AIAA SciTech Forum
4-8 January 2016, San Diego, California, USA
AIAA Infotech @ Aerospace

10.2514/6.2016-0398

Demonstration Platform for Trajectory Planning of an Autonomous Nonholonomic Skid-Steer System for Investigating Spatial Phenomena

Gordon Keller[1] *, Sebastian Hening[1] †, Sisi Song[1] ‡, Mircea Teodorescu[1] §, Abel Rodriguez[1] ¶

[1] *University of California Santa Cruz, Santa Cruz, CA, 95064*

Corey Ippolito[2] ‖, Nhan Nguyen[2] **

[2] *NASA Ames Research Center, Moffett Field, CA, 94035*

This paper presents a platform built to demonstrate the functionality of a trajectory planning algorithm intended for optimal sampling of spatial phenomena. The installation of this algorithm on a rover is the first step in a series of integration instances with various autonomous vehicles. The platform, consisting of a mobile unit (a skid-steer rover equipped with an autopilot module, onboard computer, radio frequency (RF) sensing unit, and a GPS/Magnetometer sensor) and static RF transmission modules frequently emitting empty packets, simulates expected implementation on larger-scale systems. Using the "Received Signal Strength Indication" (RSSI) from the broadcast packets received by the mobile platform from the arbitrarily placed static RF transmitters, a software recreation of the data field takes place, and a global maximum is sought by the trajectory planning algorithm to go towards during the course of traversal. Recreation of the rover's traversal demonstrates the tendency of the trajectories towards the global maximum in an efficient manner.

I. Introduction

Autonomous robotic systems have evolved into a broad spectrum of application-specific varieties, and among these, mobile systems (including land rovers, rotary systems, fixed-wing aircraft, bio-inspired locomotion, etc.) have invoked a widespread interest in path planning strategies specific to various environments.[5, 9, 14] Efficient payload aggregation and analysis methods have spurred research characterizing the parameters necessary to traverse a field of data in an optimal manner. Studies[7, 8] have been conducted regarding reoptimization of autopilot functionality when considering payload-directed navigation. A trajectory generation algorithm[10] has been proposed regarding the aforementioned goals and parameters corresponding to spatial phenomena. For this paper, a demonstration platform has been designed and developed enabling the application of the algorithm on a skid-steer rover as a proof-of-concept beyond software simulation.

The paper is organized in the following manner. Section II motivates and summarizes the trajectory algorithm used to locate the maximum of a spatial field. Section III is composed of the demonstration platforms design aspects (subdivided into the software and hardware incorporated into the mobile unit and following with the creation of the RF-generated "plume"). In Section IV, the finished demonstration platform is analyzed regarding path creation behavior, restraints corresponding to the platform's scalability,

[*] Undergraduate Researcher, ghkeller@ucsc.edu

[†] Graduate Researcher, Department of Computer Engineering, shening@ucsc.edu

[‡] Graduate Researcher, Department of Applied Mathematics and Statistics, song@soe.ucsc.edu

[§] Department of Computer Engineering, mteodore@ucsc.edu, AIAA member

[¶] Deptartment of Applied Mathematics and Statistics, abel@soe.ucsc.edu

[‖] Advanced Control and Evolvable Systems Group, corey.a.ippolito@nasa.gov

[**] Research Scientist, Lead Advanced Control and Evolvable Systems Group, nhan.t.nguyen@nasa.gov, AIAA Associate Fellow

and the effectiveness in using "received signal strength indication" (RSSI) as the payload source. Future developments in the demonstration platform and path planning algorithm are established in Section V.

II. Primary Previous Research

This paper implements an algorithm detailed in a previous conference paper,[10] which focuses on locating the maximum of an environmental field. Locating the maximum is of interest to those who study the phenomenon and can be used as a proxy for other measurements of interest, such as the source of a gas plume in the case where the source location is unknown. The algorithm uses a Gaussian process[3,4] to learn the distribution of the unknown field from the measurements taken by the vehicle and directly optimizes the vehicle trajectory for future measurements. This approach differs from other work[6,13] in that we are interested in trajectory planning rather than the locations of the vehicle at discrete and possibly distant times (a waypoint-based approach).

Because the measurements arrive sequentially, the problem becomes one of sequential experimental design, and the optimal trajectory consists of many optimal segments $\mathbf{q}_1, \mathbf{q}_2, ...$, each consisting of k locations for sampling. These segments all maximize a reward based on the data collected so far and are stitched together to create the entire trajectory. Here we parametrize the trajectory segments using constant speed circular arcs, and hence constraints manifest in the curvature of the arcs. The vehicle then follows this optimal trajectory segment to collect the k new measurements, whose information is incorporated into the posterior distribution of the field (denoted by f) and the computation of the reward to be maximized by the next optimal trajectory segment.

