# Practical application of pseudospectral optimization to robot path planning

**Steven Martin**
Queensland University of Technology
and CSIRO ICT Centre
Brisbane, Australia
steven.martin@qut.edu.au

**Nick Hillier**
CSIRO ICT Centre
Brisbane, Australia
nick.hillier@csiro.au

**Peter Corke**
Queensland University of Technology
Brisbane, Australia
peter.corke@qut.edu.au

## Abstract

To obtain minimum time or minimum energy trajectories for robots it is necessary to employ planning methods which adequately consider the platform's dynamic properties. A variety of sampling, graph-based or local receding-horizon optimisation methods have previously been proposed. These typically use simplified kino-dynamic models to avoid the significant computational burden of solving this problem in a high dimensional state-space. In this paper we investigate solutions from the class of pseudospectral optimisation methods which have grown in favour amongst the optimal control community in recent years. These methods have high computational efficiency and rapid convergence properties. We present a practical application of such an approach to the robot path planning problem to provide a trajectory considering the robot's dynamic properties. We extend the existing literature by augmenting the path constraints with sensed obstacles rather than predefined analytical functions to enable real world application.

## 1 Introduction

Path planning is a fundamental building block for robotic platforms. To move from configuration $A$ to configuration $B$, whether that be the motion of an arm, mobile robotic platform or higher-order mechanism, a valid plan must be developed. The plan provides path (or trajectory) inputs to the controllers of actuators so that the task can be executed to successful completion. This plan is subject to a wide variety of constraints that range from the kinematic constraints of the robot to dynamic effects and obstacles in the environment.

The state-of-the-art in readily available robotic planning algorithms is diverse, both in implementation and the types of constraints for which the planner can account. Inclusion of the dynamic response of the robot to the control inputs and its current state is rarely considered, nor are truly 3D representations of the environments. Instead, purely kinematic global path planners dominate the literature; such as the popular lattice based methods of A* [Hart *et al.*, 1968] (and its variants such as the iterative depending A* algorithm (IDA*) [Korf, 2001] and the incremental replanning variant D* [Stentz, 1995]), the sampling based methods such as PRMs [Geraerts and Overmars, 2003], the popular RRT [LaValle and Kuffner, 2001] and the more recent RRT* [Karaman and Frazzoli, 2010] and potential field methods such as the method of elastic bands discussed in [Quinlan and Khatib, 1993]. Practical implementations of such methods generally involve smoothing the resultant path to ensure kinematic feasibility and implementing a reactive obstacle avoidance mechanism. Some, such as the Dynamic Window Approach path follower of [Fox *et al.*, 1997] or the constrained local path exploration method of [von Hundelshausen *et al.*, 2008] include limited knowledge of the platform's kinodynamic constraints.

Accounting for dynamics locally is sufficient to solve many simple path planning problems however these approaches show reduced performance for vehicles with significant dynamics. This can lead to failures when attempting to follow a global path which is dynamically infeasible along with reduced overall performance. A globally optimal solution which is capable of accounting for vehicle dynamics and incorporating terrain information is expected to provide an ideal solution to this problem.

Solving for a global path whilst simultaneously considering the robot's dynamic properties has been often proposed, however its practical application has been limited due to the significant computational cost. Extending the search for a dynamically feasible (and preferably locally optimal) trajectory in 3D for ground robots is significantly more computationally expensive than for the 2D case and is often considered intractable due to the high dimensionality of the state-based search space. Approaches that operate on a fundamental description of the robot's environment and dynamic state constraints to solve the path-planning problem as a global optimisation have generally been shown to be computationally infeasible. One such method that showed promise is that of [Shiller and Gwo, 1991] where the terrain is approximated by a 2.5D height-field and a number of candidate trajectories are computed based on a sampled assessment of traversability. These candidates are then optimised through standard minimisation approaches to produce a locally optimal dynamic path. More recent approaches to solving this problem include the variety of non-linear programming approaches to solving the optimal control problem in an extended state-space [Howard and Kelly, 2007], polynomial approximations of the trajectories [Kelly and Nagy, 2003] and multi-resolution searches in a state-space lattice [Likhachev and Ferguson, 2009].

