

Flight Control System for NASA’s Mars Helicopter

Håvard Fjær Grip^{*,*}, Johnny N. Lam^{*,†}, David Bayard^{*,‡}, Dylan T. Conway^{*,§}, Gurkirpal Singh^{*,¶}, Roland Brockers^{*,||}, Jeff Delaune^{*,**}, Larry Matthies^{*,††}, Carlos Malpica^{*,‡‡}, Travis Brown^{*,§§}, Abhinandan Jain^{*,¶¶}, Miguel San Martin^{*,***}, and Gene Merewether^{*,†††}

^{*}Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109

^{**}NASA Ames Research Center, Moffett Field, California 94035

As part of NASA’s Mars 2020 rover Mission, an autonomous small-scale helicopter will be sent to Mars to conduct a series of demonstration flights, to validate the feasibility and utility of using helicopters for Mars exploration. In this paper, we present a high-level overview of the flight control system for the Mars Helicopter, including the Guidance, Navigation, and Control subsystems, and the implementation of these on the flight avionics hardware. We also discuss the concept of operations, and the testing, verification, and validation performed in a variety of test venues in preparation for Mars flight.

I. Introduction

The National Aeronautics and Space Administration (NASA) recently announced plans to send a helicopter to Mars as part of the *Mars 2020* rover mission. Known as the *Mars Helicopter*, it will conduct a series of flights intended to demonstrate the feasibility and utility of using helicopters for future Mars exploration.

The use of helicopters has the potential to transform Mars exploration by adding an *aerial* dimension to the capabilities available from orbiters, landers, and rovers. The ability to quickly traverse difficult terrain and take photographs from a high vantage point makes helicopters an ideal forward reconnaissance platform, and future helicopters may also carry their own science payloads to areas that are otherwise inaccessible.

The Mars Helicopter technology represents years of development at NASA’s *Jet Propulsion Laboratory* (JPL) together with partners at *NASA Ames Research Center*, *NASA Langley Research Center*, and *AeroVironment, Inc.*, to overcome the many challenges of flying an aerial vehicle on Mars. From the point of view of flight control, these challenges include

- an extremely thin atmosphere compared to Earth, which, together with reduced gravity, significantly alters the helicopter flight dynamics
- a lack of global navigation aids, such as GPS or a strong magnetic field
- large communication time lag between Earth and Mars, which makes real-time communication during flight impossible



Fig. 1 Artist’s concept of the Mars Helicopter

^{*}Mars Helicopter GNC and Aerodynamics Lead/Robotics Technologist, Guidance and Control Section, JPL; Havard.F.Grip@jpl.nasa.gov

[†]Guidance and Control Engineer, Guidance and Control Section, JPL; Johnny.N.Lam@jpl.nasa.gov

[‡]Senior Research Scientist, Guidance and Control Section, JPL; david.bayard@jpl.nasa.gov; AIAA Associate Fellow

[§]Guidance and Control Engineer, Guidance and Control Section, JPL; Dylan.T.Conway@jpl.nasa.gov

[¶]Guidance and Control Engineer, Guidance and Control Section, JPL; gurkirpal.singh@jpl.nasa.gov

^{||}Robotics Technologist, Mobility and Robotic Systems Section, JPL; Roland.Brockers@jpl.nasa.gov

^{**}Robotics Technologist, Mobility and Robotic Systems Section, JPL; Jeff.H.Delaune@jpl.nasa.gov

^{††}Senior Research Scientist, Mobility and Robotic Systems Section, JPL; lhm@jpl.nasa.gov

^{‡‡}Aerospace Engineer, NASA Ames Aeromechanics Office; carlos.a.malpica@nasa.gov; AIAA Member

^{§§}Robotic Systems Engineer, Mobility and Robotic Systems Section, JPL; Travis.L.Brown@jpl.nasa.gov

^{¶¶}Senior Research Scientist, Mobility and Robotic Systems Section, JPL; Abhi.Jain@jpl.nasa.gov

^{***}Chief Engineer of Guidance and Control Section, JPL; miguel@jpl.nasa.gov

^{†††}Robotics Technologist, Mobility and Robotic Systems Section, JPL; Gene.B.Merewether@jpl.nasa.gov

- a harsher radiation environment than on Earth, which can leave computing elements susceptible to upsets

In previous papers the authors have presented aspects of the flight dynamics [1] and the guidance and control design [2] for a preliminary Mars Helicopter concept, including a test campaign that culminated in the first helicopter flight in Martian atmospheric conditions. Since then, the concept has matured into a complete flight design, and fully functional engineering development models have been built and undergone both environmental and flight dynamics testing. The goal of this paper is to present an overview of the Mars Helicopter flight control system in its near-final state, with a particular focus on the Mars-specific design drivers and the choices made to overcome the challenges mentioned above.

II. Previous Work

Aerial vehicles for Mars have been considered at a conceptual level for decades, both fixed-wing designs and, later, rotorcraft. An early study of the implications of the Mars environment on rotorcraft design was performed by Savu and Trifu [3]. Stanford University and JPL performed a proof-of-concept test of a small rotor in Mars atmospheric conditions in 2000, as an early precursor to the current Mars Helicopter [4]. Around the same time, a number of papers were published on the technical challenges of Mars rotorcraft [5–8], and an AHS International Student Design Competition was conducted on the topic [9, 10]. Further work on the topic was conducted over the next decade at NASA Ames [11–15] and elsewhere [16, 17].

In 2014, inspired in part by JPL research on autonomous navigation of miniature terrestrial rotorcraft [18], Balaram and Tokumaru [19] presented the initial concept that would eventually evolve into the current Mars Helicopter. A major milestone in the development of this helicopter was the demonstration of controlled flight in Mars atmospheric conditions, conducted in JPL's 25-ft Space Simulator in 2016. This effort involved the design and construction of a demonstration vehicle with a full-scale rotor. Grip et al. [1] describes the modeling and identification of the flight dynamics for this vehicle, while Grip et al. [2] describes the guidance and control design.

The full flight design for the Mars Helicopter is described at a high level by Balaram et al. [20]. An aerodynamic model of the Mars Helicopter rotor for comprehensive analysis has been developed by Koning et al. [21]. An in-depth description of the Mars Helicopter navigation algorithm is given by Bayard et al. [22]. The design and fabrication of the rotor system, airframe, and landing gear is described by Pipenberg et al. [23].

III. Mission Overview

The Mars Helicopter mission is a *technology demonstration* intended to show the feasibility and utility of helicopters for Mars exploration. During flight to Mars, the helicopter will be attached underneath the *belly pan* of the Mars 2020 rover, with the landing gear partially folded. After landing on Mars, the helicopter will be deployed in an area deemed suitable for helicopter operations, and the rover will drive away to a safe distance.

After deployment, the helicopter will conduct a series of autonomous flights lasting approximately 1.5 min each. During each flight, the helicopter will take off, climb, hover, and optionally translate between a set of waypoints, before descending and landing again. Although the helicopter is entirely autonomous during flight, the waypoints will be specified from Earth prior to flight. Weather conditions on Mars will be monitored from Earth, and flights will only be commanded at times when conditions are predicted to be sufficiently benign. Because landing hazard avoidance was not prioritized for this technology demonstration, each flight will begin and end within an area that has been pre-inspected and determined to be safe in terms of obstacles and ground slope; in particular, initial deployment will occur within a 10×10 m area satisfying the requirements for takeoff and landing (see Section IV).

As will be further discussed in Section IX, a visual-inertial navigation system was developed with emphasis on robustness, but with correspondingly limited position accuracy and ability to navigate in complex terrain. In particular, the system assumes that features observed by the navigation camera lie on an approximate *ground plane* with known slope. Because of this, flights will be conducted over relatively

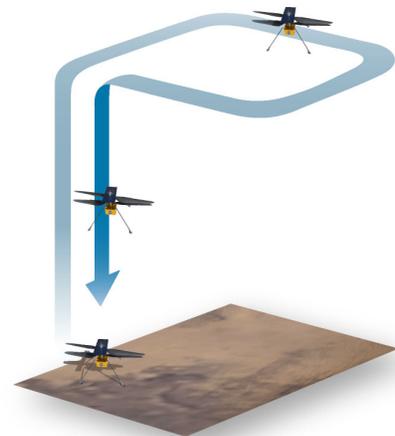


Fig. 2 Illustration of a Mars Helicopter flight, beginning and ending in the same pre-inspected safe area

flat terrain, with short-term height variations on the order of 10% of the flight height.

A summary of assumed environmental parameter values for Mars flight is given in Table 1.

Table 1 Key environmental parameters and assumed values

Parameter	Range
Atmospheric density	0.014–0.02 kg/m ³
Horizontal wind	≤ 9 m/s
Vertical wind	≤ 2 m/s
Gust component	≤ 3.5 m/s
Landing area slope at 0.6-m baseline	≤ 10°

*Approximate value

During and after each flight, the helicopter will transmit telemetry to the rover for relay back to Earth. This includes high-resolution imagery from a forward-looking 13-Megapixel color camera. Between flights, the helicopter’s onboard batteries will recharge through a solar panel mounted above the rotors.

IV. Vehicle Overview

The Mars Helicopter is a coaxial helicopter, with two counter-rotating 2-bladed rotors. The rotor is hingeless, meaning that the blades are rigidly mounted on the hub, except for bearings that allow for adjustment of blade pitch via a *swashplate* control mechanism.

Compared to Earth, the rotors are heavily oversized relative to the rest of the vehicle. This is due to the extremely low density (approximately 1–2% of Earth’s) and to a lesser extent the reduced speed of sound (approximately 76% of Earth’s), which reduce the amount of thrust that can be produced by a fixed-size rotor by two orders of magnitude. This is partially compensated for by the reduced gravity on Mars, which is 38% of Earth’s.

To minimize the chance of tipover on landing, the landing gear is designed with a large footprint, and each leg is equipped with a flexure and an aluminum damper to help soften the touchdown and prevent the vehicle from bouncing. Key properties of the vehicle, including physical dimensions and mass, are given in Table 2. The properties of the landing gear define quantitative requirements on the flight control system on touchdown, which are summarized in Table 3.



