

Dynamics and Control of a Hopping Robot for Extreme Environment Exploration on the Moon and Mars

Himangshu Kalita
University of Arizona
1130 N Mountain Ave
Tucson, AZ 85721
hkalita@email.arizona.edu

Akash S. Gholap
Indian Institute of Technology
Kharagpur, West Bengal 721302
akashgholapaero.sci@iitkgp.ac.in

Jekan Thangavelautham
University of Arizona
1130 N Mountain Ave
Tucson, AZ 85721
jekan@email.arizona.edu

Abstract— High-resolution orbital imagery from the LROC reveals evidence of subsurface voids and mare-pits on the lunar surface. Similar discoveries have been made with the HiRISE camera onboard the MRO observing the Martian surface. These accessible voids could be used for a future human base because they offer a natural radiation and micrometeorite shield and offer constant habitable temperatures. Exploration of these extreme and rugged environments remains out of reach from current planetary rovers and landers. A credible solution is to develop an architecture that permits taking high exploratory risks that translates into high reward science. Rapid advancement in electronics, sensors, actuators, and power have resulted in ever-shrinking devices and instruments that can be housed in small platforms. We propose to use a small, low-cost, modular spherical robot called SphereX that is designed to hop and roll short distances. Each robot is of several kilograms in mass and several liters in volume. Each SphereX will consist of space-qualified electronics like command & data handling board, power board for power management and s-band radio transceiver for communication. Power is provided using lithium-ion primary batteries or a PEM fuel cell power supply. Communication is established through multi-hop communication link to relay data from inside the caves to a lander outside on the planetary surface. Since the temperature inside underground lunar pits is expected at -25°C, thermal management for the space-grade electronics is minimal as they can operate up to -40°C, however thermal management for the battery pack and the propellants will be done through active and passive elements. Moreover, SphereX requires use of a propulsion system and Attitude Determination and Control System (ADCS) to perform controlled ballistic hops. Hopping on very-low gravity environments is more time-efficient than rolling due to the reduced traction. In this paper, we present detailed analysis of each subsystem of SphereX and also detailed dynamics and control simulations of SphereX for ballistic hopping and rolling mobility. For ballistic hopping control, the robot has two modes: soft landing mode for traversing long distances and entering the pit through its collapsed entrance, and a fuel-efficient hard landing mode for traversing short distances. We will then present experimental results for mapping unknown cave-like environments which is done using a quadcopter for simulating low-gravity (e.g. Moon, Mars) environments and testing the control algorithms. The quadcopter mimics the dynamics of SphereX and also carries a 3D LiDAR for mapping and navigation. 3D point cloud data collected by the LiDAR is used for performing SLAM and path planning in unknown and GPS-denied environments much like the pits, caves and lava tubes on the Moon and Mars.

TABLE OF CONTENTS

1. INTRODUCTION	1
2. BACKGROUND AND MOTIVATION.....	2
3. SYSTEM DESIGN	2
4. MOBILITY CONTROL.....	6
5. HARDWARE PLATFORM	8
6. CONCLUSION	10
REFERENCES.....	11
BIOGRAPHY.....	11

1. INTRODUCTION

The latest orbital images of the Moon and Mars taken by the Lunar Reconnaissance Orbiter (LRO) and Mars Reconnaissance Orbiter (MRO) respectively reveal hundreds of cave openings known as pits as shown in Figure 1 [1]. These pits serve as natural shelters from micro-meteoroids, cosmic radiation, and surface temperature extremes and could be used for a future human base. Mobile ground robots have become integral for surface exploration of the Moon, Mars and other planetary bodies. These rovers have proven their merit, but they are large, in the order of several hundred kilograms and house state-of-the-art science laboratories. As such, exploration of these extreme and rugged environments remains out of reach for current planetary rovers. In addition, conventional methods of path-planning and navigation used by planetary rovers are not applicable in these pits as they are sheltered by thick rock, which block sunlight and prevent radio communication with the outside world.

Exploring these pits requires a whole new mobility platform and specialized methods for path planning and navigation. With the rapid advancement in lightweight structural materials, miniaturization of electronics, sensors and actuators, it is possible to develop small, lightweight and low-cost platforms to tackle some of the hardest challenges in extreme sites of extreme importance. In this paper, we present a spherical robot platform called SphereX with several kilograms in mass and several liters in volume that can hop and roll. SphereX's flight capabilities are intended to match that of a terrestrial quadcopter but do so in low-gravity, airless or low-pressure environments.

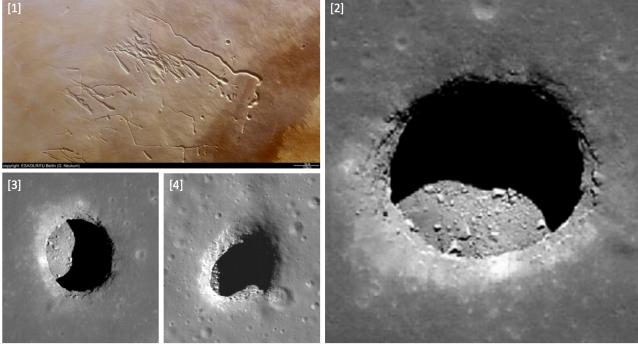


Figure 1. Lunar and Martian pits and lava tubes. (1) Lava tubes on Pavonis Mons on Mars (ESA), (2) Marius hills, (3) Mare Tranquilitatis, and (4) Mare Ingenii pit on Moon (NASA/GSFC/Arizona State University).

2. BACKGROUND AND MOTIVATION

Small spherical robots have been widely proposed in the past. Their spherical shape enables them to roll on loose, even terrain. Examples include spherical robots developed at Univ. of Sherbrooke [2], Kickbot [3] developed at MIT, Cyclops [4] at Carnegie Mellon University and inflatable ball robots developed at North Carolina State University [5] and University of Toronto [6]. Typically, these spherical robots use a pair of direct drive motors in a holonomic configuration. Others such as the Cyclops and the inflatables pivot a heavy mass, thus moving center of gravity that results in rolling. Other mobility techniques including use of spinning flywheels attached to a two-link manipulator on the Gyrover [7] or 3-axis reaction wheels to spin and summersault as with the Hedgehog developed by Stanford and NASA JPL [8]. Hedgehog's use of reaction wheels enables it to overcome rugged terrain by simply creeping over the obstacle no matter how steep or uneven. However, it's unclear if a gyro-based system can overcome both steep and large obstacles. In reality, even a gyro-based system is bound to slip on steep surfaces, but under low gravity environments such as asteroids, they may be able to reach meters in height.

An alternative to rolling and creeping is hopping. A typical approach to hopping is to use a hopping spring mechanism to overcome large obstacles [9]. One example is the Micro-hopper for Mars exploration developed by the Canadian Space Agency [10]. The Micro-hopper has a regular tetrahedron geometry that enables it to land in any orientation at the end of a jump. The hopping mechanism is based on a novel cylindrical scissor mechanism enabled by a Shape Memory Alloy (SMA) actuator. However, the design allows only one jump per day on Mars. Another technique for hopping developed by Plante and Dubowsky at MIT utilize Polymer Actuator Membranes (PAM) to load a spring. The system is only 18 grams and can enable hopping of Microbots with a mass of 100 g up to 1 m [11],[12]. Microbots are cm-scale spherical robots equipped with power and communication systems, a mobility system that enables it to hop, roll and bounce and an array of miniaturized sensors such as imagers, spectrometers, and chemical analysis sensors developed at MIT. They are intended to explore

caves, lava tubes, canyons and cliffs. Ideally, many hundreds of these robots would be deployed enabling large-scale in-situ exploration.