These works highlight the current state-of-the art in efficient and dynamically optimal approaches to solving the path planning problem in 2D and 3D environments. Solutions incorporating dynamics have been experimentally proven to not only be practical in an application, but also provide superior solutions by taking into account the platform's response as part of the planning process.

### 1.1 Pseudospectral optimisation for robotics

The recent rise of so-called pseudospectral methods for tackling the problem of optimisation in high dimensional spaces has brought renewed interest to the robotic path-planning problem. Although some may argue that the optimal control and path planning problems are sufficiently different that control optimisation methods are not amenable to domain transfer. It is easy to show that if the problem description moves beyond simplistic graph and lattice searches, posing the path-planning problem in an optimal control theoretic manner is not only attractive, but for most, if not all problems, a more natural representation. Such a formulation simply takes some (not necessarily linear) state-space representation of the robotic platform, typically it's governing dynamic representation and adds the world representation as the constraints to the problem. The world representation in this framework is flexible and can vary from simple binary descriptions of obstacles to full 3D representations and can include, for instance, material interaction properties such as friction. A cost function is then proposed for minimisation subject to the system dynamics and constraints. Typically, this minimisation function is based on time or energy.

Pseudospectral methods have been gaining widespread popularity in the domain of optimal control and the performance of such methods has been promoted by the high-profile application to space craft such as the re-orientation of the International Space Station in what was termed a zero-propellant manoeuvre [Bedrossian *et al.*, 2009], and the hybrid solar-sail pole-sitting craft proposed in [Ceriotti and McInnes, 2010]. The methods are now highly developed, allowing for not only efficient application to problems of high dimensionality with non-linear constraints, but also those with multiple *phases* whereby the constraint or state-space manifolds may be disjoint or have other non-continuous or non-smooth characteristics (provided such are known prior to run-time and can be programatically represented). For example the stepwise mass change of a chemical booster rocket being ejected during a spacecraft ascent.

Application of these approaches to the robotics path planning problem has been spearheaded by the work conducted at the Naval Postgraduate School in Monterey, USA [Bollino *et al.*, 2007; Gong *et al.*, 2009; Hurni *et al.*, 2010] who have described the advantages of such methods for robotic path planning, but results so far have been limited to theoretical applicability to wheeled robots, fixed-wing aerial robots and large naval vessels. Whilst these scenarios have been further backed up by simulation results (and in the case of [Hurni *et al.*, 2010], experimentally applied to a wheeled robot) in ideal (fully defined, analytically describable) environments, they lack results of practical application with sensed environments. In particular, these works are limited in real-world practicality by the use of obstacle representations that can be expressed as continuous analytical functions; $p$-norms are used to represent diamond, circular and square primitives (or many morphologies in between) and similar 3D extrusions of these.

Here, we intend to demonstrate that pseudospectral optimisation methods can be usefully applied to the robot path planning problem, without resorting to the analytical expression of obstacle constraints. We make no claim that the approach is better than any other path planning algorithms, indeed for the platform used in this work it is likely that standard lattice methods would perform better due to the lack of any significant platform dynamics. However confirmation that the method is indeed practical in a real-world environment is considered a worthy contribution.

## 2 Pseudospectral optimisation

A good introduction to the theory and practical implementation of pseudospectral optimisation methods can be found in the manual for the open-source PSOPT software [Becerra, 2010a] or the accompanying paper [Becerra, 2010b]. The manual also lists a number of commercial tools for solving large scale optimal control problems. An abbreviated problem formulation is as follows:

Find the control trajectories, $u^{(i)}(t)$, $t[t_0, t_f]$, state trajectories $x^{(i)}(t)$, $t[t_0, t_f]$, static parameters $p^{(i)}$, and times $t_0, t_f, i = 1, ..., N_p$ , where $N_p$ is the number of discrete problem phases, to minimise a performance index consisting of both endpoint (Mayer) and/or integral (Lagrangian) costs subject to:

- differential constraints (the dynamic model);
- path constraints (obstacles);
- event constraints (state, input and/or parameters must equal some value as a function of time, e.g. the robot must start in configuration $A$ and finish in configuration $B$);
- phase linkage constraints (how state, input and/or parameters map from disjoint spaces);
- bound constraints (typically input or state saturation limits); and,
- temporal constraints (time must flow forwards).