Fig. 3 CAD drawing of the Mars Helicopter

A. Actuation

The vehicle features a total of eight motors: two brushless direct-drive propulsion motors driving the two rotors, and six brushed servo motors for controlling blade pitch via a swashplate on each rotor. The propulsion motors are controlled to keep the rotor speed constant, at a setpoint chosen depending on the atmospheric density at the time of flight. The motion of the vehicle is controlled by modulating the blade pitch, which affects the amount of lift and drag produced by the blade. The swashplate mechanism allows for two types of blade pitch control: *collective control* changes the average pitch over a rotation, and *cyclic control* allows for periodic modulation of the blade pitch at a frequency of once per revolution, with a specified phase and magnitude.

Table 3 Touchdown requirements

Variable	Requirement
Vertical speed	≤ 2 m/s
Horizontal speed	≤ 0.5 m/s
Roll/pitch angle	≤ 15°
Angular rate	≤ 40 °/s

Table 2 Summary of key vehicle properties

Total mass*	1.8 kg
Rotor diameter	1.21 m
Rotor spacing	0.1 m
Ground clearance* (lower rotor)	0.3 m
Landing gear footprint*	0.6×0.6 m
Thrust-to-weight ratio* (density-dependent)	130% to 160%
Endurance	≥ 1.5 min
Rotor speed	≤ 2800 RPM
Collective control (both rotors)	-4.5° to 17.5°
Cyclic control (both rotors)	$\pm 10^\circ$

*Approximate value

Compared to the demonstration vehicle presented in [2], the actuation differs in two significant ways. First, the demonstration vehicle featured cyclic control only on the lower rotor. The addition of upper cyclic control provides more control authority in roll and pitch for handling the weather on Mars, as well as greater symmetry between the rotors. Second, the demonstration vehicle featured a single propulsion motor driving both rotors via a set of gears. The switch to two direct-drive motors is beneficial in terms of efficiency, robustness, and reduced gear noise, but results in greater cross-coupling between yaw and motor speed. This must be handled as part of the control architecture, as discussed further in Section X.

B. Navigation Sensors

The demonstration vehicle presented in [2] made use of a *Vicon* tracking system, consisting of a set of external cameras used to track features on the vehicle. For Mars flight, navigation is performed using the following onboard sensors:

- a Bosch Sensortech BMI-160 *inertial measurement unit* (IMU), for measuring 3-axis accelerations at 1600 Hz and angular rates at 3200 Hz
- a Garmin Lidar-Lite-V3 *laser rangefinder* (LRF), for measuring distance to the ground at 50 Hz
- a downward-looking 640×480 grayscale *camera* with an Omnivision OV7251 global-shutter sensor, providing images at 30 Hz
- a muRata SCA100T-D02 *inclinometer*, for measuring roll and pitch attitude prior to flight

These are commercial off-the-shelf (COTS) miniature sensors, largely developed for the cell phone and lightweight drone markets. The navigation algorithm using these sensors is described in Section IX.

C. Avionics and Flight Software Architecture

The Mars Helicopter avionics architecture must be lightweight and robust enough for flight on Mars, while at the same time providing enough computational throughput to process navigation images with a 30 Hz framerate and robustly stabilize the fast dynamics of the vehicle. This is accomplished by the architecture shown in Figure 4. A radiation-tolerant FPGA acts as a central switchboard, routing sensor data and traffic between other computing elements and performing low-level control of the actuators. Most of the flight control software is hosted on the *Flight Computer* (FC), which consists of an ARM Cortex-R5 microcontroller with dual-core lockstep functionality. The lockstep functionality enables cycle-by-cycle detection of errors that might arise from radiation-induced upsets. If an error is detected, the FPGA will switch to a spare version of the same microcontroller, thus allowing for uninterrupted operation of the most critical flight control functions.

To provide the computational throughput for vision-based navigation, a separate *Navigation Computer* (NC), consisting of a 2.26 GHz quad-core Qualcomm Snapdragon 801 processor, is used. On the NC, one core is devoted to camera image processing and another to the navigation filter, while the remaining cores are used for other activity, such as logging, radio communication, and command and data handling. Although the NC provides a large amount of data and computing throughput, it runs a non-realtime Linux operating system and communicates with the rest of the system through a UART

connection without realtime and reliability guarantees. As will be further discussed in Section V, the flight control system must be designed to be robust against non-realtime behavior and possible communication dropouts related to the NC.

The flight control software is implemented in C++ and is structured as a set of callbacks within the *F'* flight software framework. *F'* was developed at JPL for use on small-scale systems, and provides a framework for execution, synchronization, and communication between multiple rate groups, threads, and processes on separate hardware units, as well as mechanisms for parameter handling, logging, and event recording. For the vision processing component residing on the NC, the flight control software makes explicit use of the ARM NEON *Single Instruction, Multiple Data* (SIMD) framework, which allows simultaneous operations on up to 16 eight-bit pixels by using a set of 128-bit hardware registers.

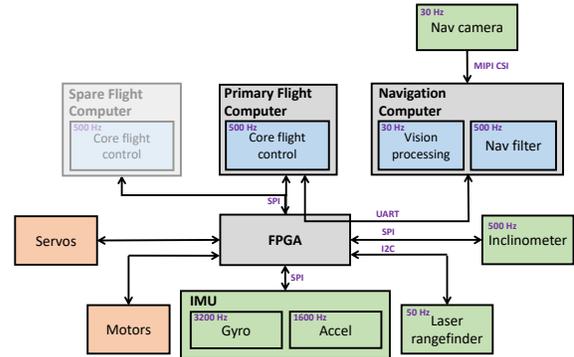


Fig. 4 The Mars Helicopter avionics architecture

D. Engineering Development Models

Two engineering development models (EDMs) of the Mars Helicopter have been built for the purpose of testing, verification, and validation. One of these, EDM-1, has been used primarily for aerodynamics and flight control testing, and the other, EDM-2, for environmental and deployment testing. Figure 5 shows EDM-1 during a recent flight test inside JPL’s 25-ft Space Simulator, using only onboard sensors for navigation. The view from the navigation camera can be seen in the upper left-hand corner.

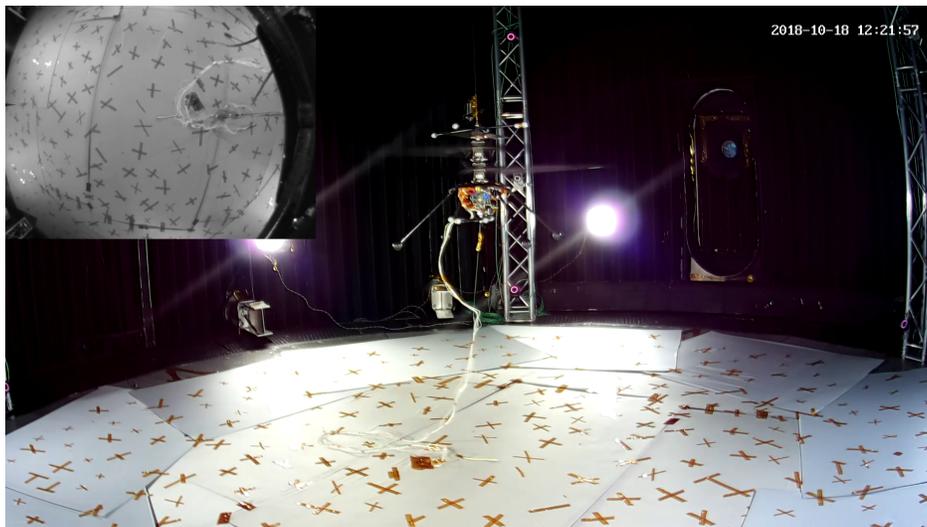


Fig. 5 EDM-1 during a recent flight test, using only onboard sensors for navigation. The view from the navigation camera can be seen in the upper left-hand corner.

V. Implementation on Flight Avionics

Once on Mars, the helicopter must survive and operate on its own, without any possibility of physical intervention. As such, the overarching concern is to satisfy the high-level functional mission requirements outlined in Section III with a maximum level of robustness—to environmental variations and disturbances (e.g., atmospheric density, temperature, wind, dust, terrain characteristics, lighting), as well as possible intermittent glitches and faults within the flight control system (e.g., communication dropouts, non-realtime behavior, radiation-induced upsets).

The flight control system can be divided into four main subsystems, as illustrated in Figure 6: the *Mode Commander*, which sets the overall mode for the flight control system, the *Guidance* subsystem, which provides

reference trajectories for the flight path, the *Navigation* subsystem, which provides an estimate of the vehicle state, and the *Control* subsystem, which commands the actuators based on the reference trajectories and the vehicle state.

Like almost all helicopters, the Mars Helicopter is open-loop unstable [1]. Because of this, closed-loop control is required at all times during flight, and ensuring uninterrupted operation of the Control subsystem is a strict requirement. The small scale of the vehicle naturally drives the dynamics to a fast time scale, with an inner-loop crossover frequency of 2.5–3.0 Hz [2]. Thus, keeping latencies in the loop at a minimum level is essential to guaranteeing robust stability across the operating envelope.

The Control system receives its state information from the Navigation subsystem; however, as described in Section IV–C, the NC and the communication link with the FC does not have realtime or reliability guarantees. This issue is resolved by splitting the Navigation subsystem between the two computers. Figure 7 illustrates how the flight control components are implemented on the flight avionics, including the flow of sensor, actuator, and state information. Non-realtime components are indicated using dashed lines. A single 500-Hz callback within the FC hosts the Mode Commander, the Guidance subsystem, the Control subsystem, as well as two components of the Navigation subsystem. The first of these components accepts IMU data that arrives at a rate of 3200 Hz (for gyro measurements) and 1600 Hz (for accelerometer measurements); subtracts a temperature-dependent bias; filters the measurements through a low-pass anti-aliasing filter as well as notch filters at harmonics of the rotor speed; and downsamples the result to 500 Hz. (Prior to flight, it also averages inclinometer data and passes the result to the NC to determine the initial attitude on the ground.) The downsampled IMU data is passed to the NC for consumption by the navigation filter; however, it is also passed to a second component of the navigation system residing on the FC. This component is responsible for receiving state estimates from the NC and propagating them to the current time using buffered IMU data.