The use of a Gaussian process prior for the field f results in a posterior distribution of the field that depends on a set of hyperparameters that are estimated from the data. Once these hyperparameters are set, the value of the field at any location follows a normal distribution with mean and variance dependent on the hyperparameters. These values are used in the reward function (1), which is based on the expected improvement criterion[11,12] and is modified for multiple points along a trajectory. An additional penalty term is included to discourage the vehicle from leaving the region of interest. See[10] for more details.

$$\begin{aligned} R(\mathbf{q}) &= \max_{i=1,\ldots,k} \mathrm{E}\left[\max\{f(\mathbf{q}(n+i)) - y^n_{\max}, 0\}\right] \qquad (1) \\ n &= \text{number of measurements so far} \\ y^n_{\max} &= \text{maximum measurement observed so far} \end{aligned}$$

III. System Architecture

A. Overview of Design Requirements Based on Algorithm

To alleviate the complications[2] corresponding to an aerial system integration in a first phase testing of the path planning algorithm, the demonstration platform is to be integrated on a ground-based vehicle. Being that the algorithm is developed specifically with nonholonomic constraints and fixed-wing behavior (gradual, radial turns with no backwards movement and a finite minimum-turn radius), a rover respecting the same limitations is selected, thus suiting the ideal ground-based mirroring of prospective aerial system implementation. Skid-steered mobile units with disabled tank turn (or pivot turn) functionality effectively cast path planning for aerial vehicles into a two-dimensional, ground-based setting.

Figure 1 displays all of the hardware components included in the demonstration platform (for both static and mobile elements). Pictured leftmost in the figure are two small units containing the RF transceivers for empty packet broadcasting to the mobile unit for power monitoring, and the rover and instruments aboard it pictured rightmost in the figure form the mobile unit. Figure 2 shows a configuration of the static nodes placed on the traversal plane with the rover in the standard starting position.

In addition to the ease-of-implementation and trajectory behavior factors mentioned previously, the scale of the rover and environment plane created to be explored by the mobile unit is intentionally small (approximately 15m x 15m) relative to future environments for monitoring (e.g. volcanic plumes) to accurately translate in shape and speed when scaling the whole system (the RF spatial distribution and mobile unit) up in size. These design aspects for the demonstration platform are paramount when selecting the components to be incorporated.

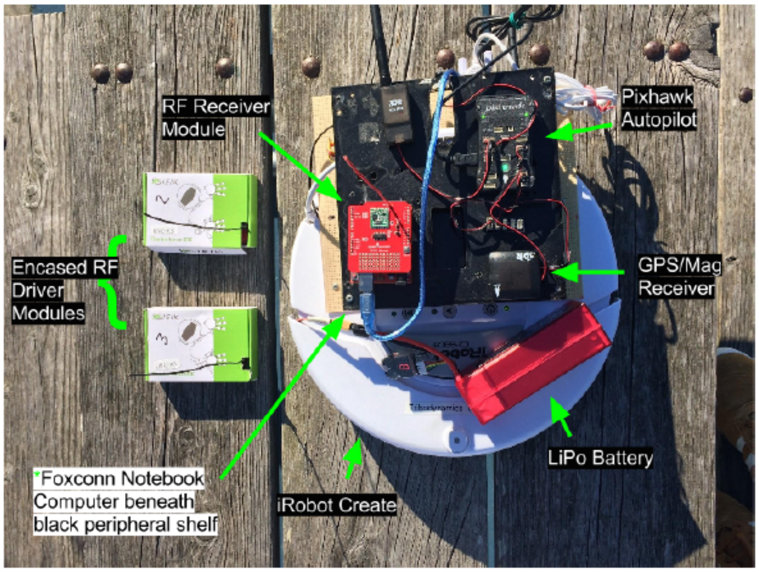

RF Receiver Module
Pixhawk Autopilot
Encased RF Driver Modules
GPS/Mag Receiver
LiPo Battery
*Foxconn Notebook Computer beneath black peripheral shelf
iRobot Create

