

USING RIVET FOR PARAMETRIC ANALYSIS OF ROBOTIC SYSTEMS

Juan Pablo Gonzalez, PhD
William Dodson
Robert Dean
Greg Kreaflle
General Dynamics Robotic
Systems
Westminster, MD

Alberto Lacaze
Leonid Saprnov
Robotics Research LLC
Gaithersburg, MD

Marshal Childers
Robotics Programs Office
Army Research Lab
Aberdeen Proving Ground, MD

ABSTRACT

Parametric analysis is an essential step in optimizing the performance of any system. In robotic systems, however, its usability is often limited by the lack of complex yet repeatable experiments required to gather meaningful data. We propose using the Robotics Interactive Visualization and Experimentation Toolbox (RIVET) in order to perform parametric analysis of robotic systems.

INTRODUCTION

Parametric analysis is an essential step in optimizing the performance of any system. In robotic systems, however, its usability is often limited by the lack of complex yet repeatable experiments required to gather meaningful data. We propose using the Robotics Interactive Visualization and Experimentation Toolbox (RIVET) in order to perform parametric analysis of robotic systems.

RIVET is a simulator that not only generates a rich world representation in real time, but also simulates sensors and the interactions of the vehicle with the terrain and other moving entities, while having the same low level interface as the actual hardware in the robotic vehicle. Through RIVET it is possible to perform closed loop simulation that includes the perception, planning and actuation components. It is therefore possible to perform parametric studies involving the complex interactions between the different subsystems of a robotic system, while still maintaining a high level of repeatability between experiments.

To illustrate the importance of RIVET as a tool for parametric analysis, we optimize the performance of a complex approach to integrating local and global navigation called the *Field Cost Interface (FCI)*. In FCI a global planner continuously generates a cost field at a radius R from the vehicle, using both prior data and sensor data. A local planner then attempts to plan paths to each point along this circle, therefore combining the kinematic constraints of the vehicle and the recommendations of the global planner. While there are many parameters involved in this approach, the most important one is the constant k that weighs the

importance of the global planner vs. the local planner. We optimize the performance of the overall system by varying this constant and measuring the quality of the resulting path.

The optimized system already accounts for many of the real-world limitations of the integrated system, and only requires small adjustments to reach peak performance in the field. This approach requires significantly less field time for optimization than attempting to optimize the system based on field experiments alone.

RIVET

RIVET is a high fidelity simulator that not only generates a rich world representation in real time, but also simulates sensors and the interactions of the vehicle with the terrain and other moving entities, while having the same low level interface as the actual hardware in the robotic vehicle.

Developed under the Army Research Laboratory's Robotics Collaborative Technology Alliance, RIVET features a game-based, interactive, multi-user modeling and simulation platform specifically suited to developing perception, planning and other robotic technologies faster and more affordably. RIVET's feature-rich environment and scalability allow users to configure the system to include multiple warfighters, as well as air, ground, and sea-based robotic assets so that realistic small team tasks can be played out safely and repetitively in the virtual world.

By leveraging the latest developments in game technology and graphics hardware RIVET is able to provide not only stunning graphics and special effects, but high fidelity physics simulation, terrain rendering, sensor simulation and

even weather effects such as fog and rain. It can also provide animation support for realistic human movement and reaction.

Simulated Environments

Rivet provides realistic 3-D environments representative of urban and cross country environments. Some of these environments are analogs of real test sites such as Fort Indiantown Gap (Fig 1), therefore allowing for realistic simulations and more direct verification. RIVET can model approximate terrain elevation, mobility obstacles such as trees, rocks, slopes, buildings, as well as humans and other vehicles.



Fig 1. Simulated images of the environment for Fort Indiantown Gap

Simulated Sensors

RIVET simulates the many existing LADAR sensors such as GDRS' Gen 4 Ladar, SICK and Hokuyo. RIVET provides an interface that matches that of the actual sensor, allowing seamless simulation of point clouds as the vehicle moves through the environment. Fig 2 shows an example environment and the corresponding simulated point cloud.

