Real-Time Escape Route Generation in Low Visibility Environments using Reinforcement Learning

Hari Srikanth
Mira Loma High School
Sacramento, California 95761
Telephone: (916) 693-0389
Email: harinsrikanth@gmail.com

***Abstract*—Structure fires are responsible for the majority of fire-related deaths nationwide. In order to assist with the rapid evacuation of trapped people, this paper proposes the use of a system that determines optimal search paths for firefighters and exit paths for civilians in real time based on environmental measurements. Through the use of a LiDAR mapping system evaluated and verified by a trust range derived from sonar and smoke concentration data, a proposed solution to low visibility mapping is tested. These independent point clouds are then used to create distinct maps, which are merged through the use of a RANSAC based alignment methodology and simplified into a visibility graph. Temperature and humidity data are then used to label each node with a danger score, creating an environment tensor. After demonstrating how a Linear Function Approximation based Natural Policy Gradient RL methodology outperforms more complex competitors with respect to robustness and speed, this paper outlines two systems (savior and refugee) that process the environment tensor to create safe rescue and escape routes, respectively.**

I. Motivation

Structure fires pose a serious danger to many communities across the world, and this problem has only grown over the past decade, with over a 33% increase in deaths from fire since 2012 (Hall and Evarts [4]). Of these, structure fires are most prominent, constituting 79% of civilian deaths and 86% of civilian injuries. These casualties stem from people being trapped within the building for an extended period of time, resulting in exposure to collapsing structures, high heat/flames, and toxic smoke. In addition, the unpredictable nature of fires reduces the efficacy of predetermined escape strategies. In order to determine safe rescue and escape routes, a live mapping of the environment with the measurement of aspects of interest is necessary. However, current methodologies are insufficient to handle this task, due to an inability to map rapidly in densely obstructed environments, as well as slow onboard machine learning.

II. Background

The first task for the system is to generate a map of the environment in real time. A well researched method for this is Simultaneous Localization and Mapping (SLAM). SLAM is a process where the agent pose is determined within an environment which is (simultaneously) being mapped. While SLAM methodologies have been well tested in regular environments, low visibility environments clouded with smoke or dust pose a problem for most systems, as they utilize optical rangefinding systems. In order to develop a resilient mapping system, various solutions have been proposed:

- SLAM with Visual and Thermal Imaging Cameras (Brunner et al. [1]): Utilizes thermal imaging in order to counterbalance visual camera obfuscation. However, increased reliance on thermal data resulted in a decrease in localization accuracy.
- SLAM with laser range finders and 94 GHz Frequency Modulated Continuous Wave Radar (Gerardo-Castro and Peynot [3]): Proposed a system with mm-wave radar to penetrate dense smoke that laser scanners cannot. Very precise, but some false negatives remained. In addition, the custom antenna system is very costly and locks mobility.
- SLAM with laser range finders and sonar sensor array (Machado Santos et al. [6]): Uses a fuzzy logic system Couceiro et al. [2] to decide between sensor inputs, reducing the risk of false negatives.

Regardless of the specific mapping methodology, the map resolution and effective range drastically decrease as the adversarial noise parameter increases. Given these limitations, the use of Multi-Robot Systems (MRS) is necessary to optimize mapping speed and robustness, with a Random Sample Consensus (RANSAC) based map unification to ensure live processing does not become too computationally expensive (Lázaro et al. [5]). The second aspect of the problem is the determination of optimal rescue or escape paths. In order to do this, the situation can be modeled through a Markov Decision Process and solved using Reinforcement Learning. Reinforcement Learning (RL) is a paradigm where an agent seeks to maximize the reward it gains through refining its policy. At each timestep t, the agent observes the environmental state and according to some policy π it takes some action. This action changes the environmental state and returns some reward, and this is used to retrain the policy. A well researched RL method is Actor Critic, which utilizes an actor that updates the policy and a critic which evaluates said policy for fast optimization. While industry standard algorithms Trust Region Policy Optimization (Schulman et al. [8]) and Proximal Policy Optimization Schulman et al. [9] utilize complex neural networks to estimate the value function, my prior research on reinforcement learning determined that a Linear Function Approximation based Natural Policy Gradient algorithm (LFA-NPG) would determine the optimal policy with much less iterations than either TRPO and PPO, in low dimensional standard and sparse reward RL benchmarks. In addition, my robustness analysis determined that LFA-NPG was significantly more noise resistant to adversarial noise than TRPO and PPO, maintaining identical performance across data sampled within 20% of the true value. Although the determination of navigation paths appears to be optimized for operations research methodology, the unique nature of each fire does not lend itself to the determination of the heuristics often employed in these algorithms.

