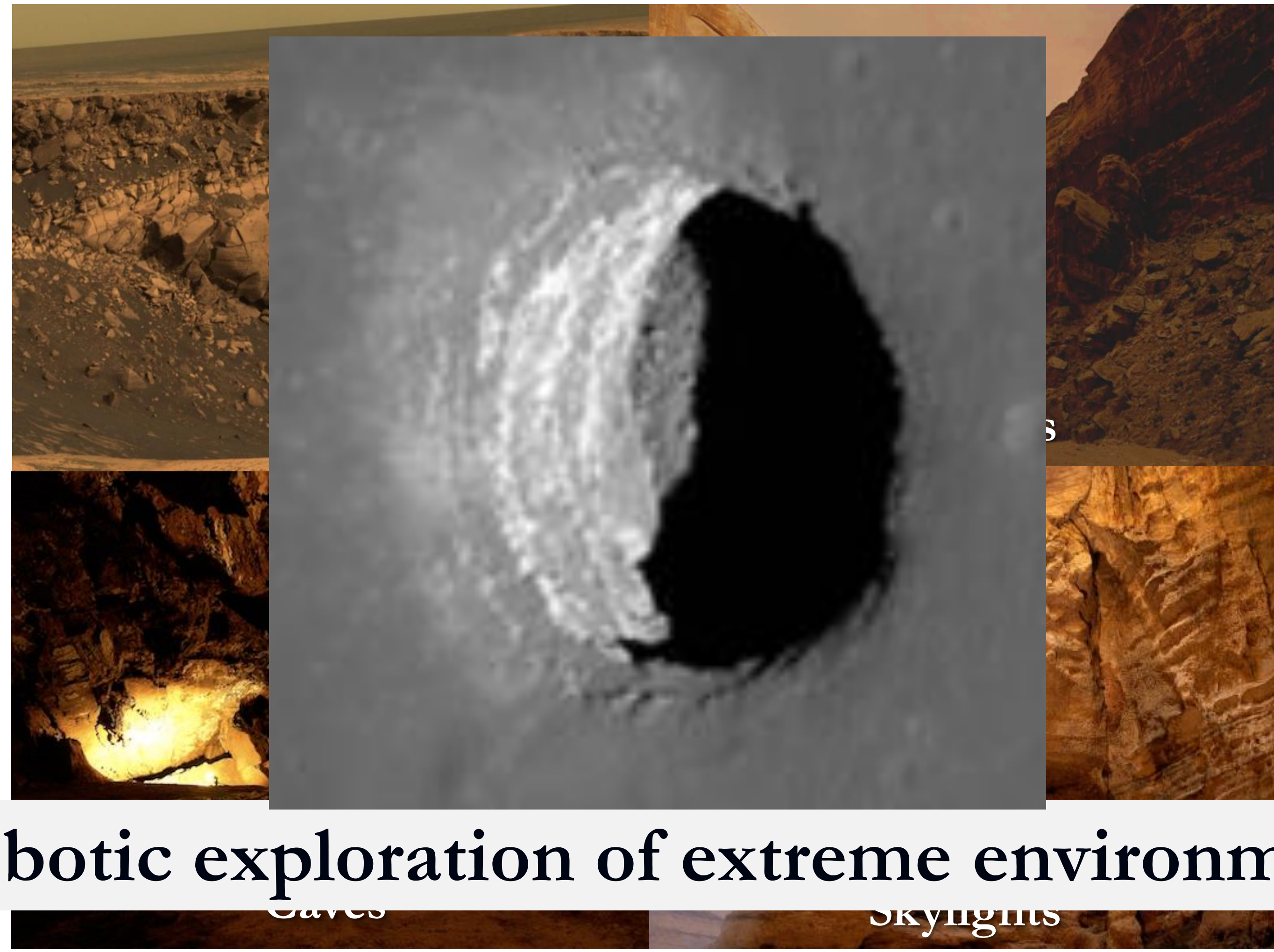
Exploring Off-World Lava Tubes and Caves Using Small Robots

Himangshu Kalita, Jekan Thangavelautham
Space and Terrestrial Robotic Exploration (SpaceTREx) Laboratory
Aerospace and Mechanical Engineering Department
University of Arizona

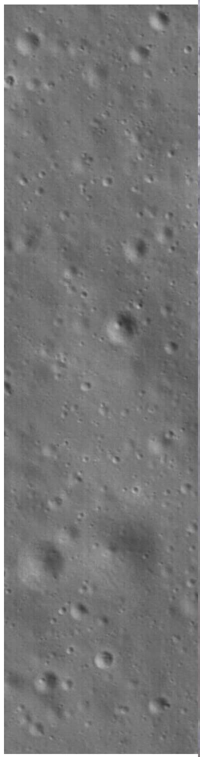
Outline

- Introduction
- Motivation
- System Overview of SphereX
- Ballistic Hopping Dynamics and Control
- Multi-robot algorithm for Navigation and Path-planning inside lava tubes
- Simulations
- Discussions
- Conclusion

Introduction



Robotic exploration of extreme environments



So

re
the
ars by

tability

nuch
and

Challenges

- Mars, Moon
  - Low-gravity, low traction
- Surface Contact Risks
  - Fine dust, static charge, porosity
- Planning, Navigation, Comms., Tracking
  - Unknown terrain, limited field of view

