in \{1,2,3\}$ and its simulated signature S_i, $i \in \{1,2,3\}$ where i refers to the three experiment runs, half the RMS error between them is returned. This serves as a measure of the inaccuracy (ϵ) encountered when replicating a drone scenario on the Stewart platform. for a specific run. This concept is extended to the three original flights, providing a measure of inconsistency between runs. This inconsistency effectively represents the tolerated inaccuracy for a given drone scenario.

$$\epsilon_i = D([O_i, S_i]) \quad \text{for } i \in \{1,2,3\} \quad (15)$$

The tolerance τ is defined as follows

$$\tau = D([O_1, O_2, O_3]) \quad (16)$$

In principle, the simulation's inaccuracy ϵ must be lower than that of the tolerance. The cumulative inaccuracy between

TABLE II: Summary of results comparing the RMS and % differences

Drone	Scenario	Run	Tolerance	Run Inaccuracy	Full Inaccuracy	Run Ratio	Full Ratio
DJI Phantom 4	Hovering on spot	1	0.002668	0.002520	0.002191	94%	82%
		2		0.001849		69%	
		3		0.002150		81%	
	Hovering 1.5m from fan	1	0.004424	0.000525	0.001210	12%	27%
		2		0.001234		28%	
		3		0.001609		36%	
DJI Matrice 600 Pro	Hovering on spot	1	0.004629	0.002096	0.002361	45%	51%
		2		0.002953		64%	
		3		0.001899		41%	
	Hovering 2.0m from fan	1	0.003162	0.002644	0.002850	84%	90%
		2		0.003183		101%	
		3		0.002692		85%	
DJI Matrice 100	Hovering on spot	1	0.007399	0.003198	0.003669	43%	50%
		2		0.003897		53%	
		3		0.003869		52%	
	Hovering 2.0m from fan	1	0.007230	0.001189	0.002284	16%	32%
		2		0.001105		15%	
		3		0.003608		50%	

the original flight and the Stewart platform across all three runs determines the overall inaccuracy $\hat{\epsilon}$ in replicating a drone scenario. This can be expressed as a ratio relative to the tolerance, where a ratio below 100% is adequate and closer to 0% is near-perfection. TABLE II summarises the results.

$$\hat{\epsilon} = \sqrt{\sum_{i=1}^{3} \frac{\epsilon_i^2}{3}} \tag{17}$$

$$\text{success } = \frac{\hat{\epsilon}}{\tau} < 100\% \tag{18}$$

The fourier analysis plots for the experiments were compared and are shown in Figure 4 and Figure 5.

