A springs and masses model for determining the lowest risk path in a threat environment

M. P. Rowe[1] G. N. Mercer[2] H. S. Sidhu[3]

(Received 14 August 2008; revised 22 January 2010)

Abstract

Finding the safest path through a threat environment is paramount for the military. This work investigates the use of a springs and masses model for battlefield applications. In particular, we examine the use of this model in scenarios when the locations of the threat environment are not necessarily known, that is, 'pop-up' threats. The strengths and weaknesses of this approach are discussed including the potential for using this model to solve safe path problems in real-time, which would allow it to be used as a decision making tool for both field and onboard system applications.

Contents

http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/1438 gives this article, © Austral. Mathematical Soc. 2010. Published February 12, 2010. ISSN 1446-8735. (Print two pages per sheet of paper.)

1 Introduction

On the battlefield, travelling on a safe path or not can be the difference between life and death. Choosing the safest path relies on a commander's training and intuition; however, the dynamic and complex nature of the modern battlefield make these decisions difficult. Furthermore, autonomous systems have no commander to make these decisions. In these situations it is essential to have the assistance of a safest path determining model to maximise the chance of survival for the people or systems involved [9].

There are many models used to determine safe paths through threat environments; however, few models are efficient at finding safe paths when the threat environment is constantly changing. This article investigates the determination of optimal routes for a single vehicle traversing a threat environment (such as a mine field) from its base to its mission objective using a springs and masses model. Previous researchers [2, 8] examined the use of this model when the threat locations are known.

Here we extend the investigation by considering 'pop-ups'. Pop-ups are threats which appear while travelling through a threat environment and were not considered when initially determining the safest path. If the pop-up is close to the safe path, then the effect on the safest path can be dramatic. Quickly determining the new safest path is vital so that the threat can be minimised [2]. Due to the dynamic nature of the battlefield, pop-up threats can be expected often. This work examines how a springs and masses model is able to rapidly adapt to changes in the threat environment due to pop-ups. We also compare different solvers in MATLAB® to determine the most computationally efficient.

The springs and masses model presented in this work can be further extended to be either a complex and realistic simulation or a real-time decision making tool. Given the limited computational power available on the battlefield, a single application that performs both tasks is not expected in the immediate future. Further work for both roles is presented.

2 Current approaches

There are numerous continuous and discrete safe path determining models. Previous work examined calculus of variations [8, 9, 10, 11, 12], Voronoi diagram search [1, 2, 7], network flow [9, 12], grid [4] and probability map [3] models. With the exception of calculus of variations, these models discretise the region and use discrete optimisation techniques to find the safest path. Discrete models do not adapt well to dynamic environments because when the threat environment changes, the safe path must be entirely recalculated. This is very inefficient, especially considering much of the work is in recalculating threat costs for path segments that are outside the range of the vehicle [4]. The calculus of variations approach also requires the safest path to be entirely recalculated when the threat environment changes. Additionally, this approach is slow to converge on a solution and is not robust in threat environments with many threats [8, 11, 12].

3 Springs and masses model

The springs and masses model is a continuous in space method for finding safe paths through threat environments. The model uses a series of masses connected by springs in a two dimensional plane. Threats repel the masses with a force that is proportional to an inverse power of distance [2]. The inverse power depends on the type of threat. For example, the inverse power for a mine threat is two [8] while for a radar it is four [4]. Threat is assumed to be the same in all directions and for this work all threats are assumed to have the same threat level, although this sameness is not required in general. The equations of motion for a spring, mass and damper system are used to determine four ordinary differential equations (ODEs) for each mass in the series, which govern the motion of the masses towards the steady state solution. Let x and y be the usual two-dimensional cartesian coordinates and u and v the velocities in the x and y directions respectively. The ODEs for the ith mass in the series being repelled by N threats are of the form

$$\dot{x}_i = u_i\,, \tag{1}$$

$$\dot{y}_i = v_i\,, \tag{2}$$

$$\dot{u}_i = \sum_{j=1}^{N} \frac{R_j(x_i - a_j)}{{d_{ij}}^{r+1}} + 2\zeta\omega_n(u_{i-1} + u_{i+1} - 2u_i) + \omega_n^2(x_{i-1} + x_{i+1} - 2x_i), \tag{3}$$

