Locomotion Mode Selection Plus (LMS+) Algorithm for Resource Efficient Outdoor Navigation

Amir Sharif[a] and Hubert Roth
Institute of Control Engineering, University of Siegen, Germany

Keywords: Mobile Robots, Path Planning, Autonomous Outdoor Navigation, Dual Mode Locomotion, Hybrid Fly-drive Robots, Locomotion Mode Selection.

Abstract: Mobile robots that can fly in air and drive on ground may offer fast and energy efficient navigation. In this paper, a unique method has been proposed for optimal outdoor navigation of robots that can fly and drive. An offline dual mode path planning algorithm is developed and it is named as Locomotion Mode Selection Plus (LMS+) algorithm. It uses a two dimensional ground route from a web based geographic map server and makes a three dimensional resource optimised path, by considering dual mode locomotion. The results showed that the LMS+ algorithm makes a path that is optimised for travel time and energy consumption. The output path can be directly given to dual mode robots for resource optimised autonomous outdoor navigation.

1 INTRODUCTION

Autonomous navigation of mobile robots is challenging field with many real world applications i.e. search and rescue missions, agriculture, package delivery, transportation, and in industry (Al-Faiz and Mahameda, 2015). Autonomous navigation requires localization, mapping, path planning, path following, and dynamic obstacle avoidance.

Autonomous navigation of mobile robots can be classified as indoor navigation and outdoor navigation. Indoor navigation is relatively difficult because the robot needs a map of the environment and then it has to localize itself in the map. Outdoor navigation is relatively easier, as GPS assisted localization and 2D map of the surroundings is easily available (Santana et al., 2015).

Mobile robots may have different methods (modes) of locomotion i.e. flying by hovering, flying by wings, driving with wheels, and walking on legs etc. The two common modes of locomotion in the field of mobile robots are: flying by hovering and driving with wheels. These locomotion modes have their own advantages and disadvantages. Flight is agile method of locomotion, but it consumes high power. Driving on the ground is slower method of locomotion and it consumes low power.

Hybrid mobile robots have multi-mode locomotion. Hybrid mobile robots with dual mode locomotion can fly in air and drive on ground. Because of additional weight, the power consumption by a robot that can fly and drive is slightly higher as compared to robots that can fly only or drive only. But fly-drive robots may offer overall optimal energy consumption and travel time during navigation.

They are recently being introduced in the field of Surveillance and package delivery i.e. AT Panther Delivery Drone developed by Advance Tactics Inc. is an autonomous vehicle that can fly and drive. Another application of dual mode locomotion is in military vehicles i.e. Flying Truck is a huge vehicle capable of driving and flying. A recent application of dual mode locomotion is in personal transport vehicles i.e. Flying Taxi developed by Audi and Airbus, can fly in air like a Quadcopter and drive on road like a car.

The challenge is to select the locomotion mode autonomously. The decision about flight or drive may depend upon type of terrain, obstacles in the path, size of obstacles and resources (energy and travel time) consumption. Our goal is to make the decision about flight or driving based on the resource consumption. The decision about flight or drive can be offline or online. We use offline planning method to decide where the robot should fly and where it should drive to optimise resource consumption. This planning method is named as LMS+ algorithm.

[a] https://orcid.org/0000-0003-1690-1594

Sharif, A. and Roth, H.
Locomotion Mode Selection Plus (LMS+) Algorithm for Resource Efficient Outdoor Navigation.
DOI: 10.5220/0008120705730579
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019)*, pages 573-579
ISBN: 978-989-758-380-3

Our focus is the outdoor navigation of hybrid mobile robots. A two dimensional map of the environment and route to the target is already available. The LMS+ algorithm makes a path using dual mode locomotion and this path will be optimal with respect to resource consumption.

1.1 Related Work

Single source shortest path search algorithms like: Breadth first search algorithm, Greedy best first search algorithm, Dijsktra's algorithm and A* algorithm are used to find the shortest or cheapest path in a grid / graph (Amgad et al., 2017).

In this paper (Araki et al., 2017), they used a 3D graph to model the environment. They used a graph search algorithm for multi robot path planning for swarm of robots that can fly and drive. Their proposed method gave optimal results for travel time and energy consumption.

