

Simulation evaluation of the takeoff threshold of an ArduPilot Gazebo quadrotor and conditions for physical implementation

Toshihiko TAKAYA

`takaya.toshihiko@gmail.com`

XXXXXXXX

Research Article

Keywords: ArduPilot, Gazebo, quadrotor, software-in-the-loop, takeoff threshold, distributed simulation, physical validation

Posted Date: October 27th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11201742/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Abstract

This study quantitatively evaluates the takeoff boundary of a quadrotor using ArduPilot software-in-the-loop coupled with Gazebo. Radio-control throttle commands from 1450 to 1500 were assessed within a common 7 s window in host log-recording time, from 1 to 8 s after each newly observed command. No altitude increase was observed at command 1485. At command 1486, both trials yielded an altitude increase of 0.090 m, with a mean telemetry climb rate of 0.02709 ± 0.00064 m/s (mean $\pm$ descriptive SD, denominator n). The observed transition between no takeoff and takeoff was therefore bracketed by integer commands 1485 and 1486 under the tested conditions. Altitude increase and mean climb rate increased at higher commands. The inertial masses specified for the components of the iris_with_gimbal model summed to 1.98 kg. This modeled mass provides an initial design reference for a physical vehicle. The results establish a baseline for repeated threshold evaluation, distributed simulation across multiple computers, and subsequent physical thrust calibration. Because the command and motor-output values have not been calibrated to physical thrust, the observed boundary is specific to the present simulation configuration.

1 Introduction

Earlier studies investigated the motion design of a small autonomous indoor blimp using proportional–integral–derivative (PID) control and a state-transition method, with position and velocity obtained primarily from charge-coupled device (CCD) camera images as state variables [1]. A subsequent study introduced flight plans with time constraints by adding time information to the state variables, enabling arrival at specified positions at specified times [2]. These approaches constructed autonomous flight behavior from externally observed changes in vehicle position. Image acquisition, state estimation, and control-output calculation required processing time, and their architecture differed from attitude control that directly uses inertial measurements acquired continuously onboard the vehicle.

For quadrotors, attitude estimation and attitude and altitude control using gyroscopes and accelerometers have been validated experimentally [3]. Attitude tracking based on the fusion of multiple inertial measurement unit (IMU) sensor signals [4], and geometric integral attitude control supported by numerical simulation [5], have also been reported. These studies have advanced flight research using onboard sensors and flight-control software. Physical flight is nevertheless affected by vehicle condition, battery condition, and external disturbances. A reproducible simulation environment is therefore useful for comparing small input differences immediately before takeoff under controlled conditions.

This study couples ArduPilot Copter software-in-the-loop (SITL) [6] with Gazebo [7] and varies the radio-control (RC) throttle command to identify the highest tested integer command at which no takeoff is observed and the lowest at which takeoff is reproduced. The computer configuration, software versions, and dependencies are documented, and the model mass and rotational-speed scale are compared with a candidate physical propulsion system. Simulator variables cannot be interpreted directly as physical

thrust. The aim is thus to establish reproducible analysis conditions and a basis for staged physical implementation.

2 Methods

2.1 Simulation environment

The simulation ran on the Intel MacBook Pro specified in Table 1. Gazebo operated in server/headless mode and communicated with ArduPilot through a JavaScript Object Notation (JSON) control interface on localhost. The vehicle configuration was Quad X, and the flight mode was STABILIZE. The interface settings were `fdm_addr = 127.0.0.1`, `fdm_port_in = 9002`, `lock_step = 1`, and `no_time_sync = 1`. The Gazebo physics settings were `max_step_size = 0.001` and `real_time_factor = 0.5`.

Table 1
Simulation computer, software versions, and dependencies

Item	Specification, version, or dependency
Host / OS	MacBookPro11,4; Intel Core i7 2.8 GHz, 4 cores; 16 GB RAM; macOS 14.8.7 (23J520); Homebrew 6.0.20 (/usr/local)
Simulation	Gazebo Harmonic / gz-sim8 8.15.0; sdfORMAT14 14.9.0; qt@5 5.15.19 and Qt 6.11.1 installed; plugin linkage verified separately
Python / MAVLink	venv Python 3.11.16; MAVProxy 1.8.74; pymavlink 2.4.49; setuptools 83.0.0
Source / plugin	ArduPilot master de5add012...; ardupilot_gazebo 082a0fe...; rebuilt plugin links gz-common5 5.9.0, gz-transport13 13.6.0, gz-msgs10 10.4.0, gz-math7 7.7.0, protobuf 36.0
Thermal conditions / ABI	CPU temperature was monitored and cooling was maintained with a high fan-speed setting. After Homebrew updates, dynamic-library dependencies and application binary interface (ABI) compatibility were checked, and ArduPilotPlugin was rebuilt when necessary.