RIVET can also simulate cameras for image-based processing in the visible range, and perform basic effects-based infrared simulation.

The simulated sensors model many of the complex aspects of perception in robotics such as occlusions, imperfect scan patterns and weather effects.

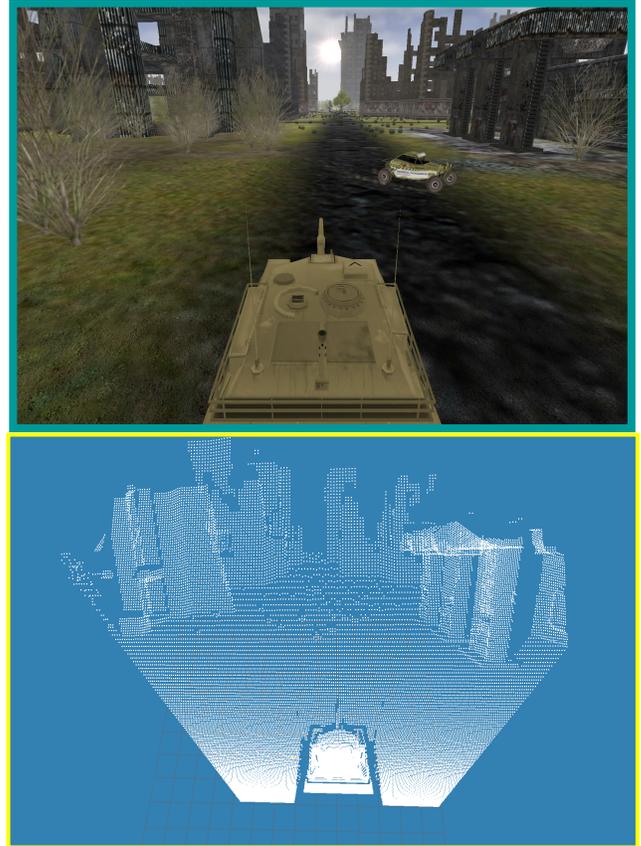


Fig 2. Simulated scene and corresponding point cloud for a LIDAR sensor.

Simulated Platforms

RIVET simulates a wide variety of air, land, and sea-based autonomous platforms, which can be simulated simultaneously in order to model complex vehicle interactions and interactions with pedestrians. Fig 3 shows some of the platforms currently available in RIVET.

The simulated platforms receive actuation commands from the autonomous mobility software and simulate the interaction of the vehicle with the environment using a real-time physics engine.



Fig 3. Some of RIVET's simulated platforms: GDRS XUV, MDARS and Talon.

USING RIVET FOR PARAMETRIC ANALYSIS OF ROBOTIC SYSTEMS

Parametric analysis is an essential step in optimizing the performance of any system. In robotic systems, however, its usability is often limited by the lack of complex yet repeatable experiments required to gather meaningful data.

Robotic Systems are combinations of complex processes and algorithms that interact with the environment. While some times the individual processes and algorithms can be

optimized in isolation, there is often a significant gap between the performance of the algorithm in isolation and its performance as part of an integrated system. The “real world” performance of a system depends greatly on factors such as accelerations, GPS errors, CPU load, network delays and bottlenecks.

Through RIVET it is possible to perform closed loop simulation that includes the perception, planning and actuation components. It is therefore possible to perform parametric studies involving the complex interactions between the different subsystems of a robotic system, while still maintaining a high level of repeatability between experiments

While no simulation can exactly replicate the conditions that a robotic system encounters in the field, we argue that using real-time hardware-in-the-loop simulation replicates most of the data flows, interactions and constraints of the actual system can be used to perform a parametric analysis of such a system. Furthermore, the results of such analysis that can be applied in real-world situations with little or no modifications, therefore speeding up the research and development cycle of robotic systems.