Usually, an optimisation problem such as that defined above is solved via regular discretization of the solution time-interval and then re-formulating the problem as one in non-linear programming (NLP). This can then be solved by any of the standard NLP approaches. The nodes of the discretization are termed *collocation* points/nodes and the continuous solution is found via some form of interpolating function (typically a low-order polynomial) constrained by these nodes.

Pseudospectral methods differ from the standard approach by utilising better approximating polynomials for the reconstruction interpolation and hence offer improved accuracy in the computation of the derivatives and integrals required in the sampling of the state space to formulate the NLP problem. The discretization is also irregular in an attempt to not only provide a better choice of nodes for the polynomial and hence increase the approximating functions' accuracy, but also to reduce the number of nodes required and hence the cost of computing a solution. In this work, we use a pseudospectral optimisation implementation found in PSOPT [Becerra, 2010a] which uses Legendre or Chebyshev polynomials and a discretization based on Gauss-Lobatto nodes.

## 3 Implementation details

We investigated the feasibility of applying the pseduospectral optimisation approach to solving the path planning problem in an indoor office environment for a platform based on the iRobot Create (see Figure 1). The platform consists of the standard Create platform, a Hokuyo scanning laser range-finder, a net book computer running the ROS middleware [Quigley *et al.*, 2009] and a modified version of the open-source Create drivers from Brown University [Brown University, 2010]. For performance and ease of debugging, the planning and localisation software were run off-board for all results presented here.

We boot-strap our implementation with the significant functionality available in the ROS navigation stack. Localisation and mapping capabilities are provided by the open-source ROS:slam_gmapping package. This package generates a map from laser scan data and odometry which is used to populate the global cost map for the ROS navigation stack. Within the navigation stack there are two path planners which are invoked, a global and local planner.

The planner proposed in this work has been integrated to work within the ROS navigation stack as the global planning algorithm, using the localisation and mapping information available from the ROS:slam_gmapping package. Execution on the robot uses the default local path planning and reactive obstacle avoidance capabilities available within the ROS navigation stack.

The overall process flow for the robot to move from a start to goal pose is as follows. The robot obtains and continuously updates information about its environment, produces a map and localises itself within the environment using the functionality available within the ROS:slam_gmapping package. This map, the current robot pose and the goal pose is passed to the planner. The planner processes the map to formulate the path constraints. The planner performs an optimisation process to produce a feasible path, as well as the associated robot state and control inputs for execution based on the robot model and constraints. Optionally, the D* planning implementation available as the default planner within the ROS navigation stack can be executed to provide an initial guess to the optimisation to improve the convergence properties of the solution. The path is then executed, utilising an appropriate local planner for path following and reactive obstacle avoidance.

A simulation environment for the Create platform was also implemented in the OpenRAVE software [Diankov and Kuffner, 2008] to ease regression testing and allow for simple experimentation with common environments that often lead to pathological behaviour or failure in naïve planning algorithms.

### 3.1 Problem postulation

For this application, the path planning problem was posed as a time-minimisation problem operating in a single phase space with the computed control inputs in velocity space (linear and rotational velocities). Event constraints were simply defined as the platform's start pose at $t_0$ and goal pose at $t_f$. Bound constraints on the state were based on the limits of the map and an arbitrary rotational limit (valid paths must not involve routes with more than a set number of full platform rotations in a single direction, chosen to be 4). In the

Figure 1: The robotic platform used for the experiments in this work, based on the iRobot Create hardware and ROS middleware.

absence of a platform with significant dynamic characteristics, the bound constraints on the inputs were chosen to limit the platform's performance and thus more easily demonstrate the method's viability, with the velocity constraints selected as 0 to 1 m/s linear (forwards only) and −4 to 4 rad/s rotational.