SphereX is the direct descendant of the Microbot platform. SphereX has the same goals as the Microbots, but with the goal of launching fewer robots, that are better equipped with science grade instruments. It consists of a mobility system to perform optimal exploration of these target environments. It also consists of space-grade electronics like computer board for command and data handling, power board for power management and radio transceiver for communicating among multiple robots. Moreover, it also consists of a power system for power generation/storage, multiple UHF/S-band antennas and accommodates payloads in the rest of the volume [13-15]. A large rover or lander may carry several of these SphereX robots that can be tactically deployed to explore and access rugged environments that would otherwise be inaccessible.

3. SYSTEM DESIGN

The design of SphereX is a complex task that involves a large number of variables and multiple engineering disciplines. It is a highly coupled problem between multiple disciplines, and it must balance payload objectives against its overall size, mass, power and control which affects its cost and operation.

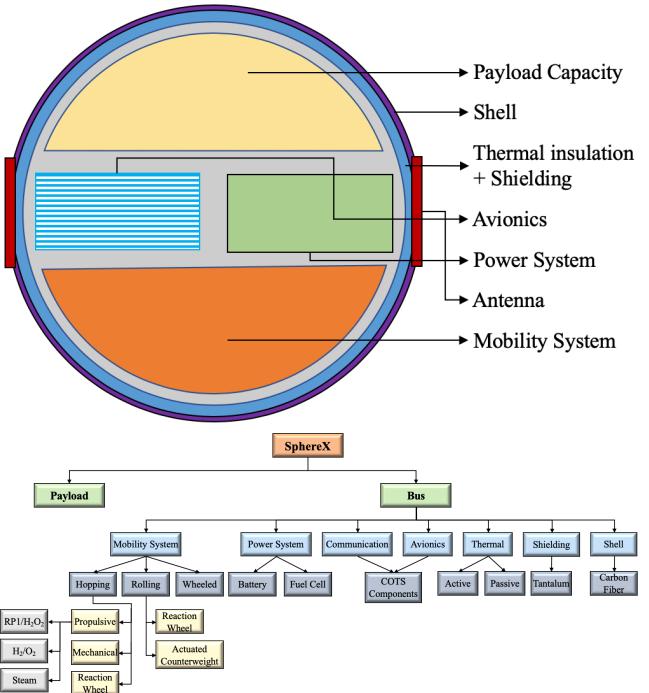


Figure 2. (Top) SphereX architecture, (Bottom) Available options for each subsystem of SphereX.

Moreover, each subsystem has multiple candidate solutions as shown in Figure 2, e.g. mobility can be achieved through hopping, rolling or wheels, power system can be design through batteries that carries all the required power or can be generated on demand through fuel cells. Similarly, the

selection of communication system and the avionics depends on numerous Commercially-Off-the-Shelf (COTS) options available, the thermal system can be designed through active, passive or a combination of both. As such, finding optimal design solutions for SphereX to meet a defined mission requirement is of paramount importance. Currently, space systems are optimized manually through evaluation of each discipline independently. With this labor-intensive approach, although feasibility is achieved, there is no guarantee for achieving optimality of the overall system. Thus, space system design could benefit from the application of multidisciplinary design optimization (MDO).

Our past work on multidisciplinary design and control optimization of SphereX [16] has shown the advantages of propulsive hopping mobility with a miniaturized propulsion system and 3-axis reaction over rolling and wheeled mobility. Moreover, it has also shown the advantages of lithium-ion batteries for shorter missions and PEM fuel cells for longer missions as the power system. Therefore, for this section, we provide a brief introduction to each subsystem of SphereX from design point of view for exploring lunar and Martian pits and lava tubes. The con-ops for performing such a mission is shown in Figure 3. A lander carrying multiple SphereX robots would descent nearby a target pit/lava tube and deploy the robots one by one. Each robot will have three phases: 1. Surface operation to approach the pit entrance, 2. Pit/lava tube entrance maneuver, and 3. Sub-surface operation to explore the pit.

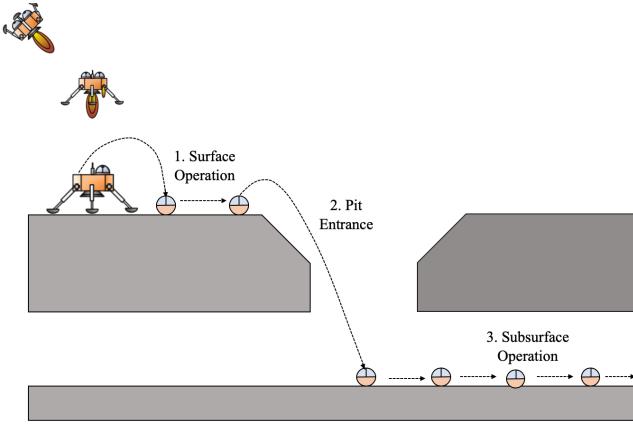


Figure 3. Concepts of operation for exploring Lunar and Martian pits and lava tubes.

Mobility – Propulsive hopping

Propulsive hopping is achieved through the combined action of a miniaturized propulsion system and a 3-axis reaction wheel system. The 3-axis reaction wheel system is used for maintaining roll, pitch, and yaw angles along its principal axes while the propulsion system providing a constant thrust for a desired time to execute a ballistic hop. During this mode of ballistic hopping the robot needs to reach a desired state $r_d = (x_d, y_d, z_d)$ from its initial state $r_0 = (x_0, y_0, z_0)$. Two modes of propulsive hopping are implemented for SphereX: a) Hard-landing mode for traversing short distances, and b) Soft-landing mode for pit entrance and traversing long

distances. The hard-landing propulsive hopping mode consists of three phases a) Attitude correction phase, b) Boost phase, and c) Ballistic trajectory phase. Apart from the above three phases, the robot has an additional soft-landing phase for soft-landing propulsive hopping. During the attitude correction phase, the robot needs to orient itself from its initial attitude states $q_0 = (\phi_0, \theta_0, \psi_0)$ to the desired attitude states $q_d = (\phi_d, \theta_d, \psi_d)$ while on ground. For the attitude correction phase a sliding-mode attitude controller is designed to maintain its desired attitude states. During the boost phase, the propulsion system provides a constant thrust $\|T\|$ for time t_b with the attitude control system maintaining the desired attitude states $q_d = (\phi_d, \theta_d, \psi_d)$. During the ballistic trajectory phase, the robot follows its dynamics due to the action of gravitational force on it. Finally, during the soft-landing phase the propulsion system again provides a constant thrust $\|T\|$ at a control angle γ such that the robot lands at zero velocity. The design of the propulsion system can be achieved through different propellants, but we have considered three options for our analysis: a) RP1/H₂O₂, b) H₂/O₂, and c) Steam.

Power system – Lithium-ion batteries and PEM fuel cells

Power required for the entire mission can be stored through lithium-ion batteries. When lithium-ion batteries are selected for the mission, the objective is to find the minimum discharge capacity of the battery system Q such that the state of charge at the end of mission $SOC(\Gamma)$ is greater than 10%, Γ being the mission time length.