The key feature of this architecture is that, despite the inherent latency in communication between the FC and the NC, and the non-realtime nature of the NC, the Control subsystem is always guaranteed to execute in a real-time manner using up-to-date state information. The navigation filter produces a full state estimate using all the available sensor information, but, even in the best case, those estimates are delayed by several milliseconds by the time they arrive at the FC. When a state estimate is received by the FC, it is therefore propagated to the current time using buffered IMU data, thereby eliminating the latency associated with the NC. As a result, the driving latency in terms of the closed-loop control is along the path from the IMU, via the FC, to the actuators (illustrated by bold arrows in Figure 7).

A. Avionics Fault Handling

Should there be a gap in the 500-Hz updates from the NC to the FC, the FC will transparently coast through the gap by continuing to propagate the most recent estimate. For best accuracy, the IMU propagation on the FC makes use of the most recent estimates of accelerometer and gyro biases, which are also computed by the navigation filter and communicated together with the state estimate. Nonetheless, the estimate will begin to drift relatively quickly if the outage is persistent, as in the case of an NC fault. Therefore, if 50 consecutive packets from the NC fail to arrive, the FC switches to a *fault response estimator* that combines the IMU-propagated state with LRF measurements in a simple observer. At the same time, an emergency landing attempt is triggered, which will cause the vehicle to touch down within a few seconds.

In case of a fault on the FC, as indicated by the lockstep functionality discussed in Section IV–C, a *hot-swap* to the spare FC will take place. From the beginning of the flight, the spare FC will have been receiving the same inputs as the primary FC and will have been executing the same navigation-related functions; thus, it immediately has access to an up-to-date state estimate. When the hot-swap takes place, the spare FC will also receive a small set of data from the primary FC, which has been securely stored on the FPGA from the last cycle prior to the fault; this includes information

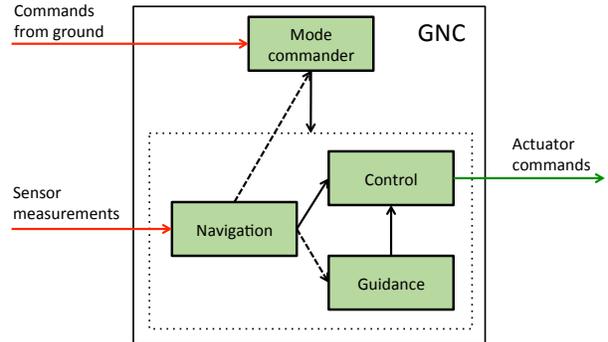


Fig. 6 High-level GNC architecture. Red arrows illustrate information flowing into the system; green arrows illustrate information flowing out of the system; solid black arrows illustrate continuous information flow within the GNC system; dashed black arrows illustrate intermittent information flow within the GNC system.

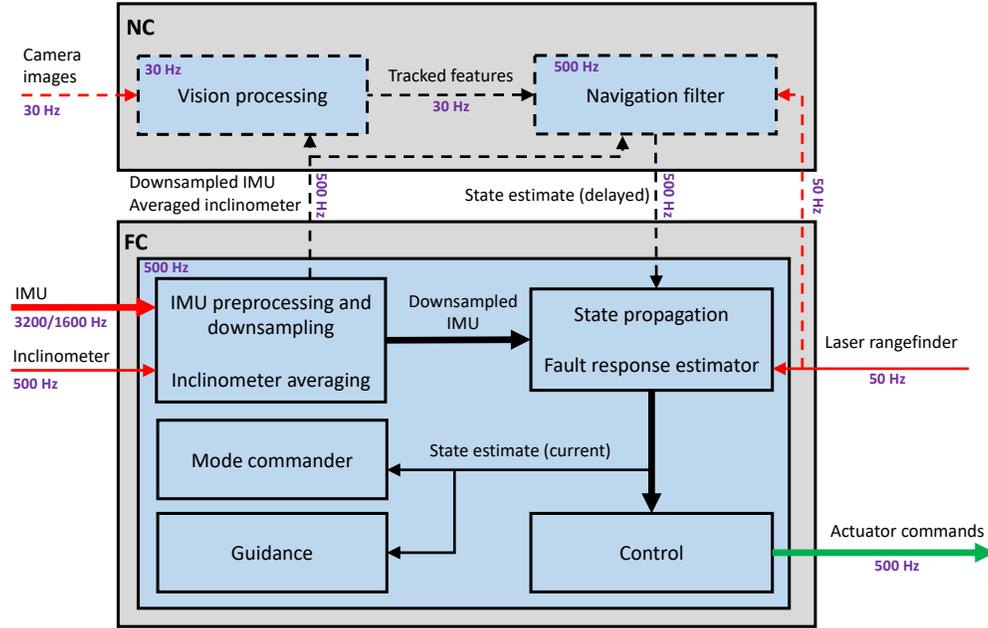


Fig. 7 Illustration of the flight control software implementation on the flight avionics, with the flow of sensor, actuator, and state estimate information. Non-realtime components are indicated with dashed lines.

about the current state of the flight control software, as well as current integrator states from the Control subsystem. This allows the spare to transparently take over and initiate a landing in response to the hot-swap.

VI. Flight Control Concept of Operations

Testing in relevant environments is far more complicated for the Mars Helicopter than for typical Earth-based drones. For this reason, the flight-control concept of operations is designed for simplicity and to minimize the flight envelope.

Each flight begins with a takeoff, followed by a vertical climb at a velocity of 1 m/s, to the flight altitude. Once the top of the climb has been reached, the vehicle begins its waypoint tracking phase. Each waypoint is specified by a position, a translation heading, a dwell time, and a dwell heading. For each waypoint, the vehicle begins by performing an altitude adjustment to the height of the waypoint, if one is needed. The vehicle then slews to the translation heading through a pure rotation about yaw, before translating along a direct line to the waypoint while maintaining the translation heading. At the waypoint, the vehicle comes to a full stop, before slewing to the dwell heading and pausing there for the specified dwell time. This process is repeated for each waypoint until reaching the last one. From the last waypoint, the vehicle descends vertically at a speed of 1 m/s until reaching the ground, at which point control is disabled and the rotors are spun down.

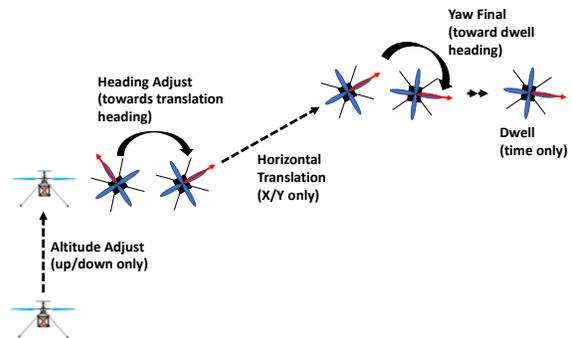


Fig. 8 Illustration of waypoint tracking phases

The nominal flight height for most flights is 5 m, which is well within the range of the LRF. The nominal horizontal translational velocity is up to 2 m/s, which is chosen to keep features on the ground from disappearing too quickly from the field of view. Depending on the expected weather conditions, the translational velocity will be limited to keep the horizontal airspeed below 10 m/s.

A. Takeoff and Landing

Takeoff and landing are the most critical phases of flight, involving several mode transitions and interactions between the Guidance, Navigation, and Control subsystems.

During the initial takeoff, a constant thrust setting is applied, corresponding to a level approximately 20% above the vehicle weight. The purpose of this is to effect a quick separation of the legs from the ground. Full six-degree-of-freedom (6-DOF) control is not applied during this phase, since the legs are initially in contact with the ground; however, limited control is applied to reduce the angular rates of the vehicle. The initial takeoff phase lasts only a short time; once the vehicle has climbed 5 cm, or a 0.4 s timeout has been reached, full control is enabled. From this point on until the end of the flight, the control system tracks a reference trajectory for position, velocity, attitude, and angular rates provided by the guidance subsystem. Dust kickup due to rotor downwash is a concern near the ground, due to potential spoofing effects on the LRF and camera. Outdoor testing of the navigation system has shown no detrimental effects in spite of sometimes significant dust kickup. Nonetheless, for added safety the navigation system does not begin consuming data from the LRF and navigation camera until the vehicle has climbed by an estimated 1 m; until this point, it relies only on IMU integration. Gyro data is integrated from the beginning of the rotor spinup, and accelerometer integration starts immediately before takeoff.

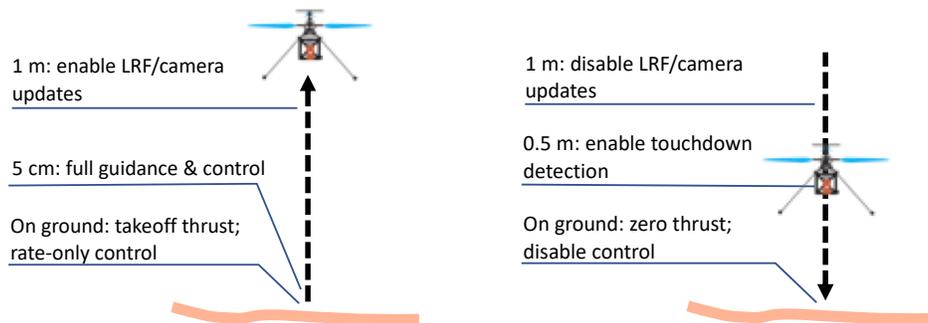


Fig. 9 Illustration of takeoff and landing sequences

On landing, updates from the LRF and camera are again turned off when the vehicle is an estimated 1 m away from ground contact. When the vehicle is an estimated 0.5 m away from ground contact, *touchdown detection* is enabled. Touchdown detection is based on monitoring the vertical velocity control error, to determine whether the landing gear has made contact with the ground. When the control error exceeds a given threshold, indicating that the vehicle is unable to continue along its downward trajectory, all control is disabled and the collective is set to the minimum value of -4.5° on both rotors. Because the navigation system operates on IMU only, potential jumps in the state estimates due to spurious measurement updates are prevented, which helps protect against erroneous touchdown detection.