In this paper (Sharif et al., 2018), we used A* algorithm for autonomous navigation with dual mode locomotion. A simulation model of Hybrid Fly-Drive Robot) HyFDR was developed. It can fly in air like a Quadcopter and drive on ground with four powered wheels. A 3D map of the virtual world was created. We mathematically calculated the energy cost of aerial and ground nodes. A* algorithm was used to find the minimum energy path in a 3D grid. The distance between nodes is same but their movement cost is different. The simulated model of HyFDR follows this path and navigates to the target avoiding obstacles. The navigation with the given path ensures minimum energy consumption.

In our previous research paper (Sharif et al., 2019), we developed a physical model of HyFDR. This robot can fly in air like a Quadcopter and also drive on ground with four wheels. The outdoor GPS waypoint navigation was tested during flight mode and drive mode. We developed LMS algorithm which decides whether to fly or drive during outdoor navigation. This algorithm can only make either flight path or drive path. The navigation with the selected path ensures either minimum energy consumption or minimum travel time.

Brute force methods search methods i.e. breadth first search, depth first search, and Dijsktra's algorithm search the whole configuration space, but they are inefficient for large configuration spaces. Heuristic based methods i.e. best first search, hill climbing, A* and genetic algorithms uses goal oriented search (Sanjoy, 2019). A unique method of generating visibility graphs in cluttered environment and finding shortest path is given in here (Welzl, 1985). Dynamic programming optimization methods such as Bellman-Ford algorithm (Bellman, 1958) is an optimization for brute force methods. Another approach is to treat a path like a polyline and then using polyline smoothing techniques (Abam et al., 2007).

For single objective trajectory optimization, Hill climbing technique (Wang et al., 2016) can be used and for multi-objective trajectory optimization there are several methods available i.e. weighted sum, Genetic algorithms, Particle swarm optimization and Ant colony optimization (Saicharanl et al., 2016).

1.2 Our Contribution

Most of methods mentioned in subsection 1.1, find optimal path usually in a single edge graph. But in case of dual mode locomotion, there can be two edges between two vertices. This makes the implementation of graph search algorithms difficult. Also, the development of a graph consisting of two different types vertices is complex task especially for large number of nodes.

Instead of graph search algorithms, we used a technique similar to polyline smoothing or trajectory smoothing to make a dual mode resource optimised path. Our proposed method to select the locomotion mode is simple, and easy to implement as compared to the graph search based methods.

1.3 Problem Statement

Given a mobile robot that can fly and drive, its energy consumption and travel time is desired to be optimized during autonomous outdoor navigation.

1.4 Solution

Our proposed solution is based on the tools that are already available and easy to use. Instead of making a new 3D map or using a 3D map of the outdoor environment, we used an open source web based 2D map server and router to get a route on ground. This approach avoids reinventing the wheel, thus saves time and it is easy to implement for real world applications. We developed an offline dual mode path planning algorithm called LMS+ algorithm. It gets a route on the ground and makes optimal path considering energy consumption and travel time. This path can be given to Hybrid Fly-Drive Robot (HyFDR), which follows the waypoints and loco-motion mode during autonomous navigation.

1.5 Assumptions

Our proposed solution will be for autonomous outdoor navigation of HyFDR. To simplify our solution, we assumed that the velocity and power consumption by HyFDR during flight and driving will remain constant. The flight path is 20 meters high above the ground and without any obstacles. The ground route is only possible path for driving on the ground; any path away from ground route will be a flight path.

2 METHODOLOGY

Open Source Routing Machine (OSRM) is an open source web user interface for map server and router. It provides 2D map of world and road network data. User can give start and target point on the map and OSRM uses routing algorithm to find the shortest route between start and target position. The route can be saved as GPX (GPS Exchange Format) file, which contains a list of GPS waypoints (latitude and longitude). Figure 1 show the output of OSRM, the black line is the ground route and the red dots are the waypoints. The start position is at first waypoint and the target position is at last waypoint.

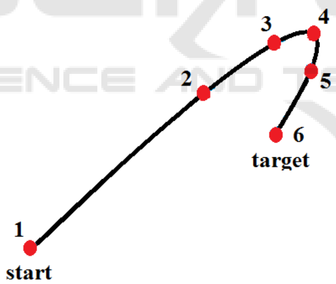

Figure 1: Ground route (black line) and waypoints (red).

