

VISUALIZATION AND SIMULATION FOR PATH PLANNING USING MATLAB

Willis Troy, Michael Thompson

Department of Electrical and Computer Engineering

Baylor University

Willis_Troy@Baylor.edu

Abstract

This work focuses on using MATLAB as an analysis tool by exploiting MATLAB's graphical functionality and modularity. The simulation and visualization capabilities of MATLAB are very useful for both engineering education and research. In this paper MATLAB is used to develop a test-bed to graphically model mission planning work to convey a visual representation of flight paths and their variability- i.e. altitude, timing, etc. The graphical representation allows visual feedback when investigating path planning algorithms that deal with escort missions. The modularity of MATLAB allows us to test different algorithms and modifications with minimal change to the framework- allowing algorithms to be chosen and updated as needed. We use the A-star algorithm, a common path planning algorithm, to illustrate the use of MATLAB, and the efficiency at which we can calculate paths and give feedback. A similar graphically enhanced simulation approach is possible for studying a wide-variety of engineering problems.

*Proceedings of the 2014 ASEE Gulf-Southwest Conference
Organized by Tulane University, New Orleans, Louisiana
Copyright © 2014, American Society for Engineering Education*

Introduction

The objective of this work is to build an infrastructure that graphically models mission planning work. We require a framework that allows the investigation of differing path planning algorithms given target locations, mission variability, and constraints. Visualization of such paths allow rapid conclusions to be made, including the verification that an algorithm works as intended, objectives are met, and whether the path overall was a success. Specifically, we are flight paths for escort aircraft in a potentially hostile environment, where there is a need to remain undetected. As such, when we are within range of an enemy target, we need to suppress detection, hence; we can define success as arriving at targets while never being detected.

MATLAB is a numeric computing environment often utilized as a tool in teaching and research settings [1, 2]. MATLAB is well known for its matrix capabilities, ability to implement algorithms, and visualization capability [1, 2, 3]. In addition, MATLAB provides toolkits tailored to specific topics, such as optimization, image processing, and statistics which have common tasks implemented per the area. Implementing algorithms is easier than conventional programming, as the programming environment allows for a higher level of abstraction. However, the level of abstraction does not sacrifice the modularity common to many languages. This modularity allows rapid prototyping. MATLAB code can be run immediately without the need to compile, and often notifies the origin of the error allowing quick debugging. In addition, MATLAB's visualization allows us to model complex tasks graphically, where a numeric representation might fall short.

The modeling investigated in this paper is for specifying escort missions in which a protected entity (PE) with specified waypoints needs aid to suppress detection. Specifically, our PE is an aircraft and we need to suppress its detection from threat radars. The suppression is done by another aircraft via electronic warfare (EW), namely radar jamming [4]. This EW aircraft is referenced as an electronic attack (EA) aircraft and escorts the PE while residing outside of a target's lethal range and suppressing its detection as needed [4, 5]. A target's lethal range is defined as the area of radar coverage that will result in the removal of a discovered entity, for example, with the use of a surface to air missile. Geometrically, a target's lethal range is smaller than that of a radar's coverage, and assumed, for our study, to be centered with the radar. The threat radars are stationary and currently we assume that all radars are known and we have the required information to accurately model their area of detection and associated lethal range. Hence, we know the regions the EA cannot enter, know where the PE and EA may be detected, and can conclude the proximity we need the EA to be to cloak the PE.

As mentioned previously, the EA aircraft is responsible for suppressing detection of the PE via jamming. Jamming is the use of directed electromagnetic waves to saturate the radar's receiver with noise, thereby detrimentally affecting the received information [4]. The EA's jamming capability is limited by its alignment and proximity with respect to the PE and the radar. Ideally, the EA aircraft would always reside adjacent to the PE, but the EA's proximity to the threat is explicitly constrained to reside outside of the lethal zone. However, the EA may enter threat coverage under the constraint that the jamming alignment and technique used will cloak both itself and the PE. This constrained area of successful jamming techniques that can be used is referred to as the Jamming Acceptability Region (JAR). In essence, the JAR is an

overlay of possible jamming techniques. Obviously, some techniques are more rigid than others in terms of the proximity and geometry required.

