Kobu: An efficient extraterrestrial space filling curve based multi-robot coverage planner

Raghav Thakar[1], Akshat Pandey[1] and Gokul P[1]

***Abstract*— Multi-robot teams have been used for a variety of navigation and mapping applications. They have proven to provide better time efficacy and output than using a singular robot in doing the same. A place where such an implementation may have a lot of potential is in the exploration of planets and other similar terrestrial bodies, which is what this study aims to focus on. With space research thriving more than ever, a decentralized multi-robot system has the potential to deliver results many more times than a single ground robot would. This paper introduces an optimized space-filling technique and tries to achieve autonomous multi-robot navigation without any path overlap.**

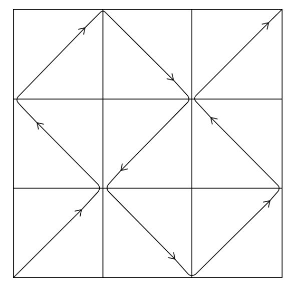
I. BACKGROUND AND INTRODUCTION

Planetary exploration is an area of interest that has become quite prevalent. Gathering information on other possible terrestrial bodies in the hope of someday finding an environment where the human race could manage to survive has been a goal of many space researchers. Taking Mars as an example, we have come a long way from the initial mars rover, **Pathfinder** in 1997 to **Perseverance** and the **Ingenuinity** helicopter in 2020. Over these 24 years, much has been learned about the martian terrain, and many enhancements have been made to technology and research, but is there another way to explore planetary terrains? This study presents an approach to utilizing the idea of a multi-UGV (Unmanned Ground Vehicle) system in a *dispersion* phase to help traverse the terrain workspace a lot faster and efficiently. This work implements a space-filling curve-based planner so that no area is left unexplored and avoids redundant navigation to already visited locations simultaneously. Using a multi-robot system would help in the rapid gathering of basic data and provide for a fail-safe compensation by other robots in case one member of the team were to lose functionality somehow. The environment that has to be explored and the rate at which information retrieval is needed can be core deciding factors as to how many members one would want to implement in the system; in future cases, such a system could be made compatible with other types of robotic entities to formulate a fully autonomous multi-robot extraterrestrial exploration network.

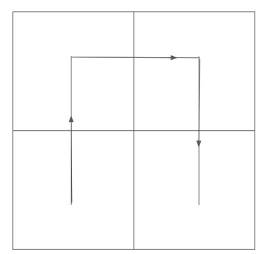
II. METHODOLOGY

A space-filling curve is a two-dimensional curve that passes through every reachable location of an N-dimensional region. It is a parameterized function that maps a unit line segment to a continuous curve in the unit square, cube, hypercube, etc., which gets arbitrarily close to a given point in the unit cube as the parameter increases.

There are many different variations of Peano's Curve. We have leveraged the Hilbert's Curve in addition to Peano's curve for our purpose. The unit square outputs for their implementation are shown in Figure 1



(a) Unit square output for Peano curve



(b) Unit square output for Hilbert curve

Fig. 1: Unit square output for Hilbert's and Peano's curves.

1) Environment setting:

- Robot properties: There are n robots that are initially considered to be dispersed.
- Sensor properties: Taking into consideration the amount of noise that might be present when we are traversing to the target waypoint,

$$\hat{d_i} = d_i + \epsilon_i \tag{1}$$

where ϵ_i is the noise component of the i^{th} robot. $\hat{d_i}$ has a Gaussian distribution with mean error $m_i = m_{LOS}log(1 + d_i)$ and a variance σ_i^2. The probability density function of $\hat{d_i}$ is given as:

$$p(d_i \hat{|} d_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp[-\frac{(\hat{d_i} - d_i - m_i)^2}{2\sigma_i^2}] \tag{2}$$

- Environment properties: The environment is unbounded and the particles are randomly initialized in the environment with velocities v_i.

[1]The authors are affiliated to Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India

2) General construction: The basic idea for constructing Peano's Curve is to continuously divide each square into smaller squares and determine a path or curve to go through each square.

The Peano curve can be built via a succession of stages whereby a set S_i of squares and a sequence P_i of centers in the squares are built by the i^{th} step in the previous step.

In stage i each S_{i-1} square will be divided into nine smaller equal squares, and its center point c will be replaced with a continuous series of these nine smaller squares' centers. It consists of grouping n of the smaller squares in $log_2 n$ columns. In each column, the centers are ordered contiguously, and the columns are then ordered from one side of the square to the other in a fashion equivalent to the length of the side of the smaller squares.