Figure 1. Entire Hardware Suite for Demonstration Platform

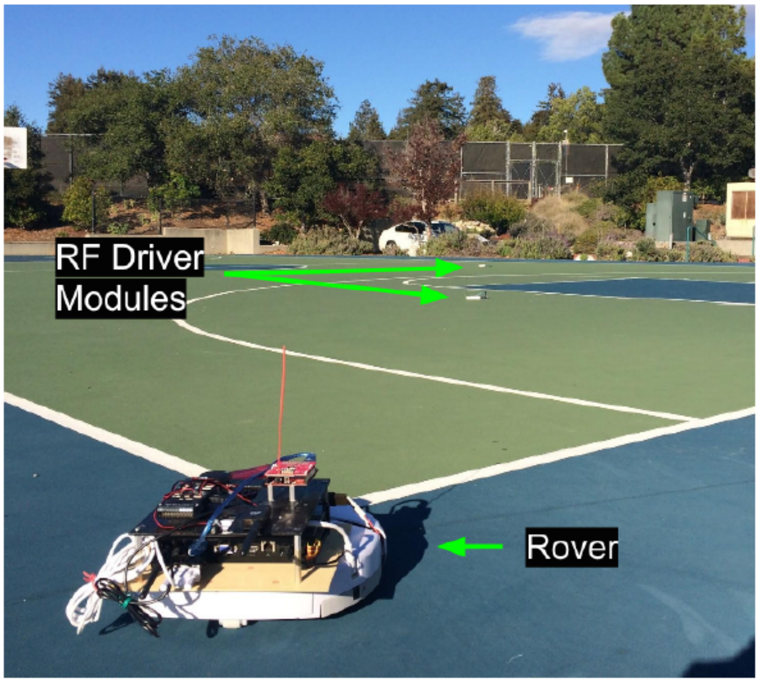


Figure 2. Starting System Configuration

B. Mobile Platform Design

1. Hardware

An iRobot Create skid-steer rover serves as the base for the mobile system. In establishing a minimum turn radius and disabling tank turn capabilities, we intend to closely mimic nonholonomic fixed-wing flight patterns (with the caveat of disregarding yaw or tilt factors present in future aerial implementations). A Pixhawk autopilot module is utilized to interact with and interpret the GPS and magnetometer onboard the rover. This specific autopilot is selected for compatibility with "Mission Planner": waypoint entry software interfacing with a Python scripting feature. This software and its pertinence to the system operation is discussed more thoroughly in Section 3.1.2 ("Software"). To host all of the software elements, a Foxconn notebook computer executes all necessary scripts and programs and supports a local TCP network wherein each script or program acts as a socket. A 14.8 V LiPo battery powers the rover which routed the power through to the computer. Figure 3 depicts a block diagram of the hardware components for the mobile unit. Lastly, an IEIK Uno R3 microcontroller mounted with an RF22S transceiver shield is the data collection component in the system, and its contribution to the demonstration platform is discussed in Section 3.2 ("RF RSSI-based Spatially Distributed Environment").

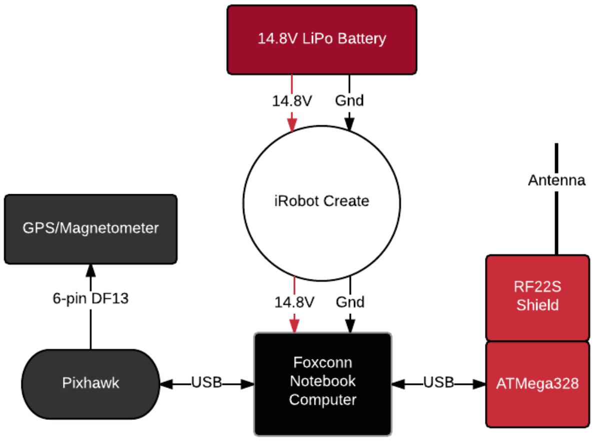


Figure 3. High-level Block Diagram for Hardware Comprising Mobile Unit

2. Software

For compartmentalization, the demonstration platform is projected to be composed of distinct software modules and, following this, would necessitate an onboard communication network. In modifying each of the software modules for TCP by adding a socket to each script or program, a network is established within which all functionality concerning transferral of data and commands for the mobile unit would be achieved. This mesh network links all software outside of Mission Planner.

As seen in Figure 4, the central Python script passes waypoint assignments from the path generation program to Mission Planner and waypoint visitation receipts (i.e. the coordinates with data collected counting as "waypoint hit") to the path generation program to be processed. Additionally, the central script passes RC motor values to the script dedicated to rover motor control for the iRobot Create. Being linked to every software module, this script is the TCP server while the others (script for motor control and program containing path generation algorithm) act as clients.