Case Study: Integrating Local and Global Navigation Using the Field Cost Interface

Few examples illustrate the importance of using Rivet in the research and development of robotic systems better than the integration of local and global navigation. This task requires complex interactions between the perception system, a local planner and a global planner.

In this case study we will use an approach to combine local and global navigation called the *Field Cost Interface* (FCI).

In FCI [1] a global planner continuously generates a cost field at a radius R from the vehicle, using prior data, sensor data and information on the dynamic tactical environment. The global planner uses the Geometric Path Planner (GPP) [2] in order to generate global routes that consider tactical mission requirements such as travel time, mobility cost, exposure risk and coverage; it uses Multi Resolution Field D^* [3] as a planning engine which is able to quickly evaluate paths over an area of several kilometers while still maintaining a relatively high resolution representation of environment. The global planner, however, does not consider the kinematic constraints of the vehicle.

The local planner then attempts to plan paths to each point along this circle, thereby combining the kinematic constraints of the vehicle and the recommendations of the global planner. The local planner is an ego-graph-based planner [4] that considers the kinematic and non-holonomic constraints of the vehicle.

The main challenge with using FCI is that it requires combining different cost metrics for the global and local planner. In order to combine these costs, the planner first

scales both the global and local costs by calculating the average cost assigned to a given section of the route by both planners. This produces a cost metric that has similar scales and that adapts to different terrain types. The planner then combines the scaled costs as follows

$$C_{total} = C'_{local} + k \cdot C'_{global} \quad (1)$$

where C'_{local} and C'_{global} are the scaled local and global path costs, and k is a constant that is determined experimentally. This constant defines the relative weight of the global costs with respect to the local costs and is the most important parameter for the performance of the algorithm, as it compensates for any systematic differences between the local and global costs.

Fig 4 shows how the FCI evaluates routes. The cyan lines show the global paths being used to generate the field costs, and the yellow line shows the local path chosen after considering both the global and local costs.

Although many parameters influence the performance of the system when using *FCI*, the most important parameter is the constant k . Optimizing this parameter requires running both the global and the local planner, which in turn requires providing prior data as well as sensed data which comes from the perception system. Because the planner commands the motion of the vehicle, it affects the perception system as well. For this reason, any meaningful analysis of this system needs to include at least these three subsystems, plus the ability to control the vehicle (as opposed to just playing back recorded data).

The value of k affects many aspects of the behavior of the robot through the FCI. The main two aspects are the ability of the robot to turn away from a local obstacle when the global goal is in the direction of the obstacle (as in Fig 4), and the ability of the robot to turn around when the local planner is trapped in a local minimum. Small values of k give more control to the local planner, which makes it easier for the robot to move away from a local obstacle, but also makes it harder to escape a local minimum. Conversely, larger values of k give more control to the global planner, which makes it harder to move away from a local obstacle while enabling easier escape from local minima.

In order to identify the ideal value of k in a typical real-world scenario, a test scenario was setup that would test the robot's ability to detect and move away from a local obstacle (sensed blockages on the road) and the ability to navigate to the goal in a complex environment that should provide local minima for the local planner as well (Fig 5)

A total of 30 runs covering over 40 kilometers of autonomous navigation were performed in RIVET and in the field, with k taking the values 0.01, 0.05, 0.1, 0.2, 0.4, 0.8, and 1.6. From these runs, we selected two relevant metrics to

evaluate the performance of the planner: total execution time and total distance traveled.