The jamming techniques fall into two main categories: preemptive and reactive jamming. The difference is that preemptive assumes *a priori* knowledge of threat radar characteristics, and reactive indicates a response to an unexpected threat emitter. Under these two categories are three types of jamming which are dependent on the alignment: out-of-alignment, side-lobe alignment, and main-lobe alignment. The out-of-alignment technique is dependent only on proximity to the threat. In this case, the EA jams all radar detection within a hemisphere, but at the cost of range. The side-lobe-alignment allows a longer range, but requires the PE to be located between the EA and threat within a certain angle. The main-lobe-alignment follows the same pattern, however, the allowable range is the furthest and the alignment angle is the tightest. Generally speaking the use of preemptive out-of-alignment jamming (PO) is the focus for our study. Although this requires a closer proximity to the PE and threat, this method allows more flexibility in jamming techniques in case the EA needs to distance itself later. However, the reactive out-of-alignment technique covers a larger area in comparison due to the reactive technique focusing only on the detected frequency, as opposed to PO which jams the entire known bandwidth. As such, we currently have RO as our desired jamming technique.

Given the background, the necessity for an EA path should be obvious. However, determining the path for optimal jamming can be time consuming. Hence the need exists for a framework that can allow us to test algorithms, various terrains, and various paths. In this paper, we discuss our success of using the A-star algorithm [6, 7, 8], a common path planning algorithm, and the benefits MATLAB provides.

This work illustrates simulation techniques and design that can be extrapolated to a variety engineering problems. While the subject matter is particular to path planning, it is common that students and researchers alike need to develop a framework that can rapidly model systems for classes and research. MATLAB provides numerous tools for rapidly developing graphically based simulations of reasonably complex system. Similar designs are feasible for undergraduates to implement in a semester long project. Additionally, this type of work could be implemented in teams to show the importance of specifications for group projects. Graphically based simulations are effective tools for facilitating deeper understanding of the material. For example, in this work, the simulations provide insight into the relationship between the PE path and EA path that would otherwise be difficult to characterize.

Method

MATLAB's strength comes from the multiple levels of interaction a user can have with the software [1, 9, 2, 3]. On a rudimentary level, MATLAB can be used purely for numeric computation, such as basic arithmetic in the command window. For repetitious tasks, a variable can be used for reassignment in a formula and recomputed, or done with matrix algebra. Given that a formula or a set of computations may be repetitively used, scripts and functions can be formed to save the tasks and executed as needed. The intermediate steps in calculations can also be suppressed, allowing only the desired data to be outputted. MATLAB's functionality obviously builds off of basic levels, allowing a user to create with various degrees of complexity.

Data is generally conveyed to the user in the console window in a numeric form.

However, numeric calculations with numerous results can be hard to comprehend or grasp in a

timely manner. For example, the results of taking the sine for degrees 1 to 360 would result in 360 possible solutions, but without knowing the mathematical operation would not result in you understanding what was done to the data. However, if the data is visually represented in a plot, the mathematical transformation is clear. Our framework makes great use of plots to convey information readily- paths are made much clearer when shown on a map then as a set of vectors. In addition, geometric limitations become much clearer when visually represented, for example plotting 3 points to gauge the alignment instead of performing regression.

Following the functionality, the interaction with MATLAB is not left only to the console window. While we currently use the console to query user defined constraints for data display, we hope to make a more aesthetic system for ease of use. MATLAB allows the creation of graphical user interfaces (GUIs) [3, 9], which allows a systematic layout that makes the options clearer while keeping the screen usage minimal. GUIs also make visualizations the focal point for viewing, as you have more control over where and when to display them.

MATLAB has built numerous toolboxes for engineering and researcher needs. These tool boxes contain implemented formulas, functions, and algorithms related to the fields they are named after. For example, in this application, we utilize the image processing toolbox to convert polygon images to binary masks. This binary representation allows us to treat the image as a matrix of zeroes and ones that can be used a graph representation for our path planning algorithm. MATLAB's toolboxes are an invaluable source for implementing new ideas without recreating foundational parts.

Application of Methods

*Proceedings of the 2014 ASEE Gulf-Southwest Conference
Organized by Tulane University, New Orleans, Louisiana
Copyright © 2014, American Society for Engineering Education*

To model the mission planning work we require three input files. First, the PE's flight path (pe_flight_path.txt) which gives the PE's waypoints and time from start to be at those waypoints. The waypoints are currently in a generic (x y z) format which corresponds to (latitude longitude altitude). Second, the parameters needed to model the radars are in threat_parameters.txt, this file contains the necessary information to calculate the radar coverage using the JATO equation and gives a radar's associated lethal range. Finally, the Electronic Order of Battle (EOB) is stored in EOB.txt, and defines the locations of the radars. These files constitute the variable constraints of the mission.