Software flow charts for each of the scripts and programs is shown in Figure 5. Initialization for all modules is sequentially set with the central script first executed (run within Mission Planners custom console), the motor control script being executed next (within a Python console), and the path production algorithm being executed last (run in a Unix shell). With all modules initiated, the system enters the closed loop.

Downloaded by Abel Rodriguez on December 31, 2018 | http://arc.aiaa.org | DOI: 10.2514/6.2016-0398

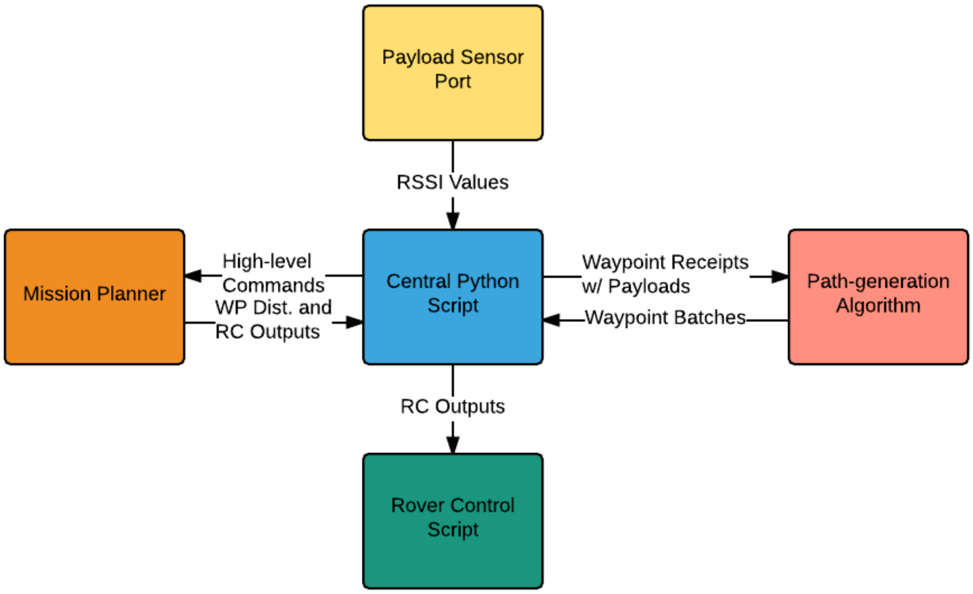


Figure 4. High-level Block Diagram of Software Architecture for Mobile Unit

C. RF RSSI-based Spatially Distributed Environment

A supplementary system is incorporated into the demonstration platform that serves as the quasi-continuous spatial data; RF transceivers utilizing Zigbee protocols create an effective "radio plume" from which payloads could be sourced. By frequently (every tenth of a second from initialization) broadcasting empty messages from each of the static RF nodes in the mesh, received transmission values from each message received represent a magnitude (i.e. simulating matter density or temperature in real applications) which, after being converted from the Radiohead API's[1] RSSI in dBm power units to an exponential of milliwatts using