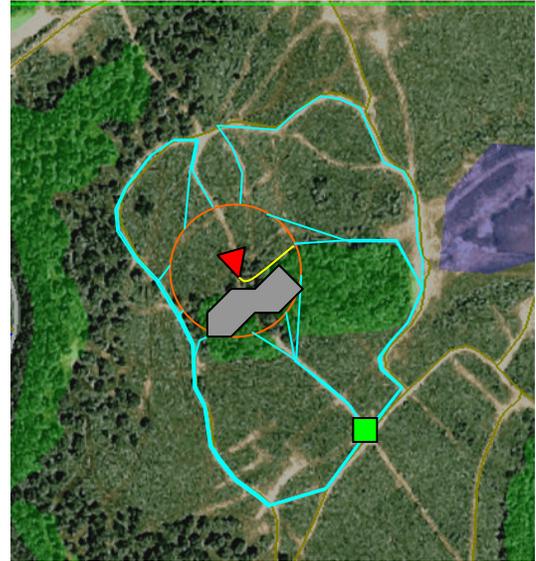


Fig 4. Blockage on route and alternatives passed to the local planner by the FCI (cyan). The new route is selected considering the global paths and the local kinematic constraints of the vehicle.



Fig 5. Test scenario for optimizing FCI's performance. The blue line shows the original path, while the yellow line shows one of the possible alternate paths for the robot to follow after the blockages in the main route are detected (red crosses). The area shown is 600x600 meters.

RESULTS

Fig 6 shows the plot of the total distance for each completed run for different values of k , for RIVET and field tests. In RIVET, the best values of k are 0.1 and 0.2, with almost exact performance. Values of k lower than 0.1 usually cause the robot to favor too much the local planner and often take longer detours because of local mobility minima (the robot avoids locally expensive trajectories such as turning). Also, the total distance becomes more spread going from very short runs (similar to the best values of k) to fairly large values not seen in the best cases. Values of k greater than 0.2 fail to reach the goal in all attempts.

In a similar fashion, Fig 7 shows the plot of the total execution time for each completed run for different values of k for RIVET and field tests. The total execution time is best with $k = 0.2$, being slightly longer for $k = 0.1$. As with total distance, smaller values of k produce longer execution times and less predictable results. When $k \geq 0.4$ the robot is unable to reach the goal in all attempts.

The field experiments match RIVET in a number of ways, yet differ in others. Field tests confirm the RIVET simulation indicating that the goal cannot be reached when $k \geq 0.4$. They also confirm that the best performance is achieved with $k=0.2$. In field tests, however, $k=0.1$ performs significantly worse than $k=0.2$. This is likely due to the presence of more vegetation and mobility obstacles in the field than in RIVET, which causes the robot to follow locally attractive paths instead of being more assertive in its goal seeking behavior. RIVET runs are also shorter than field runs. This is because the RIVET environment is not an exact match of the field, and because many areas of the field have vegetation that is not accurately modeled in RIVET.

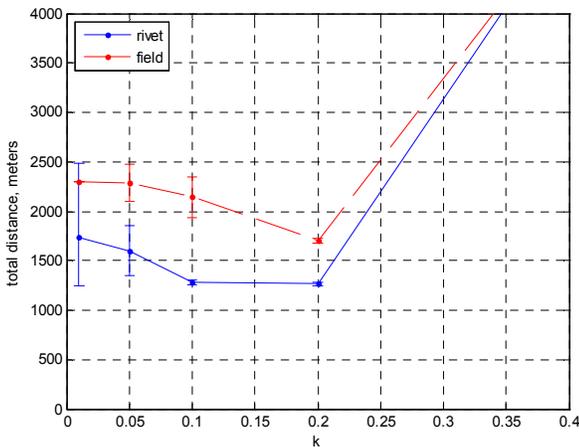


Fig 6. Total distance for RIVET and field runs. All runs failed to complete for $k \geq 0.4$ in RIVET and in the field. While field distances are greater than in RIVET, both graphs show the same trends.

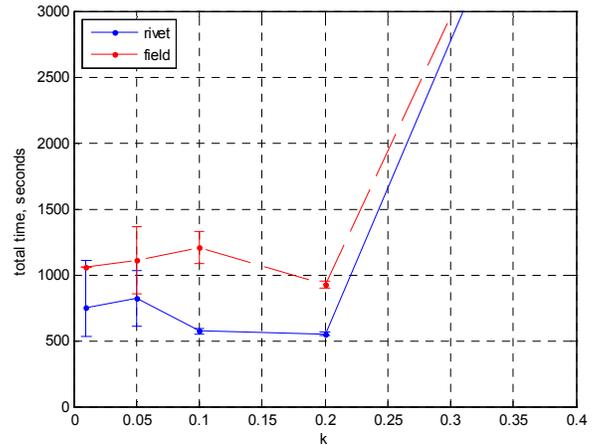


Fig 7. Total execution time for RIVET and field runs. All runs failed to complete for $k \geq 0.4$ in RIVET and in the field. While field times are greater than in RIVET, both graphs show the same trends.