On macOS, Homebrew updates can alter the combination of Gazebo, Qt, protobuf, and associated dynamic libraries. Checking dependencies with tools such as `otool -L` and rebuilding the plugin when necessary are important for reproducibility. Prolonged simultaneous execution of Gazebo and SITL imposes sustained central processing unit (CPU) load. Thermal throttling may change wall-clock execution time, the real-time factor, and communication intervals. CPU temperature was therefore monitored, and cooling conditions were kept as consistent as practicable.

2.2 Input conditions and data analysis

The evaluated RC throttle commands were 1450, 1475, 1485, 1486, 1487, 1488, 1490, and 1500. The analysis divided RC_CHANNELS messages into consecutive segments whenever `chan3_raw` changed. The origin of each segment was the host log-recording timestamp of the first RC_CHANNELS message containing the new value, not the time at which the operator issued the command. Segments were

retained if the interval from their first to last RC_CHANNELS message was at least 8 s. This yielded 12 segments in one MAVLink telemetry log: two each at RC1450, RC1486, RC1488, and RC1500, and one each at the other commands. Each retained segment is termed a trial.

RC_CHANNELS, GLOBAL_POSITION_INT, TERRAIN_REPORT, VFR_HUD, and SERVO_OUTPUT_RAW messages were extracted from the Micro Air Vehicle Link (MAVLink) log. For each trial, messages with host log-recording timestamps between 1 and 8 s after the segment origin, inclusive, were selected. This is a 7 s selection window in host time; it is not a 7 s interval in the vehicle time_boot_ms clock. Each message stream was selected using its own recording timestamps without endpoint interpolation.

Altitude increase Δh was calculated as the last minus the first GLOBAL_POSITION_INT.relative_alt value within the window, converted from millimeters to meters. Mean climb rate was the unweighted arithmetic mean of the selected VFR_HUD.climb values in meters per second; it was not calculated as Δh divided by 7 s. The mean motor output was calculated by averaging the nonzero SERVO_OUTPUT_RAW.servo1_raw through servo4_raw values separately over the window and then averaging the four motor means.

For RC conditions with two trials, trial-level altitude increases, mean climb rates, and mean motor outputs were averaged with equal weight. The reported $\pm$ values for altitude increase and climb rate are descriptive standard deviations across trial-level values, calculated with denominator n using statistics.pstdev. They are not sample standard deviations with denominator $n - 1$ or standard errors. Single-trial conditions are reported as individual values. Table 2 reports the mean motor output without a between-trial standard deviation.

The experimental observation criterion for takeoff was sustained positive vertical displacement. The table-generation procedure assigned Takeoff when the RC-group mean altitude increase was positive and No takeoff otherwise; this numerical rule did not independently test ground contact or a specified duration of upward motion. In the selected RC1485 window, all 14 relative-altitude samples were zero. The two RC1486 windows contained 14 and 13 samples, respectively, each progressing nondecreasingly from 0 to 90 mm. RC commands and simulator motor-output values were not treated as physical quantities calibrated to thrust in newtons. Original logs, analysis files in comma-separated values (CSV) format, figures, tables, and related files were retained with Secure Hash Algorithm 256-bit (SHA-256) integrity checks.

Reprocessing the archived telemetry log reproduced all 204 numerical fields in the saved trial-level CSV file (12 trials and 17 fields per trial), within absolute and relative tolerances of 10^{-10} and 10^{-12} , respectively. Aggregation and rounding reproduced all entries in Table 2. The SHA-256 checksums of the six archived source files matched the saved checksum manifest. These checks establish consistency between the archived data and the reported analysis; they do not constitute independent experimental replication.

2.3 Use of artificial intelligence assistance

ChatGPT (OpenAI) was used as a supplementary reference during construction of the simulation environment and data analysis, including consideration of the evaluation interval and sampling interval. The author made the research judgments, selected the analysis conditions, and checked the results. AI assistance was also used to translate and edit the manuscript in English and adapt its formatting.