$$\dot{v}_i = \sum_{j=1}^{N} \frac{R_j(y_i - b_j)}{{d_{ij}}^{r+1}} + 2\zeta\omega_n(v_{i-1} + v_{i+1} - 2v_i) + \omega_n^2(y_{i-1} + y_{i+1} - 2y_i), \tag{4}$$

where R_j is the threat level for the jth threat which is located at (a_j, b_j), d_{ij} is the distance of that threat to the mass, and r is the inverse power of the threat type [8]. The spring, mass and damper system has a natural frequency, ω_n, and damping ratio, ζ. The second and third terms in each of equations (3) and (4) represent the force balance of spring and repulsive

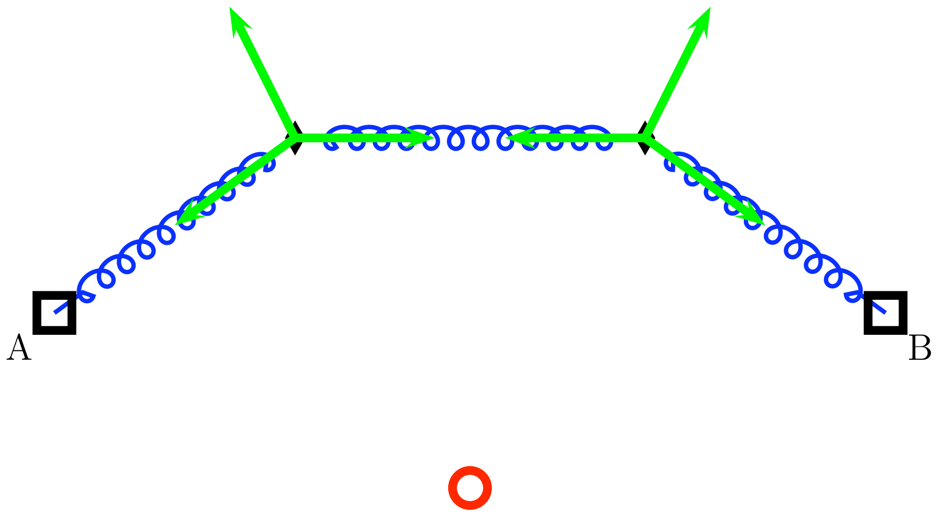

Figure 1: A schematic diagram showing an example of the springs and masses model. The safe path around the threat (red circle) between the start (A) and end (B) points (black squares) is indicated by the springs (blue coils) which connect two masses (black diamonds). The green arrows show the forces acting on the masses.

forces. If there are M masses, then the system consists of $4M$ first order ODEs. This system of first order ODEs is solved using numerical techniques (in this case using MATLAB®) to find the steady-state solution [2, 8]. The final locations of the masses are the waypoints on the safest path through the threat environment. A schematic representation of this model is shown in Figure 1. The springs and masses model offers improvements over other models; however, it does have some drawbacks, which we outline in the next section.

The main advantage of the model is that it handles dynamic environments efficiently. The computational time required to solve a system of ODEs to steady state depends on how close the initial values are to the final solution. If the initial values are close, then the computational time is reduced. Using the springs and masses model allows us to recycle the previous safe path as the initial conditions for a changed threat environment, which reduces the computational time. This potentially means that once the initial path is found, any changes to the path can be computed in real-time.

3.1 Implementation issues

Total threat Total threat is not explicitly calculated by this model. Since the location of each mass is known and each is connected by straight springs, the total threat can be found by numerically integrating the threat between each mass and finding the sum of these segments.

Range input The safe path is not useful if it exceeds the range of the vehicle that must travel along it. The length of the path is not explicitly an input to the springs and masses model. However, making range an input can be achieved by calculating the path length during each iteration of the integration. The natural frequency of the system is dynamically changed so that, as the length of the path increases, the spring constant increases, and therefore the spring force becomes much larger. This results in the range not exceeding a given maximum. For this work the natural frequency was adjusted using the equation

$$\omega_n = -1 - \log\left(\frac{l_{\text{actual}}}{l_{\text{maximum}}}\right), \tag{5}$$

where l_{actual} is the calculated path length for the iteration and l_{maximum} is the maximum allowable vehicle range.