LMS+ algorithm (detail in section 2.3) runs on Matlab. It loads the ground route and uses a technique called 'Best shortcut in polyline' to make roughly optimal path. This technique is explained in next section.

2.1 Best Shortcut in a Polyline

Best shortcut in a polyline is similar to polyline smoothing method. It uses brute force to find the best shortcut in polyline. It is not required to retain the original shape of polyline, so the resultant may not look like the original polyline. Figure 2 shows the same ground route as in shown in figure 1, but now it is treated as a polyline (black line) with total length L_{16}. A shortcut from point 1 to point 6 in this polyline is d_{16} (blue line). It is the shortest distance between point 1 and point 6.

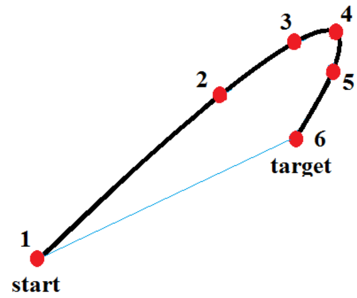

Figure 2: A polyline (black line) and shortcut (blue line).

To measure the quality of a shortcut, we have defined a constant called 'length ratio'. It is the ratio of shortest distance between two points and the total length of the curve between two corresponding points. It is represented by symbol σ and can be calculated using following equation:

$$\sigma = \frac{d}{L} \tag{1}$$

where d is the shortest distance between two points, L is the total length of curve between two corresponding points. A smaller length ratio σ means a better shortcut. If there are multiple points in a polyline curve then there will be multiple shortcuts available. All possible shortcuts in the polyline are shown in figure 3.

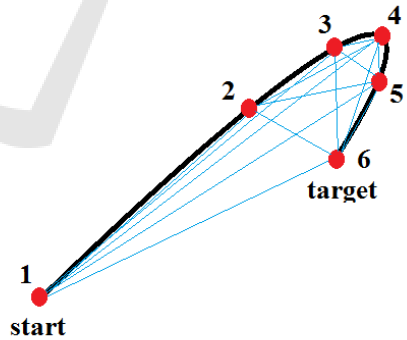

Figure 3: All possible shortcuts in polyline.

To find best shortcut in the polyline of figure 3, we need selection criteria. A best shortcut in a polyline is the one with minimum length ratio σ as compared to the length ratios of other shortcuts. The length ratio σ should be lower than the arbitrary threshold ε. To find the best shortcut, we need to calculate the length ratio σ for all shortcuts of the polyline in figure 3. The value of ε is usually set to

0.5. It is obvious from figure 3 that d_{26} is the best shortcut (rough estimate). The best shortcut will be combined with the original line from start point and to the target point as shown in the figure 4. The resultant line does not look like original long polyline, nor it is the shortest distance from start point to destination; instead it has moderate length.

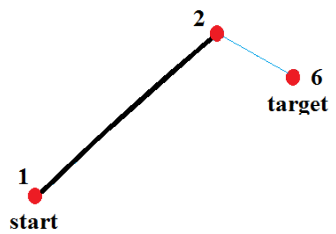

Figure 4: The resultant line and best shortcut (blue line).

We shall use this 'best shortcut in the polyline method' in LMS+ algorithm to find the roughly optimal path. The best shortcut (blue line in figure 4) will be named as intermediate flight path; the original polyline (black line in figure 4) will be the drive path and the resultant polyline is the hybrid path.

2.2 Travel Cost Calculation

The travel cost is the weighted sum of energy cost (energy consumption) and time cost (travel time) of a given path. The equation for travel cost calculation is derived from the 'weighted sum' method for multi-objective optimisation (Athan, 1996). It can be calculated using following equation:

$$C = \alpha \frac{E}{E_{max}} + (1-\alpha)\frac{T}{T_{max}} \tag{2}$$

where C is the travel cost of flight path, α is the scaling factor, T is the time cost, E is the energy cost, E_{max} is the maximum energy and it can be obtained from the maximum value of flight energy consumption and driving energy consumption, T_{max} is the maximum of time cost and it can be obtained from maximum value of flight time cost and driving time cost. The time cost can be calculated by following equation:

$$T = \frac{L}{V} \tag{3}$$

where L is the total length of the flight path, V is the average velocity of HyFDR. The energy cost can be calculated using following equation:

$$E = TP \tag{4}$$

where P is the average power consumption by HyFDR during flight.