The landing strategy has been updated compared to the landing strategy used for the demonstration vehicle [2], where the vehicle simply dropped to the ground from an altitude of approximately 10 cm. Although the drop landing method is less complex, it requires good knowledge of the height above ground to prevent the drop from happening too early or too late. With touchdown detection, relatively large errors in altitude can be tolerated, making the system more robust to LRF and IMU bias.

B. Fault response

Because the helicopter has limited endurance, and because it is unstable in open loop, it does not have a true *safe mode* that can be entered in case of an anomaly. Instead, the response to any significant anomaly is to initiate a landing at the current location. This response is not without risk, since the current location may be less suitable than the nominal landing location. Thus, early landing is triggered only by serious anomalies, such as a persistent loss of sensor information or the failure of a computing element.

VII. Modeling, Simulation, and System Identification

The main tool for flight control-related modeling and simulation of the Mars Helicopter is called *HeliCAT*, and was built based on the *Darts/Dshell* multibody simulation framework developed at JPL (see, e.g., [24]). As described in [1],

HeliCAT is used extensively for flight dynamics analysis and control design, and features a detailed rotor model that includes aerodynamic forces from CFD-generated airfoil tables, dynamic inflow, and structural flexibility represented by hinges with springs and dampers. In addition, HeliCAT includes features that enable time-domain simulations in a variety of setups, as well as end-to-end flight simulations. Key features for flight simulation include

- generation of terrain with randomized topography, arbitrary overlaid textures, and adjustable lighting
- ground contact dynamics between terrain and landing gear
- actuator and sensor models, including a camera model that can replicate the properties of the navigation camera and an LRF model for measuring distance to the terrain
- simulation of arbitrary wind and gust disturbances
- flight software integration

Figure 10 shows an example screenshot from a simulated flight, with the navigation camera image in the bottom right-half corner.



Fig. 10 Screenshot from a simulated flight in HeliCAT. The view from the navigation camera is shown in the bottom right corner.

Using the above features and others, HeliCAT is used to perform Monte-Carlo simulations of end-to-end flights for comparison with requirements. Figure 11 shows the simulated trajectories for 247 flights that include a short lateral displacement, with random variation in simulation parameters including

- atmospheric density and wind disturbances
- terrain topography, texture, and surface contact properties
- actuator properties, such as servo bandwidth and propulsion motor characteristics
- sensor misalignments and properties such as camera focal length, image blurring, and IMU and LRF bias
- helicopter mechanical properties, such as landing gear damping
- latencies due to non-realtime properties of avionics related to the NC

Also shown in Figure 11 are horizontal and vertical velocities, as well as the norm of the roll/pitch angle and angular velocities, which can be compared to the requirements in Table 3. The parameter ranges are generally chosen to be bounding, and in particular include more terrain variation than expected. This affects both the spread in vertical position (because the vertical position estimate is based on the LRF, it is ground-relative), and in horizontal position (because the ground-plane assumption is violated). Nonetheless, the system functions robustly, with the vehicle state at touchdown well within the requirements. For this set of runs, wind conditions were taken from a dataset produced by atmospheric simulation. The winds are in many cases stressing in terms of the magnitude of the low-frequency contents and variation

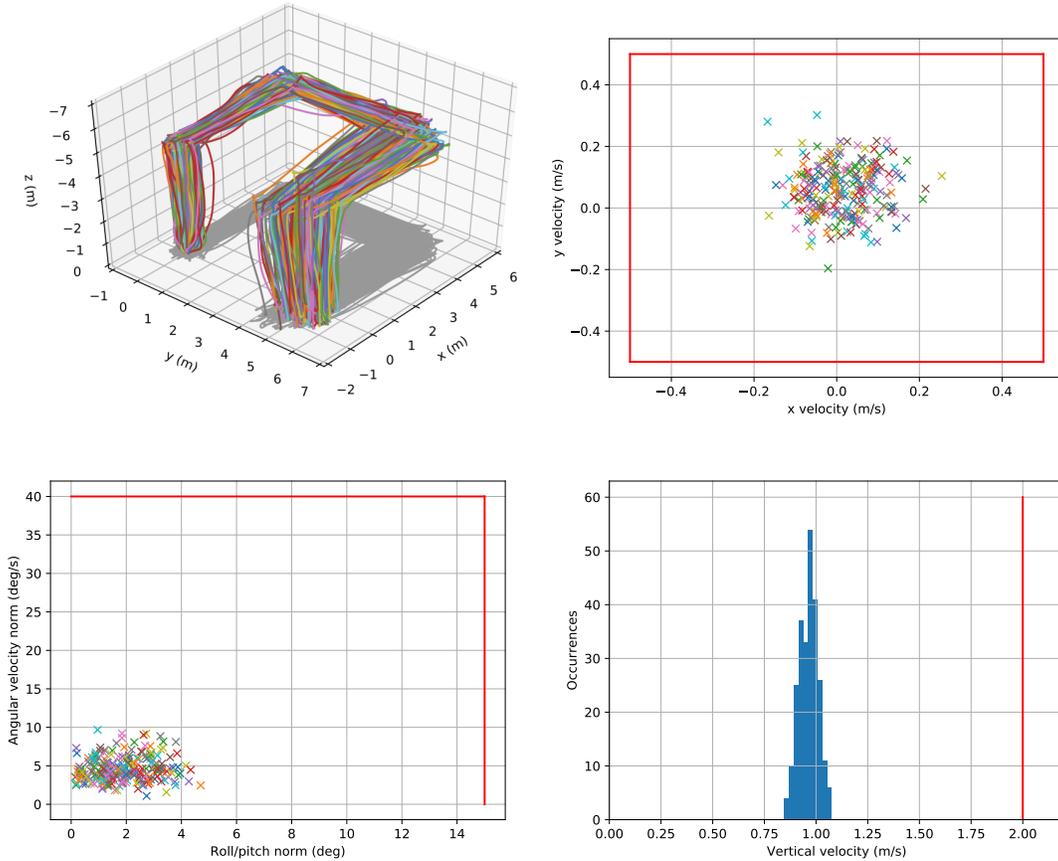


Fig. 11 Clockwise from top left: simulated trajectories, horizontal velocities at touchdown, vertical velocities at touchdown, and norm of roll/pitch and angular velocities at touchdown, for a Monte-Carlo simulation consisting of 247 flights. The touchdown requirements from Table 3 are indicated by red lines.

up to approximately 0.25 Hz. However, the data set lacks richness at higher frequencies, and as a result, the spread in roll/pitch and angular velocity magnitudes tends to be underestimated in this set of simulations.

A. System Identification

Helicopter dynamics is difficult to model due to the complex aerodynamics and interactions with the structural dynamics of the rotor. Mars Helicopter dynamics is particularly challenging due to the lack of prior results. As a result, first-principles modeling cannot be relied upon to provide a fully accurate model for control design, and instead, the actual dynamics of the vehicle must be identified.

Grip et al. [1] describes a process of *system identification* for the early demonstration vehicle, using a combination of three configurations at Mars atmospheric density: a *locked-down* configuration, where the helicopter controls are exercised while the helicopter is attached to force-torque sensor; a *swinging-arm* configuration, where the helicopter is moved by an actuated arm to generate airflow over the rotor; and a *gimbal* configuration, where the helicopter is free to rotate in roll and pitch.

Because helicopter flight dynamics tends to change as the helicopter transitions to from *hover* to *forward flight*, system identification in forward flight is also necessary.* This was recently accomplished through a similar set of experiments as in [1], using a set of approximately 900 small commercial fans to generate wind speeds up to approximately 11 m/s around the rotor. Experiments were performed with the vehicle in a level orientation and with the vehicle tilted to provide a combination of horizontal and vertical flow. Key risks addressed in the forward-flight system identification

*The term *forward flight* refers to the helicopter moving horizontally relative to the surrounding air, whether due to wind or due to helicopter translation.

campaign were

- sensitivity of pitch moments to vertical airspeed perturbations (high sensitivity could lead to a fast pitch-heave instability)
- variation in dynamic model parameters between hover and forward flight (large variation could lead to difficulty designing a single fixed-gain controller for the whole flight envelope)
- cross-axis coupling in forward flight (large cross-axis coupling could have adverse effects on stability margins)

With respect to these risk factors, the test results were in line with expectations based on prior modeling. Based on the identified parameters from the campaign, the roll/pitch inner loop controller was retuned to optimize gain and phase margins, but further design changes were not required.

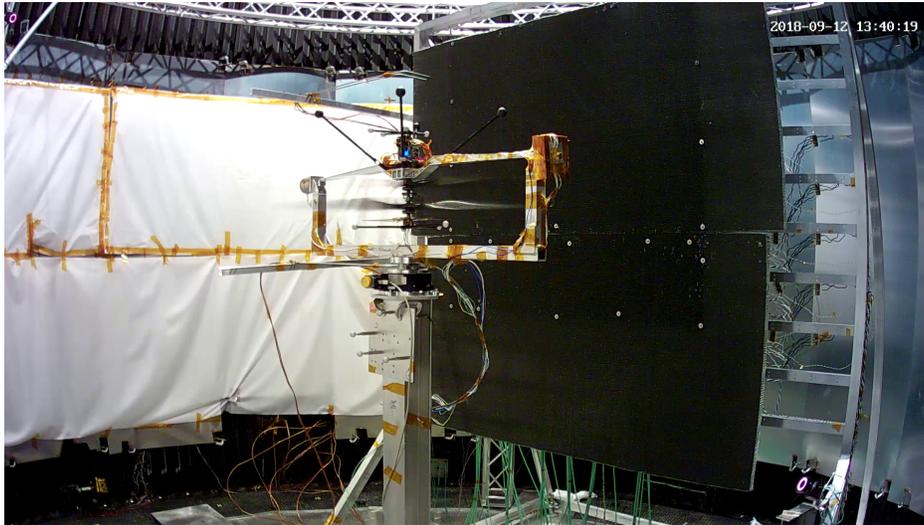


Fig. 12 EDM-1 during a system identification experiment in front of the *wind wall* inside JPL’s 25-ft Space Simulator. The helicopter is mounted upside-down on a pitch gimbal, and is under closed-loop control to maintain stability. A cyclic frequency sweep is applied to induce pitch variation.