Our framework allows information to be relayed in two ways- textually and graphically. The text information consists of a jamming requirements list (JRL), which specifies the EA's jamming technique, which threat it is jamming, and the time at which it is jamming the threat. This information can be suppressed, printed per PE waypoint, or printed per a specified step size. Graphically, we have the option of 4 figures which can be viewed in any combination.

The first figure displays the PE traversing the enemy terrain, with the enemy radars depicted as red stars. When the PE enters threat coverage a red line is drawn between the PE and radar, and a composite JAR is overlaid on the radar. This composite JAR represents the ranges the emitter can be jammed from, independent of the lethal range and PE location. Within this composite JAR is a green JAR, which represents jamming techniques that could be used for the PE's current location. The PE location is represented by a black dot, and all radars for which the PE is outside of the threat coverage have a dashed yellow line connecting the two points. This dashed yellow line visually represents the alignment between the two objects in case jamming was needed. Figure 1 displays a typical screenshot from runtime.

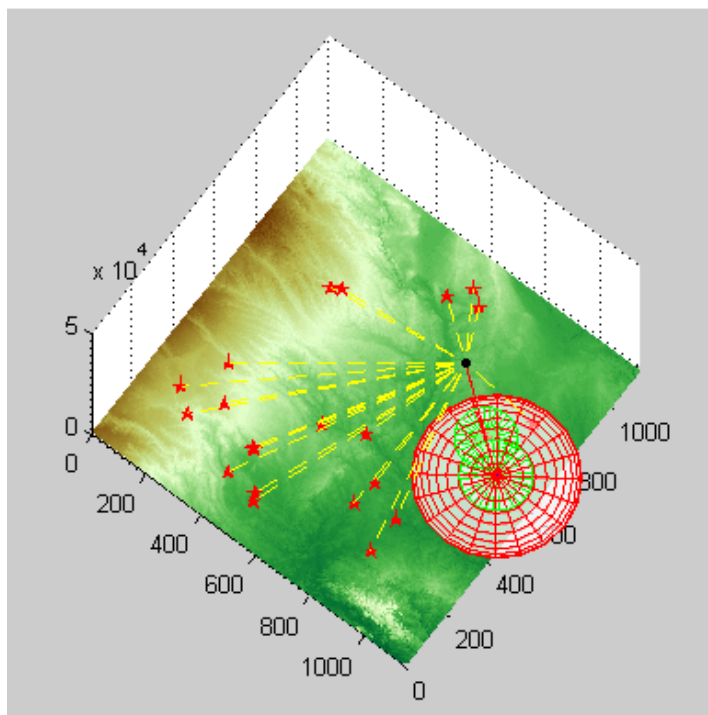


Fig. 1: PE Location and Runtime Activity

The second figure for display illustrates the 6 different available jamming techniques given the PE location and threat to jam. This information is overlaid on the radar's associated lethal kill zone to illustrate what such a technique might look like.