Local minimums The solution generated by the model is not always the global minimum threat path because the path may settle in a local threat minimum if the initial conditions are sufficiently close to it. For example, a threat environment with two threats will have three local minimum threat paths: one which keeps both threats on the left, one which keeps both threats on the right, and one which passes between the two threats. To determine which is the global minimum the path can be calculated multiple times with different initial conditions. If there are multiple solutions, then the total threat for each path is calculated to determine which path is the safest. Intelligent selection of initial conditions can also assist with finding the global minimum. If there is a human 'in the loop', then their intuition is used to input a guess of where the safe path will be. If there is no human 'in the loop', then starting with the masses in random locations throughout the region will reduce the probability of finding local minimums.

Bunching Ideally there should be more masses in areas of high threat to provide better resolution. However, in this model we find that masses bunch together in areas of low threat because the repulsive force of the threats is greater than the spring force. We overcame this by dynamically adjusting the natural frequency based on threat level. If a mass is in an area of high threat, the spring force should be increased by increasing the natural frequency. This differs from the implementation of maximum vehicle range because the natural frequency is changed locally for an individual mass, not globally for the whole chain of masses.

3.2 Improving the efficiency

In order to optimise the springs and masses model, several different ODE and non-linear equation (NLE) solvers were compared to find the fastest method of solving the large system of ODEs generated by the model. Many of these solvers are specific to MATLAB®, so the applications for use in the field are limited. However, the analysis gives an insight into the best methods of solving this large system of ODEs. The solvers that were tested were the standard MATLAB® solvers `ode45`, `ode23`, `ode113`, `ode15s`, `ode23t` and `ode23tb` (`ode23s` was excluded because it was significantly slower than the others). Details of these solvers can be found in the documenation for MATLAB® [6]. An Euler time-step solver was also tested, as well as two NLE solvers: `broy` and `mmfsolve`, both of which use the quasi-Newton Broyden method and are available freely from MATLAB® Central [5]. Full details of these methods including how the initial Jacobian matrix is constructed is available from MATLAB® Central [5].

The solvers were tested in randomly generated threat environments. The initial conditions were generated by randomly placing 100 masses in a rectangular region around the fixed start and finish points. Each of the solvers then solved the identical problem and the time taken for each to find a steady-state solution was measured. Steady-state was found by measuring the sum of the speeds for all the masses until it was below a given threshold, 100 for this work. This process was repeated 933 times, each with a unique threat environment and initial conditions. No solution was checked for sensible output; the times measured were times taken to generate *an* answer, not necessarily *the* answer. The NLE solvers sometimes produce unstable steady-state solutions to the system if the initial guesses are not close to the final stable solution, so this certainly means that 'incorrect' answers were generated. In extreme cases of bad initial conditions the NLE solvers produced an error and failed. Of the 933 trials, all nine solvers generated solutions in 803 trials. The times for each function in each trial were converted to a fraction of the time taken for the `broy` function in that particular trial. This was done to allow for direct comparison between trials where the initial conditions were closer to the final solution than others, which affects the computational time. The average for each function over all the trials was then taken. The results showing the comparisons between different solvers are in Table 1.

`Broy` was determined to be significantly faster than other solvers; however, `Broy` will not always find stable steady state solutions. While a fast solution is desirable when in the field, it absolutely must be the correct solution every time. The second fastest solver was `ode113`. While this solver is slower than

Table 1: Computational time of different solvers compared to `broy`.

Solver	Relative time
`ode23tb`	5.36
`euler`	5.13
`ode23t`	4.13
`mmfsolve`	2.97
`ode15s`	2.70
`ode23`	2.51
`ode45`	2.35
`ode113`	1.76
`broy`	1.00

`broy`, `ode113` will always produce a stable steady state solution. A balance was found by using `ode113` first to find good initial conditions which were then used by `broy` to solve for the required stable solutions.

3.3 Results

Figure 2 is an example solution. In this example 20 randomly placed threats were positioned in the domain and the safe path determined from points A to B. A pop-up threat is chosen randomly to be near the safest path at some random time (point M on the figure). The new safe path is then determined. This figure shows that the emergence of a pop-up threat can significantly change the safe path. The new safe path passes through different threats, showing that the model can find global minimum threat paths. Clearly the springs and masses model is capable of finding safe paths and adapting to changes in the threat environment.