Tackling all of these challenges using a systems engineering approach

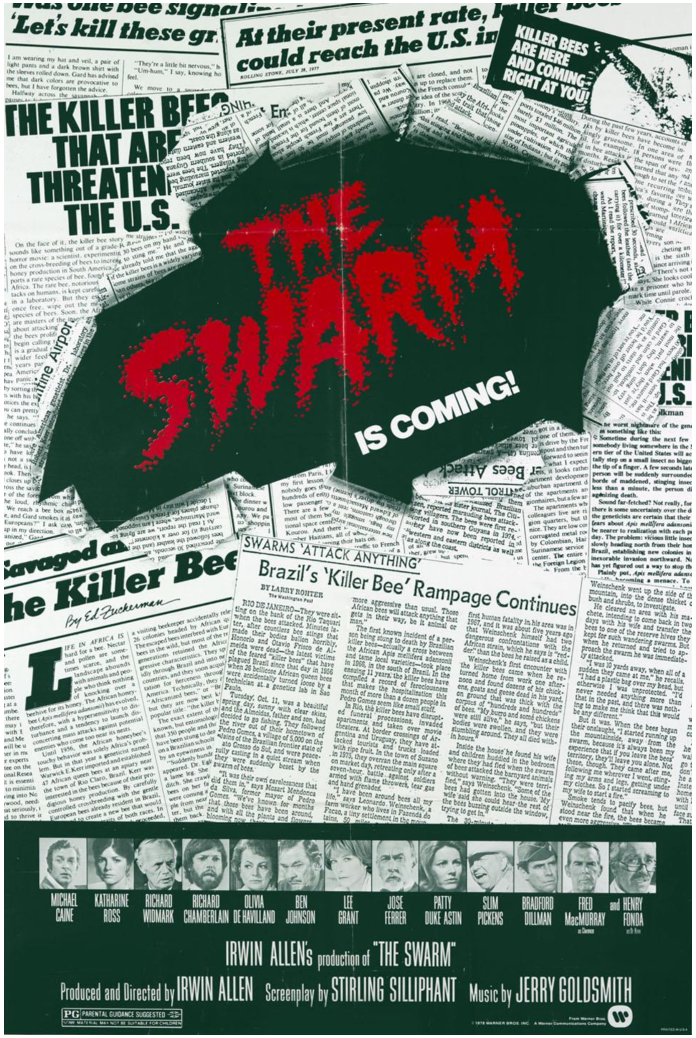
Low-Gravity Environments

Ease of mobility by hopping…

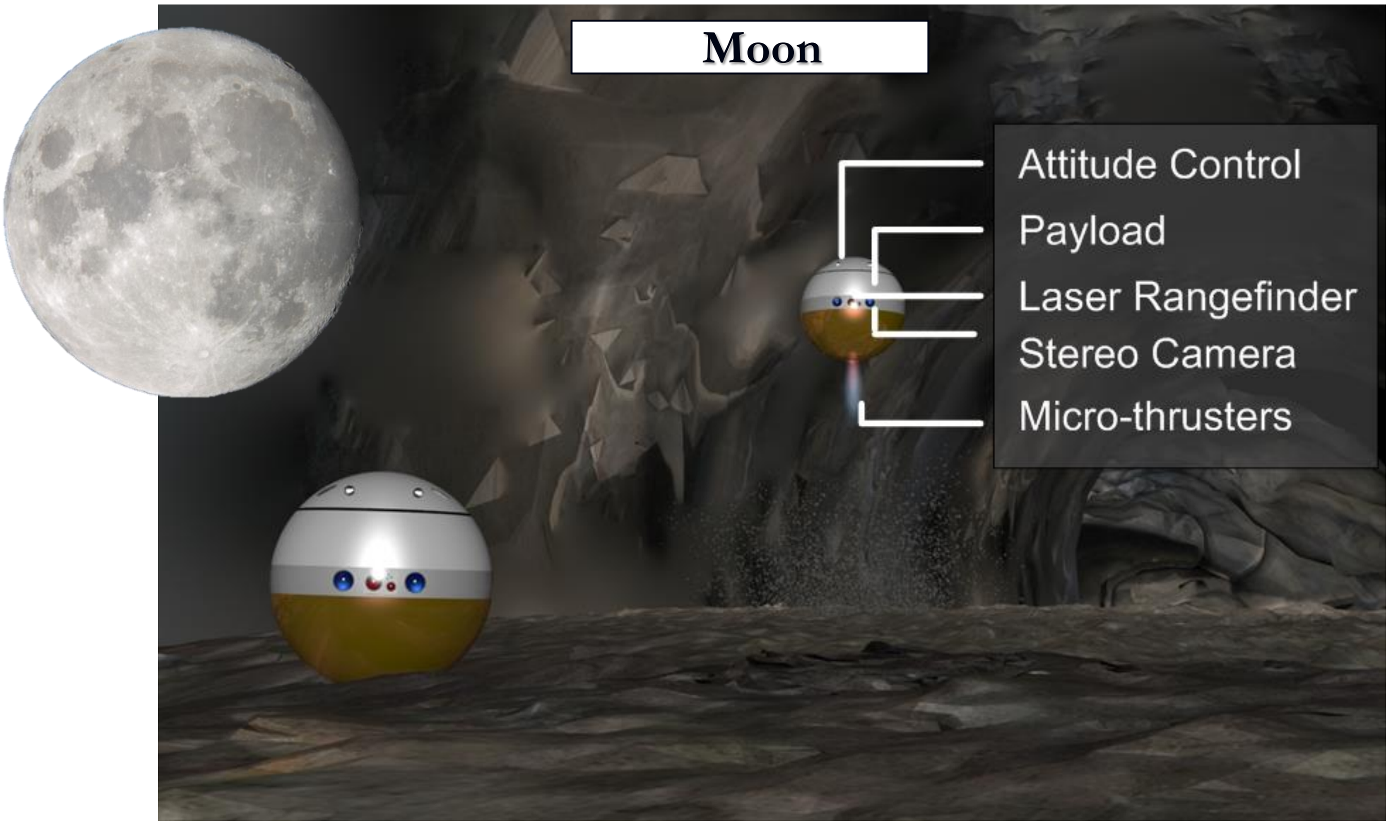
Approach

- The high risk involved all but eliminates use of a large lander to do it all.
- Use swarms of disposable robots that perform distributed science.

"Failure is an Option!"