$${P_{mW}}^{1/10} = 10^{P_{dBm}/100}/1000,$$

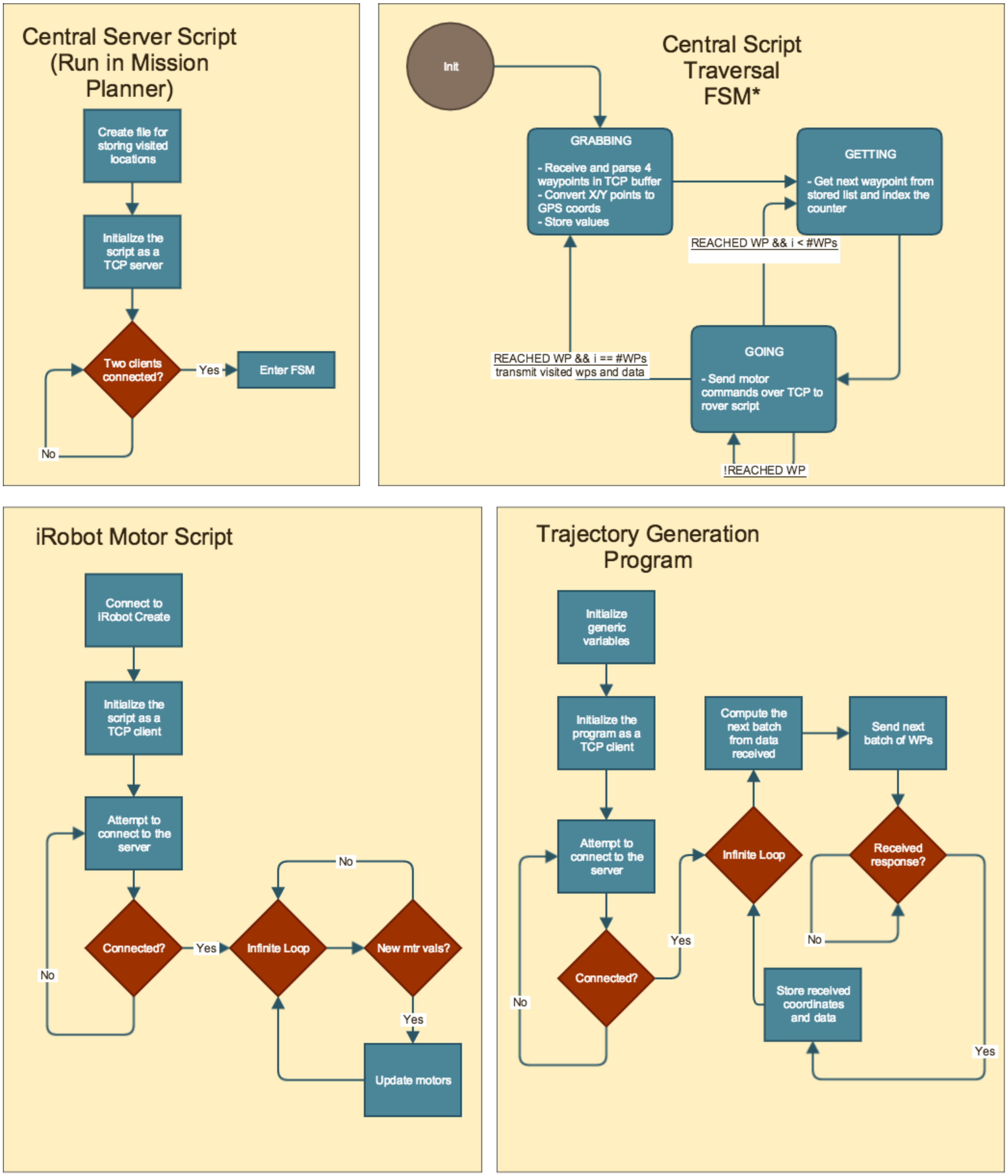
are summed and applied as the data of interest. In effect, the distributions resulting from each node's transmission power would appear as spatially dispersed gradients concentric about various maxima (viewable as colormapping in Figures 6 and 8). This quasi-continuous distribution of electromagnetic transmission strength recreates the field patterns used in the prior research paper's computer simulations[10] with inherent noise, and from it the rover would have an environment to recreate internally. Using this recreation, the payloads aggregated would exercise the path generation algorithm. The optimal path segments, containing the k next locations to sample, provide the waypoints for the rover.

As seen in Figure 6 and Figure 7, the shape of the radio plume is similar to the simulated plume in the previous work.[10]

IV. Results and Discussion

A. Nomenclature Reference/Clarification of Values

- **Instructed vs. visited:** These terms refer to the coordinates issued by the path generation algorithm (initially in two-dimensional Cartesian space and converted to bounded GPS) and the coordinates returned to the path generation algorithm (initially in bounded GPS and converted to Cartesian space).
- **"Application(s)":** Unless otherwise specified, the use of the term "application" encompasses utility of the algorithm for research and/or implementation wherein the the algorithm is a means towards a end as opposed to being solely exercising the algorithm for testing.


Central Server Script (Run in Mission Planner)
Create file for storing visited locations
Initialize the script as a TCP server
Two clients connected?
Yes
No
Enter FSM
Init
Central Script Traversal FSM*
GRABBING
- Receive and parse 4 waypoints in TCP buffer
- Convert X/Y points to GPS coords
- Store values
GETTING
- Get next waypoint from stored list and index the counter
REACHED WP && i < #WPs
REACHED WP && i == #WPs
transmit visited wps and data
GOING
- Send motor commands over TCP to rover script
!REACHED WP
iRobot Motor Script
Connect to iRobot Create
Initialize the script as a TCP client
Attempt to connect to the server
Connected?
Yes
No
Infinite Loop
New mtr vals?
No
Yes
Update motors
Trajectory Generation Program
Initialize generic variables
Initialize the program as a TCP client
Attempt to connect to the server
Connected?
Yes
No
Infinite Loop
Compute the next batch from data received
Send next batch of WPs
Received response?
No
Yes
Store received coordinates and data