The mathematical formulation of Peano's curve utilizes a periodic function on the real line. Let

$$p(t) = \begin{cases} 0 & for \quad 0 \le t \le 1/3 \\ 3t-1 & for \quad 1/3 \le t \le 2/3 \\ 1 & for \quad 2/3 \le t \le 1 \end{cases} \tag{3}$$

and extend $p(\cdot)$ to the entire real line by taking $p(-t) = p(t)$ and $p(t+2) = p(t)$. Define

$$x(t) = \sum_{k=1}^{\infty} \frac{p(3^{2k-2}t)}{2^k}, y(t) = \sum_{k=1}^{\infty} \frac{p(3^{2k-1}t)}{2^k} \tag{4}$$

for each $t \in [0, 1]$.

We have also extrapolated our work for the Hilbert Curve, which is a variant of the Peano curve where the square is divided into 4 instead of 9 and the method is continued. The formula for Hilbert's curve is given as follows. Let $t \in [0, 1]$ be represented in its base four expansion,

$$t = 0_4 q_1 q_2 q_3 ...,$$

and,

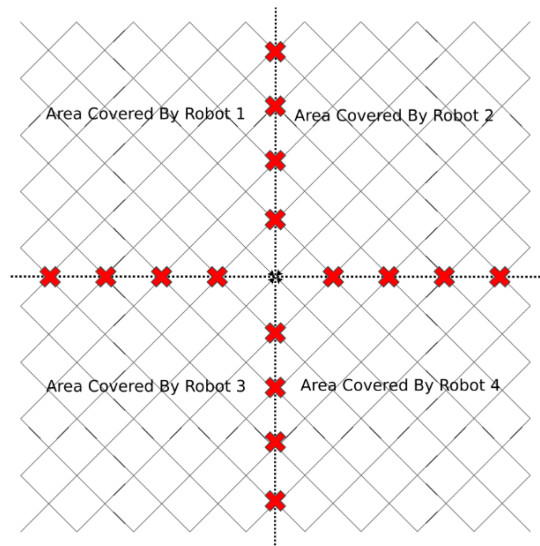
$$sign(x) = \begin{cases} 1 & if \quad x > 0 \\ 0 & if \quad x = 0 \\ -1 & if \quad x < 0 \end{cases} \tag{5}$$

then,

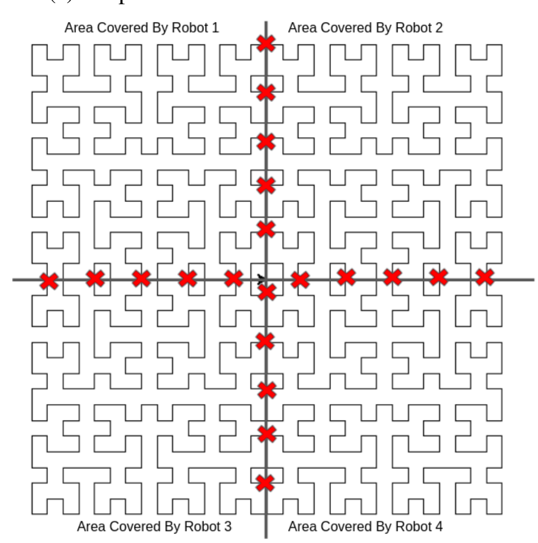
$$h(t) = \sum_{j=1}^{\infty} \frac{(-1)^{e_{0j}}}{2^j} sign(q_j) \begin{pmatrix} (1-d_j)q_j - 1 \\ 1 - d_j q_j \end{pmatrix} \tag{6}$$

where e_{kj} denotes the number of k's preceeding q_j and $d_j = e_{0j} + e3j$.

3) Robot coordination: To avoid a negative impact on the benefits, multiple robots must be able to communicate effectively. This can be very useful to mitigate issues like multiple robots mapping the same area and even when any one of the robots fails, other robots can take its place. Because of the symmetric nature of the Peano Curve, and considering the velocity of the robots to be the same, as shown in Figure 2, each pair of robots will meet at every symmetric point on the entire environment, this has been leveraged to exchange information about the areas covered by the robots to not go over the same area multiple times. To account for uncalled circumstances, the robots wait at the meeting point until the paired robot comes into contact or a certain amount of time lapses. The convergence also serves as validation in cases where any of the robots malfunction or stop working altogether. In such cases, the paired robot takes up the exploration area of the malfunctioned robot; this is done after the robot completes covering the area assigned to it and then moves on to the next area.



(a) Output from the visualiser for Peano Curve



(b) Output from the visualiser for Hilbert Curve

Fig. 2: This figure depicts the output from the visualizer. The points marked in red are some of the points of intersection of the robots. These act as information transfer centers for the robots.

III. Solution Quality and ongoing research

This section presents the initial results of our abstract. Robot exploration and terrain mapping in an extra-terrestrial environment is the main aim of the project. To validate our results, we have tested it on Gazebo 9 as the simulator with ROS as the middleware. Hence, to ensure that no edge case is left hanging, we have pushed the limits of our simulation to extreme scenarios. The simulations show 4 robots trying to explore the environment based on the algorithms discussed below. To make it realistic to the real-world environment, we have also added noise to the sensors. As the output from the simulations and the visualizer shows, the robots can successfully explore the entire environment including uneven terrains and craters. To add to our algorithm, we are currently working to include collision avoidance in the robots. We also plan to extrapolate the results to compare against some of the well-known space-filling curves including the Lebesgue Curve.