2.3 LMS+ Algorithm

The flowchart of LMS+ algorithm is shown in figure 5. The LMS+ algorithm works in Matlab software. In the first step the LMS+ algorithm loads the GPX file into Matlab working space. It extracts the waypoints from GPX file. Matlab has a built-in function called 'gpxread' to read the GPX file. The waypoints are saved in the form of a list containing latitude and longitude. After that it converts the GPS coordinates of each waypoint in the track to UTM coordinates. The UTM coordinates are measured in metric units (meters) which makes and further calculations simple.

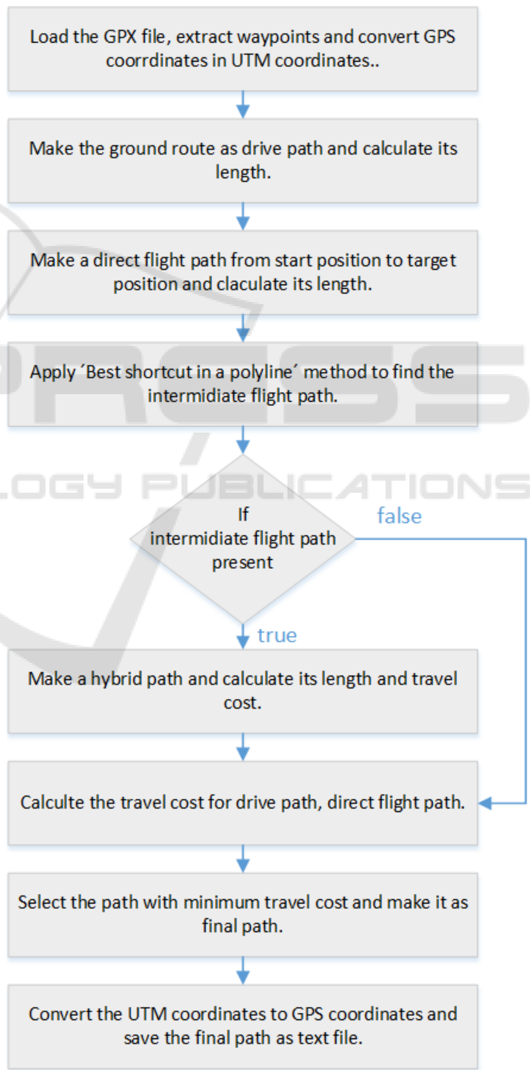

Figure 5: LMS+ algorithm flowchart.

In the next step it converts ground route into drive path by including a mode value equal to zero with each waypoint in the list. The mode value represents the locomotion mode and it is zero for drive path. In the next step it makes a direct flight path from start position to target position. The direct flight path includes horizontal length, takeoff height and landing height.

After that the algorithm saves the points corresponding to the shortcut path and joins them with the drive path from start position and target position on both ends to make a hybrid path. It is assumed that driving is only possible on the drive path; the shortcut path is always for flight. If the best intermediate flight path is present and at least one intermediate drive path is also present than the algorithm combines the intermediate flight and drive paths to make a hybrid path. After that it calculates the length and travel cost of hybrid path, direct flight path and drive path. If there is no intermediate flight path present then the algorithm only calculate the length and travel cost of direct flight path and drive path.

3 EXPERIMENT AND RESULTS

We have created three test cases, in each test case the start and target positions are different which makes three different scenarios and three different types of ground routes.

3.1 Test Case 1

Test case 1 is shown in figure 6. Start position is marked by green point and target position is marked by red point on the map provided by web user interface of OSRM. The blue line is the shortest ground route made by OSRM. The route was downloaded as a GPX file.