III. Design

The main necessities of a real world robotics system are as follows:

- Speed: Data must be sampled at a high rate, in order for the system to react appropriately
- Robustness: Algorithms must be resistant to adversarial noise, an inherent factor when live systems are considered

In addition to these two criteria, the search and rescue application motivating this paper requires the system to have some measure of its own accuracy. Fig. 1 details the proposed system, taking these criteria into consideration.

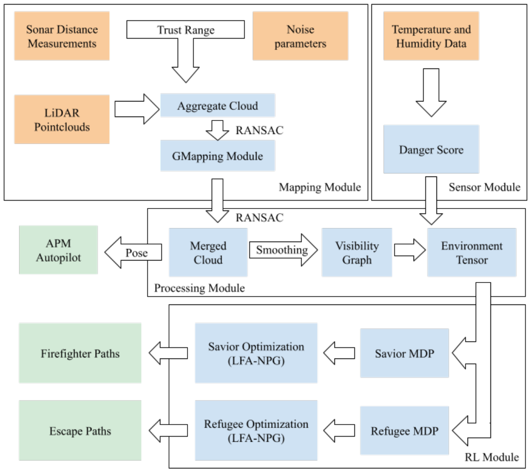


Fig. 1. System Organization

A. Data Acquisition & Processing

Given that each second the system takes to find an exit path is a second of exposure to hazardous conditions, the key priority of my solution is speed. Fast navigation, fast mapping, fast processing, and fast path determination take precedence over exact optimization or high resolution. In order to map at this rate, the system will utilize a fleet of autonomous drones. Each drone will be outfitted with a mapping module (consisting of a LiDAR rangefinder and four sonar scanners) and a sensor module (to collect temperature and humidity data, as well as determine a quantitative adversarial noise parameter). Using the temperature and humidity data, the map will then be populated with scores quantifying the danger of each area of the building. From here, the LFA-NPG agent will determine the optimal paths, both for rescue (for the firefighters) and for escape (for the inhabitants). The base hardware for each device is a light quadrotor with ROS on a RPi running ubuntu. Using 3D printed parts to keep the weight as low as possible, the drone will have a target thrust/weight of 2:1. Each RPi interfaces with a Navio 2 flight control unit running an ardupilot APM autopilot. ROS features a variety of nodes to run SLAM calculations, such as HectorSLAM, GMapping, KartoSLAM, CoreSLAM, and LagoSLAM. Of these, GMapping is the most optimal for mapping in a smoky environment, as it remains robust and has a large amount of support.

The raw data that the mapping module collects from each drone is laser scan information from the LiDAR, four sonar distance measurements, and a measure of adversarial noise. The core of the mapping system is the LiDAR data: even though its accuracy is compromised in smoky environments, it is the most high-resolution sensor available within the price range. The sonar data comes from four sensors at 90 degrees to one another. The sonar data is then transformed to a laser scan frame to directly evaluate it in comparison to the LiDAR data. In order to determine which sensor data should be used (LiDAR for high resolution vs Sonar for higher accuracy) when constructing the map, the device incorporates a fuzzy logic system, through a measurement of confidence called the Trust Range. The trust range is determined by the difference between laser and sonar data and the adversarial noise parameter. In a design intended for real world use, the adversarial noise parameter in the use case of a fire would be determined by a particulate matter sensor. However, the logistics of producing real smoke multiple times for testing is a safety concern, and as such the effect of smoke is simulated through the use of a smoke machine. Because smoke machines do not produce particulate smoke, but rather vaporize alcohol-based fog solution, the noise parameter is instead determined through the use of an alcohol sensor (no difference in algorithm, just uses a different function and hardware when calculating). After the trust range is determined, the points outside the trust range are eliminated from the data and points within are added to an aggregate point cloud. This aggregate point cloud includes data from each drone exploring the system. The data from each point cloud is then passed into GMapping, creating a map. The set of maps from each drone is then fed into the processing module.