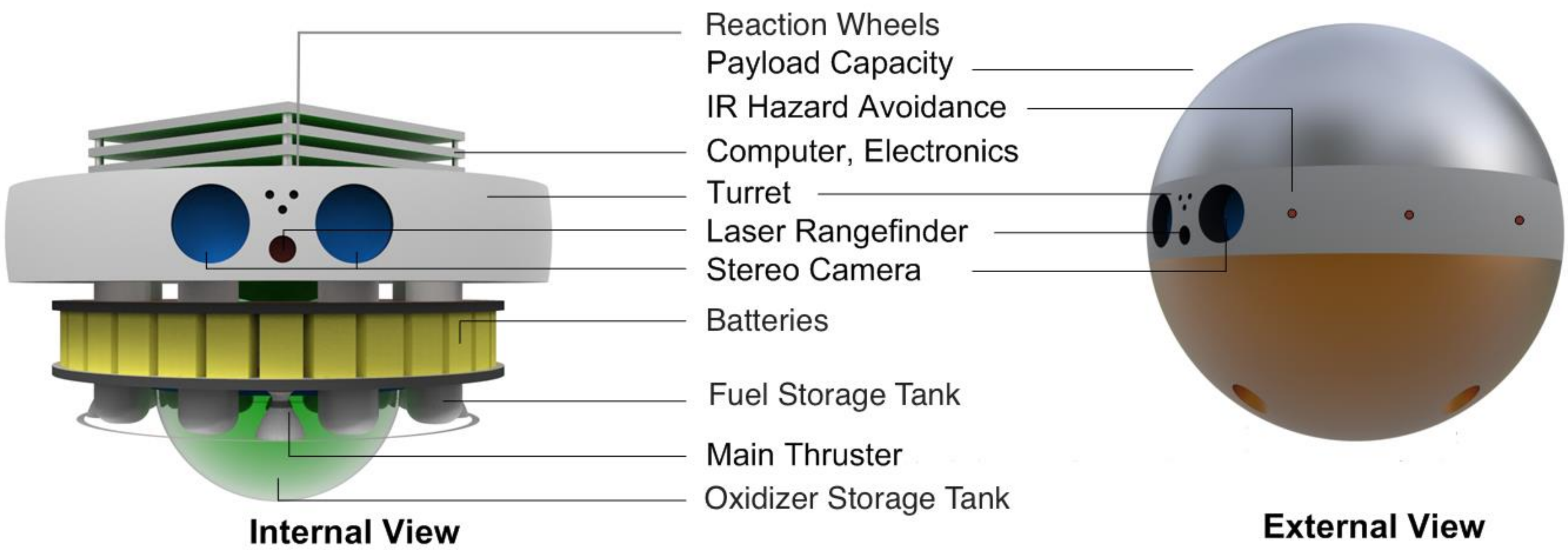


SphereX Robots



3 kg mass, 30 cm diameter robot that can fly, hop and roll

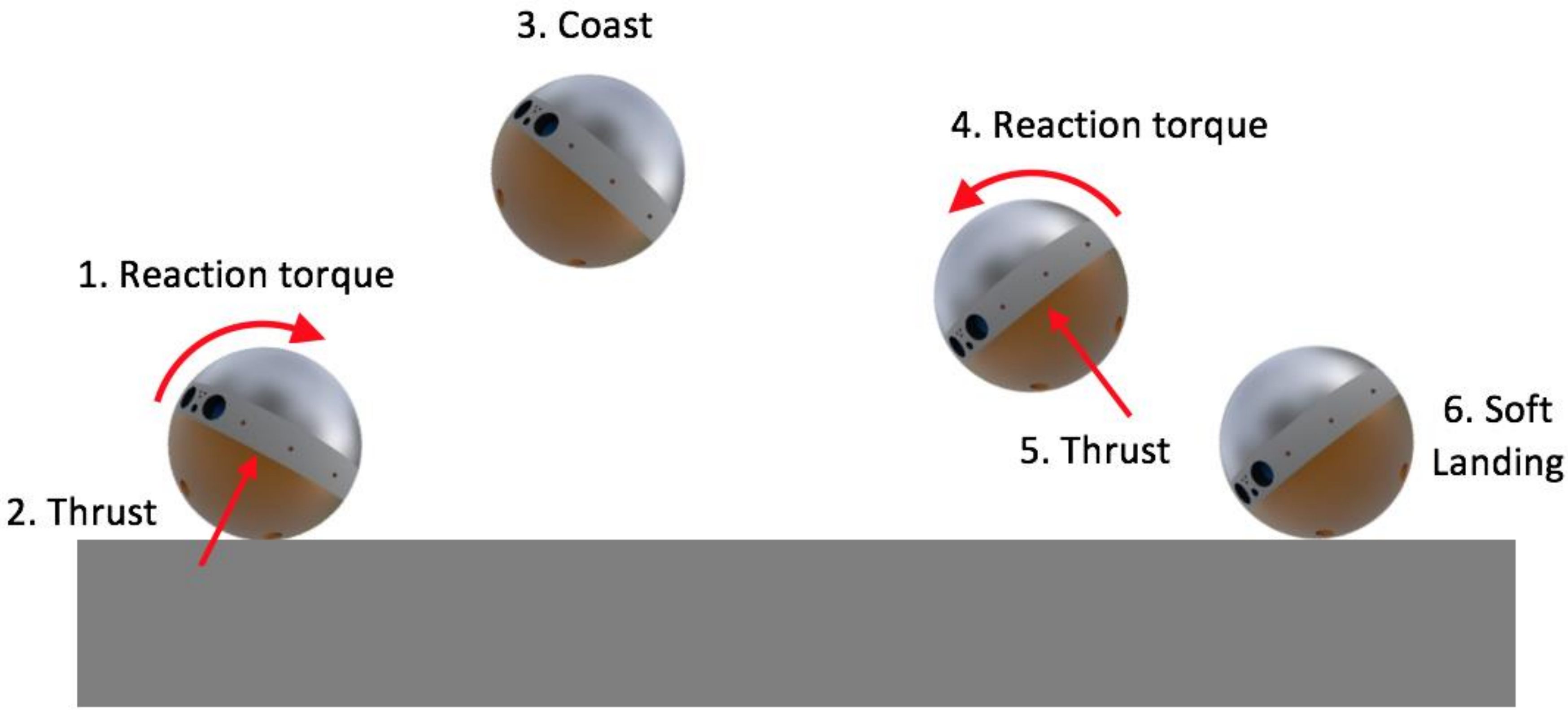
SphereX Robots



- **Attitude Control: Reaction-wheels, IMU**

Smartphone powered drone for off-world environments

Ballistic Hopping



Ballistic Hopping Dynamics

- Simplified model and calculation of initial velocity
  - Robot needs to hop from rest position r_{t0} with velocity v_{t0} and impact position r_{tf} with velocity v_{tf}
  - $d = d_x\hat{\imath} + d_y\hat{\jmath}$ is the vector connecting the initial position to the final position
  - g is the acceleration due to gravity vector and τ is the transfer time
  - $v_{t0} = v_x\hat{\imath} + v_y\hat{\jmath} + v_z\hat{k}$, where $\boldsymbol{v_x} = \frac{d_x}{\tau}$, $\boldsymbol{v_y} = \frac{d_y}{\tau}$, $\boldsymbol{v_z} = \frac{g\tau}{2}$

Trajectory Optimization

- **Pinpoint soft landing can be achieved with two impulsive thrust to achieve initial and final delta v of**

$$\Delta v_1 = v_{t0} \text{ and } \Delta v_2 = -v_{tf}$$

- **Optimization objective is to minimize the fuel consumption and the optimal index can be expressed as**

$$J = \int_0^{\tau} \|T\| \, dt$$

Desired Orientation

- The normalized cross product of the z-axis of the inertial frame and the delta v vector provides the orientation axis of rotation.

$$n = [0\ 0\ 1]^{\mathrm{T}} \times \Delta v,\ u = \frac{n}{\|n\|}$$

- The angle of rotation, φ is calculated as

$$\varphi = \cos^{-1}\left(\frac{[0\ 0\ 1]\cdot\Delta v}{\|v_{t0}\|}\right)$$

- And finally the desired quaternion is calculated as

$$e_0 = \cos\frac{\varphi}{2},\ \ e = \begin{Bmatrix} e_1 \\ e_2 \\ e_3 \end{Bmatrix} = u \sin\frac{\varphi}{2},\ p = \{e_0 \quad e_1 \quad e_2 \quad e_3\}$$

Ballistic Hopping Dynamics

- **Attitude control system maintains the desired Euler angles and Angular velocities to perform ballistic hopping**