Power can also be generated on demand with fuel cells, that consists of a negatively charged electrode (anode), a positively charged electrode (cathode), and an electrolyte membrane. Hydrogen is oxidized on the anode and oxygen is reduced on the cathode. Protons are transported from the anode to the cathode through the electrolyte membrane, and the electrons are carried to the cathode over the external circuit. The fuel cell layers are the PEM, gas diffusion layer (GDL) and catalyst layers as shown in Figure 4 (Top-Left) [17]. For a power demand P , each cell operating at a voltage V , the rate of oxygen and hydrogen used are expressed as Equation (1).

$$\dot{O}_2 = 8.29 \times 10^{-8} \frac{P}{V} \frac{\text{kg}}{\text{s}}, \quad \dot{H}_2 = 1.05 \times 10^{-8} \frac{P}{V} \frac{\text{kg}}{\text{s}} \quad (1)$$

In a PEM fuel cell, water is produced as a byproduct and the rate of water generated is given by Equation (2).

$$\dot{H}_2O = 9.34 \times 10^{-8} \frac{P}{V} \frac{\text{kg}}{\text{s}} \quad (2)$$

The hydrogen and oxygen required for the operation of the fuel cell are produced on demand by the hydrolysis of LiH and catalytic thermal decomposition of LiClO₄. Therefore, the design of the PEM fuel cell power system consists of a LiH tank, water tank and a LiClO₄ tank. The water produced as a byproduct of the operation of the fuel cell will be stored in the water tank for hydrolysis of LiH. Moreover, the

operating voltage of a PEM fuel cell is less than the theoretical value due to losses or irreversibilities in the form of activation losses, fuel crossover and internal currents, ohmic losses and mass transport or concentration losses [17]. Adding all these losses, the operating voltage of a fuel cell at a current density i can be expressed as Equation (3).

$$V = E_{oc} - ir - A \ln(i) + me^{(ni)} \quad (3)$$

Where, E_{oc} is the open circuit voltage, r is the area-specific resistance, A is a constant denoting the slope of the Tafel line for the particular fuel cell, and m and n are constants for mass-transfer losses. Thus, for the design of the fuel cell system, the goal is to find the optimal number of fuel cells and current density such that the mass of oxygen and hydrogen for constant power demand P for a mission lifetime Γ is minimized. Figure 4 (Right) shows the optimal values of n and i for different power demand. The different subsystems of SphereX operates at different cycles and as such there is a varying power demand. A varying load connected directly to fuel cell would cause voltage oscillations that reduce the life of the fuel cell. Hence the power system of SphereX consists of a fuel cell-battery hybrid system. The fuel cell constantly charges the battery at constant power, and the battery along with a power management board handles the high and varying demands of the load as shown in Figure 4 (Bottom-Left).

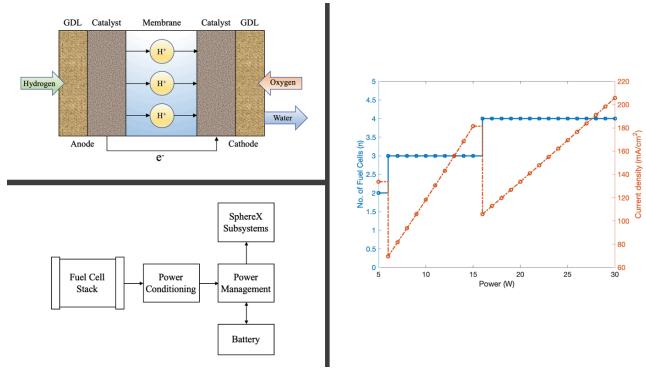


Figure 4. (Top-Left) A Polymer Electrolyte Membrane (PEM) fuel cell and its major components, (Right) Optimal values of number of fuel cells and current density for different power demands, (Bottom-Left) Fuel cell-battery hybrid system for power management.

Communication

For a team of N robots to cooperatively explore through a target area, efficient communication is a key factor. If the robots transmit information periodically, without any synchronization, all robots might try to transmit at the same time causing communication delays. Furthermore, collisions are very likely to occur, further delaying communication. The robots need to transmit information with synchronization among them. To transmit these a protocol is used that depends on a user defined configuration parameter T_{up} (update period) in which all robots transmit once as shown in Figure 5 (Top). The update period is equally subdivided in

slots for each robot of duration $T_x = T_{up}/N$. Robots transmit at the beginning of their respective slots, thus spacing the transmissions as much as possible. Each robot uses only a fraction of its slot, the remaining time is used to accommodate delays in transmission. When a robot transmits at time t_{now} it sets its own transmission instant $t_{next} = t_{now} + T_{up}$. Now, with the instruments of each N robots taking samples (images/3D-point cloud) at a frequency f , and size of each sample D_{sam} , the desired data rate is calculated as $B_r = (D_{sam}fN)/T_{up}$. The power consumed by the antenna is a function of the antenna gain and desired data bit rate as shown in Equation (4).

$$P_{comm} = \frac{16\pi^2 f_r^2 k T_s (\text{SNR}) B_r d^2}{c^2 G_r G_t L_l \eta_p} \quad (4)$$

Where, c is the speed of light, G_r and G_t are the receiver and transmitter gain, L_l is a system line loss factor, f_r is the frequency, k is the Boltzmann constant, T_s is the system noise temperature, B_r is the data bit rate, d is the inter robot distance, η_p is the communication efficiency and SNR is the signal-to-noise ratio of the system. To accommodate this protocol, the communication subsystem consists of multiple UHF/S band patch antenna and a radio transceiver. The patch antennas are designed as an array along the circumference of the robot such that the total half power beam width (THPBW) and the bandwidth (BW) of the array is maximized. The THPBW is calculated by adding the HPBW of each lobe of the array. Thus, for the design of the antenna array, the goal is to find the optimal frequency and number of patch antennas in the array such that the THPBW and BW is maximized.

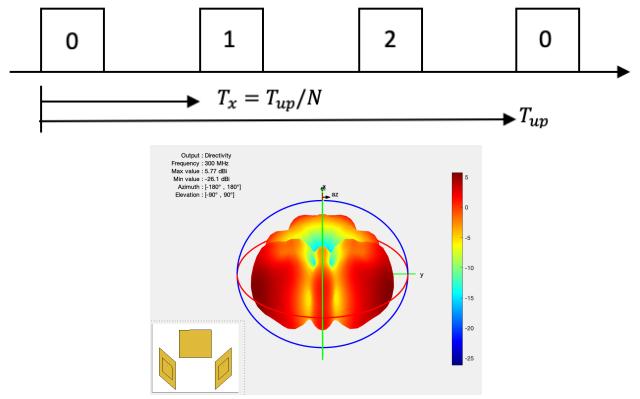


Figure 5. (Top) Communication protocol for efficient communication among N robots, (Bottom) Radiation pattern of an array of 3 patch antennas along the circumference of SphereX designed at 300 MHz.

Thermal

In order to prevent the robot from freezing and overheating without compromising thermal reliability and stability, we implement a thermal control architecture that relies on both active and passive thermal control elements. The proposed thermal control architecture relies on a low emissive silver coating finish and a low conductive silica aerogel insulation

layer along with thermal control heat rejection/generation mechanism composed of a variable emittance coating (VEC), a heat switch and an electric heater. Figure 6 shows the thermal model of the robot with all heat transfer modes. The model consists of a spherical shell of mass m_s and specific heat $c_{p,s}$, representing the robot shape, with a low outer emissivity coating ϵ_s and a thick inner insulation layer with conductivity λ_{in} and thickness x . The temperature of the outer shell is represented as T_s , and that of the surrounding as T_a . All the internal components of the robot are assumed to be a homogeneous spherical body with temperature T_b , mass m_b and specific heat $c_{p,b}$. The VEC has a maximum emissivity of $\epsilon_{s,vec}$ and a surface area of A_{vec} . The heat switch has a thermal conductivity λ_{sw} , contact area A_{sw} , and switch length x_{sw} and the electric heater provides heat at a rate \dot{Q}_{eh} . Moreover, heat is generated during the operation of the robot by different subsystems (mobility (propulsion), power (fuel cell), communication) which is denoted by \dot{Q}_{gen} . As such we implement the thermal model with two state variables T_b and T_s and the differential equations governing them is shown in Equation (5)

$$\begin{aligned} m_b c_{p,b} \dot{T}_b &= \dot{Q}_{gen} - \dot{Q}_{cond,s} - \dot{Q}_{cond,g} - \dot{Q}_{cond,sw} + \dot{Q}_{eh} \\ m_s c_{p,s} \dot{T}_s &= \dot{Q}_{sun} + \dot{Q}_{cond,s} - \dot{Q}_{rad,[vec]} - \dot{Q}_{conv} \end{aligned} \quad (5)$$

Thus, for the design of the thermal system, the goal is to find the optimal values of $x, A_{vec}, \epsilon_{s,vec}, A_{sw}$ and x_{sw} such that the total mass of the thermal system and power consumed by the electric heater \dot{Q}_{eh} is minimized.