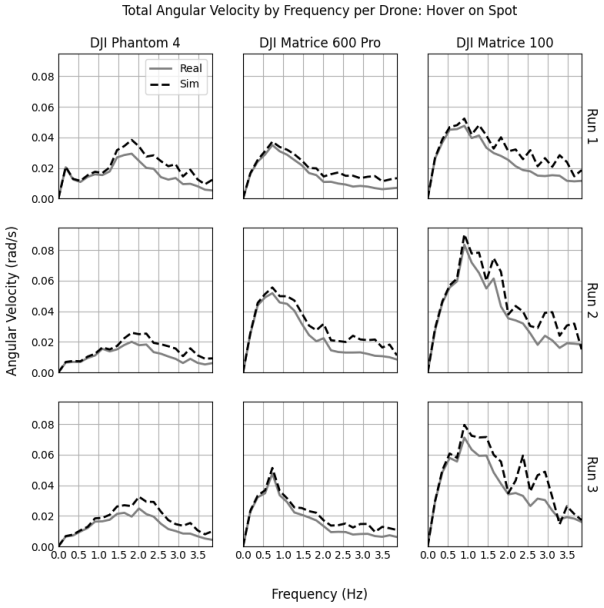

Figure 4: Results in the absence of artificial wind

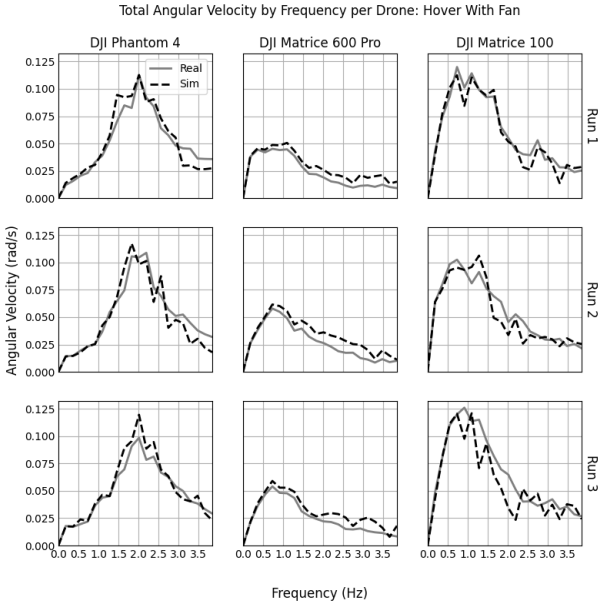

Figure 5: Results in the presence of artificial wind

Based on the established criteria, the Stewart platform successfully replicated each drone scenario in all but one of the runs. Notably, drones exhibiting the most pronounced movement, such as the DJI Phantom 4 and DJI Matrice 100 under fan disturbance, were replicated more accurately. Analysis of the plots indicates that the Stewart platform tends to overestimate low amplitude movements. This is especially evident with the DJI Matrice 600, which, being the heaviest of the three drones, was the least affected by wind disturbance.

V. CONCLUSION

The goal of this research is to develop a reliable, accurate, safe and replicable means to simulate drone motion in real life, outside of computer simulations, which is crucial to many

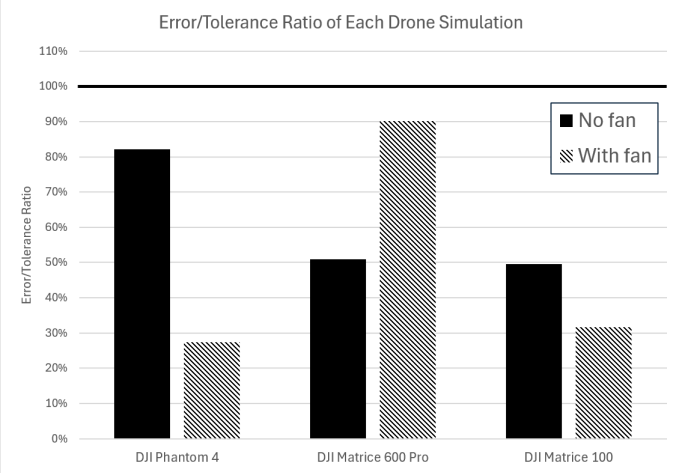

Figure 6: Ratio of error over tolerance

use cases such as vibration analysis and damping, control systems development, payload testing, etc. The results show that with cheap linear lead-screw based geared actuators, the simulator can replicate low-frequency drone motion accurately. Drone simulation using the Stewart platform eliminates the risks of flying real UAVs with fast-moving parts, making it safe and ideal for research and student learning environments. Future extensions to this research might explore upgrading the actuators of the simulator to pneumatic or hydraulic ones, that have better dynamics and response. Addition of high frequency vibration generation capabilities along with better actuators will exponentially enhance the realism of the simulated drone motion thus increasing the usability of the Stewart platform based simulator for more advanced research and development. Future works will include the use of AI for fusing environmental models into the UAV flight for the Stewart platform.

VI. Acknowledgement

The authors would like to thank Dr Lounis Chermak for the use of the Aviation Hall at the Defence Academy of the UK and Dr Luke Feetham for assisting with piloting the UAVs for this research.

References

[1] D. Axe, "A grenade-armed ukrainian drone spotted two russian soldiers. one russian lit a smoke, and asked the drone-operator to kill his comrade first.," *Forbes*, Jan. 2024. [Online]. Available: https://www.forbes.com/.

[2] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics: Results of the 11th International Conference*, Springer, 2018, pp. 621–635.