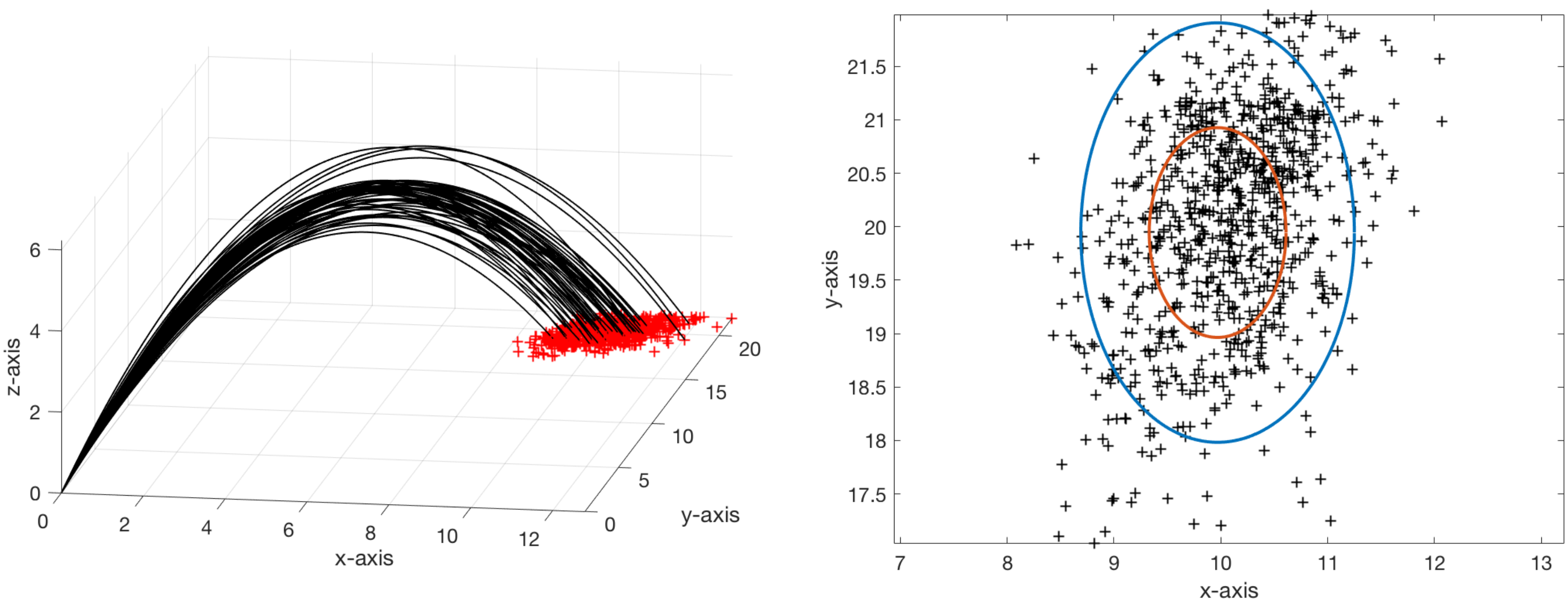
$$\dot{\vec{\omega}} = J^{-1}\left[-\vec{\omega} \times \overrightarrow{H_{tot}} + \overrightarrow{\tau_{rw}} + \overrightarrow{\tau_{dist}}\right]$$

- **Developed PD control algorithm that generate control torque inputs as a function of attitude errors**

$$\overrightarrow{\tau_{rw}} = -K_p(e_{des} - e_{act}) - K_d(\omega_{des} - \omega_{act})$$

- **Main thruster provides thrust along +z axis of body frame**

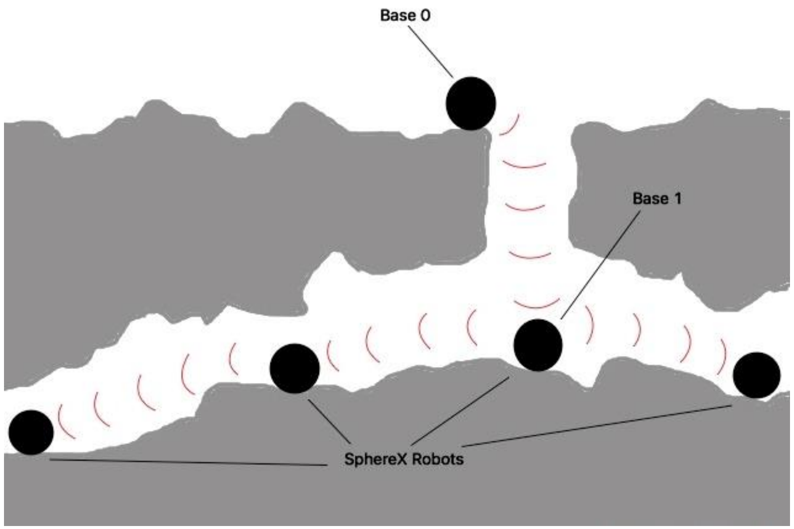
Ballistic Hopping Dynamics



Monte Carlo histories: (left) three-dimensional ballistic hopping trajectories, (right) landing ellipse (1σ and 2σ values)

Navigation Inside Caves/Lava Tubes

- No line of sight
- Communication signals are blocked due to rocks in the way
- Requires setting up communication relays
- We envision that the robots need to cooperate in the form of a bucket brigade to establish a multi-hop communication link



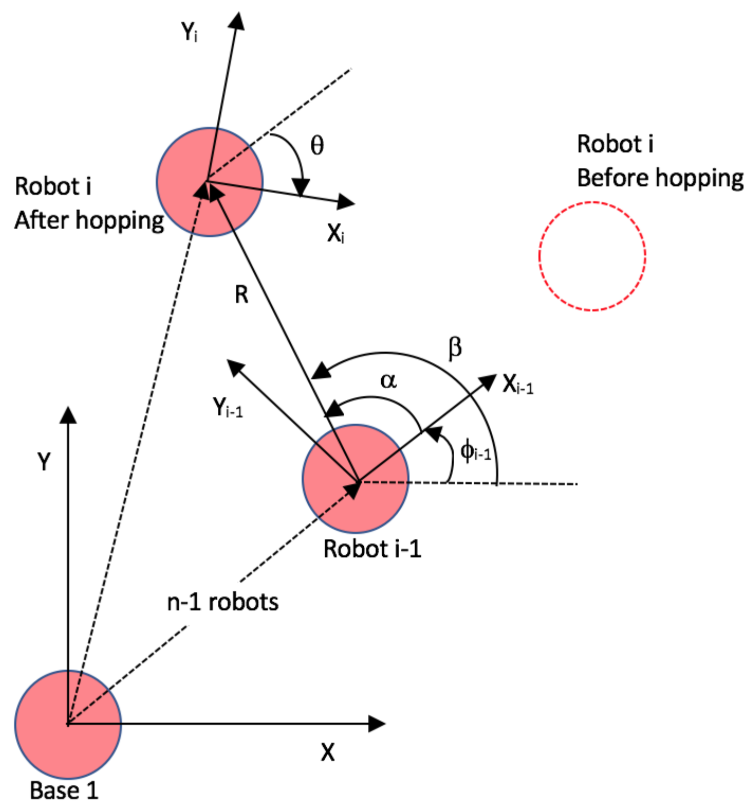
Multi-hop communication link strategy

Robot Position Measurement and Localization

- Inside lava tubes localization system such as GPS are not available
- Each robot is equipped with 2D laser scanner mounted on a servo
- Global frame (X,Y) is constructed w.r.t the fixed robot (Base 1)
- At any instant the relative position and orientation of a robot is measured w.r.t a neighboring stationary robot and then can be converted to global position and orientation

$$\beta = \alpha + \phi_{i-1}$$

$$\begin{pmatrix} x_i \\ y_i \\ \phi_i \end{pmatrix} = \begin{pmatrix} x_{i-1} \\ y_{i-1} \\ \phi_{i-1} \end{pmatrix} + \begin{pmatrix} R\cos\beta \\ R\sin\beta \\ \theta \end{pmatrix}$$