The dynamic constraints follow a simplistic model for a differential-drive robot in planar operation with 3 degrees of freedom ($x$,$y$,$\theta$) for 2 control inputs ($u_v$,$u_\theta$):

$$\dot{x} = u_v \times \cos(\theta) \quad (1)$$
$$\dot{y} = u_v \times \sin(\theta) \quad (2)$$
$$\dot{\theta} = u_\theta \quad (3)$$

### 3.2 Path constraints

The defined path constraints differ from the analytical expressions presented in earlier works [Bollino *et al.*, 2007; Gong *et al.*, 2009; Hurni *et al.*, 2010] and it is this empirical description that allows for practical application of the method with real sensor data and without significant intermediate structural interpretation of the sensor data to present the planner with an idealised and analytically differentiable world model.

The environment was defined by a standard occupancy grid, populated with obstacle data from the scanning laser range-finder. The ROS navigation stack creates a global cost map where zero cost denotes free space with increasing values for the defined clearance region, possible collision, collision and unknown cell states. All cells in the grid other than known free space are treated as obstacles for the path planning implementation.

Such a representation is a poor candidate for input into an optimisation routine due to the grid-based discretisation of the environment and the discontinuities between unobserved, free and obstacle space. Furthermore, these regions are flat and thus do not provide information to the optimiser on how to exit obstacle/unobserved space, nor does it provide information about the location of nearby obstacles given a point in free space. To overcome these problems, a simple constant-gradient representation was employed in a manner similar to defining a linear potential field around the obstacles.

Each cell in the occupancy grid is thus assigned a value based on a linear 2D distance transform, decreasing from obstacle boundaries into free space and increasing otherwise, to create a distance cost grid. To achieve a continuous representation, each point in the map queried by the solver is assigned a value based on bilinear interpolation of the nominal values at the centre of the four closest distance cost grid cells. A comparison of the two representations is presented in Figure 2.

The path constraint ($h$) is then specified by an inequality based on the cost of the distance cost grid at the obstacle boundaries ($C_b$) minus any additional cost increment used to specify a clearance offset that may be desired ($C_c$).

$$-\infty <= h <= C_b - C_c \quad (4)$$

### 3.3 Solving the path planning problem

To seed the solver, an initial guess was provided in the form of the default path solution provided by the ROS navigation stack, a D* [Stentz, 1995] implementation. This assists in rapid convergence and reduces susceptibility of the solver to convergence to infeasible local minima, but is not imperative (the planner often works with a zero'd seed state). Whilst a D* implementation contains unnecessary any-time incremental re-searching capabilities, its ready availability within the ROS framework is attractive for the feasibility study conducted in this work.

The solver was then run, utilising a numerical differentiation routine, with multiple mesh refinement steps which are used to improve the accuracy of the solution.

## 4 Results

A variety of simple tests were conducted to evaluate the feasibility of the generated paths, and this section presents the results of these.

### 4.1 Simulation Results

Three environments were implemented in the simulator, including two common degenerate scenarios (a culdesac and U-shaped corridor) and a simplified and slightly modified version of an office space environment used in testing of the real platform (see Figure 3). The culdesac scenario tests for failure due to either poor initial guess plans or the inability of the planner to escape local minima. The U-shaped corridor plan is a favourite