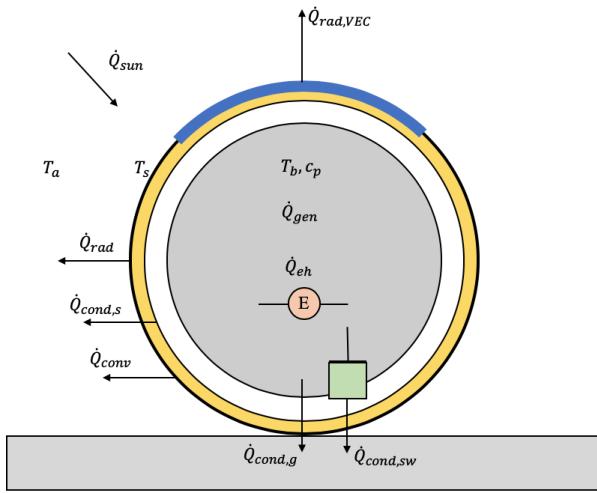


Figure 6. Modes of heat transfer involved in the operation of SphereX.

Avionics

The main computer selected is Rincon Research's AstroSDR which is a complete RF payload: software-defined radio (SDR), FPGA signal processor, ARM processor, and data storage. The single board computer contains the Dual-core ARM Cortex A9 with NEON processor that can operate at up to 733 MHz and a Xilinx Zynq 7045 FPGA. It also contains

512 Mbyte DDR3 RAM memory and 2 GByte Flash for radiation-tolerant OS storage and an option for 64 GByte eMMC flash storage. The tuning range for the receiver and transmitter is 70 MHz to 6 GHz with a maximum bandwidth of 56 MHz. It also has 30 pins 1.8 V GPIO and 24 pins 3.3 V GPIO interfaces. The dimension of the board is 90 x 90 mm, weighs only 95 g, consumes 5.5 W power under nominal conditions and has an operating temperature range of -40°C to 85°C. For power management, the GomSpace NanoPower P31u board is selected that is configurable with lithium-ion batteries. It features a microcontroller that provides maximum power-point tracking (MPPT) capability, measures and logs voltages, currents and temperatures of the system. With an I2C interface, it is possible to read out measurements, control the on/off-state of 3.3 V and 5 V busses, switch on/off the MPPT and to set/read various parameters. The incoming power from the batteries is used to feed two buck-converters supplying a 3.3 V @ 5 A and a 5 V @ 4 A output bus. It also contains six individually controllable output switches with over-current shut-down and latch-up protection, each separately configurable to either 3.3 V or 5.0 V output. The dimension of the board is 89.3 x 92.9 x 15.3 mm, weighs only 100 g and consumes 0.165 W power under nominal conditions. Moreover, for controlling the reaction wheels, the MAI400 Controller Board is selected that incorporates four microcontrollers with a 3-axis MEMS accelerometer, 3-axis MEMS gyroscope and 3 sets of reaction wheel drive circuitry. One of the microcontrollers is implemented as the ADCS processor while the remaining three microcontrollers used for reaction wheel control. It has multiple communication interfaces available such as UART, RS232, I2C, SPI. The dimension of the board is 86 x 88 mm, weighs only 47 g and consumes 0.45 W power under nominal conditions.

Shielding and Shell

The shielding of the robot is done through aluminum whose thickness is calculated based on the incident radiation dose at the target environment. For structural robustness, the shell of the robot is made of carbon fiber whose thickness is calculated such that the deformation on impact during hard-landing hops is within 1 mm.

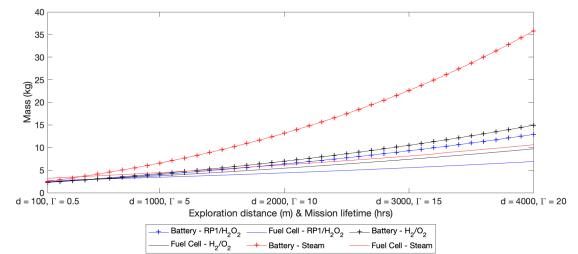


Figure 7. Mass of the robot for all combinations of propulsion and power system for varying exploration distance and mission time.

Figure 7 shows the mass of SphereX for different combinations of propulsion and power system with varying

exploration distance and mission time on the surface of the Moon. The payload for this design is considered to have a mass of 1 kg, volume of 1liter, and power demand of 10 W. It can be seen that for an exploration objective of 100 m and 0.5 hrs, the system with lithium-ion batteries and RP1/H₂O₂ propulsive mobility is the optimal choice, however as the exploration objective increases to 4000 m and 20 hrs, the system with fuel cells and RP1/H₂O₂ propulsive mobility is the optimal choice.

4. MOBILITY CONTROL

As discussed in section 3, mobility of the robot is achieved through ballistic hopping with the help of a miniaturized propulsion system and 3-axis reaction wheels. Two modes of ballistic hopping are identified for the robot to be able to explore extreme environments: a) Hard-landing mode for exploring short distances, and b) Soft-landing mode for pit entrance and exploring long distances whose time diagrams are shown in Figure 8. Moreover, with the 3-axis reaction wheels, rolling mobility can also be achieved when needed. Here we present the details of the different controllers used and their simulation results.

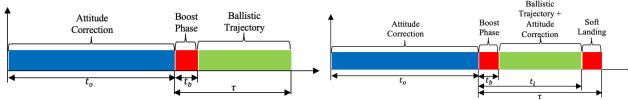


Figure 8. Time diagram of (Left) Hard-landing hopping mode, (Right) Soft-landing hopping mode.

Attitude Controller

The attitude dynamics of the robot actuated by a set of 3 reaction wheels are described by Equation (6).

$$\dot{\omega}_s = -J_s^{-1}\tilde{\omega}_s(J_s\omega_s + h_{rw}) + J_s^{-1}(\tau_{rw} + \tau_{ext}) \quad (6)$$

where, ω_s is the angular velocity of the robot in its body fixed frame (\mathcal{B}), $\tilde{\omega}_s$ is the cross-product matrix of ω_s , J_s is the robot's inertia matrix, h_{rw} is the net reaction wheel angular momentum, τ_{rw} is the net reaction wheel torque, and τ_{ext} is the external torque. The dynamics are second order and can be written as: $\dot{\omega}_s = f + Bu$, where $f = -J_s^{-1}\tilde{\omega}_s(J_s\omega_s + h_{rw}) + J_s^{-1}\tau_{ext}$ is the drift function, $B = J_s^{-1}E_{rw}$ is the input matrix, and $u = \tau_{ext}$ is the control input. The robot's attitude is represented by the Modified Rodrigues Parameters (MRP), σ . Now, let's define the sliding variable, s as in Equation (7).

$$s(\delta\omega, \delta\sigma) = \delta\omega + \Lambda\delta\sigma + \Lambda_I \int_0^t \delta\sigma dt \quad (7)$$

where, Λ and Λ_I are 3x3 symmetric positive-definite constant matrices, $\delta\omega = \omega_s - {}^{\mathcal{B}\mathcal{R}}R^{\mathcal{R}}\omega_d$ is the difference in the robot's angular velocity and the desired angular velocity with ${}^{\mathcal{B}\mathcal{R}}R$ as the rotation matrix from the reference frame (\mathcal{R}) to the body frame (\mathcal{B}), and $\delta\sigma = \sigma - \sigma_d$ is the difference between the robot's MRP vector and desired MRP vector. The control input is $u = u_{eq} - u_{rob}$ as shown in Equation (8).

$$u_{eq} = B^{-1} \left(-f + {}^{\mathcal{B}\mathcal{R}}R^{\mathcal{R}}\dot{\omega}_d - \tilde{\omega}_s {}^{\mathcal{B}\mathcal{R}}R^{\mathcal{R}}\omega_d - \frac{1}{4}\Lambda B(\delta\sigma)\delta\omega - \Lambda_I\delta\sigma \right)$$

$$u_{rob} = B^{-1}K\text{sign}(s(\delta\omega, \delta\sigma)) \quad (8)$$

where, K is a diagonal matrix.