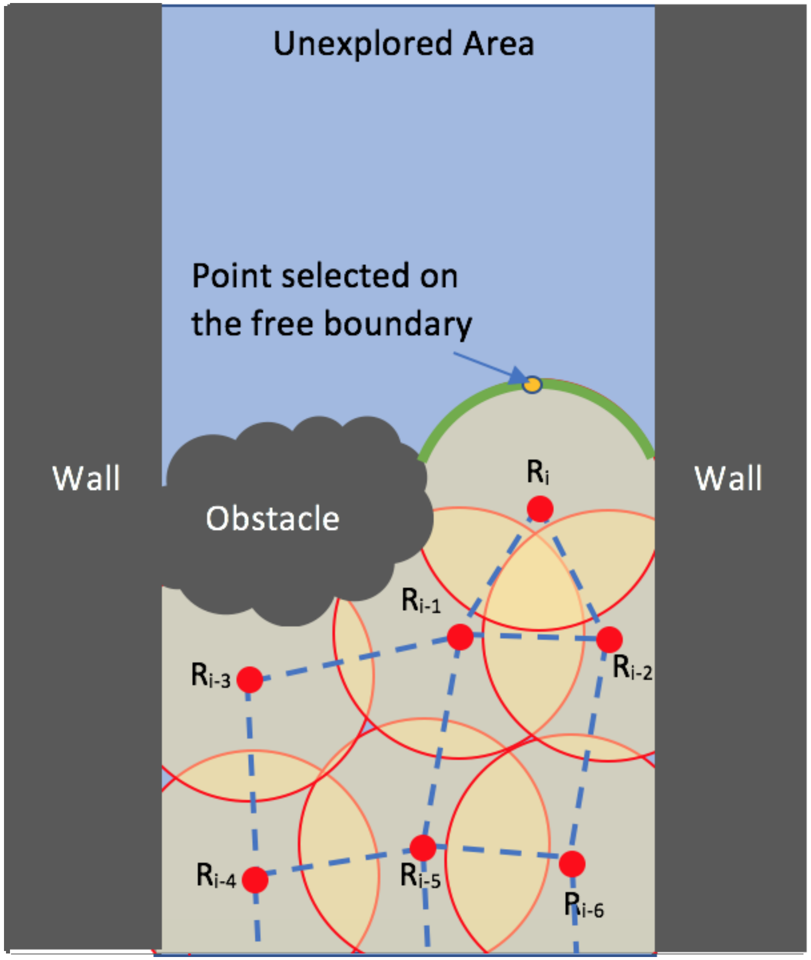


Multi-robot Algorithm

- A network of robots uses data from the already explored area for path planning and moves forward one at a time
- The network has to maintain communication link with the base station
- The cave environment is modeled as a grid with circular obstacles of different size
- Each cell of the grid is termed explored once it falls within the sensing radius of any robot
- The exploration is directed through the selection of free boundaries from a given configuration

Multi-robot Algorithm

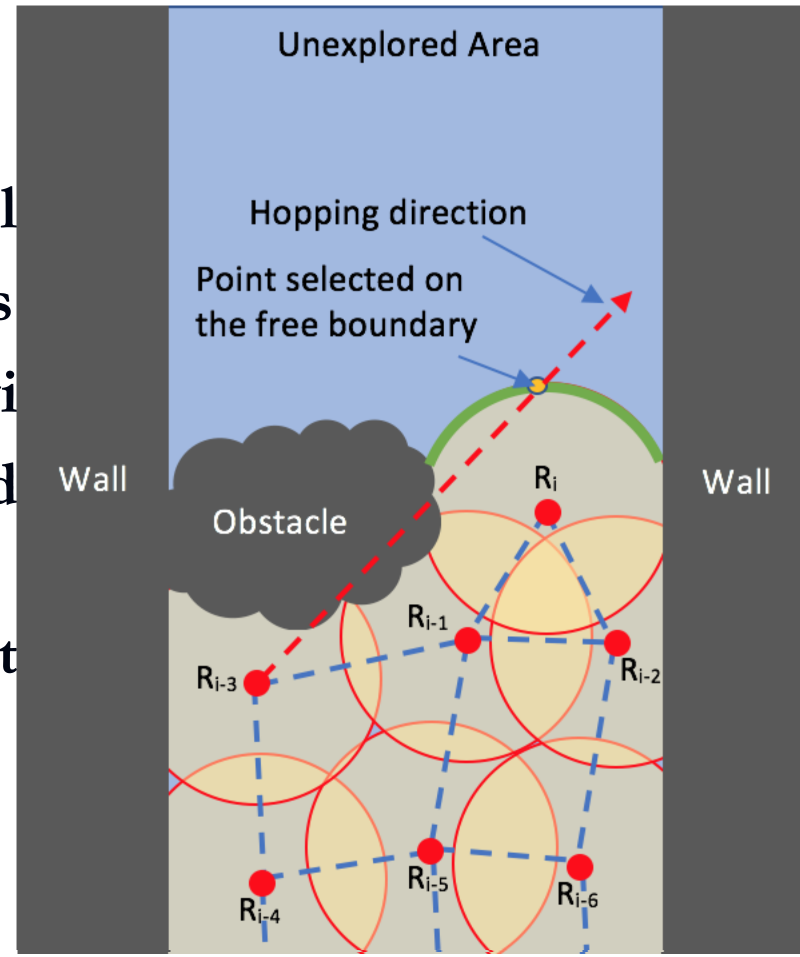
- Cave Environment built and modeled as a grid
- Robot sensors detects surrounding
- Unexplored free boundary identified
- Random point selected on free boundary



Multi-robot Algorithm

- Unit vector between selected point and robot defines hopping direction
- Robot hops to its new position
- The unexplored free boundary is updated
- Moreover, at every instant, the algorithm validates its hopping direction and distance.