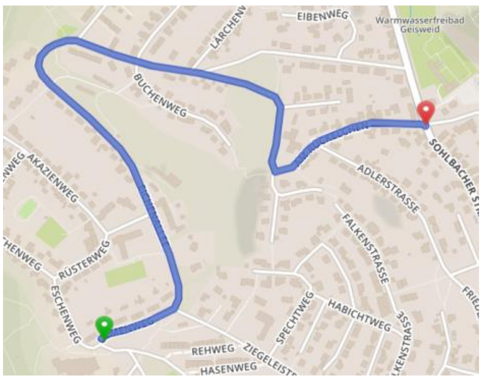

Figure 6: Ground route.

The LMS+ algorithm loads the GPX file and extracts the waypoints. The ground route is stored as a list of waypoints consisting of latitude and longitude. The output of LMS+ algorithm is a hybrid path in this case. As shown in figure 7, the green path is the drive path, the blue colour line is the direct flight path and the red path is the intermediate flight path. By combing the best intermediate flight path with the drive path the algorithm made a hybrid path. In the figure 7 the hybrid path is driving from start position to best intermediate flight path and when the best intermediate flight path is finished then the drive path will lead to the target position.

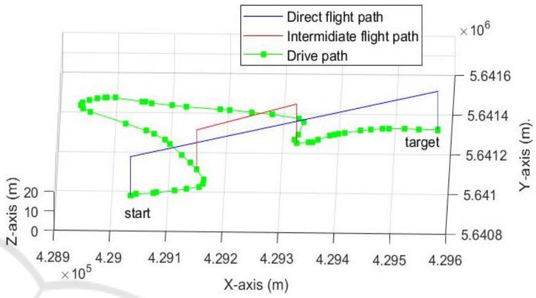

Figure 7: Drive path, flight path and hybrid path.

The algorithm calculated the total length, and travel cost for drive path, flight path and hybrid path. Hybrid path has minimum travel cost as shown in figure 8. The travel cost of driving path and direct flight path is higher than hybrid path. The LMS+ algorithm selected the hybrid path and made it as final path. By following the hybrid path the robot will consume less energy as compared to direct flight path and less time as compared to drive path. The length of hybrid path is less than drive path and more than direct flight path.

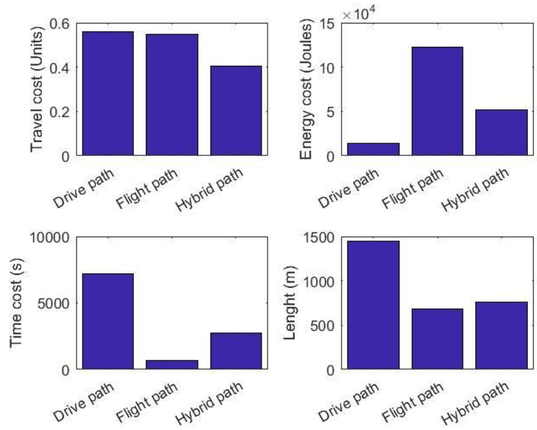

Figure 8: Travel cost (top left), Energy cost (top right), time cost (bottom left, and total length (bottom right).

The time cost of hybrid path is more than direct flight path and less than the drive path. The result shows that following hybrid path reduces the time of travel and energy consumption.

3.2 Test Case 2

In this test case the start and target position are selected in the straight line on the map as shown in figure 9. The green point is the start position and the red point is the target position. The blue line is the ground route generated by OSRM.

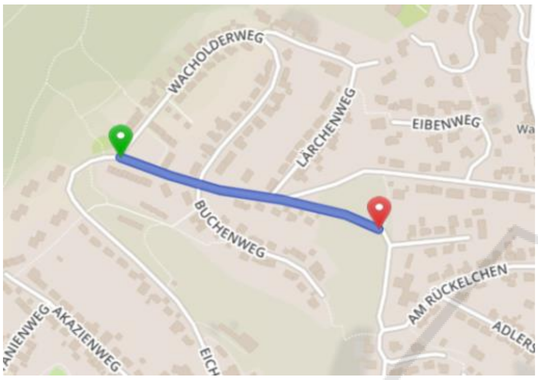

Figure 9: Ground route.

The GPX file is downloaded from the web server and uploaded to Matlab. Figure 10 shows the drive path and the direct flight path. The intermediate path is not present because there is no curve in the path where shortcut can be made. The threshold (limit) in the algorithm makes sure that only best shortcut is selected whose value is above the given limit. In this case the travel cost of drive path is less than the direct flight path so the LMS+ algorithm selected the drive path and made it as final path.