Hard Landing Maneuver

During the boost phase of ballistic hopping, the thruster provides a constant thrust $\|T\|$ for time t_b with the attitude control system maintaining the desired attitude states $q_d = (\phi_d, \theta_d, \psi_d)$. The dynamic equations governing the motion of the robot can be expressed as Equation (9).

$$\dot{r} = v, \quad \dot{v} = \begin{cases} g + \frac{T}{m}, & \text{if } t < t_b \\ g & \text{if } t_b < t < \tau \end{cases} \quad (9)$$

The optimization objective is to minimize the fuel consumption during the boost phase which is equivalent to minimizing the total thrust during burn time t_b . Two constraints are added such that the burn time $t_b < \tau$ and the final position of the robot is equal to its desired position as $\|r(\tau) - r_d\|^2 = 0$. The objective of the optimization problem is to find the optimal values of the design variables $\mathbf{d}_{hl} = [t_b, \tau]$ such that the optimal index $f(\mathbf{d}_{hl})$ is minimized and is mathematically expressed as Equation (10).

$$\min f_{hl}(\mathbf{d}_{hl}) = \int_0^{t_b} \|T\| dt \quad (10)$$

subject to $\begin{cases} g_{hl1}(\mathbf{d}_{hl}) \equiv t_b - \tau < 0 \\ g_{hl2}(\mathbf{d}_{hl}) \equiv \|r(\tau) - r_d\|^2 = 0 \end{cases}$

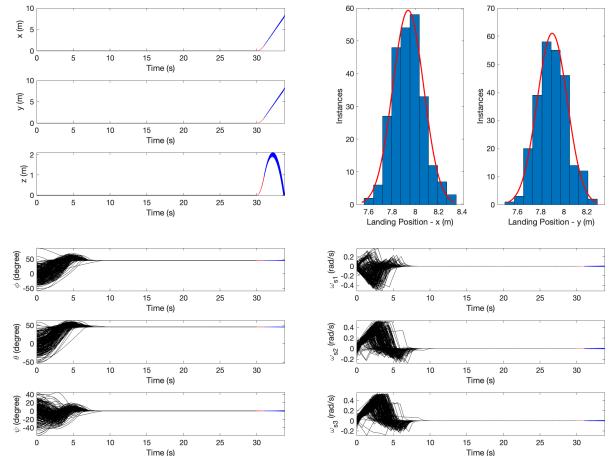


Figure 9. Monte Carlo time history of (Top-Left) Position of the robot when commanded to hop from $r_0 = (0, 0, 0)$ to $r_d = (8, 8, 0)$, (Bottom-Left) Euler angles in 'ZYX' Euler angle notation, (Bottom-Right) Angular velocities in the body fixed frame, and (Top-Right) Histogram of the landing position of the robot.

A Monte Carlo hopping control simulation was performed with 250 numerical simulations to check the robustness of the

controllers used with various input parameter combinations. The constants that vary between each simulation are the initial conditions (angular position, angular velocity) of the robot, inertia matrix terms, reaction wheel spin axis inertia, reaction wheel spin axis, external torque, and magnitude of thrust. Figure 9 shows the Monte Carlo time history of the position, Euler angles, and angular velocities in the body fixed frame along with the histogram of the landing position of the robot when commanded to hop from its initial position $r_0 = (0,0,0)$ to its desired position $r_d = (8,8,0)$. It can be seen that the mean of the landing position of the robot along x and y-axis are 7.95 and 7.91 m and the standard deviations are 0.145 and 0.146 m.

Soft Landing Maneuver

For entering Lunar and Martian pits and lava tubes and also to travel long distances, the robot needs a soft-landing maneuver. During the soft-landing phase, the control angle γ for the applied thrust can be derived as $\gamma = \tan^{-1}(v_z/v_x)$. The dynamic equations governing the motion of the robot are expressed as Equation (11).

$$\dot{r} = v, \quad \dot{v} = \begin{cases} g + \frac{T}{m} & t < t_b \\ g & t_b < t < t_l \\ g + \frac{T}{m} & t_l < t < \tau \end{cases}, \quad \dot{m} = \begin{cases} -\frac{\|T\|}{I_{sp}g_0} & t < t_b \\ 0 & t_b < t < t_l \\ -\frac{\|T\|}{I_{sp}g_0} & t_l < t < \tau \end{cases} \quad (11)$$

The optimization objective is to minimize the fuel consumption and the optimal index can be expressed as $f(\mathbf{d}_{sl}) = \int_0^{t_b} \|T\| dt + \int_{t_l}^{\tau} \|T\| dt$. Four constraints are added, with the design variables $\mathbf{d}_{sl} = [t_b, t_l, \tau]$ and the optimization problem is mathematically expressed as Equation (12).

$$\min f_{sl}(\mathbf{d}_{sl}) = \int_0^{t_b} \|T\| dt + \int_{t_l}^{\tau} \|T\| dt$$

subject to $\begin{cases} g_{sl1}(\mathbf{d}_{sl}) \equiv t_b - t_l < 0 \\ g_{sl2}(\mathbf{d}_{sl}) \equiv t_l - \tau < 0 \\ g_{sl3}(\mathbf{d}_{sl}) \equiv \|r(\tau) - r_d\|^2 = 0 \\ g_{sl4}(\mathbf{d}_{sl}) \equiv \|v_{\tau}\|^2 = 0 \end{cases} \quad (12)$

Figure 10 shows the Monte Carlo time history of the position and velocity in x, z-axis, Euler angles along with the histogram of the landing position in x direction, velocity in x and z-axis of the robot when commanded to perform a soft-landing hop from its initial position $r_0 = (0,0,0)$ to its desired position $r_d = (55,0,-100)$ to enter a pit. It can be seen that the mean of the landing position of the robot along x-axis is 54.42 m with a standard deviation of 0.323 m and the bandwidth for the kernel distributions are 0.017 and 0.475.

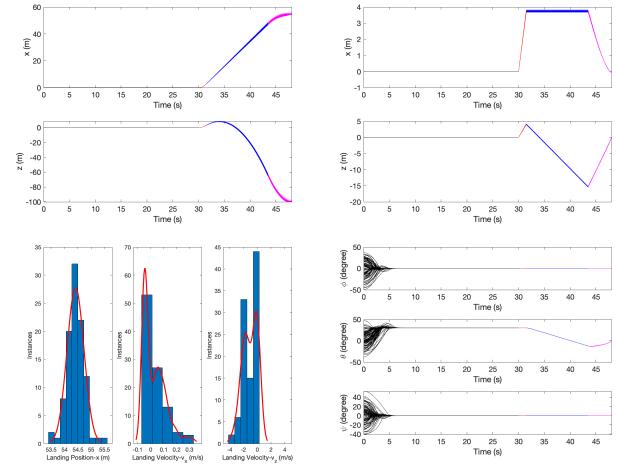


Figure 10. Monte Carlo time history of (Top-Left) Position of the robot when commanded to hop from $r_0 = (0, 0, 0)$ to $r_d = (55, 0, -100)$, (Top-Right) Velocities in x and z-axis, (Bottom-Right) Euler angles in ‘ZYX’ Euler angle notation, and (Bottom-Left) Histogram of the landing position in z-axis, landing velocity in x and z-axis of the robot.