- Next the hopping distance is cal
  - The communication graph is
  - The final position must lie wi
  - The hopping distance should range of SphereX
- The algorithm computes the opt velocity v_{t0} and v_{tf}



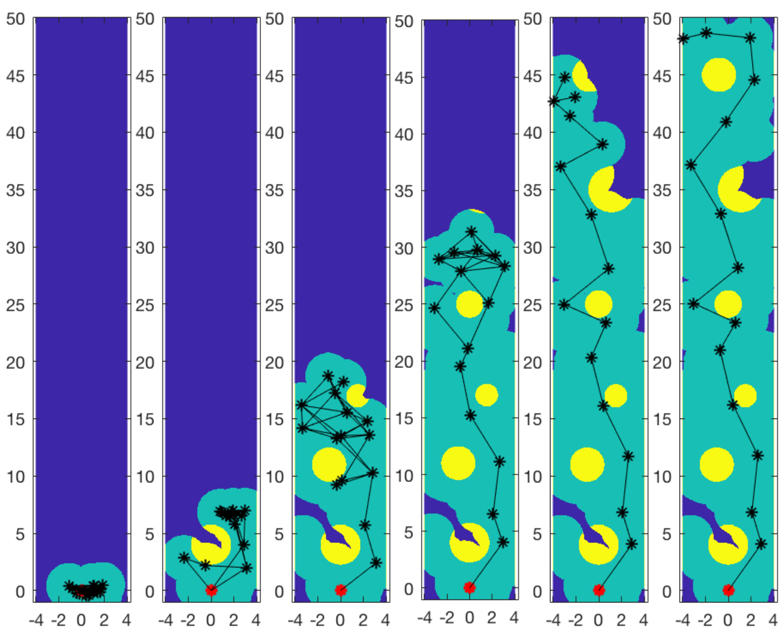
Multi-robot Algorithm

Algorithm: Multi-robot path planning for hopping robots
Require: Initial position, orientation for each robot;

```
for k = 0 to K do
    for i = 1 to N do
        Update explored grid cells;
        Identify obstacles;
        Compute free boundary;
        Select random point on free boundary;
        Compute hopping direction;
        Verify hopping direction;
        Compute hopping distance;
        Verify hopping distance;
        Move robot i to new position;
        Update explored grids and obstacles;
        Compute new free boundary;
        Set i = i+1;
    end for
    Set k = k+1;
end for
```

Simulations

Scenario I: The robots should always be in connection with the base robot

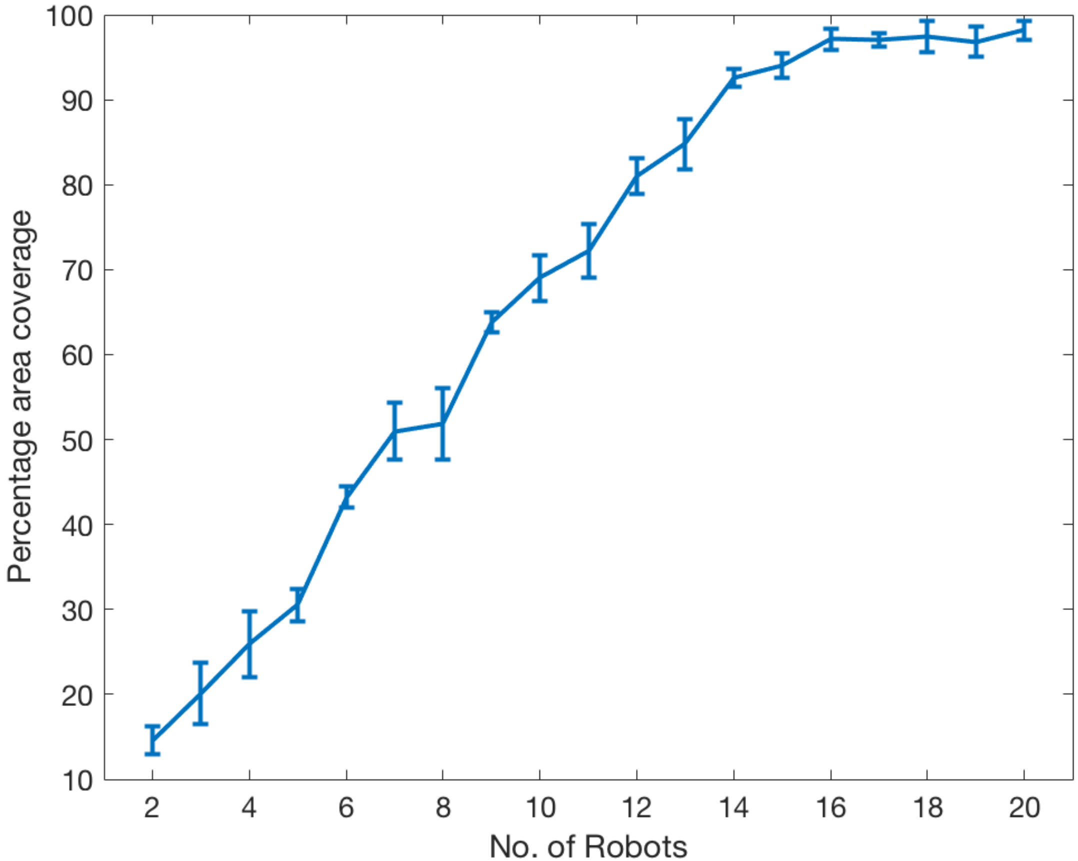


Simulation of a system of 15 robots and a base robot at timestep 0, 2, 5, 10, 15 and 20

- **The environment is built as a grid structure of length 50 units and width 8 units represented by 800x5000 grids.**
- **It consists of 6 circular obstacles of different radius.**
- **Each robot is considered to have a vision radius of 2 units, communication range of 5 units and a hopping range of 7 units**