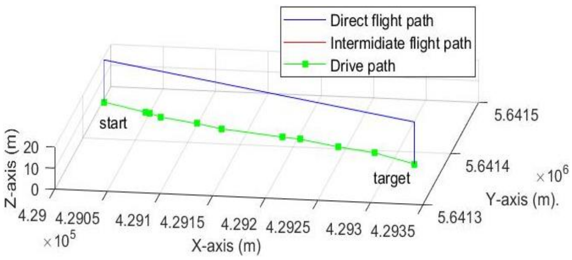

Figure 10: Direct flight path (blue) and drive path green).

The results are shown as bar plot in figure 11. As there is no intermidiate flight path that is why the hybrid path is not created. The travel cost of drive path is lower than the travel ocst of direct flight path htta is wh y the LMS+ algorithm selected the drive path and made it as final path. The drive path has requires more time to reach the destination but the enegy spent is much lower as compared to the flgith path. The length of drive path is also less as compared to the flight path.

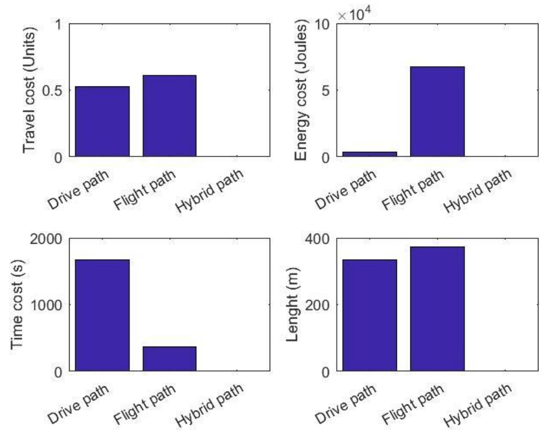

Figure 11: Travel cost (top left), Energy cost (top right), Time cost (bottom left, and total length (bottom right).

3.3 Test Case 3

Figure 12 shows the scenario of third test case. In this test case the start and target position are placed in such a way that the direct flying distance is much less than the driving path's length. The green point shows the start and the red point shows the target position. The blue line is the ground route generated by the OSRM.

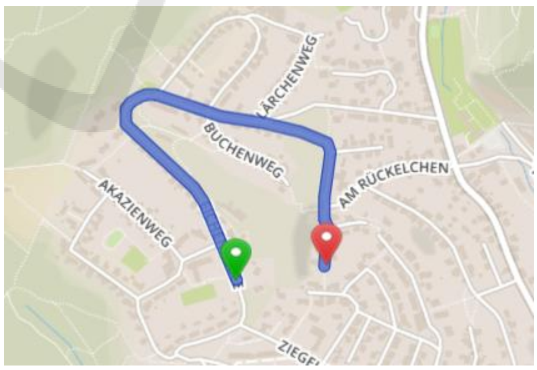

Figure 12: Test case 3 (ground route made by OSRM).

The GPX file of the ground route is loaded into Matlab. The green color line in figure 13 is the drive path and the green points are the GPS waypoints on the ground route. The blue color line is the direct flight path. The LMS+ algorithm found the best intermidiate flight path (best shortcut path) that is same as the direct flight path. There is no hybrid path because there is no intermidiate drive path and the best intermidiate flight path starts and finishes at the end of the drive path.

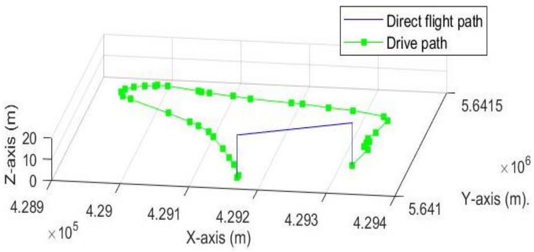

Figure 13: Flight path (blue) and drive path (green).

As shown in figure 14, the travel cost of the direct fligth path is less than the drive path so the LMS+ algorithm selected the direct fligth path. It requires more energy consumption as compared to the drive path but the time to reach the target is less as compared to the drive path. The total length of the drive path is more than the total length of the direct flight path because flight path uses a shortcut upto the target position.