The processing module begins by merging each of the individual maps together, using the algorithm outlined below:

1) Use a correlative scan matcher to identify matching edges of each scan

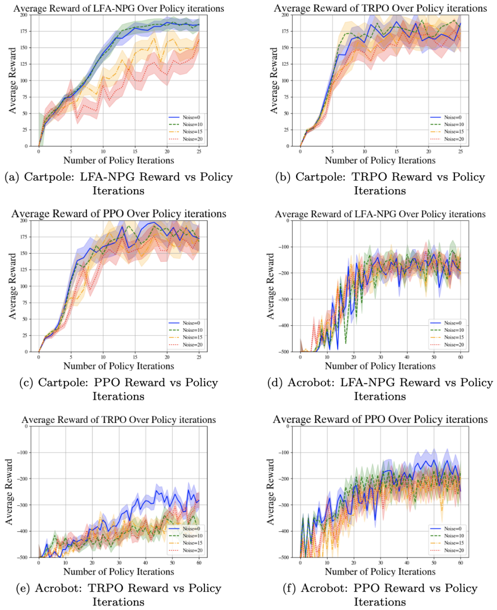
2) Add each edge solution to a pool of candidates
3) For any two maps, determine a translation between them such that one candidate edge is identical (zero error).
4) Determine the errors of all other candidate edges. Depending on if the error of the other edges is high or low, it is possible to determine inliers and outliers
5) Once the correct translation has been determined, merge the two maps into a global map. This global map can then be merged with the next map, which can repeat until all maps have been integrated into the global map.

After the global map has been created, the pose of each drone is passed to the flight controller, completing the localization aspect of SLAM. While this map could just be exported as is, more processing and simplification is necessary to determine exit paths. First, the merged map is smoothed into a visibility graph, which maps complex environmental features to a significantly more condensed set of traversable nodes. Each node is then assigned a danger score based on temperature and humidity data. For nodes where no temperature data has been collected, data from nearby nodes is weighted and used to estimate the conditions at the location. The complete environment tensor with traversable nodes and dangers is then ready to be fed into the RL training models.

B. RL Data Analysis

LFA-NPG is an policy-space RL Paradigm that utilizes the Natural Actor Critic (Peters and Schaal [7]) Architecture, optimizing the policy through natural gradient descent while estimating the value function (how well a given policy will produce reward) through Linear Function Approximation. LFA-NPG was evaluated on standard RL benchmarks, Cartpole, and Acrobot (a sparse-reward environment, which is more common in real world robotics systems). Its performance is as shown in Fig. 2.

As shown, LFA-NPG converges to the optimal policy even as the noise parameter (represented by the true state sampling error passed through to the model) increases up to 20%. This is due to its simple value estimation method: in contrast to the complex neural networks present in other forms of RL, LFA-NPG's simpler value estimation methodology enables it to be more resistant to fluctuations in input. In addition to robustness, LFA-NPG also is much faster than competing algorithms in low dimensionality applications (such as CartPole and Acrobot). It reaches a similar level of reward compared to PPO and TRPO, with a much faster and more logarithmic convergence, ideal for high speed use cases. This is evidenced in Fig. 3.



(a) Cartpole: LFA-NPG Reward vs Policy Iterations

(b) Cartpole: TRPO Reward vs Policy Iterations

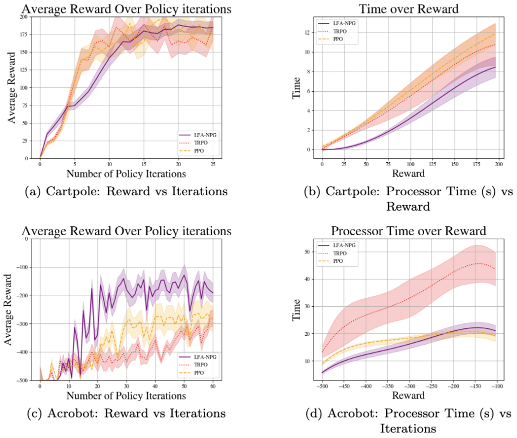
(c) Cartpole: PPO Reward vs Policy Iterations

(d) Acrobot: LFA-NPG Reward vs Policy Iterations

(e) Acrobot: TRPO Reward vs Policy Iterations

(f) Acrobot: PPO Reward vs Policy Iterations

Fig. 2. LFA-NPG Robustness Analysis



(a) Cartpole: Reward vs Iterations

(b) Cartpole: Processor Time (s) vs Reward

(c) Acrobot: Reward vs Iterations

(d) Acrobot: Processor Time (s) vs Iterations

Fig. 3. LFA-NPG Model Convergence w.r.t Iterations & Time

Considering that the data that can be collected during a live system activity is limited in scope, it will tend to be sufficiently low dimensionality enough for LFA-NPG to function at an optimal level. Moreover, RL methodology has been shown to be competitive with leading operations research algorithms currently used for traversal problems, and since the structure of an MDP is more cohesive with raw sensor inputs than the heuristic information necessary for OR algorithms, it is uniquely positioned as the optimal real time path planner for real world robotics.