3 Results

Table 2 summarizes altitude increase, mean climb rate, mean output of the four motors, and takeoff classification for each RC command. No altitude increase was observed at RC1450, RC1475, or RC1485. At RC1485, the altitude increase was 0.000 m and the mean climb rate was 0.00006 m/s. At RC1486, both trials yielded an altitude increase of 0.090 m and the mean climb rate was 0.02709 ± 0.00064 m/s. The transition from no takeoff to positive upward displacement was therefore observed between RC1485 and RC1486.

Table 2
Takeoff response during the common evaluation interval

RC	n	Δh (m)	Mean climb rate (m/s)	Mean motor output	State
1450	2	0.000 ± 0.000	0.00000 ± 0.00000	1526.12	No takeoff
1475	1	0.000	0.00000	1550.96	No takeoff
1485	1	0.000	0.00006	1560.77	No takeoff
1486	2	0.090 ± 0.000	0.02709 ± 0.00064	1561.55	Takeoff
1487	1	0.180	0.05875	1562.25	Takeoff
1488	2	0.265 ± 0.005	0.08282 ± 0.00456	1563.04	Takeoff
1490	1	0.520	0.17159	1565.33	Takeoff
1500	2	1.710 ± 0.020	0.54039 ± 0.03120	1575.18	Takeoff

Values of Δh and mean climb rate are mean $\pm$ descriptive SD (denominator n) for n = 2 and individual values for n = 1. The window is defined in host log-recording time. Motor-output values are uncalibrated simulator outputs; between-trial SDs are not reported.

Figure 1 shows mean climb rate as a function of RC throttle command. The response changed to a clearly positive value at RC1486 and increased with command at RC1487, RC1488, RC1490, and RC1500. Mean climb rates were 0.08282 ± 0.00456 m/s at RC1488 and 0.54039 ± 0.03120 m/s at RC1500.

Figure 2 shows altitude increase as a function of RC throttle command. Altitude increase changed from 0.000 m at RC1485 to 0.090 m at RC1486. It subsequently reached 0.180 m at RC1487, 0.265 ± 0.005 m

at RC1488, 0.520 m at RC1490, and 1.710 ± 0.020 m at RC1500. Vertical displacement therefore increased in the region above the observed takeoff boundary.

Figure 3 shows the relationship between RC command and mean output of the four motors. The mean output was 1560.77 at RC1485 and 1561.55 at RC1486, a difference of 0.78. Motor output increased smoothly across the tested commands, while the vertical response changed from no takeoff to takeoff between RC1485 and RC1486.

4 Discussion

4.1 Interpretation of the observed takeoff boundary

No altitude increase was observed at RC1485, whereas two trials within the same recording session at RC1486 yielded the same altitude increase of 0.090 m. The mean climb rate at RC1486 was 0.02709 ± 0.00064 m/s. Although this was a weak upward response immediately above the observed boundary, the two trials gave similar results. Positive vertical motion at RC1486 was therefore repeatable within the two analyzed segments of this session.

The two RC1486 windows yielded altitude increases of 0.090 m and mean VFR_HUD.climb values of 0.02644843 and 0.02773594 m/s. The elapsed time between the selected first and last altitude samples was 6.66794 and 6.26594 s in host log-recording time, but 3.250 and 2.999 s in the vehicle time_boot_ms clock. Thus, dividing Δh by the nominal 7 s host-time window does not reproduce the reported mean climb rate. The difference follows from the use of distinct measured quantities, clock bases, and discrete sample endpoints. The observed time ratios are consistent with the configured real_time_factor of 0.5, but the recorded time spans, rather than a fixed factor of 0.5, should be used for quantitative comparisons.

The difference in mean motor output between RC1485 and RC1486 was only 0.78, and Fig. 3 shows a smooth increase in motor output with RC command. These observations do not indicate a discontinuity in motor output. Rather, they describe the response near the transition from ground contact to upward motion. Because motor-output values were not calibrated against actual thrust, the difference of 0.78 cannot be interpreted as a thrust difference in newtons.