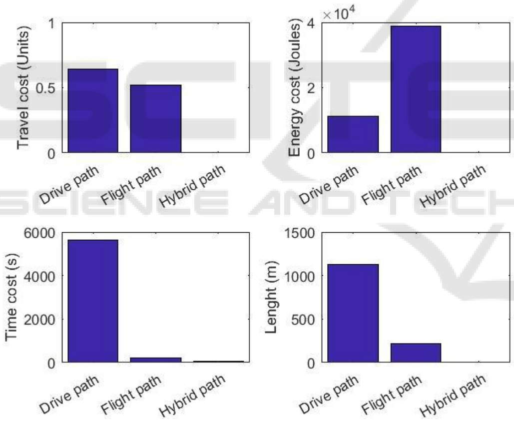

Figure 14: Travel cost (top left), Energy cost (top right), Time cost (bottom left, and total length (bottom right).

4 CONCLUSIONS

The proposed LMS+ algorithm is a simple and unique method that converts a two dimensional ground route into a three dimensional dual mode path. This path is optimised for travel time and energy consumption. We just have to give start and target position on the web user interface and the LMS+ algorithm will generate a path that can be directly given to a robot which can fly and drive.

REFERENCES

Al-Faiz, Z., M., Mahameda, E., G. 2015. GPS based Navigated Autonomous Robot. In *2015 International Journal of Emerging Trends in Engineering Research*, Volume 3, No. 4.

Santana, V., L., Brandão, S., A., and Sarcinelli-Filho, M. 2015. Outdoor waypoint navigation with the AR. Drone quadrotor. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS*), Denver, CO, pp. 303-311.

Amgad, M., Aref, G., W., Rehman, U., F., Rahman, A., M., and Basalamah M., S. 2017. A Survey of Shortest-Path Algorithms, *CoRR abs*/1705.02044.

Araki, B., Strang, J., Pohorecky, S., Qiu, C., Naegeli, T., and Rus, D. 2017. Multi-robot path planning for a swarm of robots that can both fly and drive. In *2017 International Conference on Robotics and Automation (ICRA),* Singapore, pp. 5575-5582.

Sharif, A., & Herath, L., Herath, S., Roth, H. 2018. Energy Efficient Path Planning of Hybrid Fly-Drive Robot (HyFDR) using A* Algorithm. In *2018 Proceedings of International Conference on Informatics in Control Automation and Robotics,* 201-210.

Sharif, A., Choi, S., Roth, H. 2019. A new algorithm for autonomous outdoor navigation of robots that can fly and drive. In *2019 Proceedings of 5th International Conference on Mechatronics and Robotics Engineering,* 141-145.

Sanjoy, D., Rosli, O., Latip, A., Badariyah, N. 2019. A Review on Energy Efficient Path Planning Algorithms for Unmanned Air Vehicles. In *2019 International Carnahan Conference on Security Technology*, Kota Kinabalu, Malaysia.

Welzl, E. 1985. Constructing the visibility graph for n-line segments in O(n2) time. In *1985 Information Processing Letters*, Volume 20, Issue 4, pp 167-171.

Bellman, R. 1958. On a routing problem. *Quarterly of Applied Mathematics*, 87–90.

Abam, M., A., de Berg, M., Hachenberger, P., Zarei, A. 2007. Streaming algorithms for line simplifications. In *1958 ACM Symposium on Computational Geometry (SoCG),* pp. 175–183.

Wang, S., Hussein, A., M., Baudoin, G., Venard, O., and Gotthans, T. 2016. Comparison of hill-climbing and genetic algorithms for digital predistortion models sizing. In *2016 International Conference on Electronics, Circuits and Systems (ICECS),* Monte Carlo, 2016, pp. 289-292.

Saicharanl, B., Tiwari, R., Roberts, N. 2016. Multi Objective Optimization based Path Planning in Robotics using Nature Inspired Algorithms. In *2016 Proceeding of International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES).*

Athan, W. T., Papalambros, Y., P., 1996. A note on weighted criteria methods for compromise solutions in multi-objective optimisation. *Engineering Optimization*, 155-176.