VIII. Mode Commanding and Guidance

The Mode Commander is implemented as a finite state machine, in which the states define the current operating mode of the flight control system. The Mode Commander interacts with a smaller state machine within the Navigation subsystem, which defines the operating mode of the navigation filter.

The Mode Commander states are as follows:

- *Idle*: do nothing
- *Inclinometer averaging*: gather and average inclinometer data, to pass to the Navigation subsystem
- *Initialize estimator*: wait for Navigation to finish initialization
- *Spinup*: rotor spinup to setpoint speed
- *Takeoff preparation*: wait for Navigation to start full 6-DOF IMU integration
- *Takeoff*: climb to 5 cm altitude or until timeout reached, with rate-only control
- *Climb*: climb to flight altitude with full control
- *Waypoint tracking*: translate between waypoints
- *Descent*: descend while maintaining horizontal position
- *Landing*: continue descent with touchdown detection armed
- *Touchdown*: disable control, set collectives to lowest setting
- *Spindown*: spin down rotors

The states of the Navigation subsystem are as follows:

- *Estimator idle*: do nothing
- *Estimator initialization*: gather and average IMU data; determine initial attitude and bias estimates
- *Estimator spinup*: integrate gyros

- *IMU ascent*: integrate gyros and accelerometers
- *Free-running*: full filter operation with LRF and camera measurements
- *IMU descent*: integrate gyros and accelerometers

The states and the transitions of the Mode Commander and the Navigation state machines are illustrated in Figure 13. Also shown in Figure 13 is a state machine within the Guidance subsystem, which is active during the *Waypoint tracking* mode to manage the maneuvers associated with each waypoint.

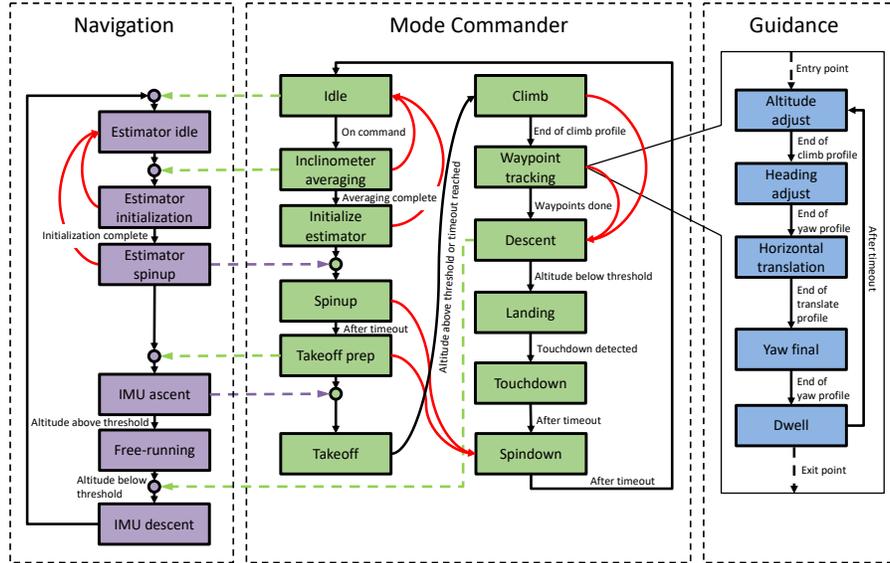


Fig. 13 Illustration of Mode Commander and state machines in the Navigation and Guidance subsystems. The dashed arrows between the Mode Commander and the Navigation subsystem indicate points at which a transition is blocked until a certain state has been reached in the other subsystem. The red arrows indicate transitions triggered in case of a fault.

The Guidance subsystem provides reference trajectories and actuator feedforward signals for the Control subsystem. As described in [2], the guidance algorithm is based on a simplified reference model that treats the helicopter as a thrust vector that can be tilted to produce horizontal accelerations. For translation (or yaw), trajectories are generated for position (or heading), velocity, acceleration, and jerk, while respecting limits on velocity, acceleration, and jerk. In the case of horizontal translation, the acceleration and jerk profiles are translated into equivalent profiles for roll/pitch and roll/pitch rate. Except for profiling of the rotor speed, Guidance is only activated once the Mode Commander enters *Climb*, at which point it is active until *Touchdown* is entered.

IX. Visual-Inertial Navigation

The primary purpose of the navigation system is to provide real-time estimates of the vehicle state—specifically, position, velocity, attitude, and angular rates—to the control system. The state estimate is based on fusing information from the onboard IMU, inclinometer, LRF, and navigation camera. In the following, a high-level description of the navigation algorithm is given; see [22] for a more in-depth description.

A. Principle of Operation

Prior to flight, when the vehicle is at rest on the ground, the inclinometer is used to estimate the initial roll and pitch attitude. Based on this, initial estimates of the accelerometer and gyro biases are also obtained.

Once the vehicle is in motion, integration of the IMU measurements is used to estimate changes in position, velocity,

and attitude. Due to the low accuracy inherent in MEMS-based IMUs, however, additional navigation aids are needed to bound the growth in the navigation errors. The LRF provides range measurements between the vehicle and terrain, which provides observability of vertical velocity and position. The navigation camera is used to track visual features on the ground, under the assumption that all of the features are located on a ground plane with a known slope. This provides observability of horizontal velocity as well as roll and pitch attitude, and helps limit the drift in horizontal position and yaw angle. Nonetheless, no absolute reference for horizontal position and yaw angle is available, and these estimates are subject to long-term drift. Therefore, shortly before touchdown at the end of each flight, a navigation camera image is stored for later transmission on Earth, so that an absolute position and heading fix can be obtained by comparison to the known terrain.

B. MAVeN Algorithm

The MAVeN navigation algorithm used for the Mars Helicopter was originally developed as part of a JPL research project on comet exploration [25], and then adapted for the Mars Helicopter [22]. The MAVeN algorithm is based on the tracking of visual features, which are first identified by a *feature detector* in a *base image*. Using a camera calibration model, the image-space pixel location of each feature can be converted to a unit vector from the vehicle to a patch of terrain. Given an estimate of position and attitude at the time of the base image, the position of the terrain patch can be estimated by intersecting the unit vector with an assumed ground plane. The features identified in the base image are tracked in subsequent *search images* using a *feature tracker* and compared to their predicted image-space locations based on the sensed vehicle motion since the last image. The difference between the prediction and the location determined by the feature tracker provides an innovation that can be used to correct the estimated state.

MAVeN is implemented as an Extended Kalman Filter (EKF) that generates and makes use of the innovations described in the previous paragraph, as well as innovations from the difference between the predicted and measured LRF range. The Mars Helicopter implementation of MAVeN has a state vector with seven components: position, velocity, attitude, IMU accelerometer bias, IMU gyro bias, base image position, and base image attitude, for a total of 21 scalar components. The last two components are referred to as *clone states*, because they are copies of other state components obtained at the time of a base image. These are needed in the state vector because the visual feature innovations depend on both the position and attitude at the time of a search image *and* at the time of the corresponding base image. The IMU is used to integrate the non-clone states, and when an LRF measurement or search image measurement is received, an EKF update occurs to correct the full state.

MAVeN only tracks features between the current search image and the base image. Because the base frame is frequently reset as features are lost, MAVeN is effectively a long-baseline visual odometry algorithm: the relative position and attitude between the two images are measured, but not the absolute position and attitude. A typical consequence of visual odometry algorithms is that the absolute position and attitude error can grow over time. In the implemented navigation system, this is true for horizontal position and yaw. The LRF provides direct observability of vertical position which bounds vertical position error. In addition, the visual features and flat plane assumption provide observability of absolute pitch and roll when the vehicle is moving (but not when the vehicle is stationary). Although absolute pitch and roll are not directly measurable in hover flight, the growth in pitch and roll error is much slower than the gyro angle-random-walk rate. This is because rotational and translational motion are coupled in the IMU integration equations. The result is that pitch and roll error grow at a rate proportional (by the inverse of gravity) to the accelerometer bias growth rate.

A key advantage of MAVeN over other *Simultaneous Localization and Mapping* (SLAM) algorithms is that the state only needs to be augmented with six scalar elements—three for position and three for attitude. Other SLAM algorithms have a state augmentation scheme that scales with the number of features tracked (see [26, 27]). The use of an LRF and an assumed ground plane enables MAVeN to estimate 3D position and velocity without introducing a scale ambiguity, resulting in reduced computational cost and improved numerical conditioning. Furthermore, unlike some loosely coupled architectures (see, e.g., [28]), MAVeN immediately benefits from feature data once it begins receiving images, without requiring a dedicated initialization step of the vision subsystem. Finally, unlike many other methods (see, e.g., [29]), MAVeN does not require vehicle motion to maintain observability, which makes it a good fit for a hovering helicopter.

The two main disadvantages of MAVeN are sensitivity to rough terrain, due to the ground-plane assumption, and long-term drift in position and heading. For the Mars Helicopter technology demonstration, this is an acceptable tradeoff, because accuracy degradation is graceful and the algorithm has proven to be highly robust in both simulation and experiments.

C. Feature Detection and Tracking

Feature detection in base images is performed with an implementation of the *FAST algorithm* [30], which selects corner-like features that have sufficient contrast between a center pixel and a contiguous arc surrounding the center pixel. From an initial large set of identified features, the 28 highest-contrast features within each of nine equally sized *image tiles* are selected.