***On central script termination, values for waypoints instructed, visited, RSSI, and path taken written out to text files for results**

Figure 5. Flow charts for modules in TCP mesh

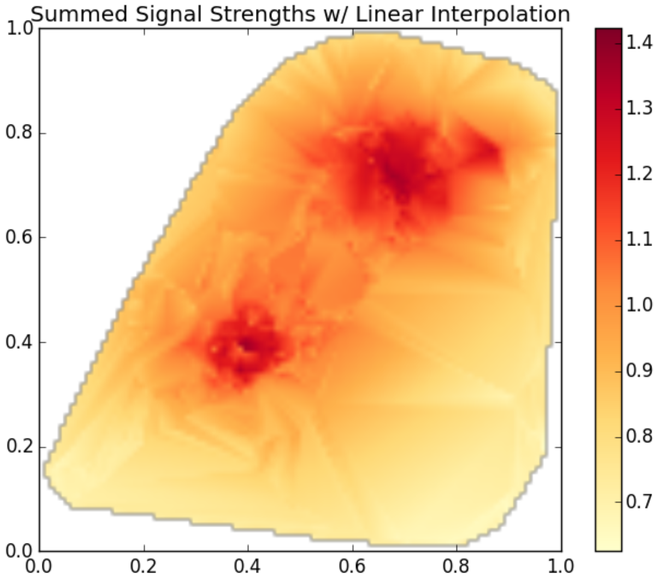


Figure 6. RF-Simulated Plume for a Two-Node Configuration

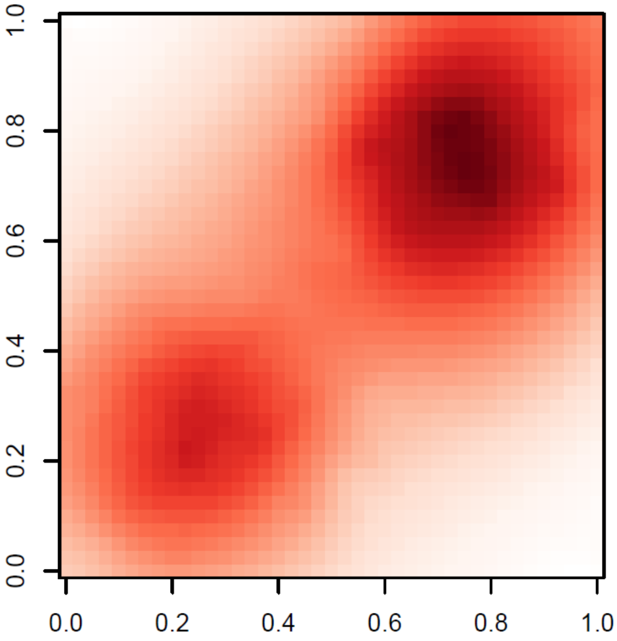


Figure 7. Example of Simulated Plume in Prior Research[10]

- **Value without units:** Unless denoted with a unit, the values corresponding to charts and colorbars do not represent physical measurements, rather they are projections of scaled values meant to closely resembled the simulations of the previous research.[10]

- **RF node transmission power values:** In all of the grids presented with gradients produced by RF RSSI values, the assigned powers for each of the nodes to be transmitting at are +20 dB for the larger Gaussian blot and +8 dB for the smaller Gaussian blot.

- **Waypoint batch iterations per rover run:** Unlike prior research,[10] the number of iterations of the path generation algorithm are not affixed to a constant number; the lengths of the runs in terms of waypoints issued by the algorithm were terminated when the person(s) operating the rover deemed the path suitable to constitute a thorough run or when a "return-to-launch" might be issued in an autopilot system (e.g. for violating a bounded traversal area).

B. Demonstrations with Platform for Trajectory Planning Algorithm

The routes exercised by the rover in a collection of separate runs is depicted in the panel of images in Figure 8 wherein the blue points represent waypoint visitation receipts, the continuous line corresponds to the actual path of traversal, and the background gradient represents a Gaussian interpolation of the data points collected. Several variations of node placement corresponding to the local maxima in the field are applied to experiment with the waypoints instructed and the paths traversed in these different plume shape scenarios.