Simulations

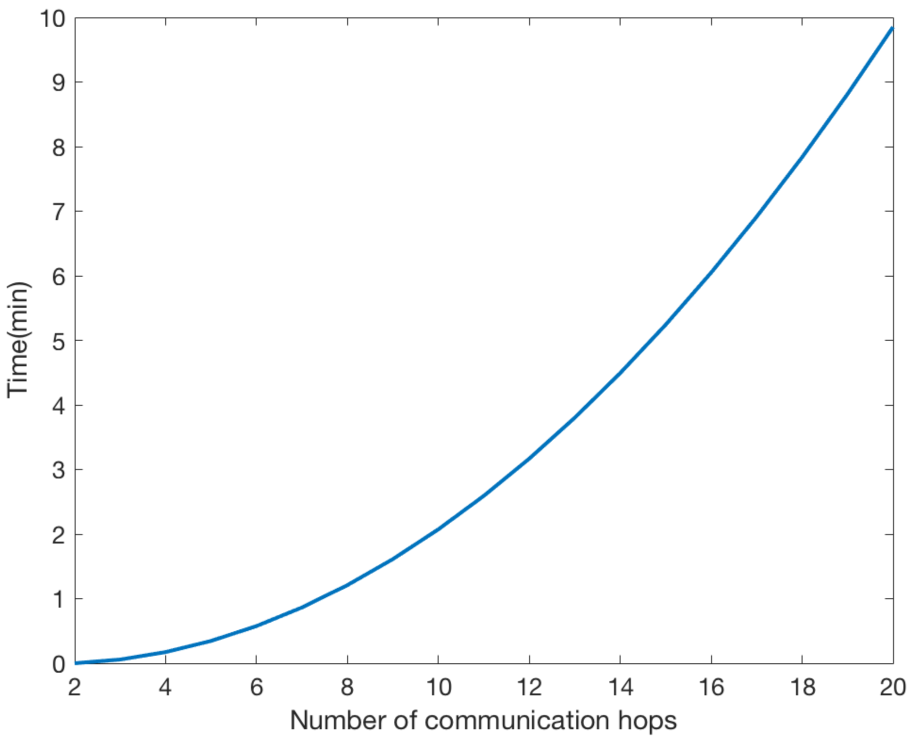
Simulations



Average percentage of area covered and standard deviation over 10 runs

Multi-robot Algorithm

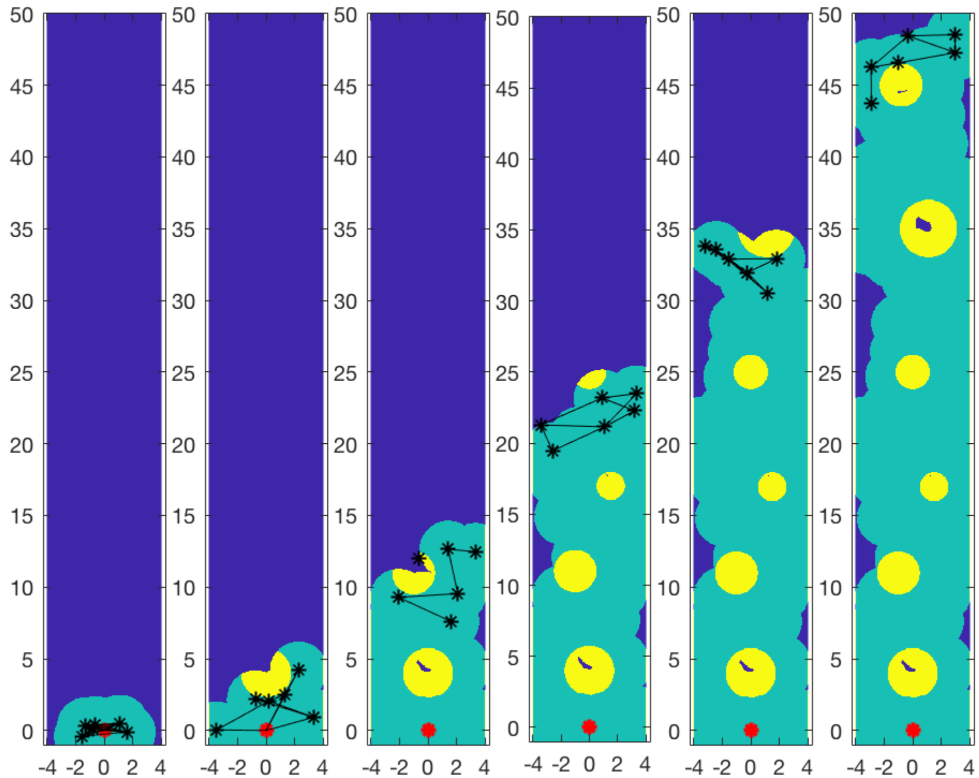
Parameter	Value
No. of Communication hops	2-20
Data size	1 MB
System Noise Temperature	200 K
Minimum Eb/No for reception	10 dB
Antenna range	500 m
Channel bandwidth	20 kHz
Pointing loss of antenna	18 dB
Data packet size	1024 bits



Transmission time for 1MB data through multiple hops

Multi-robot Algorithm

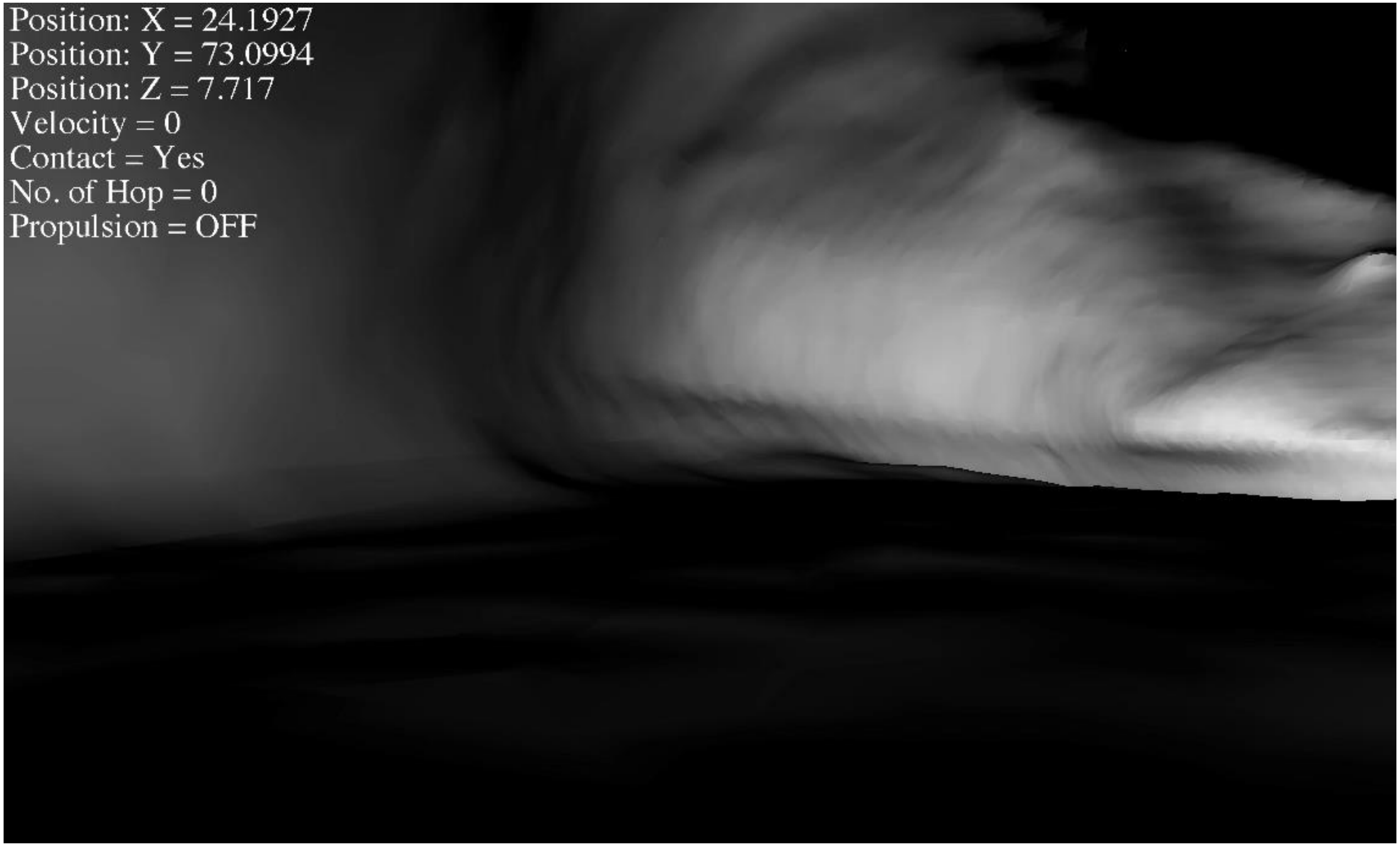
Case II: The robots can explore as a swarm without maintaining connection with the Base robot