4.2 Model mass and conditions for physical implementation

The iris_with_gimbal model combines iris_with_standoffs and gimbal_small_3d [8–10]. In the Simulation Description Format (SDF) file for iris_with_standoffs, base_link has a mass of 1.50 kg, the IMU link has a mass of 0.15 kg, and each of the four rotors has a mass of 0.025 kg [8]. The base, yaw, roll, and pitch links of gimbal_small_3d have masses of 0.20, 0.01, 0.01, and 0.01 kg, respectively, giving a gimbal mass of 0.23 kg [9]. The sum of the specified inertial masses is consequently $1.50 + 0.15 + 0.10 + 0.23 = 1.98$ kg.

The value of 1.98 kg is the sum of configured SDF link masses, not the measured mass of a physical vehicle. It is used here as the nominal design mass for an initial physical prototype. The ArduPilotPlugin configuration in iris_with_gimbal specifies a motor-speed command multiplier of 838 [10], corresponding to a rotational-speed scale of approximately 8.0×10^3 rpm when interpreted in rad/s. This is a configured command scale, not a measured rotor speed at RC1486. A propulsion system for a vehicle in the approximately 2 kg class is therefore considered.

Figure 4 provides schematic orthographic views of the model used in the simulation [10]. The top view shows four rotors in a Quad X arrangement around the central body and a gimbal positioned toward the front. The front view shows the vertical arrangement of the body, rotor plane, landing gear, and gimbal. The side view illustrates the relative placement of the forward gimbal, rotor plane, and landing gear near the vehicle center of mass. These views explain the model configuration and are not dimensioned engineering drawings. The nominal mass of 1.98 kg was calculated from SDF inertial masses [8, 9], independently of the drawing scale. The views provide a reference for comparing rotor layout, landing gear, gimbal placement, and center-of-mass location during future physical design.

Figure 5 conceptually illustrates the observed transition. Panel (a) represents the no-takeoff condition corresponding to RC1485, panel (b) represents the boundary region between RC1485 and RC1486, and panel (c) represents RC1486 immediately after the onset of upward motion. This computer-generated illustration explains the observed state transition and does not present additional measurements.

Table 3 compares the model mass and command-speed scale with a commercial propulsion example. The theoretical hover requirement of approximately 495 gf per motor lies between the manufacturer’s reported thrust values at 50% and 55% throttle, and the maximum static thrust is 1332 gf per motor [11]. Four motors at this maximum thrust give a nominal thrust-to-weight ratio of approximately 2.7 for the 1.98 kg design mass. The example thus has thrust margin suitable for consideration as a candidate for the first physical prototype.

Table 3
Model mass and a commercial propulsion example

Item	Simulation or candidate physical configuration
Nominal vehicle mass	1.98 kg; sum of SDF inertial masses: 1.50 + 0.15 + 0.10 + 0.23 kg [8, 9]
Required thrust	Hover: approximately 4.85 N (495 gf) per motor. Initial prototype design target: at least 9.81 N (1.0 kgf) per motor.
Commercial example	T-MOTOR AIR2216II-KV920 motor with T1045 propeller and 4S supply [11]. Thrust: 447 gf at 50%, 536 gf at 55%, 993 gf at 80%, and 1332 gf at 100% throttle; maximum power: 272 W.
Speed comparison	Model command scale: 838 rad/s, approximately 8000 rpm [10]. AIR2216II/T1045 at 75% throttle: 8096 rpm [11]. Dimensions and aerodynamic coefficients differ.

Manufacturer throttle percentages are distinct from the RC command values in Table 2. Commercial data identify a design candidate; they do not establish equivalence to the SDF model. Physical thrust must be calibrated with a thrust stand.

Propeller diameter, aerodynamic coefficients, electronic speed controller (ESC) characteristics, and vehicle geometry differ between the commercial example and the SDF model. The comparison identifies a propulsion system in a similar vehicle class; it does not equate simulator motor-output values with the thrust of the commercial motor.

4.3 Limitations

RC1475, RC1485, RC1487, and RC1490 were each evaluated in only one trial. In particular, the no-takeoff side of the observed boundary at RC1485 has $n = 1$. The present results therefore describe an observed boundary between tested integer RC commands under a common evaluation procedure, rather than a probabilistic threshold estimate. No significance tests were performed. More trials near RC1485 and RC1486 are needed to assess takeoff probability and relate attitude, differences among motor outputs, and vertical acceleration on a common time basis. The results are restricted to the simulation configuration and initial conditions used in this study. The repeated trials are segments within one recording session, not demonstrated independent repetitions after a complete simulator restart. A disarm/arm sequence separates the two RC1486 trials, but this does not establish identical initial conditions. The host-time window also covers different vehicle-time spans among trials; a future analysis using a common vehicle-time window is needed to separate response variability from timing effects.