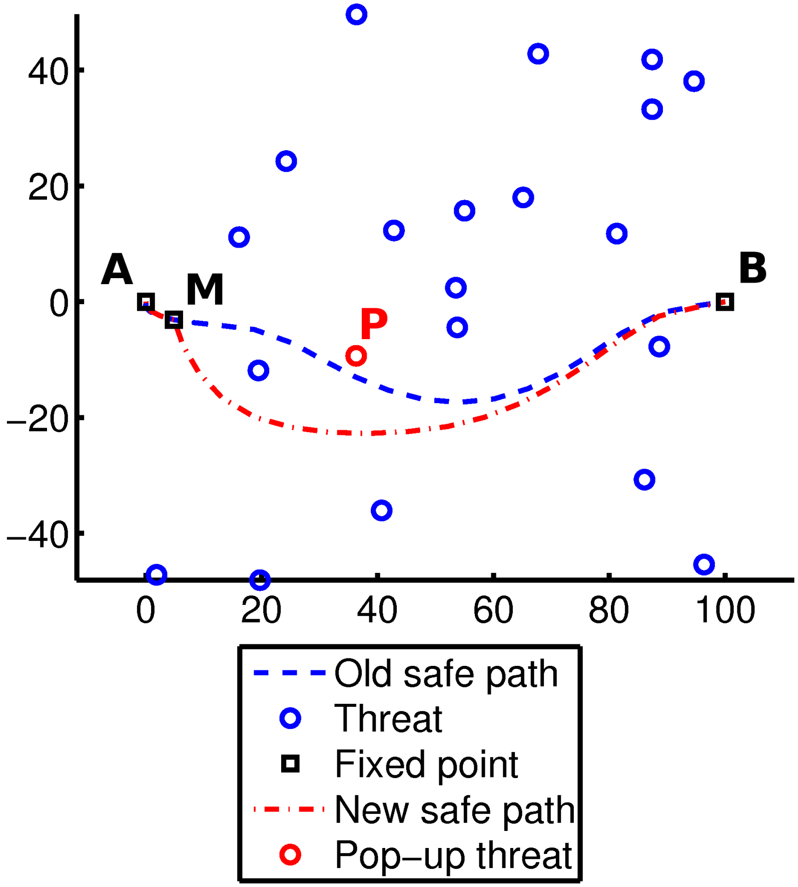

FIGURE 2: An example problem. The initial safe path (blue dashed line) is calculated based on the known (random) threat environment of 20 threats (blue circles) between two fixed points (black squares A and B). Also shown is the scenario of a vehicle that has travelled along this path to black square M when a new threat appears (red circle P). The new safe path (red dot-dashed line) is calculated using the previous safe path as the initial conditions for the masses. The code is completely automated and requires no human input.

3.4 Further work

There are many ways to improve the model's usefulness for military applications. As well as refining the model, this work has implemented additional fixed points. Fixed points are points that must be travelled over. Some models are limited to only a start and end point, so the addition of these extra fixed points makes the springs and masses model more useful for military applications. The implementation essentially involves having a mass that is not moved by spring forces or repelled by threats. This is done by setting both the accelerations and velocities for that mass to zero. When the system of ODEs is solved, the corresponding mass does not move.

The springs and masses model presented in this work is by no means the perfect solution to all path determining problems. There are still many improvements that can be made to the model to increase its usefulness in real-world applications. Some of the improvements that we are currently examining are 'moving' fixed points and a real-time model. Moving fixed points are waypoints that simulate situations such as moving targets or moving airfields (for example, an aircraft carrier). Furthermore, the model must also provide solutions in near real-time to enable it to be used as a decision making tool by commanders rather than a tool for checking the decisions that have already been made.

4 Conclusion

Finding the safest path through threats is paramount on the battlefield and can be the difference between life and death. The modern battlefield is a complex environment, so there is a need for safe path determining models for applications both with and without a human 'in the loop'.