The center pixel and a square area surrounding it form a *template* that is tracked in subsequent search frames. High-contrast corners tend to generate good templates by appearing locally unique and being well localized (i.e., they can be accurately correlated between two images). An implementation of the *Kanade–Lucas–Tomasi (KLT)* algorithm is used to track features between images. The KLT algorithm estimates the displacement of a template from one image to the next, using a gradient-based search algorithm that minimizes the difference in pixel intensity [31, 32]. An example set of feature correspondences between two images is shown in Figure 14.

Visual feature tracking algorithms like KLT can give a small number of erroneous template matches. The *RANSAC* algorithm is used to identify and discard poor measurements, to prevent them from corrupting the navigation estimates [33]. The Mars Helicopter RANSAC implementation assumes that the base image features and search image features are related by a *homography*, which implies that all features lie on a single plane [34]. In addition to discarding erroneous matches, the homography-based RANSAC has the added benefit of removing features that significantly violate the ground-plane assumption.

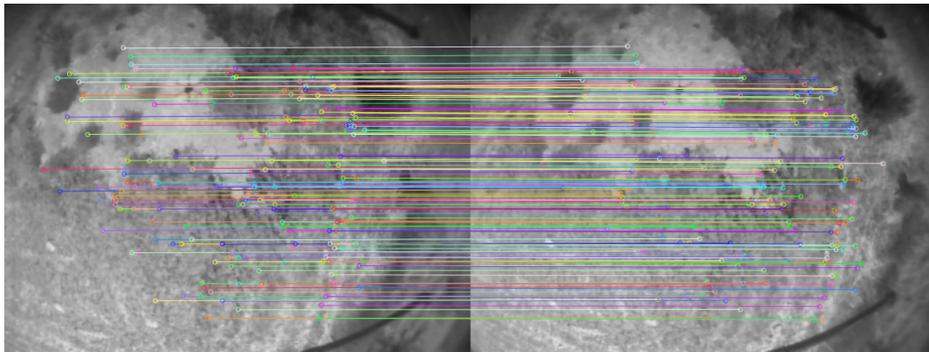


Fig. 14 Feature tracks between a base image (left) and search image (right) during an outdoor test flight

The KLT tracking algorithm has an inherently limited radius of convergence. Two measures are taken to increase the likelihood of correctly tracking a feature in subsequent frames. First, in a process known as *derotation*, gyro measurements from the IMU are used to improve the initial guess of each feature’s image-space location, by displacing the previously observed location based on the difference in attitude since the last observation (see [35]). Second, the KLT tracker operates on an *image pyramid* that consists of the full-resolution image as well as half- and quarter-resolution versions of the same image. By performing the tracking at consecutively higher levels of resolution, the radius of convergence is increased [36].

D. Experimental Testing

The navigation system has undergone extensive testing in an outdoor environment, part of which was conducted with Mars Helicopter sensors and avionics attached on a commercial hexacopter platform and with ground truth provided by RTK-GPS. A key takeaway from testing is that the navigation system performs robustly, and degrades gracefully and predictably when the system is stressed beyond the normal flight envelope. The main effect in those circumstances is a degradation in horizontal position accuracy, which is of secondary importance to velocity estimates in terms of maintaining vehicle stability. Results from a typical flight, which includes a 90-m excursion and return to the original location, are shown in Figure 15.

X. Control

Grip et al. [2] describes the control design for the demonstration vehicle that was flown in Mars atmospheric conditions in 2016. The starting point for that design is a static mixing of the control input channels, to achieve a pairing of inputs and controlled outputs such that the pairs are predominantly decoupled from each other. The approximate

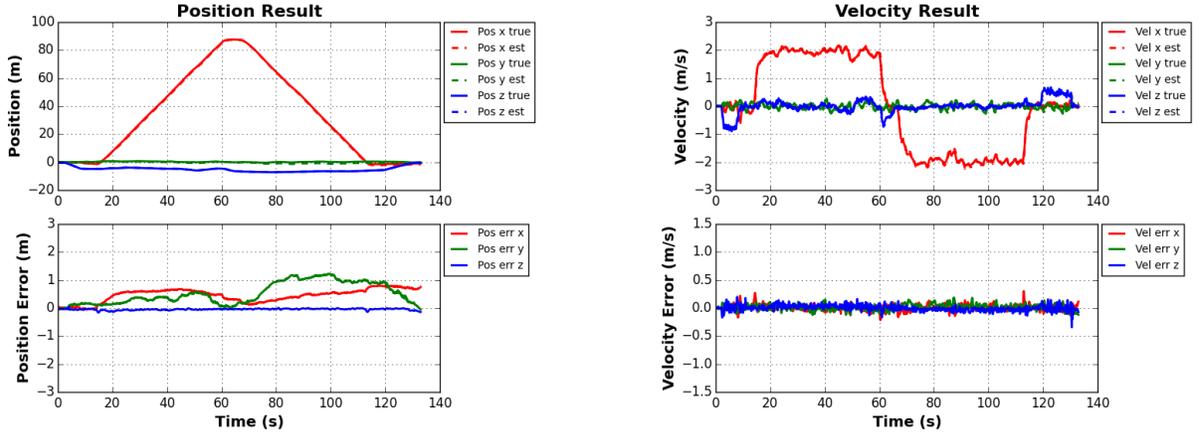


Fig. 15 Navigation results for a sample test flight: position (left) and velocity (right). The top subplots show the state estimate and a RTK-GPS-based ground truth estimate. The bottom plots show the difference between the navigation solution and the ground truth solution. The navigation solution is IMU-only below 1 m altitude. On final descent, this occurs at approximately 125 seconds and continues for 2–3 seconds after touchdown.

input-output pairing is as follows:

- *Heave* (vertical translation) is controlled approximately by *symmetric collective*, which is defined as the average of the upper and lower collective.
- *Yaw* (heading) is controlled approximately by *antisymmetric collective*, which is defined as one-half the difference between lower and upper collective.
- *Roll* and *pitch* are controlled by *lower cyclic*.

After closing the control loop for roll and pitch, *horizontal translation* is controlled by an outer loop that treats the commanded roll and pitch angles as the control input. Control laws are designed using single-input single-output (SISO) design techniques in the frequency domain. Robustness of the overall system is then evaluated within a multiple-input multiple-output (MIMO) framework by using *multiloop disk margins* that account for simultaneous variations in gain and phase across the input channels (see [37]).

The control design for the flight vehicle follows the same strategy as above, while accounting for two significant differences between the demonstration vehicle and the flight vehicle: (i) the flight vehicle is equipped with upper cyclic control, in addition to lower cyclic; and (ii) the flight vehicle is equipped with two direct-drive propulsion motors instead of a single motor with gears. The upper cyclic control is simply mixed with the lower cyclic control to provide a single (approximately) roll-aligned input channel as well as a pitch-aligned input channel. The consequences of the change in the propulsion motor configuration are more complex, as described in the next section.

A. Coupling with Propulsion Motor Dynamics

The rotor speeds can be decomposed as an average rotor speed and a differential rotor speed. In a single-motor configuration, the differential rotor speed is identically zero, but for the two-motor configuration on the flight vehicle it represents an additional degree of freedom. As a result, a net aerodynamic yaw moment, achieved approximately by actuation of antisymmetric collective, no longer results in a direct moment on the helicopter fuselage; instead, it affects the differential rotor speed by increasing the speed on one rotor and decreasing the speed on the other. Yaw moments are applied on the fuselage only as a secondary effect, when the motor speed controller acts to restore the setpoint speed, and the motor control loops are not fast enough for the effect to be considered instantaneous. Attempting to control fuselage yaw with differential collective therefore results in a strong coupling with the differential rotor speed, which is detrimental to MIMO robustness margins when the motor control loops are included in the analysis.

This issue is solved by the addition of a feedforward path from the collective control to the motor input voltages, which are used to control the torque applied to the motors. When actuating the collective control, the voltages are modulated in anticipation of the change in rotor drag, thus nullifying the effect of the collective input change on the rotor speeds. Figure 16 illustrates the control architecture.

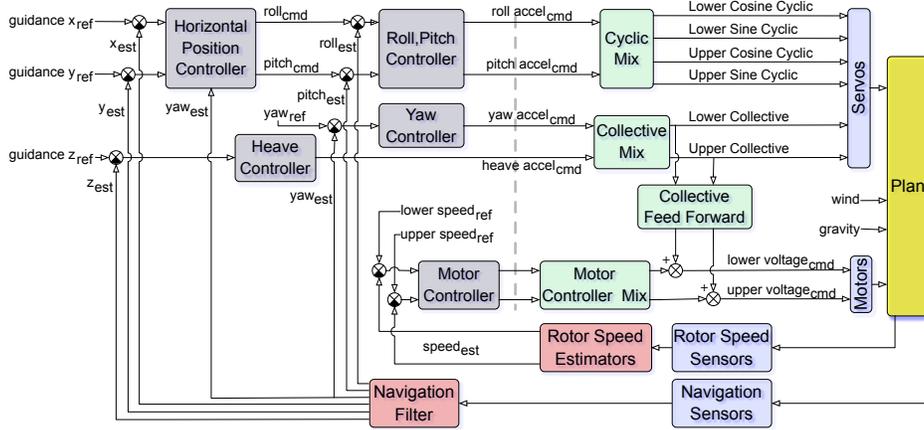


Fig. 16 Illustration of control architecture, including the feedforward from collective to motor input voltage

As a result of the additional feedforward, the same input-output pairings as before can be used in the control design, and MIMO robustness margins are recovered. However, the decoupling is not exact, due to actuator dynamics and errors in the modeled aerodynamic drag. Thus, evaluation of MIMO margins must now include the motor control loops, and the design of the motor controller parameters must be integrated with the rest of the control design to achieve the best possible margins.