Algorithm 1: Multi-robot exploration based on Hilbert Curve

```
Function HilbertWaypoint(level, angle, ln)
    if level=0 then
        Move robot forward by length=ln;
    else
        RobotTheta=90;
        Robot right by angle = RobotTheta;
        HilbertWaypoint(level-1,-angle, ln);
        Move robot forward by length=ln;
        Robot left by angle = RobotTheta;
        HilbertWaypoint(level-1, angle, ln);
        Move robot forward by length=ln;
        HilbertWaypoint(level-1, angle, ln);
        Robot left by angle = RobotTheta;
        Move robot forward by length=ln;
        HilbertWaypoint(level-1, angle, ln);
        Robot left by angle = RobotTheta;
Function Communication(waypoint_1, waypoint_2)
    if waypoint_1 ≠ waypoint_2 then
        Hold at the point of symmetry;
    else
        continue
    if time ≥ max_time then
        Interrupt;
        Reallocate area;
    else
        continue
    while i ≤ n do
        HilbertWaypoint(level, angle, ln)
```

The video of our submission can be viewed: Youtube Video- https://youtu.be/NxmvM7g0AX4

The code has been released as an open-sourced repository: Github Repository- https://github.com/raghavthakar/kobu

Algorithm 2: Multi-robot exploration based on Peano Curves

```
Function PeanoWaypoint(level, ln)
    if level=0 then
        Move robot forward by length=ln;
    else
        RobotTheta=90;
        PeanoWaypoint(level-1, ln/3);
        Robot right by angle = RobotTheta;
        PeanoWaypoint(level-1, ln/3);
        while m < 2 do
            while n < 3 do
                Robot left by angle = RobotTheta;
                PeanoWaypoint(level-1, length/3);
            RobotTheta = -(RobotTheta);
        Robot left by angle = RobotTheta;
        PeanoWaypoint(level-1, length/3);
Function Communication(waypoint_1, waypoint_2)
    if waypoint_1≠ waypoint_2 then
        Hold at the point of symmetry;
    else
        continue
    if time ≥ max_time then
        Interrupt;
        Reallocate area;
    else
        continue
    while i ≤ n do
        PeanoWaypoint(level, ln)
```

References

[1] J. E. de Freitas, R. F. de Lima, and D. T. dos Santos, "The n-dimensional Peano Curve," São Paulo Journal of Mathematical Sciences, vol. 13, no. 2, pp. 678–688, Apr. 2019, doi: 10.1007/s40863-019-00132-9.

[2] K. M. Wurm, C. Stachniss, and W. Burgard, "Coordinated multi-robot exploration using a segmentation of the environment," 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sep. 2008, doi: 10.1109/iros.2008.4650734.

[3] S. Sand et al., "Swarm exploration and navigation on mars," 2013 International Conference on Localization and GNSS (ICL-GNSS), Jun. 2013, doi: 10.1109/icl-gnss.2013.6577272.

[4] "On the geometrization of the Peano curve and the arithmetization of the Hilbert curve," International Journal of Mathematical Education in Science and Technology, 2021. https://www.tandfonline.com/doi/abs/10.1080/0020739920230309.

[5] S. V. Spires and S. Y. Goldsmith, "Exhaustive geographic search with mobile robots along space-filling curves," Collective Robotics, pp. 1–12, 1998, doi: 10.1007/bfb0033369.

[6] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," IEEE Transactions on Robotics, vol. 21, no. 3, pp. 376–386, Jun. 2005, doi: 10.1109/tro.2004.839232.

[7] P. S. Schenker et al., "Robotic automation for space: planetary surface exploration, terrain-adaptive mobility, and multirobot cooperative tasks," Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision, Oct. 2001, doi: 10.1117/12.444181.

[8] J. J. Lopez-Perez, U. H. Hernandez-Belmonte, J.-P. Ramirez-Paredes, M. A. Contreras-Cruz, and V. Ayala-Ramirez, "Distributed Multirobot Exploration Based on Scene Partitioning and Frontier Selection,"

Mathematical Problems in Engineering, vol. 2018, pp. 1–17, Jun. 2018, doi: 10.1155/2018/2373642.
[9] D. Saur and K. Geihs, “IMPERA: Integrated Mission Planning for Multi-Robot Systems,” Robotics, vol. 4, no. 4, pp. 435–463, Oct. 2015, doi: 10.3390/robotics4040435.
[10] Bader, M., 2012. Space-filling curves: an introduction with applications in scientific computing (Vol. 9). Springer Science Business Media.
[11] Sagan, H., 2012. Space-filling curves. Springer Science Business Media.