Rolling Controller

For rolling control, the robot needs to reach a desired state $r_d = (x_d, y_d)$ from its initial state $r_0 = (x_0, y_0)$. By imposing the no slip constraints, the linear velocity of the robot is computed as, $v = {}^R\omega \times r e_3$, where ${}^R\omega$ is the angular velocity of the robot in the reference frame. The objective is to stabilize the states $r = (x, y)$ to the desired equilibrium states $r_d = (x_d, y_d)$. We use a control law for this objective to determine the desired angular velocities ${}^R\omega_d$ in the reference frame (R) as in Equation (13).

$${}^R\omega_d = \left(\frac{k}{r}(y - y_d), -\frac{k}{r}(x - x_d), 0 \right) \quad (13)$$

where $k > 0$ are free and is a linear state-feedback control law but when transformed into body coordinate velocities ω_s , it is nonlinear. The closed-loop system with this control law is globally exponentially stable with respect to $e_{xy} = (x - x_d, y - y_d)$ and $k > 0$. Considering the Lyapunov function $V: \mathbb{R}^2 \rightarrow \mathbb{R}$ defined by $V = (1/2)e_{xy}^T e_{xy} > 0$ which is a positive definite function, the derivative of V along the trajectories of the closed loop system is, $\dot{V} = -k e_{xy}^T e_{xy} = -2kV < 0$. Thus, the closed-loop system is globally exponentially stable. Now, with the desired angular velocities determined, we use the sliding-mode controller-based attitude controller to control the actuators. Figure 11 shows the Monte Carlo time history of the position, Euler angles, and angular velocities in the body fixed frame of the robot when commanded to roll from its initial position $r_0 = (0,0)$ to its desired position $r_d = (1,-1)$.

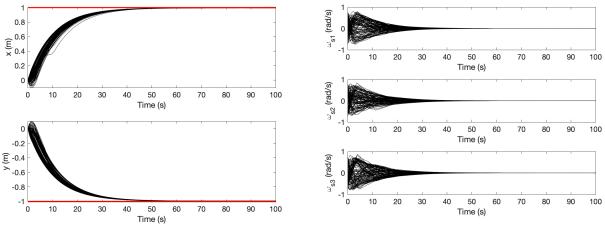


Figure 11. Monte Carlo time history of (Left) Position of the robot along x and y-axis when commanded to roll from $r_0 = (0, 0)$ to $r_d = (1, -1)$, (Right) Angular velocities of the robot in the body fixed frame.

5. HARDWARE PLATFORM

For exploring and mapping unknown cave-like environment, SphereX will employ a Hop+Scan→Stop→Process cycle. During the Hop+Scan phase, the robot hops from its current position to a desired position while a LiDAR collects 3D scans of the environment. After the hop is completed, the robot stops and processes the collected data to generate a global 3D map of the environment, localize the robot and select the next hopping location. To demonstrate this exploration and mapping cycle, we developed a hardware platform for testing the hopping and mapping sequences of SphereX that consists of a DJI Matrice 100 quadcopter, a six-core arm based single board computer (SBC), a LiDAR, an ultrasonic sensor, a microcontroller board and an optional stereo camera module as shown in Figure 12.

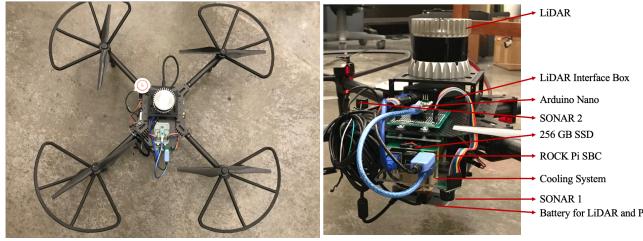


Figure 12. Hardware platform consisting of a drone, LiDAR, SBC and ultrasonic sensors to test hopping and mapping sequences of SphereX.

A single board computer system was carefully selected to serve any uncalled need for computational capability and simultaneously considering power requirements. Using six cores, channels a huge advantage to the ROS platform which relies on multithreading for its fluent functioning. The flight controller module communicates with the external SBC to provide velocity and angular velocity information. Raw point cloud packets from the LiDAR are sent to the SBC for processing, localization and deciding the next hopping location. Moreover, raw measurements from two ultrasonic sensors are first sent to the microcontroller board for pre-processing then a parsed data is sent to the SBC to decide on soft landing maneuver and the extent of hopping range based on the roof height of the cave/lava tube environment as shown in Figure 13. We have used Robot Operating System (ROS) based software architecture to weave intercommunicating nodes together, with the source code

developed in C++ as shown in Figure 14. The details of each node in the software architecture is provided below.

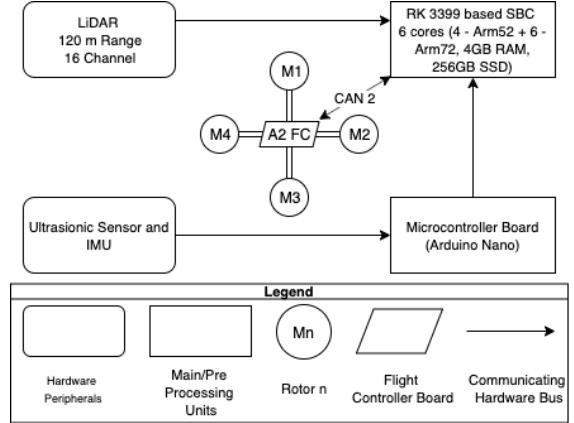


Figure 13. System architecture of the hardware test platform.

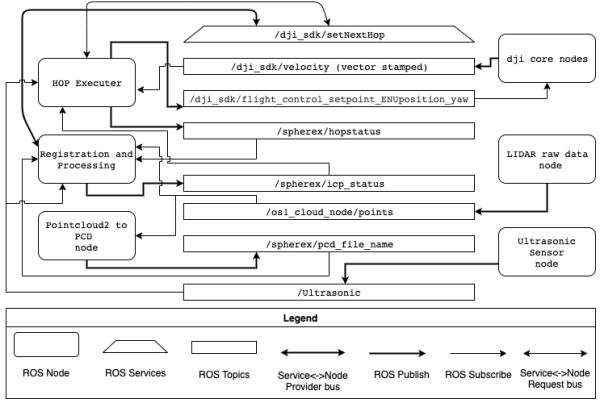


Figure 14. Software architecture implemented for the hardware test platform.

HOP Executer: This node is responsible for the overall execution of hopping and landing sequence of the drone; this node publishes velocity commands onto the flight control topic which is then subscribed by dji's core control node to serially transmit the control commands to the flight controller board A2.

Registration and Processing: This node is responsible for the point cloud processing which includes filtering and registration for mapping and navigation, and path planning to provide subsequent hopping location service to the Hop executer node.

DJI Core Nodes: These includes communication nodes, service nodes, and various flight status nodes.

Ultrasonic Sensor Node: This node communicates with the microcontroller board through the serial ports and publishes range data onto the topic which is then used to select hopping range and soft landing.

LIDAR Raw Data node and Point cloud to PCD node: These two nodes are responsible for the raw point cloud packets

processing and point cloud to pcd file generation respectively.

Registration and Pose Estimation

With the LiDAR generating 3D maps at a frequency f , successive 3D maps are registered and merged into one coordinate system using the point-to-plane iterative closest point (ICP) algorithm. Given a source map S and a destination map D , each iteration of the ICP algorithm first establishes a set of pair-correspondences between point in the source and point in the destination using kD-trees. With the point-to-plane iterative closest point (ICP) algorithm used, the objective is to find a transformation $T: S \rightarrow D$ that minimize the sum of the squared distance between each source point and the tangent plane at its correspondence destination point. More specifically, if $s_i \in S$ is a source point, $d_i \in D$ is the corresponding destination point, and n_i is the unit normal vector at d_i as shown in Figure 15, then the minimization problem can be written as Equation (14) [18].

$$\min_{T:S \rightarrow D} \sum_i ((Ts_i - d_i) \cdot n_i)^2 \quad (14)$$

$T: S \rightarrow D$ is a 4×4 3D rigid-body transformation matrix that is composed of a rotation matrix R and a translation matrix t . Thus, the rotation matrix R and a translation matrix t contains the estimate of the orientation and position of the drone with respect to its initial orientation and position. After finding the optimal transformation, the global map is updated as $\mathcal{M} \leftarrow D + TS$.