B. Robustness Margin Evaluation

When evaluating robustness for flight on Mars, it is necessary to consider not just a single model, but models representing different flight regimes and different densities. Moreover, errors in the nominal flight dynamics model, due to inaccuracies in the modeling and system identification, must be accounted for. Multiloop disk margins are therefore evaluated for a set of models spanning the various operating points, while randomly perturbing the model parameters according to an assumed uncertainty model. Figure 18 shows the results for the endpoints of the density range $0.017\text{--}0.02\text{ kg/m}^3$, which is covered by a single rotor speed, for three different flight regimes (hover, forward flight, and forward flight with updraft). Parameters identified through experiments are varied in magnitude, with 3σ bounds in the range of $20\text{--}30\%$, depending on the quality of the experiment. Furthermore, for pairs of parameters that are naturally defined in terms of an amplitude and phase angle (e.g., torques in response to cyclic input), the phase angles are varied with 3σ bounds in the range of $10\text{--}20^\circ$. Small model-based parameters are subject to 100% variation.

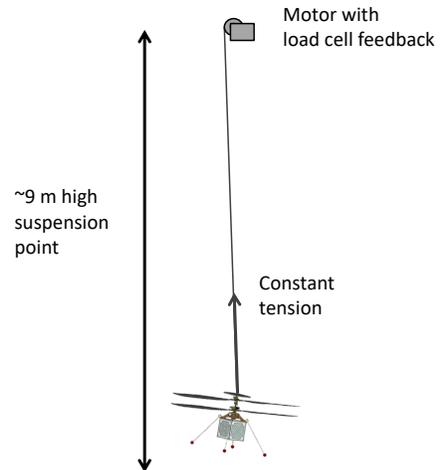


Fig. 17 Illustration of gravity offload system

C. Gravity-Offloaded Flight Testing

The demonstration vehicle described in [2] was designed to be light enough to be flown in the low density of Mars and the high gravity of Earth. This was achieved by delivering power through an electrical cable, rather than onboard batteries, and by not carrying components such as the solar panel or onboard avionics. By contrast, *gravity offloading* is needed for flight testing of the full vehicle.

The gravity offload system, which is illustrated in Figure 17, consists of an electrical motor attached to the vehicle via a lightweight tether. The motor is controlled by feedback from a load cell, in order to maintain constant tension equal to the difference between Earth and Mars gravity. The tether is attached to the helicopter at the top of the mast, above the solar panel.

Because the tether is attached to a fixed point in the chamber, it applies a lateral force on the vehicle when it moves

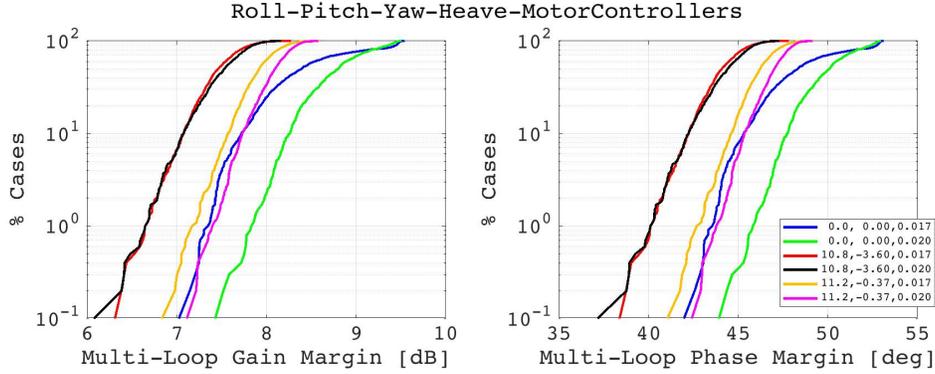


Fig. 18 Gain and phase margin evaluations for different operating points (hover, 11.2 m/s horizontal combined with a 0.37 m/s updraft, and 10.8 m/s horizontal combined with 3.6 m/s updraft). The margins are derived from multiloop disk margins. Each plot shows the cumulative percentage-wise probability of the margin falling below a certain value.

off-center. Also, because the attachment point on the vehicle is above the center of mass, the tether applies roll and pitch moments when the vehicle is tilted. These forces are restoring and have a stabilizing effect on the dynamics. This is undesirable from a testing perspective; however, the effects are well-understood and easily modelable, and they are therefore considered acceptable for testing.

The gravity offload system introduces several other complications:

- *Limited horizontal travel:* in order to translate off-center, the vehicle must fight the restoring forces from the tether. This limits the range of travel and degrades the transient behavior during translation. During testing, commanded horizontal travel is limited to 0.5 m from the center.
- *Interference with vertical dynamics:* The gravity offload system cannot enforce a perfectly constant tether tension. To minimize the effect on the helicopter dynamics and prevent the potential for destabilizing interference, the gravity offload system is designed to meet a frequency-dependent requirement on the transfer function from the vertical helicopter displacement to the applied tension (which should ideally be zero).
- *Pogo mode:* The gravity offload system has a mode in which the helicopter and the gravity offload pulley oscillates together due to flexibility of the tether. The mode is above 20 Hz due to the high modulus of the tether and the small inertia of the pulley, and does not pose a risk to the stability of the helicopter.
- *String mode:* The tether has modes at multiples of approximately 6 Hz, due to vibrations of the tensioned tether, similar to a guitar string. These modes are nominally poorly damped and are coupled with the roll and pitch dynamics of the helicopter. To prevent the modes from impinging on the robustness margins, additional damping is introduced by attaching a bundle of loose strings near the middle of the tether, which absorbs energy from the vibrating tether.

XI. Verification and Validation

Flight control verification and validation (V&V) for the Mars Helicopter is performed in multiple venues, and builds upon component-level testing of the mechanical system, sensors, and actuators that has been performed to verify both performance and robustness in the Mars environment. Because the full range of Mars conditions cannot be replicated in a single test venue on Earth, the goal of the V&V program is to perform multiple sets of tests with partial replication of the Mars environment, which together validate the operation of the whole system. Key aspects of the V&V program include

- simulation of end-to-end flights using HeliCAT (see Section VII)
- system identification in JPL's 25-ft Space Simulator (see Section VII-A)
- gravity-offloaded flight testing in JPL's 25-ft Space Simulator (see Section X-C)
- outdoor testing of the navigation system using a surrogate commercial platform (see Section IX-D)
- gravity-offloaded drop testing of the landing gear, performed by AeroVironment

This partitioning of tests reflects the reality that the aerodynamic environment on Mars cannot be combined with a photorealistic environment for navigation testing, the space necessary for large lateral translations, or realistic ground

surface properties. Simulation in HeliCAT allows for the various aspects of the Mars environment to be combined, subject to the limitations of modeling accuracy, and for parameters to be varied over a large number of simulations in order to stress the system and identify significant edge cases.

Although the navigation system is largely tested separately from the guidance and control system, it is necessary to verify that all of the flight control subsystems can be successfully combined. For this reason, flight tests have also been performed in Mars atmospheric conditions with the visual-inertial navigation system. Figure 5 in Section IV shows EDM-1 during a flight test in which the entire flight control system is operating as intended on Mars, except for minor changes to the guidance parameters. Figure 19 shows results from the test, including ground truth data from the Vicon system. In terms of velocity, the visual-inertial solution is virtually indistinguishable from the Vicon-based solution, except for a small error during the initial IMU-only phase. Horizontal position drift on the order of a few centimeters is observed, as well as a persistent error of about 6 cm due to LRF bias.

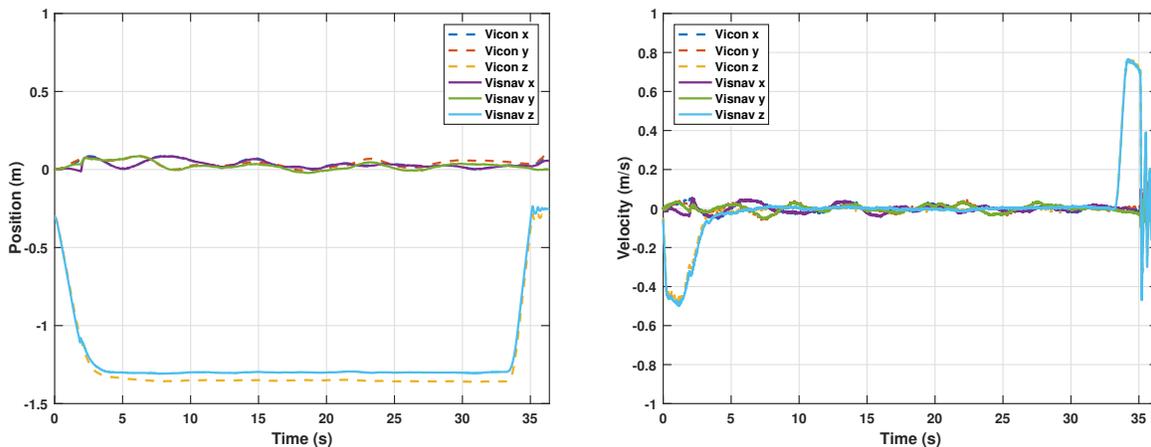


Fig. 19 Position (left) and velocity (right) estimates from onboard visual-inertial navigation and Vicon-based solution, for an EDM-1 test flight in the 25-ft Space Simulator.

XII. Conclusion

In this paper we have given a high-level overview of the Mars Helicopter flight control system in its near-final state, and discussed the testing, verification, and validation performed on the system to date. At the current point in time, the flight vehicle is being assembled, and preparations are being made for final integration and testing. This includes verification of the flight dynamics, through tests similar to the ones discussed in Section VII–A, as well as limited closed-loop testing.