Fig 8 and Fig 9 illustrate these differences: The RIVET run takes a number of shortcuts that make the overall run shorter and faster. In the field, these shortcuts are either blocked by vegetation, or surrounded by vegetation in a way that makes it very costly for the local planner to navigate through them.

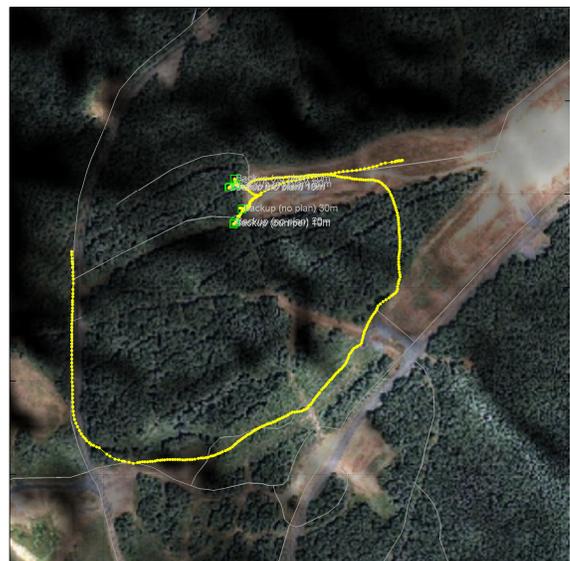


Fig 8. Sample run in RIVET with $k=0.2$.



Fig 9. Sample run in the field with $k=0.2$. Notice how the robot does not take the shortcut that was taken in the RIVET run.

CONCLUSIONS

The results presented here show that RIVET is a viable alternative for performing parametric analysis of robotic systems. Even though RIVET is not an exact replica of the field environment, the results obtained through it closely match those obtained in the field. Rather than performing complete parametric analyses in the field, it is now possible to perform an initial analysis in RIVET, which allows for greater repeatability and requires less time and resources. Once an approximate operating point has been determined in RIVET, field experiments can be used to confirm and further refine the parameters of a system.

However, not all robotic systems can be correctly modeled in RIVET. Because of limitations in processing power and rendering engines, some local mobility characteristics are not accurately modeled. Also, some sensors such as infra-red and radar are particularly difficult to accurately model in simulation. RIVET-based analysis of robotic systems that depend heavily on these sensors is likely to be inaccurate and may not be representative of the behavior in the field.

Future developments in RIVET and in game-based technology are likely to increase the array of systems and sensors that can be modeled in RIVET, therefore extending the results presented here to other robotic applications.

REFERENCES

- [1] A. Lacaze, "Hierarchical planning algorithms". Proceedings of SPIE, SPIE, 2002, 4715, 320
- [2] J.P. Gonzalez, B. Nagy and A. Stentz, "The Geometric Path Planner for Navigating Unmanned Vehicles in Dynamic Environments". Proceedings ANS 1st Joint Emergency Preparedness and Response and Robotic and Remote Systems, 2006.
- [3] D. Ferguson and A. Stentz, "Multi-resolution Field D*," Proceedings of the International Conference on Intelligent Autonomous Systems (IAS), March, 2006.
- [4] A. Lacaze, Y. Moscovitz, N. DeClaris, and K. Murphy, "Path planning for autonomous vehicles driving over rough terrain ". Intelligent Control (ISIC), 1998. Held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Intelligent Systems and Semiotics (ISAS), Proceedings, 14-17 Sep 1998, 50-55