Simulation of a system of 6 robots at timestep 0, 2, 6, 11, 15 and 21

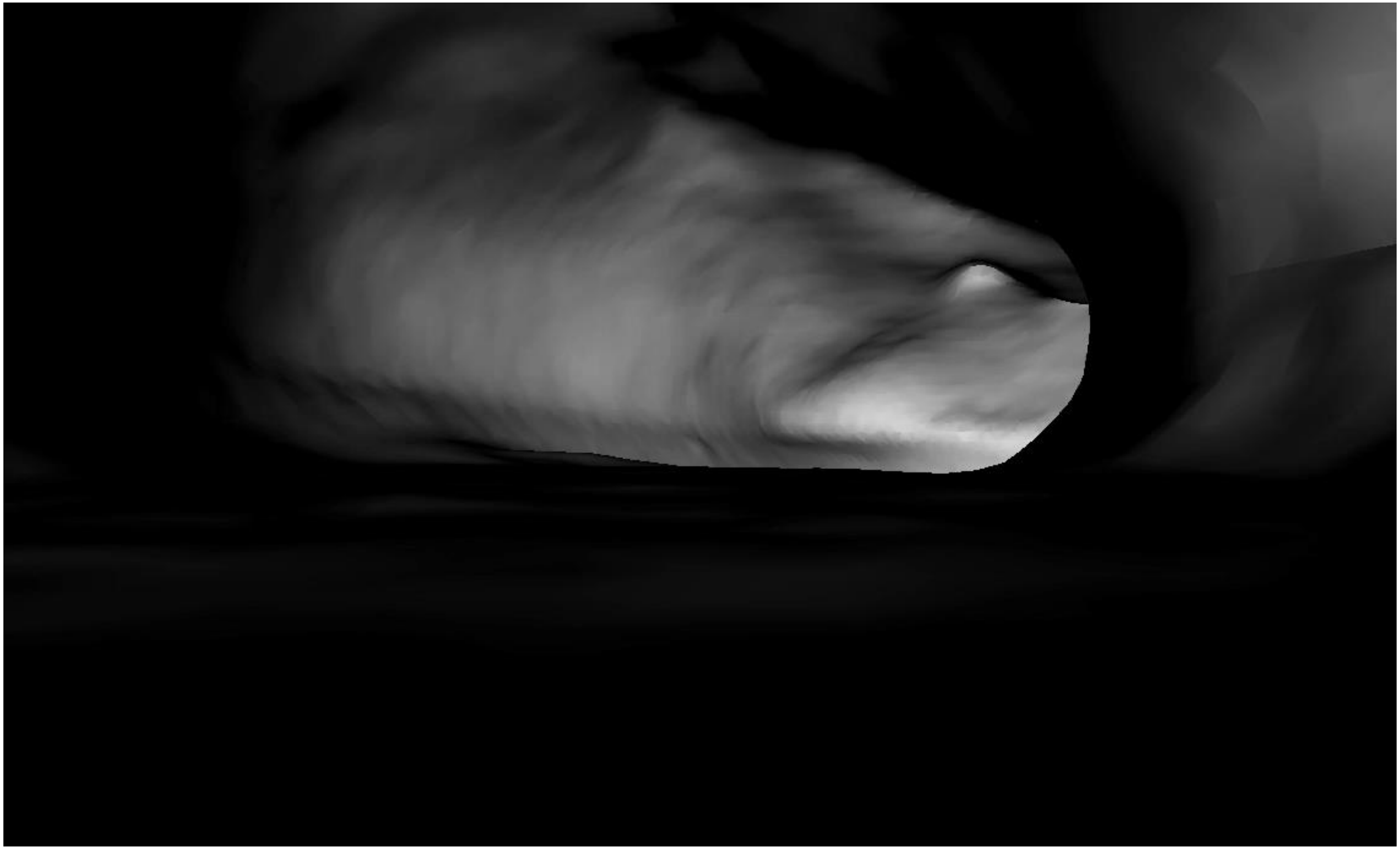
Simulations

Animation


Position: X = 24.1927
Position: Y = 73.0994
Position: Z = 7.717
Velocity = 0
Contact = Yes
No. of Hop = 0
Propulsion = OFF

SpaceTREx

Animation



SpaceTREx

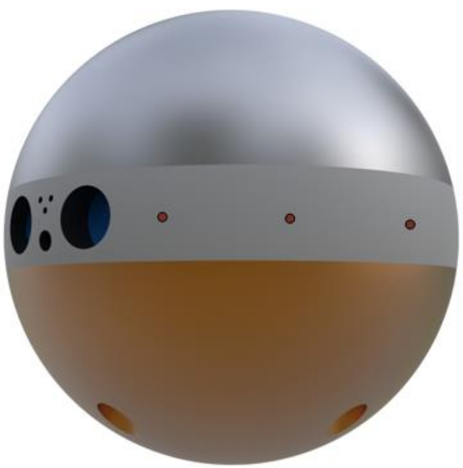
Conclusions

- Presented a spherical robotic platform (SphereX)
- Detailed dynamics and control for mobility
- Proposed an algorithm for multi-robot navigation and path-planning in unknown environments like caves and lava tubes
- The proposed concept will allow mapping of these extreme environments compiled into a 3D point cloud
- Future work will include testing the algorithm on real cave environments along with hardware demonstrations

Thank You

Questions ?

Mass Budget for Each SphereX



Major Subsystem	Mass (kg)
Computer, Comms, Electronics	0.2
Power	0.3
Stereo Camera, Laser Rangefinder	0.3
Propulsion	1.2
ADCS	0.4
Payload	0.6
Total	**3.0**

Smartphone-class electronics, sensors

Trajectory Optimization

- Distance travelled per hop has a huge impact on the number of hops possible and the total distance covered.
- Fig. for 1 Kg of propellant with Isp 350s on Moon and Mars.