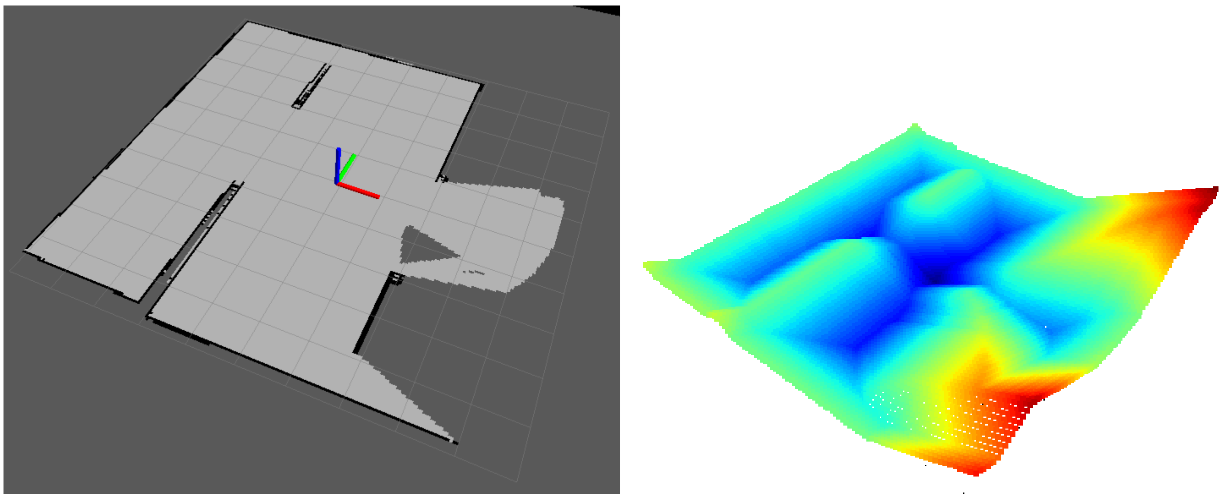

Figure 2: A comparison between the basic global map provided by the ROS navigation stack for a simple indoor environment (left) and the distance cost grid used in the path constraint representation for the same environment (right). The ROS map distinguishes between free space (light grey), obstacles (black cells) and unexplored space (dark grey). The map shows a 1m grid overlaid to indicate scale and consists of 0.05m cells.