In the presented use case, the environment tensor can be used to inform both the optimal strategy for a firefighter and the optimal strategy for a trapped civilian. As such, two RL models will train simultaneously using the same base state space, with different reward functions. The savior system determines the strategy for the firefighter: Given the locations of possible entryways into the building, the system then determines the optimal paths to navigate to any point within the building, in the form of a tree. The refugee system is used to determine the escape routes. The starting nodes are all lethal areas (determined by danger score) and have a reward function dependent on duration and level of exposure to hazardous conditions. These two objectives are each used to formulate distinct MDPs, which LFA-NPG based solvers optimize through the use of the actor critic architecture.

In order to determine the feasibility of this technology in real world applications, a minimum of 2 drones must be utilized (to evaluate the efficacy of the merged map) in a simulated indoor fire. This indoor fire will be simulated through the use of a smoke machine, which the glycol sensor uses to determine the danger parameter. As it is difficult to simulate the high temperatures of a fire, the environment tensor will instead be fed a possible matrix of danger scores within the building.

IV. Conclusion

This work considers the relevant Search and Rescue use case for autonomous robot systems for the traversal of a burning building. In addition to discussing the efficacy of current low visibility perception systems, this work proposes a novel perception system that makes use of a multi agent mapping array. Each individual agent prioritizes certain flows of data through a trust range, determined by the measure of adversarial noise, which is additionally used to evaluate the safety of each location in the map. It goes on to discuss a possible way in which this data can be analyzed, namely for the generation of escape/rescue routes that maximize safety. In order to accomplish this, the paper recommends the use of a linear function approximation based natural policy gradient reinforcement learning methodology, demonstrating its high speed and strong resistance to adversarial noise in sufficiently low dimensional systems.

References

[1] Christopher Brunner, Thierry Peynot, Teresa Vidal-Calleja, and James Underwood. Selective combination of visual and thermal imaging for resilient localization in adverse conditions: Day and night, smoke and fire. *Journal of Field Robotics*, 30(4):641–666, 2013. doi: https://doi.org/10.1002/rob.21464. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21464.

[2] Micael S. Couceiro, J.A. Tenreiro Machado, Rui P. Rocha, and Nuno M.F. Ferreira. A fuzzified systematic adjustment of the robotic darwinian pso. *Robotics and Autonomous Systems*, 60(12):1625–1639, 2012. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2012.09.021. URL https://www.sciencedirect.com/science/article/pii/S0921889012001753.

[3] Marcos Paul Gerardo-Castro and Thierry Peynot. 01 2012. URL https://www.researchgate.net/publication/264545306_Laser-to-radar_sensing_redundancy_for_resilient_perception_in_adverse_environmental_conditions.

[4] Shelby Hall and Ben Evarts. Fire loss in the united states during 2021, 2021. URL https://www.nfpa.org/~/media/fd0144a044c84fc5baf90c05c04890b7.ashx#:~:text=In%202021%2C%20local%20fire%20departments,14%2C700%20reported%20civilian%20fire%20injuries.

[5] M. T. Lázaro, L. M. Paz, P. Piniés, J. A. Castellanos, and G. Grisetti. Multi-robot slam using condensed measurements. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1069–1076, Nov 2013. doi: 10.1109/IROS.2013.6696483.

[6] João Machado Santos, Micael S. Couceiro, David Portugal, and Rui P. Rocha. Fusing sonars and lrf data to perform slam in reduced visibility scenarios. In *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 116–121, 2014. doi: 10.1109/ICARSC.2014.6849772.

[7] Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7):1180–1190, 2008. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2007.11.026. URL https://www.sciencedirect.com/science/article/pii/S0925231208000532. Progress in Modeling, Theory, and Application of Computational Intelligenc.

[8] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017. URL https://arxiv.org/abs/1502.05477.

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.