[3] E. Ebeid, M. Skriver, K. H. Terkildsen, *et al.*, "A survey of open-source uav flight controllers and flight simulators," *Microprocessors and Microsystems*, vol. 61, pp. 11–20, 2018.

[4] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, vol. 3, 2004, pp. 2149–2154.

[5] R. J. Brooks and A. M. Tobias, "Choosing the best model: Level of detail, complexity, and model performance," *Mathematical and computer modelling*, vol. 24, no. 4, pp. 1–14, 1996.

[6] A. P. Cracknell, "Uavs: Regulations and law enforcement," *International Journal of Remote Sensing*, vol. 38, no. 8-10, pp. 3054–3067, 2017.

[7] R. V. Petrescu, R. Aversa, A. Apicella, *et al.*, "Inverse kinematics of a stewart platform," *Journal of Mechatronics and Robotics*, vol. 2, no. 1, pp. 45–59, 2018.

[8] H. Yun, L. Liu, Q. Li, *et al.*, "Development of an isotropic stewart platform for telescope secondary mirror," *Mechanical Systems and Signal Processing*, vol. 127, pp. 328–344, 2019.

[9] X. Yang, H. Wu, B. Chen, *et al.*, "Dynamic modeling and decoupled control of a flexible stewart platform for vibration isolation," *Journal of Sound and Vibration*, vol. 439, pp. 398–412, 2019.

[10] J. Jiao, Y. Wu, K. Yu, and R. Zhao, "Dynamic modeling and experimental analyses of stewart platform with flexible hinges," *Journal of Vibration and Control*, vol. 25, no. 1, pp. 151–171, 2019.

[11] D. Stewart, "A platform with six degrees of freedom," *Proceedings of the institution of mechanical engineers*, vol. 180, no. 1, pp. 371–386, 1965.

[12] Z. Chen, J. Li, J. Wang, *et al.*, "Towards hybrid gait obstacle avoidance for a six wheel-legged robot with payload transportation," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 3, p. 60, 2021.

[13] J. Si, B. Li, L. Wang, *et al.*, "A uav autonomous landing system integrating locating, tracking, and landing in the wild environment," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 2, p. 51, 2024.

[14] E. Alvarado Requena, A. Estrada, G. T. Ramírez, *et al.*, "Control of a stewart-gough platform for earthquake ground motion simulation," in *Latin American Symposium on Industrial and Robotic Systems*, Springer, 2019, pp. 138–146.

[15] K. Cetin, H. Tugal, Y. Petillot, *et al.*, "A robotic experimental setup with a stewart platform to emulate underwater vehicle-manipulator systems," *Sensors*, vol. 22, no. 15, p. 5827, 2022.

[16] C. Villacís, M. Navarrete, I. Rodriguez, *et al.*, "Real-time flight simulator construction with a network for training pilots using mechatronics and cyber-physical system approaches," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, IEEE, 2017, pp. 238–247.

[17] Y.-S. Kim, H. Shi, N. Dagalakis, *et al.*, "Design of a six-dof motion tracking system based on a stewart platform and ball-and-socket joints," *Mechanism and machine theory*, vol. 133, pp. 84–94, 2019.

[18] H.-L. Tseng and I.-K. Fong, "Implementation of a driving simulator based on a stewart platform and computer graphics technologies," *Asian Journal of Control*, vol. 2, no. 2, pp. 88–100, 2000.

[19] J. Pradipta, M. Klünder, M. Weickgenannt, and O. Sawodny, "Development of a pneumatically driven flight simulator stewart platform using motion and force control," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, IEEE, 2013, pp. 158–163.

[20] Y. Yang, S. T. Zheng, and J. W. Han, "Motion drive algorithm for flight simulator based on the stewart platform kinematics," *Key Engineering Materials*, vol. 460, pp. 642–647, 2011.

[21] K. Kido, *Digital Fourier analysis: fundamentals*. Springer, 2014.

[22] P. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on audio and electroacoustics*, vol. 15, no. 2, pp. 70–73, 1967.