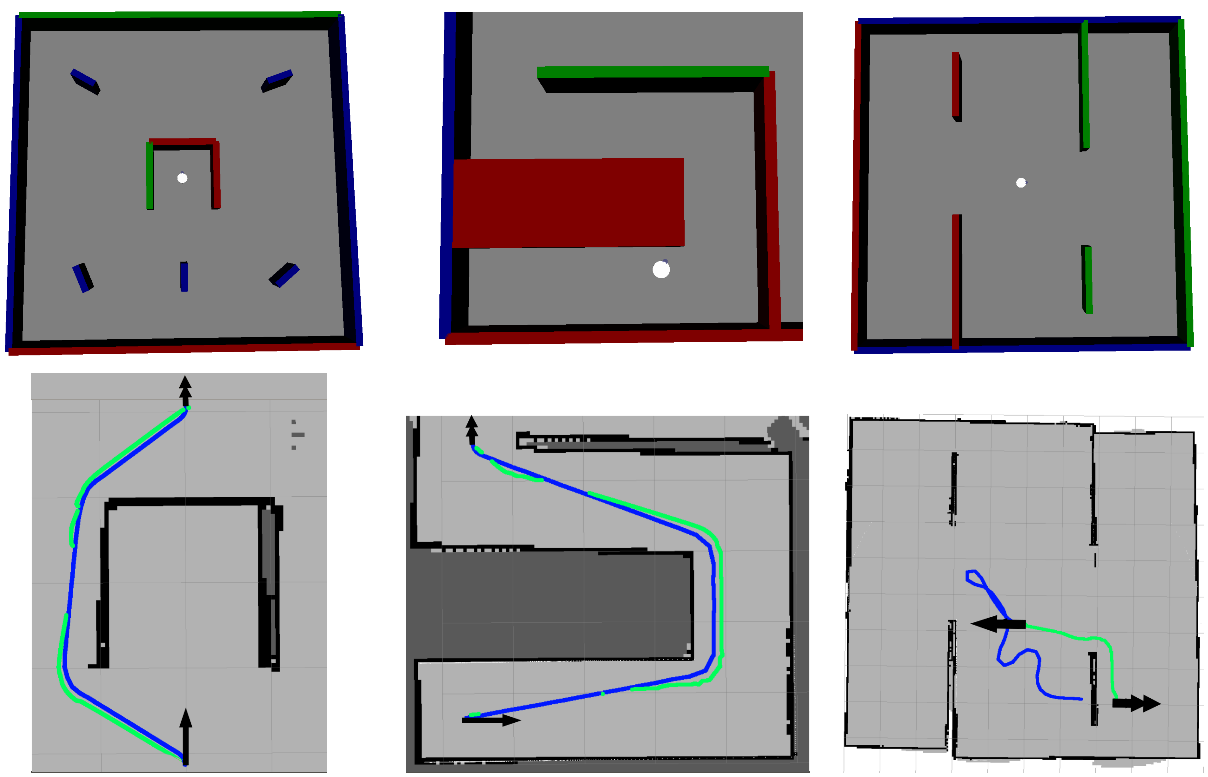

Figure 3: From left to right: (top row) the culdesac simulation environment (note extra features to assist in the SLAM computations), the corridor simulation environment and the office simulation environment. (bottom row) The map provided by the ROS navigation stack with some paths produced by the planner (blue) and the D* solution from the ROS navigation stack (green) for comparison. Note the poor result of the dynamic planner in the simulated office environment, believed to be due to a poorly mapped environment resulting in poor constraint modelling.

amongst the literature and tests planning viability in a constrained environment with large deviation from a simple straight-line start to goal plan.

Plots of some typical results are presented in Figure 3. Whilst the final paths are generally quite similar to the solutions provided by the D* solution from the ROS navigation stack implementation, due mostly to the fact that the platform model does not contain any significant dynamic terms, three characteristics of the dynamically optimised path are of note.

Firstly, the D* paths contain no heading information, whilst not only do the dynamically optimised paths contain this information, and hence the paths contain curvature elements near the ends to ensure proper orientation at the end-pose, but the dynamically optimised paths also contain estimates of the full state including the control inputs required to follow the paths (not plotted). This information can be used to advantage in several ways, for example to seed the path following controller, such that the controller essentially becomes a disturbance rejection mechanism, or to provide ideal-case state and input estimates for an on-line parameter estimator (e.g. for tyre force or terrain slip estimation).

Secondly, the D* paths are distance optimised, whilst the dynamically optimised paths presented here are time optimised. This can lead to significantly different solutions. The result in Figure 4 is a classic case-in-point. Here the additional information from the dynamic model, in particular the heading information, allows for more natural paths with minimal turn-in-place operations. This is further exemplified in the result presented in Figure 5 whereby the desired pose is directly behind the robot and in the same direction as the start pose.

Finally, the dynamic planner is susceptible to poor or non-convergence in environments with poor models of the path constraints. The results for the simulated office environment of Figure 3 show how the planner may provide a poor solution in such circumstances. Here, the wall obstacles that are being considered have been mapped whilst the robot had a poor localisation solution. This results in the obstacles containing (false) observed space and discontinuities corrupting the distance cost grid.

### 4.2 Experimental Results

The robot was operated in a simple, but cluttered indoor office environment and a series of point-to-point path planning tests conducted. Performance was near identical to that observed in the simulator. The planner was able to successfully plan dynamically feasible paths around obstacles as shown in Figure 6, however showed similar susceptibility to non-convergence in cases with poor modelling of path constraints, Figure 7. Figure 8 shows the costmap for the office environment.

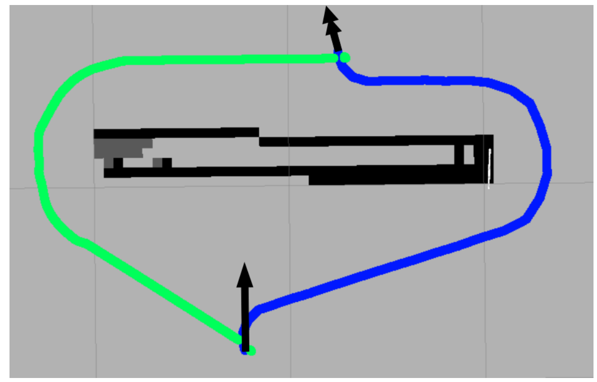

Figure 4: Comparison between a minimum distance plan (green) and the generated time-optimal dynamically feasible plan (blue) for a goal pose on the otherside of a rectangular obstacle.