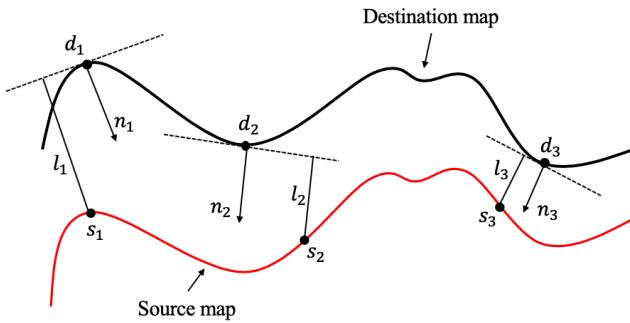


Figure 15. Point-to-plane error between two surfaces.

Path Planning

For the robot to perform exploration, it needs to find the next hopping location avoiding collision with obstacles after completing each hop. We have developed an algorithm that relies on the current global map and current position of the robot to determine the next hopping location as discussed below.

With the current global map \mathcal{M} and the current position of the robot r_t known, the first step is to find the indices of the neighboring points within a radius \mathcal{R} and form a map $\mathcal{M}' \subseteq \mathcal{M}$. Next we identify the ground plane \mathcal{G} by fitting a plane below the robot on \mathcal{M}' . The obstacles are then defined as $\mathcal{O} = \mathcal{M}' - \mathcal{G}$.

Algorithm: Autonomous exploration through hopping in unknown environments

Require: Current global map \mathcal{M} , current position of the robot r_t

1. Find neighbors within a radius \mathcal{R} from the current robot position: $\mathcal{M}' \subseteq \mathcal{M}$;
2. Identify ground plane \mathcal{G} on \mathcal{M}' ;
3. Identify obstacles $\mathcal{O} = \mathcal{M}' - \mathcal{G}$;
4. Perform v vertical layer segmentation \mathcal{V} on \mathcal{O} ;
5. **for** each vertical layer i
6. Perform n sector segmentation on \mathcal{V}_i ;
7. **for** each sector j
8. Sector indexing $I_j^i = \begin{cases} 1 & \text{free sector;} \\ 0 & \text{obstacle} \end{cases}$;
9. $I_j = \prod_{i=1}^n I_j^i$;
10. set $j = j + 1$;
11. **end for**
12. set $i = i + 1$;
13. **end for**
14. Possible sectors $\mathbb{I} = \{j: I_j = 1\}$

Next we divide the obstacle map \mathcal{O} into v layers $\mathcal{V}_i, i = 1, 2, \dots, v$. Each vertical layer \mathcal{V}_i is then divided into n sectors defined by radius r and angle $\theta = 2\pi/n$. The selection of r and n depends on the hopping range and size of the robot. Next for each sector $j = 1, 2, \dots, n$ on vertical layer \mathcal{V}_i , an index I_j^i is defined. The value of the index I_j^i is equal to 1 if no points from the obstacle map \mathcal{O} lies inside the sector, while it is equal to 0 if any points from the obstacle map \mathcal{O} lies inside the sector. Next, the indexes of each sector across all v vertical layers are multiplied together to find the final index I_j . Finally, the index set of all the sectors whose indexes are equal to 1 are defined as $\mathbb{I} = \{j: I_j = 1\}$. If there is only one sector that lies on \mathbb{I} that sector is selected for the next hop. However, if there are multiple sectors that lies on \mathbb{I} , the sector for the next hop is selected based on the previous position of the robot r_{t-1} . An exploration unit vector $e_{exp} = r_t - r_{t-1}$ and another unit vector for each sector in \mathbb{I} , $e_{sk}, k = 1, 2, \dots, |\mathbb{I}|$ are defined and the sector that lies closest to e_{exp} , $\{k: \min(\cos^{-1}(e_{exp} \cdot e_{sk}))\}$ is selected as the sector for the next hop. Once the sector is selected, the next hopping location is defined by a tuple (d, ϕ) , where d is determined by the hopping range of the robot and ϕ is the orientation of the unit vector of the selected sector.

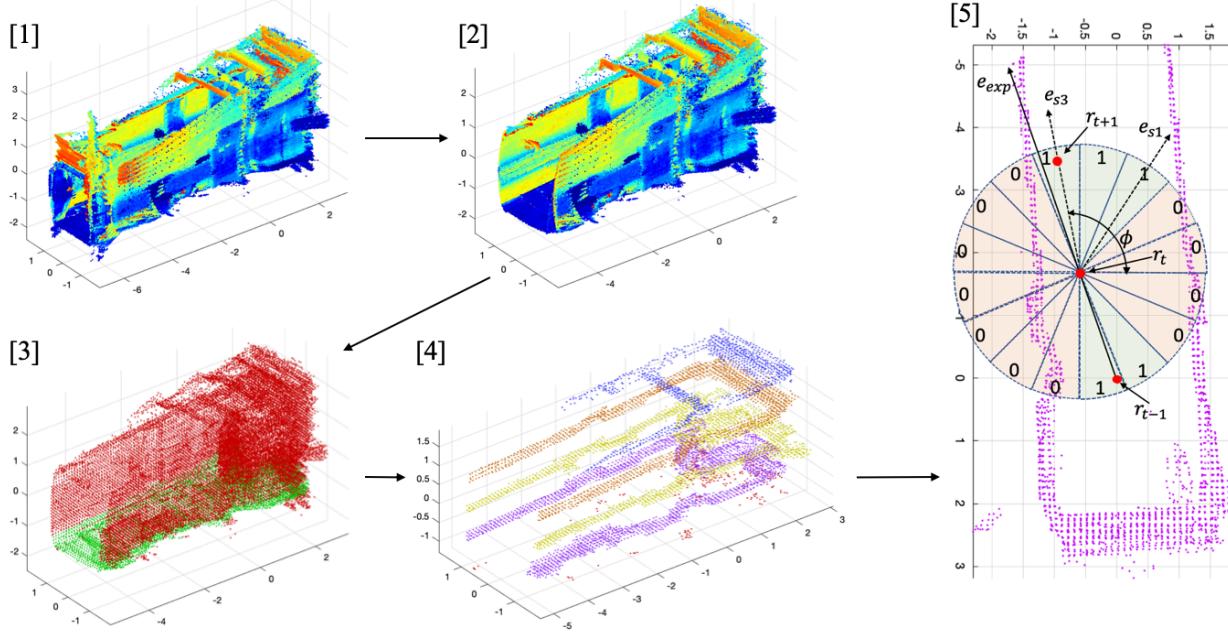


Figure 16. Graphical representation of the algorithm (autonomous exploration through hopping in unknown environments) (1) Current global map \mathcal{M} , (2) Map of neighbors \mathcal{M}' within a radius \mathcal{R} from the current position of the robot r_t , (3) Ground plane map \mathcal{G} shown in green and obstacle map \mathcal{O} shown in red, (4) Vertical layer segmentation \mathcal{V} on \mathcal{O} , and [5] Sector segmentation on \mathcal{V}_i along with the indexes for each sector. The red dots show the previous position of the robot r_{t-1} , current position of the robot r_t and the location selected to perform the next hop r_{t+1} .

Figure 16 shows a graphical representation of each step of the algorithm. Using this algorithm, we performed a mapping experiment on the hallways of the Aerospace and Mechanical Engineering Department of the University of Arizona. Figure 17 shows the map generated of one of the hallways using our approach. The robot started at position (0,0,0) and the final position of the robot was (-9.8,22.6) with the units in meters.