4.4 Future work

Future work will compare simultaneous execution of Gazebo, ArduPilot SITL, and analysis on one computer with execution distributed across multiple computers. Gazebo physics and visualization will run on one computer, while ArduPilot SITL, future Robot Operating System 2 (ROS 2) integration, or log analysis will run on another. Comparisons will include wall-clock execution time, real-time factor, CPU utilization, CPU temperature, latency and loss in JSON/MAVLink communication, and repeatability of the RC1485/RC1486 boundary. Distributed execution should therefore be assessed in terms of thermal load, communication delay, and time-synchronization stability as well as processing time.

For physical implementation, 1.98 kg will serve as the initial design mass. An ArduPilot-compatible flight controller and a four-cell series-connected (4S) power system will be considered. Evaluation will proceed through thrust calibration, controlled low-altitude flight tests, and model identification before the simulation results are used to guide physical flight settings.

5 Conclusions

This study evaluated the relationship between RC throttle command and takeoff response in an ArduPilot–Gazebo quadrotor simulation. No altitude increase was observed at RC1485. At RC1486, both

trials yielded an altitude increase of 0.090 m, and the mean climb rate was 0.02709 ± 0.00064 m/s. Above this boundary, altitude increase and mean climb rate increased with command from RC1487 to RC1500.

Within the tested configuration and initial conditions, the highest tested integer RC command with no observed takeoff was 1485, and the lowest with repeated takeoff was 1486. The observed integer-command takeoff boundary was therefore bracketed by these two values. The total SDF inertial mass of 1.98 kg provides an initial reference for physical vehicle design. Further trials near the boundary, evaluation of distributed computing and thermal load, physical thrust calibration, and low-altitude flight tests are required to connect the simulation results to a physical vehicle.

Abbreviations

ABI	application binary interface
CCD	charge-coupled device
CG	computer-generated
CPU	central processing unit
CSV	comma-separated values
ESC	electronic speed controller
IMU	inertial measurement unit
JSON	JavaScript Object Notation
MAVLink	Micro Air Vehicle Link
PID	proportional–integral–derivative
RC	radio control
ROS 2	Robot Operating System 2
SD	standard deviation
SDF	Simulation Description Format

SHA-256

Secure Hash Algorithm 256-bit

SITL

software-in-the-loop.

Declarations

Availability of data and materials

The datasets generated and analyzed during the current study, including the original telemetry logs and analysis CSV files, are available from the corresponding author on reasonable request.

Competing interests

The author declares that he has no competing interests.

Funding

This research was self-funded by the author.

Authors' contributions

TT conducted the research and prepared the manuscript.

Acknowledgements

The author acknowledges the assistance of ChatGPT (OpenAI) in the activities described in the Methods section.

References

1. Takaya T, Kawamura H, Yamamoto M, Ohuchi A (2008) Autonomous indoor blimp robot motion design based on state value. Ricoh Technical Report 34:119–130.
2. Takaya T, Kawamura H, Yamamoto M, Ohuchi A (2008) Control of indoor balloon robot with time constraints. IEEJ Transactions on Electronics, Information and Systems 128(9):1455–1461.
3. Wang W, Ma H, Xia M, Weng L, Ye X (2013) Attitude and altitude controller design for quad-rotor type MAVs. Mathematical Problems in Engineering 2013:587098. <https://doi.org/10.1155/2013/587098>
4. Koksall N, Jalalmaab M, Fidan B (2019) Adaptive linear quadratic attitude tracking control of a quadrotor UAV based on IMU sensor data fusion. Sensors 19(1):46. <https://doi.org/10.3390/s19010046>
5. Eslamiat H, Wang N, Hamrah R, Sanyal AK (2022) Geometric integral attitude control on $SO(3)$. Electronics 11(18):2821. <https://doi.org/10.3390/electronics11182821>