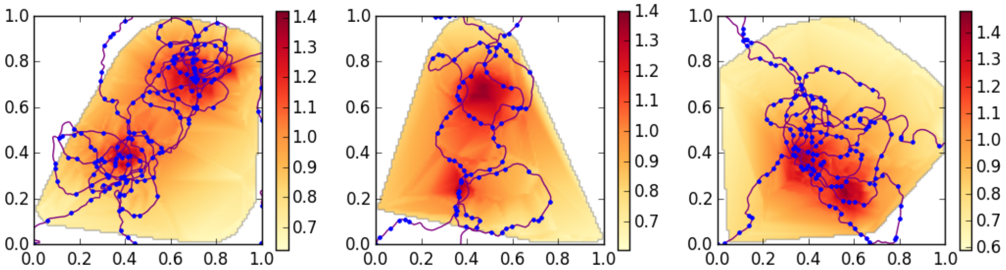


Figure 8. Rover Exploration Runs in RSSI-generated Distribution

The winding in the paths traveled by the rovers is indicative of the asynchronous nature of the TCP mesh wherein the motor values dictated by Mission Planner would experience significant latency before being registered by the motor control script. The weaving indicative of this drawback conflicts with the underlying theory incorporated by the path generation algorithm which operates assuming uniform directional change. The paths created in the algorithm tangentially flow between waypoint batches which, when significant discrepancies surface between the instructed and visited waypoints, is compromised. However, the tendency of the path to hover most about the global maximum in the electromagnetic distribution demonstrates the robustness of the algorithm; the tolerance for moderate divergence from the ideally radial paths created by the algorithm suggests relative indifference to the actual nonholonomic nature of the mobile unit.

The effectiveness of the unit as a variably precise actuation instrument is exhibited in the magnitude of the drift between instructed and visited waypoints. This value tracking for the system is depicted in Figure 9 by the bars separating green circles and red stars (representing the instructed and visited waypoints respectively). The size of the distances are determined by the waypoint radius (or "hit zone": the circle within which a waypoint would be deemed visited if the rover entered it). The autopilot's firmware returns rover distances in whole-numbered meters, and due to this limitation, the waypoint visitation receipts are estimated to average at approximately a full meter away from the intended target when issued. The buffer with which the rover could count a waypoint as having been hit therefore is predicted as nominally a circular area of $3m^2$. This element in tandem with the latency of motor correction commands (optimized with PID tuning) warped the desired smoothness of the radial path creation, and the nominal value resulted as being slightly larger than predicted. Averaging over the discrepancy magnitudes of the 50 waypoint assignments denoted in Figure 9 and scaling the average by $15m$ (the length of the physical side to the grid) results in an average of $1.59m$.

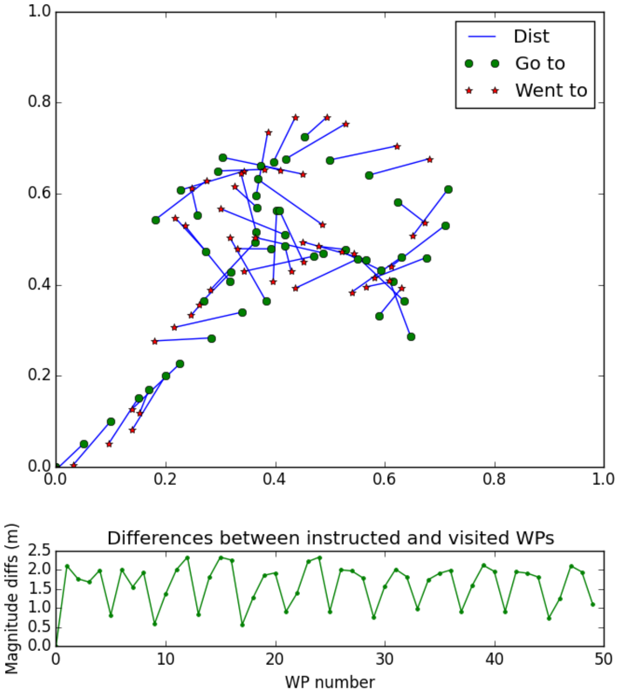


Figure 9. Magnitudes of WP Receipt Distances

Regarding high-level system functionality, the final software architecture for the demonstration platform is conducive to the autonomy of the mobile unit; once each of the modules is initialized in the case of any given run, the system runs independently until all processes are to be killed at the end of operation (i.e. when a “return-to-launch would be issued in application). Keeping a closed loop on the operation sequence upheld the feedback control paradigm expressed throughout the simulation-based research of the previous paper, alternating between generating batches of waypoints and traversing them to aggregate payloads.