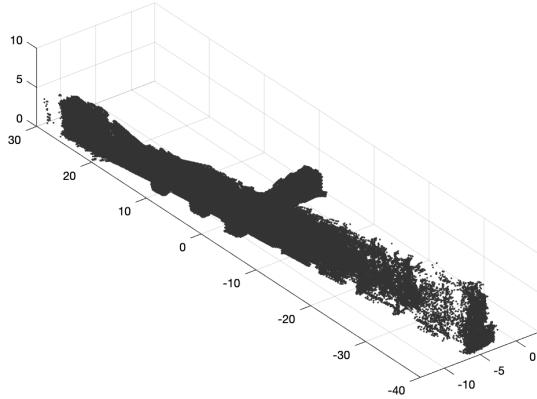


Figure 17. Map of a hallway at the Aerospace and Mechanical Engineering department of the University of Arizona generated by using the algorithm discussed above.

6. CONCLUSION

This paper presented the SphereX robot that uses a miniaturized propulsion system along with a 3-axis reaction wheel system to perform controlled ballistic hopping and rolling for exploring extreme off-world environments like caves, pits and lava tubes on the surface of the Moon and Mars. We explored three propellant for the design of SphereX and found advantages of using RP1/H₂O₂ over H₂/O₂ and steam. However, for exploring pits and caves on the surface of the Moon and Mars, the hardest challenge is to enter through the pit/lava tube entrance as the distance from the entrance to the floor are \sim 100 meters. As such, SphereX is equipped with two modes of hopping mobility: soft-landing and hard-landing mode. The soft-landing mode will be used to enter the pits and lava tubes so that the robot can land on the floor of the entrance at approximately zero velocity. It can also be used to traverse large distances when needed. Comparatively, the hard-landing mode is much more efficient in terms of fuel use and will be used to traverse shorter distances in the order of 10 meters. Thus, the shell of the robot is designed such that it can absorb the impact shock during short range hard-landing hops. Moreover, the 3-axis reaction wheel system can also be used for rolling mobility when needed. For power, the robot will use PEM fuel cells to produce power on demand using a fuel cell-battery hybrid system. To avoid cryogenic storage of hydrogen and oxygen for the fuel cell, hydrogen is generated through the hydrolysis reaction of lithium hydride and oxygen is generated through thermal decomposition reaction of lithium perchlorate. However, use of lithium-ion batteries as a power source is more advantageous when the robot is designed for short

mission lifetime. For inter-robot communication, the robot has an array of patch antennas around its circumference so that the directivity is maximized. We then presented detailed Monte Carlo simulations of the hopping and rolling controllers that showed their robustness towards dynamics uncertainties and external disturbances. Although SphereX can be used for various science motivated missions, our primary objective is to map unexplored environments like pits and lava tubes. As such we presented an algorithm for autonomous exploration of unknown environments using hopping mobility. To test the algorithm, we developed a hardware platform that consists of a drone and a 3D LiDAR that uses our proposed algorithm to navigate and map unknown environments. Our preliminary tests inside the hallways of the AME department of University of Arizona shows promising results and we are planning to test it on real cave environments in Arizona and New Mexico.

REFERENCES

- [1] R.V. Wagner and M.S. Robinson, "Distribution, formation mechanisms, and significance of lunar pits," *Icarus*, Vol. 237, pp. 52-60, 2014.
- [2] F. Michaud, J. de Lafontaine, S. Caron, "A Spherical Robot for Planetary Surface Exploration," 6th International Symposium on Artificial Intelligence and Robotics & Automation in Space, June 2001.
- [3] C. Batten, D. Wentzlaff, "Kichbot: A Spherical Autonomous Robot," MIT Technical Report, 2001.
- [4] B. Chemel, E. Mutschler, H. Schempf, "Cyclops: Miniature Robotic Reconnaissance System," IEEE International Conference on Robotics & Automation, May 1999.
- [5] Jeffrey Antol, "A New Vehicle for Planetary Surface Exploration: The Mars Tumbleweed," 1st Space Exploration Conference, 2005.
- [6] Goran Jurisa Basic, "Power-scavenging Tumbleweed Rover," MSc Thesis, University of Toronto, 2010.
- [7] Y. Xu, K. W. Au, G. C. Nandy, H. B. Brown, "Analysis of Actuation and Dynamic Balancing for a Single Wheel Robot," IEEE/RSJ International Conference on Intelligent Robots and Systems, October 1998.
- [8] M. Pavone, J. C. Castillo-Rogez, I. A. D. Nesnas, J. A. Hoffman, N. J. Strange, "Spacecraft/Rover Hybrids for the Exploration of Small Solar System Bodies," IEEE Aerospace Conference, 2013.
- [9] D. H. Kim et al., "Mechanism, control, and visual management of a jumping robot," Mechatronics, 2008.
- [10] E. Dupius, S. Montminy, P. Allard, "Hopping robot for planetary exploration," 8th iSAIRAS, September 2005.
- [11] S. Dubowsky, K. Iagnemma, et al., "A concept Mission: Microbots for Large-Scale Planetary Surface and Subsurface Exploration," Space Technology and Applications International Forum, 2005.
- [12] S. B. Kesner, J. Plante, P. J. Boston, T. Fabian, S. Dubowsky, "Mobility and Power Feasibility of a Microbot Team System for Extraterrestrial Cave Exploration," IEEE Int. Conf. on Robotics and Automation, 2007
- [13] J. Thangavelautham, M. S. Robinson, A. Taits, et al., "Flying, hopping Pit-Bots for cave and lava tube exploration on the Moon and Mars" 2nd International Workshop on Instrumentation for Planetary Missions, NASA Goddard, 2014.
- [14] H. Kalita, A. Ravindran, S. Morad, J. Thangavelautham, "Path Planning and Navigation Inside Off-World Lava Tubes and Caves," IEEE/ION PLANS Conference, 2018.
- [15] H. Kalita, J. Thangavelautham, "Multirobot Cliff Climbing on Low-Gravity Environments," 11th NASA/ESA Conference on Adaptive Hardware and Systems, Pasadena, USA, 2017, 24-27 July.
- [16] H. Kalita, J. Thangavelautham, "Automated Multidisciplinary Design and Control of Hopping Robots for Exploration of Extreme Environments on the Moon and Mars," 70th International Astronautical Congress (IAC), Washington D.C., USA, 2019, 21-25 October.
- [17] J. Larminie, A. Dicks, "Fuel Cell Systems Explained," Wiley, 2003.
- [18] K-L. Low, "Linear Least-Squares Optimization for Point-to-Plane ICP Surface Registration," Technical Report, University of North Carolina, 2004, February.

BIOGRAPHY



Himangshu Kalita received a B.Tech. in Mechanical Engineering from National Institute of Technology, Silchar, India in 2012. He is presently pursuing his Ph.D. in Mechanical Engineering from the University of Arizona in the Space and Terrestrial Robotic Exploration (SpaceTREx) Laboratory. His research interests include dynamics and control, space robotics, machine learning and automated design.



Akash S. Gholap is presently a final year student at the Indian Institute of Technology (IIT), Kharagpur, India pursuing B. Tech and M. Tech dual degree in Aerospace Engineering. He worked as a summer research intern at the Space and Terrestrial Robotic Exploration (SpaceTREx) Laboratory at the University of Arizona in the summer of 2018 and 2019. His research interests include dynamics and control, space robotics and guidance for unmanned aerial vehicle.



Jekanthan Thangavelautham has a background in aerospace engineering from the University of Toronto. He worked on Canadarm, Canadarm 2 and the DARPA Orbital Express missions at MDA Space Missions. Jekan obtained his Ph.D. in space robotics at the University of Toronto Institute for Aerospace Studies (UTIAS) and did his postdoctoral training at MIT's Field and Space Robotics Laboratory (FSRL). Jekan Thanga is an assistant professor and heads the Space and Terrestrial Robotic Exploration (SpaceTREx) Laboratory at the University of Arizona. He is the Engineering Principal Investigator on the AOSAT I CubeSat Centrifuge mission and is a Co-Investigator on SWIMSat, an Airforce CubeSat mission concept to monitor space threats.