6. ArduPilot Development Team. SITL Simulator (Software in the Loop). ArduPilot Developer Documentation. <https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html>. Accessed 31 Aug 2026.
7. Open Robotics. Gazebo Documentation. <https://gazebo.org/docs/>. Accessed 31 Aug 2026.
8. ArduPilot. iris_with_standoffs model.sdf. ardupilot_gazebo. https://github.com/ArduPilot/ardupilot_gazebo/blob/main/models/iris_with_standoffs/model.sdf. Accessed 1 Sep 2026.
9. ArduPilot. gimbal_small_3d model.sdf. ardupilot_gazebo. https://github.com/ArduPilot/ardupilot_gazebo/blob/main/models/gimbal_small_3d/model.sdf. Accessed 1 Sep 2026.
10. ArduPilot. iris_with_gimbal model.sdf. ardupilot_gazebo. https://github.com/ArduPilot/ardupilot_gazebo/blob/main/models/iris_with_gimbal/model.sdf. Accessed 1 Sep 2026.
11. T-MOTOR. Air Gear 450 II Multi-Rotor UAV Combo / AIR2216II-KV920 and T1045 test data. <https://store.tmotor.com/jp/product/air-gear-450II-combo-set-no-esc.html>. Accessed 1 Sep 2026.

Figures

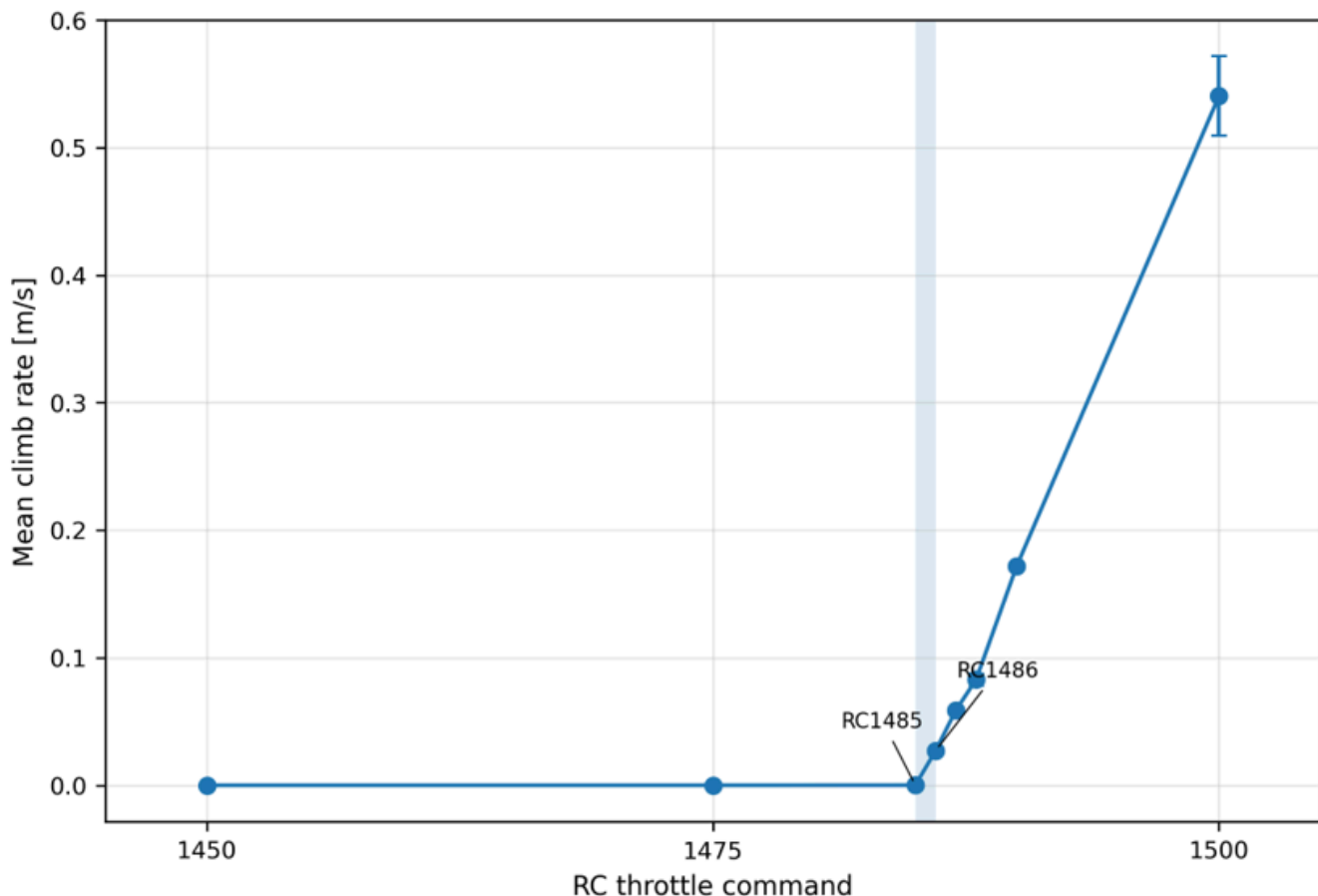


Figure 1

RC throttle command and mean climb rate. The shaded band marks the observed integer-command boundary between RC1485 and RC1486. Summary values are listed in Table 2.