V. Future Work

Following the results of this research, an aerial vehicle for small-scale testing in a more realistic application setting will be be outfitted with approximately the same algorithm. Based on future developments projected to evolve in the algorithm,[10] a three-dimensional implementation with the same objectives will be produced. The complications involving GPS resolution coarseness for the ground-based system will be addressed by incorporating a specialized localization system fit to a grid of similar size to that traversed in this papers research. Additionally, a more sophisticated PID control system tailored for tank drive rovers will be implemented to resolve weaving due to slow speeds.

VI. Conclusion

The demonstration platform developed to incorporate a path generation algorithm specialized for optimal data collection for spatial phenomena into a rover unearthed the relevance of trajectory adherence in physical applications. In particular, the ability of the mobile platform to strictly follow the curves generated by the path creation program moderately confounded the projection of what airborne vehicles paths would look like; the weaving inherent to current mobile unit caused relatively inaccurate trajectory matching, thus hindering comparisons between the simulated and actuated paths. However, the functionality of the system remained sound despite restraints inherent to various components (e.g. the limitation on GPS resolution leading to large hit zones for waypoints, the latency in our TCP network, etc.) in the trajectory proximity waning as each rover run progressed. The presence of the RF distribution measured in terms of RSSI by the rover mirrored a field of Gaussian distributions to simulate spatial distributions with the imperfections of electromagnetic noise lending to the distribution randomness expected in an application field. The tendency of the trajectories to draw tighter and tighter into the global maximum of the environment reflects the intentions in the development of the algorithm and the prospect of integration into real-world applications.

VII. Acknowledgments

This work was supported by awards DMS-1322216, NASA/UARC NAS2-03144/TO.030.18.MD.D. and University of California, Santa Cruz's Center for Information Technology Research in the Interest of Society (C.I.T.R.I.S.), SURF-IT for funding, and the Open-source Radiohead RF library authors and contributors

References

[1] Rh_rf22 header file.

[2] Balaban, E., Narasimhan, S., Daigle, M., Roychoudhury, I., Sweet, A., Bond, C., and Gorospe, G. Development of a mobile robot test platform and methods for validation of prognostics-enabled decision making algorithms. *International Journal of Prognostics and Health Management 4*, 1 (2013).

[3] Banerjee, S., Gelfand, A. E., and Carlin, B. P. *Hierarchical Modeling and Analysis for Spatial Data.* CRC Press, 2004.

[4] Cressie, N. *Statistics for Spatial Data.* Wiley, 1993.

[5] Dhariwal, A., Sukhatme, G., and Requicha, A. Bacterium-inspired robots for environmental monitoring. In *IEEE Int. Conf. Robotics and Automation* (2004), vol. 2, pp. 1436–1443.

[6] Graham, R., and Cortés, J. Cooperative adaptive sampling of random fields with partially known covariance. *Int. J. Robust and Nonlinear Control 22*, 5 (2012), 504–534.

[7] Ippolito, C., Fladeland, M., and Yeh, Y. H. Applications of payload directed flight. In *Aerospace conference, 2009 IEEE* (2009), IEEE, pp. 1–15.

[8] Ippolito, C., Yeh, Y.-H., and Campbell, C. A trajectory generation approach for payload directed flight. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition* (2009), p. 1351.

[9] Rimon, E., and Koditschek, D. Exact robot navigation using artificial potential functions. *IEEE Trans. Robot. Autom. 8*, 5 (Oct 1992), 501–518.

[10] Song, S., Rodriguez, A., and Teodorescu, M. Trajectory planning for autonomous nonholonomic vehicles for optimal monitoring of spatial phenomena. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on. IEEE, 2015* (2015), IEEE.

[11] Vazquez, E., and Bect, J. Convergence properties of the expected improvement algorithm with fixed mean and covariance functions. *Journal of Statistical Planning and Inference 140*, 11 (2010), 3088–3095.

[12] Wagner, T., Emmerich, M., Deutz, A., and Ponweiser, W. On expected-improvement criteria for model-based multi-objective optimization. In *Parallel Problem Solving from Nature*, vol. 6238 of *Lecture Notes in Computer Science*. Springer, 2010, pp. 718–727.

[13] Xu, Y., and Choi, J. Adaptive sampling for learning gaussian processes using mobile sensor networks. *Sensors 11*, 3 (2011), 3051–3066.

[14] Yilmaz, N., Evangelinos, C., Lermusiaux, P. F. J., and Patrikalakis, N. Path planning of autonomous underwater vehicles for adaptive sampling using mixed integer linear programming. *IEEE J. Ocean. Eng. 33*, 4 (Oct 2008), 522–537.