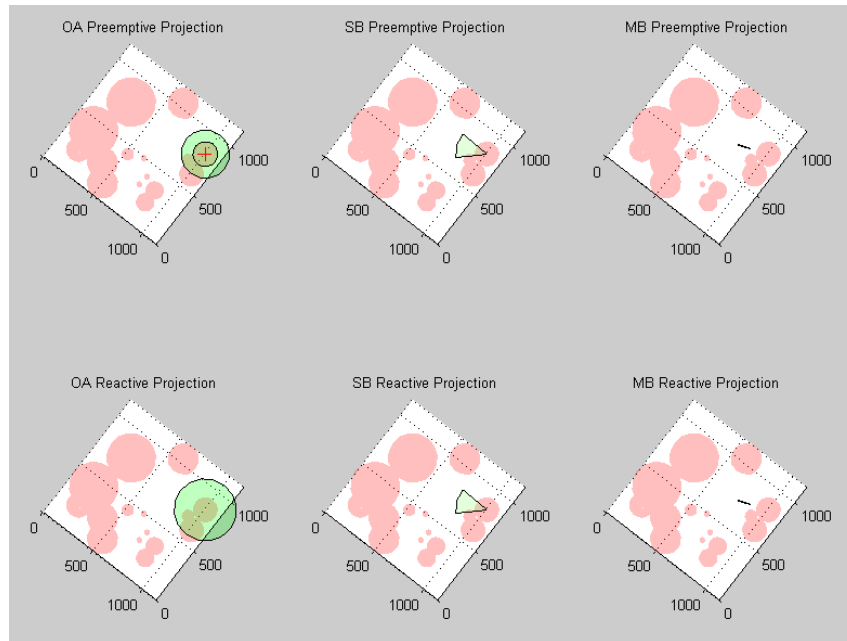
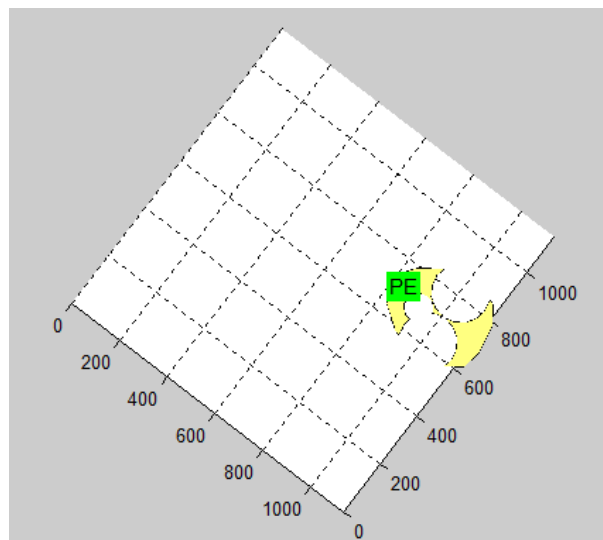


Fig. 2: Jamming Envelopes

We also have the option to display the PE in relation to the available composite JARs with the lethal range removed. Basically this figure shows the set of all possible safe areas for the EA given the set of all possible jamming techniques. This is displayed in figure 3.



faster about framework development and end results. This work is in the relatively early stages and we continue to learn about possible ways to improve both the mission-planning framework and to improve search algorithms for accomplishing mission planning. For example, future work is planned for making improvements to our methods for displaying JARs, lethal ranges, and threat coverages.

Acknowledgment of Support

This publication results from research supported by the Naval Postgraduate School Assistance Grant/Agreement No. N00244-13-2-0011 awarded by the NAVSUP Fleet Logistics Center San Diego (NAVSUP FLC San Diego). The views expressed in written materials or publications, and/or speakers, moderators, and presenters do not necessarily reflect the official policies of the Naval Postgraduate School nor does mention of trade names, commercial practices, or organizations imply endorsement by the U.S. Government.

References

- [1] The MathWorks, Inc, *MATLAB Release 2013a*, Natick, Massachusetts.
- [2] M. A. Majid, Z. A. Huneiti, M. A. Al-Naafa and W. Balachandran, "A Study of the Effects of Using MATLAB as a Pedagogical Tool For Engineering Mathematics Students," *Interactive Collaborative Learning (ICL), 2012 15th International Conference*, Sep. 2012.
- [3] The MathWorks, Inc., "MATLAB® Creating Graphical User Interfaces," 2013. [Online]. Available: http://www.mathworks.com/help/pdf_doc/matlab/buildgui.pdf.
- [4] Department of Defense, *Electronic Warfare and Radar Systems Engineering Handbook*, REV. 4 ed.,

United States, 1 June 2012.

- [5] J. Dark, J. Buscemi and S. Burkholder, "Aircrew Aid To Assess Jam Effectiveness". United States Patent 7427947 B1, 23 Sep. 2008.
- [6] L. Xia, J. Xie, C. Manyi, X. Ming and Z. Wang, "Path planning for UAV based on improved heuristic A* algorithm," *Electronic Measurement & Instruments, 2009. ICEMI '09. 9th International Conference*, pp. 3-488, 3-493, 16-19, Aug. 2009.
- [7] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics SSC4*, pp. 100-107, 1968.
- [8] B. M. Sathyaraj, L. C. Jain, A. Finn and S. Drake, "Multiple UAVs path planning algorithms: a comparative study," *Fuzzy Optimization and Decision Making*, vol. 7, no. 3, pp. 257-267, Sep. 2008.
- [9] The MathWorks, Inc., "GUI Building Basics," [Online]. Available: <http://www.mathworks.com/help/matlab/gui-building-basics.html>. [Accessed 2014].