Acknowledgments

The work of Håvard Fjær Grip, Johnny Lam, Gurkirpal Singh, David Bayard, Dylan T. Conway, Roland Brockers, Jeff Delaune, Larry Matthies, Travis Brown, Abhinandan Jain, Miguel San Martin, and Gene Merewether was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The authors would like to thank current and former members of the Mars Helicopter team: MiMi Aung, Bob Balaram, Jeff Simmonds, Chris Bang, Wayne Johnson, Daniel P. Scharf, Milan Mandić, Larry Young, Michael Kokorowski, Susan Gorton, William Warmbrodt, Jeff Umland, Joe Melko, Karen Lee, Matt Keennon, Ben Pipenberg, Bart Hibbs, Jeremy Tyler, Sara Langberg, Jeff York, Saverio D’Agostino, Katie Beckwith, Daniel Fugett, Ned Ferraro, Teddy Tzanetos, Fernando Mier-Hicks, Eric Archer, Nacer Chahat, Ed Satorius, Lauren McNally, Matthew Chase, Anthony Mittskus, Dave Hansen, Massimo Tinto, Brandon Burns, Greg Carr, Steve Dawson, Marshall Smart, Joseph Zitkus, Dhack Muthulingam, Christian Liebe, Patrick Meras, Gabrielle Massone, Bill Whitaker, Ted Kopf, Tien Nguyen, Gorang Gandhi, Jaakko Karras, Gerik Kubiak, Dima Kogan, Timothy Canham, Courtney Duncan, Matt Golombek, Justin Maki, Amelia Quon, Ryan Stern, David Zhu, Stephan Weiss, Jason Carlton, Ed Konefat, Chris Lefler, Parviz Danesh, Jonathan

Denison, Mary Romejko, Chris Richardson, Brent Tweddle, John Liu, Lia Altenbuchner, Chris Porter, Steve Yows, Jack Pempejian, Josh Ravich, Taryn Bailey, Faramarz Keyvanfar, Brandon Metz, Charles Wang, Michael Pauken, Stefano Cappucci, Jennifer Miller, Tyler Schmidt, Pradeep Bhandari, Eric Sunada, Mike Mischna, Dan Tyler, Jeff Barnes, Mark Richardson, James Seastrom, Karan L'Heureux, Ella Seal, Tom Tourigny, Keith Fields, Brad Kinter, Brian Allan, Bérénice Mettler, and Olivier Toupet.

We would also like to thank the members of the Mars Helicopter Advisory Group: Naomi Palmer, Leslie Livesay, Gentry Lee, Howard Eisen, Rob Manning, Keyur Patel, and Richard Cook.

References

- [1] Grip, H. F., Johnson, W., Malpica, C., Scharf, D. P., Mandić, M., Young, L., Allan, B., Mettler, B., and San Martin, M., "Flight dynamics of a Mars Helicopter," *Proc. European Rotorcraft Forum*, Milan, Italy, 2017.
- [2] Grip, H. F., Scharf, D. P., Malpica, C., Johnson, W., Mandić, M., Singh, G., and Young, L., "Guidance and Control for a Mars Helicopter," *Proc. AIAA Guidance, Navigation, and Control Conference*, Kissimmee, FL, 2018.
- [3] Savu, G., and Trifu, O., "Photovoltaic Rotorcraft for Mars Missions," *Proc. Joint Propulsion Conf. and Exhibit*, San Diego, CA, 1995.
- [4] Kroo, I., and Kunz, P., "Development of the Mesicopter: A Miniature Autonomous Rotorcraft," *Proc. American Helicopter Society Vertical Lift Aircraft Design Conf.*, San Francisco, CA, 2000.
- [5] Aiken, E. W., Ormiston, R. A., and Young, L. A., "Future Directions in Rotorcraft Technology at Ames Research Center," *Proc. American Helicopter Society Annual Forum*, Virginia Beach, VA, 2000.
- [6] Young, L. A., "Vertical Lift – Not Just For Terrestrial Flight," *Proc. AHS/AIAA/Rae/SAE International Powered Lift Conference*, Arlington, VA, 2000.
- [7] Young, L. A., et al., "Use of Vertical Lift Planetary Aerial Vehicles for the Exploration of Mars," *NASA Headquarters and Lunar and Planetary Institute Workshop on Mars Exploration Concepts*, Houston, TX, 2000.
- [8] Young, L. A., Chen, R. T. N., Aiken, E. W., and Briggs, G. A., "Design Opportunities and Challenges in the Development of Vertical Lift Planetary Aerial Vehicles," *Proc. American Helicopter Society International Vertical Lift Aircraft Design Conference*, San Francisco, CA, 2000.
- [9] Thompson, B., "Full Throttle to Mars," Rotor & Wing, Phillips Business Information, LLC, Potomac, MD, 2001.
- [10] Datta, A., Roget, B., Griffiths, D., Pugliese, G., Sitaraman, J., Bao, J., Liu, L., and Gamard, O., "Design of the Martian Autonomous Rotary-Wing Vehicle," *Proc. AHS Specialist Meeting on Aerodynamics, Acoustics, and Test and Evaluation*, San Francisco, CA, 2002.
- [11] Young, L. A., and Aiken, E. W., "Vertical Lift Planetary Aerial Vehicles: Three Planetary Bodies and Four Conceptual Design Cases," *Proc. European Rotorcraft Forum*, Moscow, Russia, 2001.
- [12] Young, L. A., et al., "Engineering Studies into Vertical Lift Planetary Aerial Vehicles," *Proc. AHS International Meeting on Advanced Rotorcraft Technology and Life Saving Activities*, Utsunomiya, Tochigi, Japan, 2002.
- [13] Young, L. A., Aiken, E. W., Gulick, V., Mancinelli, R., and Briggs, G. A., "Rotorcraft as Mars Scouts," *Proc. IEEE Aerospace Conf.*, Big Sky, MT, 2002.
- [14] Young, L. A., Aiken, E. W., and Briggs, G. A., "Smart Rotorcraft Field Assistants for Terrestrial and Planetary Science," *Proc. IEEE Aerospace Conference*, Big Sky, MT, 2004.
- [15] Young, L. A., Lee, P., Briggs, G., and Aiken, E., "Mars Rotorcraft: Possibilities, Limitations, and Implications For Human/Robotic Exploration," *Proc. IEEE Aerospace Conference*, Big Sky, MT, 2005.
- [16] Tsuzuki, N., Sato, S., and Abe, T., "Conceptual Design and Feasibility for a Miniature Mars Exploration Rotorcraft," *Proc. International Congress of the Aeronautical Sciences*, 2004.
- [17] Song, H., and Underwood, C., "A Mars VTOL Aerobot – Preliminary Design, Dynamics and Control," *Proc. IEEE Aerospace Conference*, 2007.
- [18] Brockers, R., Humenberger, M., Weiss, S., and Matthies, L., "Towards autonomous navigation of miniature UAV," *Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops*, 2014.

- [19] Balam, J., and Tokumaru, P. T., "Rotorcrafts for Mars Exploration," *Proc. International Planetary Probe Workshop*, Pasadena, CA, 2014. LPI Contribution No. 1795.
- [20] Balam, J., Canham, T., Duncan, C., Golombek, M., Grip, H. F., Maki, J., Quon, A., Stern, R., and Zhu, D., "Mars Helicopter Technology Demonstrator," *Proc. AIAA Atmospheric Flight Mechanics Conference*, Kissimmee, FL, 2017.
- [21] Koning, W., Johnson, W., and Grip, H. F., "Improved Mars Helicopter Aerodynamic Rotor Model for Comprehensive Analyses," *Proc. European Rotorcraft Forum*, Delft, The Netherlands, 2018.
- [22] Bayard, D. S., Conway, D. T., Brockers, R., Delaune, J., Matthies, L., Grip, H. F., Merewether, G., and San Martin, A. M., "Vision-Based Navigation for the NASA Mars Helicopter," *Proc. AIAA Scitech Forum and Exhibition*, 2019.
- [23] Pipenberg, B. T., Keennon, M. T., Tyler, J. D., Langberg, S. A., Hibbs, B., Balam, B., Grip, H. F., and Pempejian, J., "Design and Fabrication of the Mars Helicopter Rotor, Airframe, and Landing Gear Systems," *Proc. AIAA Scitech Forum and Exhibition*, 2018.
- [24] Lim, C., and Jain, A., "Dshell++: A Component Based, Reusable Space System Simulation Framework," *Proc. Third IEEE International Conference on Space Mission Challenges for Information Technology*, Pasadena, CA, 2009.
- [25] San Martin, A. M., Bayard, D. S., Conway, D. T., Mandić, M., and Bailey, E. S., "A Minimal State Augmentation Algorithm for Vision-Based Navigation without Using Mapped Landmarks," *10th International ESA Conference on Guidance, Navigation & Control Systems*, ESA, 2017.
- [26] Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O., "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , No. 6, 2007, pp. 1052–1067.
- [27] Montiel, J. M., Civera, J., and Davison, A. J., "Unified inverse depth parametrization for monocular SLAM," *Robotics: Science and Systems*, 2006.
- [28] Klein, G., and Murray, D., "Parallel tracking and mapping for small AR workspaces," *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, IEEE, 2007, pp. 225–234.
- [29] Kottas, D. G., Wu, K. J., and Roumeliotis, S. I., "Detecting and dealing with hovering maneuvers in vision-aided inertial navigation systems," *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, 2013, pp. 3172–3179.
- [30] Rosten, E., and Drummond, T., "Machine learning for high-speed corner detection," *European Conference on Computer Vision*, Springer, 2006, pp. 430–443.
- [31] Shi, J., and Tomasi, C., "Good features to track," Tech. rep., Cornell University, 1993.
- [32] Lucas, B. D., and Kanade, T., "An iterative image registration technique with an application to stereo vision," *Proc. Intl. Joint Conf. on Artificial Intelligence*, Vancouver, British Columbia, 1981.
- [33] Fischler, M. A., and Bolles, R. C., "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, Vol. 24, No. 6, 1981, pp. 381–395.
- [34] Hartley, R., and Zisserman, A., *Multiple view geometry in computer vision*, Cambridge university press, 2003.
- [35] Hwangbo, M., Kim, J.-S., and Kanade, T., "Gyro-aided feature tracking for a moving camera: fusion, auto-calibration and GPU implementation," *The International Journal of Robotics Research*, Vol. 30, No. 14, 2011, pp. 1755–1774.
- [36] Bouguet, J.-Y., "Pyramidal implementation of the affine Lucas Kanade feature tracker: Description of the algorithm," *Intel Corporation*, Vol. 5, No. 1-10, 2001, p. 4.
- [37] Blight, J. D., Dailey, R. L., and Gangsaas, D., "Practical control law design for aircraft using multivariable techniques," *Int. J. Control*, Vol. 59, No. 1, 1994, pp. 93–137.