Orthographic projection of the quadrotor model used in the present simulation

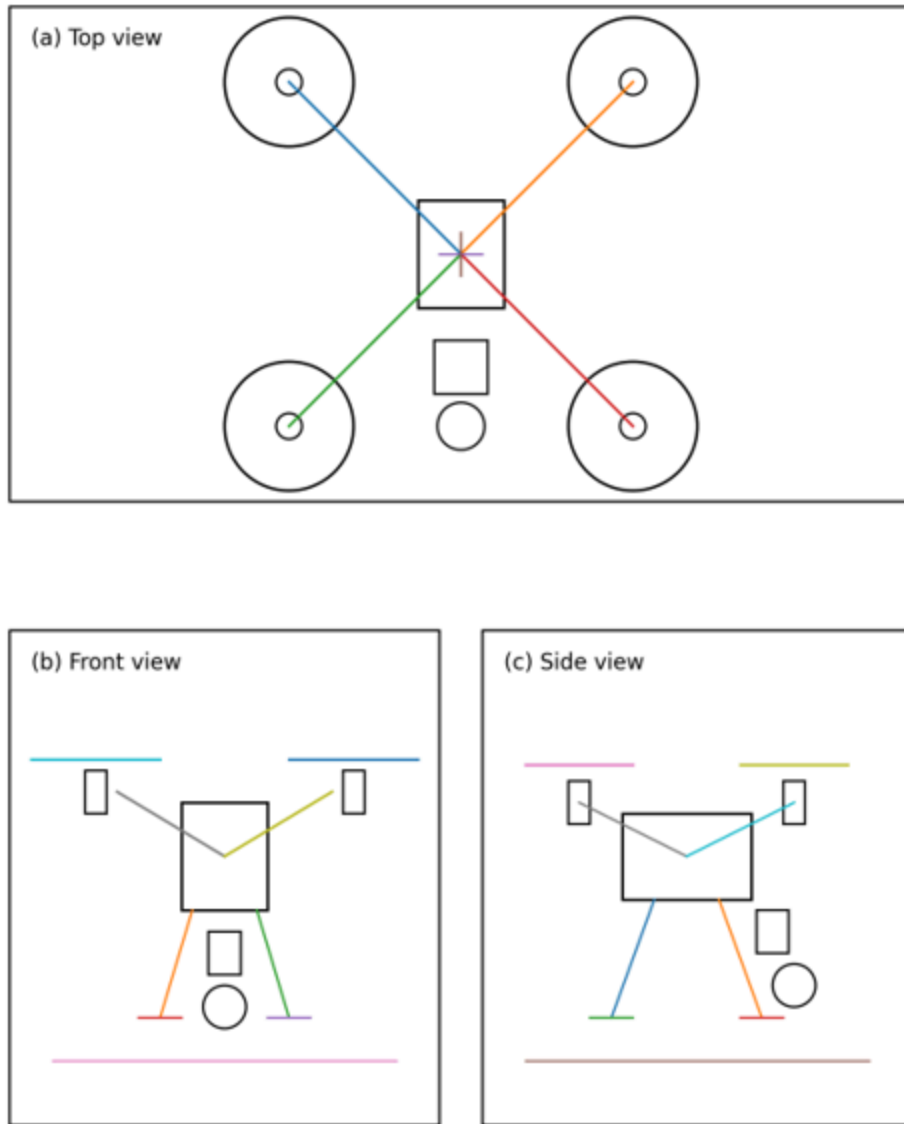


Figure 4

Schematic orthographic views of the simulated quadrotor. (a) Top view. (b) Front view. (c) Side view. The views illustrate the model configuration and are not dimensionally exact. Model mass is obtained from SDF link masses, not from this drawing.