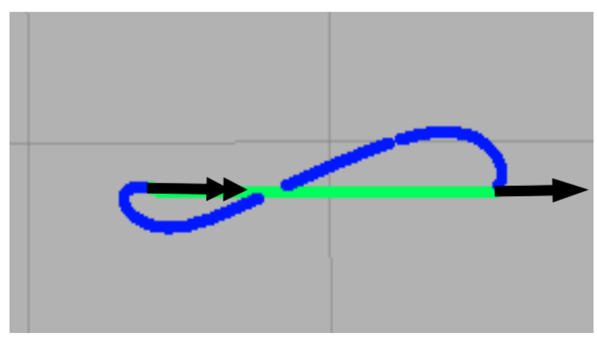

Figure 5: Comparison between a minimum distance plan (green) and the generated time-optimal dynamically feasible plan (blue) for a goal pose directly behind the start pose and aligned in the same direction.

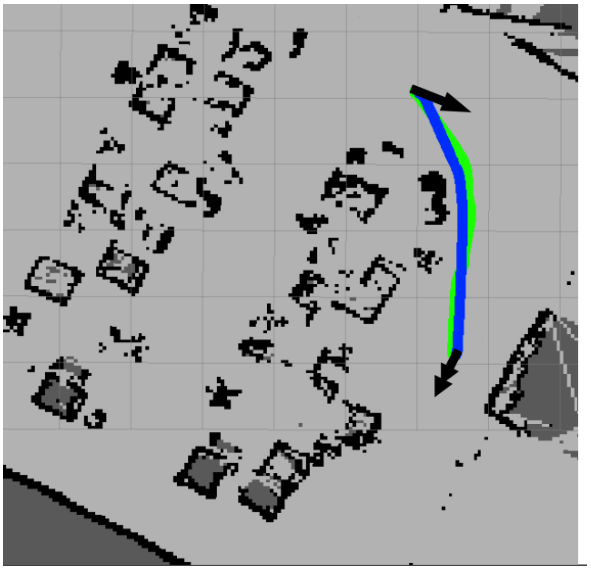

Figure 6: Experimental comparison between a minimum distance plan (green) and the generated time-optimal dynamically feasible plan (blue) in a cluttered office environment.

To improve the convergence properties additional preprocessing steps to remove unreachable free space and to perform additional smoothing of the path constraints has been proposed. Such operations are expected to provide a significant reduction in local minima leading to better convergence but have not been tested on the current implementation.

Typically, the predicted execution time for the paths was an order of magnitude shorter than the computation time required to calculate the paths (e.g. a predicted 7 second execution time took about 1 minute to compute) when calculated on a machine based on a Intel Core2 Duo T9300 running Ubuntu 10.04.

## 5 Conclusions

This work has investigated the feasibility of pseudospectral optimisation methods to the robot path-planning problem. This method plans trajectories that are optimal against a defined criterion (here, time) and are dynamically feasible. This paper extends the current literature concerned with pseudospectral optimisation to the problem of robotic path planning with the use of measured environments, as opposed to those which have been pre-programmed and are described through analytical expression of the path constraints.

We conclude that the method is indeed feasible for computation of trajectories for robotic platforms while accounting for the platform dynamics with sensed environments. The computational cost is significant, but pseudospectral methods have been shown to provide

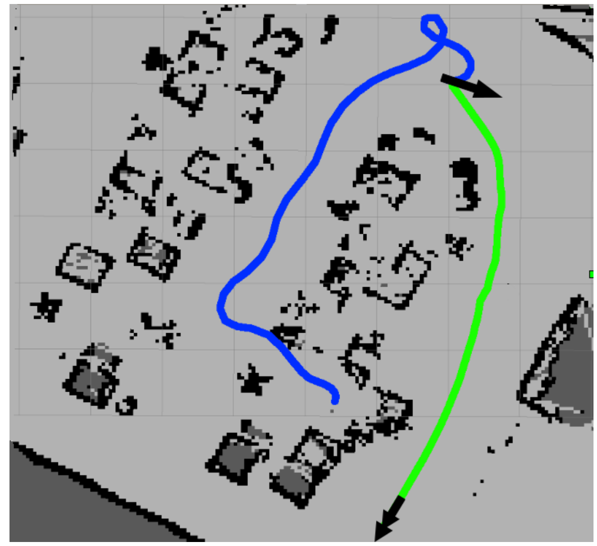

Figure 7: Failure of planner (blue) in an office environment.