The springs and masses model uses the equations of motion of masses connected in a chain by springs to form a system of ODEs which are then numerically solved. The closer the initial conditions are to the final solution, the less the computational time that is required to solve the problem to steady-state. As a result, the model can recycle an old safe path as the initial guess for a new safe path if the threat environment changes. This is very important for adapting to pop-up threats and means that new safe paths can be found quicker using this approach than other techniques, such as spatial discretisation models.

The total threat of a path has been incorporated into the model, and this has been used to distinguish between different solutions to find the safest path. Further improvements that have been implemented include the addition of range as an input to the model to account for vehicle limitations, and preventing masses from bunching together in areas of low threat by adjusting the spring forces acting on each mass based on the threat level.

References

[1] R. W. Beard, T. W. McLain, M. Goodrich, and E. P. Anderson. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Transactions on Robotics and Automation*, 18:911–922, 2002. http://ieeexplore.ieee.org/iel5/70/25967/01159009.pdf. C1068

[2] S. A. Bortoff. Path planning for UAVs. In *Proceedings of the American Control Conference*, number 6, pages 364–368, Sep 2000. doi:10.1109/ACC.2000.878915. C1067, C1068, C1069, C1070

[3] M. Jun and R. D'Andrea. Path planning for unmanned aerial vehicles in uncertain and adversarial environments. In S Butenko, R Murphey, and P Paralos, editors, *Cooperative Control: Models, Applications and Algorithms*, chapter 6, pages 95–110. Kluwer Academic Publisher, 2003. http://www.seas.ucla.edu/coopcontrol/papers/02cn04.pdf. C1068

[4] J. J. Leary. Search for a stealthy flight path through a hostile radar defense network. Master's thesis, Naval Postgraduate School, Monterey, CA, USA, March 1995. http://handle.dtic.mil/100.2/ADA297669. C1068, C1069

[5] The MathWorks. MATLAB central. http://www.mathworks.com/matlabcentral/. C1073

[6] The MathWorks. MATLAB documentation. http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/math/f1-662913.html. C1073

[7] T. W. Mclain and R. W. Beard. Trajectory planning for coordinated rendezvous of unmanned air vehicles. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 1247–1254, 2000. http://citeseer.ist.psu.edu/306568.html. C1068

[8] G. N. Mercer and H. S. Sidhu. Two continuous methods for determining a minimal risk path through a minefield. In W Read and A J Roberts, editors, *Proceedings of the 13th Biennial Computational Techniques and Applications Conference*, volume 48, pages C293–C306, July 2007. http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/56. C1067, C1068, C1069, C1070

[9] R. Murphy, S. Uryasev, and M. Zabarankin. Trajectory optimization in a threat environment. Research report 9, Department of Industrial and Systems Engineering, University of Florida, July 2003. http://www.ise.ufl.edu/uryasev/Trajectory_optimization.pdf. C1067, C1068

[10] M. C. Novy. Air vehicle optimal trajectories for minimization of radar exposure. Master's thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, USA, March 2001. http://handle.dtic.mil/100.2/ADA390154. C1068

[11] H. S. Sidhu, G. N. Mercer, M. J. Sexton, N. A. Ansari, and Z. Jovanoski. Optimal path trajectories in a threat environment. *Journal of Battlefield Technology*, 9(3):33–39, November 2006. http://www.argospress.com/jbt/. C1068

[12] M. Zabarankin, S. Uryasev, and P. Pardalos. Optimal path risk algorithms. In R Murphey and P Pardalos, editors, *Cooperative Control and Optimization*, chapter 13, pages 273–298. Kluwer Academic Publisher, 2002. doi:10.1007/0-306-47536-7-13. C1068

Author addresses

1. **M. P. Rowe**, School of Physical, Environmental and Mathematical Sciences, University of New South Wales at the Defence Force Academy, AUSTRALIA.
2. **G. N. Mercer**, School of Physical, Environmental and Mathematical Sciences, University of New South Wales at the Defence Force Academy, AUSTRALIA; and National Centre for Epidemiology and Population Health, Australian National University, AUSTRALIA.
   mailto:Geoff.Mercer@anu.edu.au
3. **H. S. Sidhu**, School of Physical, Environmental and Mathematical Sciences, University of New South Wales at the Defence Force Academy, AUSTRALIA.
   mailto:h.sidhu@adfa.edu.au