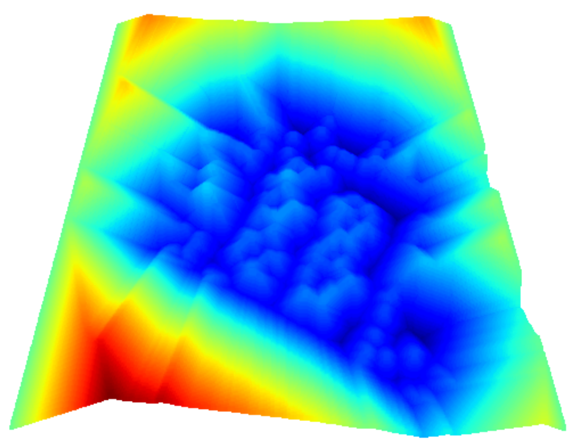

Figure 8: Distance costmap of the office environment presented in Figure 7.

a much reduced computational burden over alternate state-space trajectory optimizers with path constraints. The approach is not particularly advantageous for the platform and application shown here and traditional kinematic-based methods would likely result in faster execution with significantly less computational cost. Other works have shown that path planning based on full dynamic characterisation of the platform can provide significant benefits for a large, fast moving or low power-to-weight-ratio platforms and the proposed method has the potential for planning paths over 3D terrain while also accounting for various slip and stability constraints. These latter effects are particularly important for fast moving platforms which is our research focus. Adapting the technique to new platforms becomes a simple matter of redefining the platform's model since it does not require parameterised local path representations or other similar abstractions to define the robot's behaviour.

## 6 Future Work

The approach here has shown that pseudospectral optimisation has potential to be applied in a path planning context however further development is still required. The convergence properties of the algorithm are not sufficient to provide a real time and reliable planning solution, additional pre-processing to produce smoother obstacle map is expected to improve convergence rate and reliability of the algorithm.

The optimisation is also susceptible to local minima, while in simple environments the minimum distance seed path from D* algorithm is expected to be close to the global minima however this may not be the case in real world environments. To account for this a number of candidate paths may be assessed or a near optimal path identified before the final optimisation such as in [Iagnemma *et al.*, 2008]. A near optimal path is also expected to show rapid convergence.

The planner also computes the control inputs for path execution and this information can be utilised by path following algorithms. This is of particular interest for vehicles travelling at high speeds and over varying terrain and could provide control input estimates to reduce the required online capabilities of path followers. This is expected to reduce path following error and allow higher speeds to be safely attained.

## Acknowledgments

The authors wish to thank the following for their support. QUT's CyPhy Lab and CSIRO's ICT Centre and Minerals Down Under Flagship for providing funding and equipment for this work. Julian Ryde, Paul Flick and Reece Wadeson for assistance in getting base functionality available on the iRobot Create platforms. Victor Becerra for providing a quality open-source pseudospectral optimisation package.

## References

[Becerra, 2010a] V. M. Becerra. *PSOPT Optimal Control Solver User manual*. University of Reading, School of Systems Engineering, 2010.

[Becerra, 2010b] V. M. Becerra. Solving complex optimal control problems at no cost with psopt. In *Proc. IEEE Multi-conference on Systems and Control*, pages 1391–1396, Yokohama, Japan, September 7-10, 2010 2010.

[Bedrossian *et al.*, 2009] N. S. Bedrossian, S. Bhatt, W. Kang, and I. M. Ross. Zero propellant maneuver guidance. *IEEE Control Systems Magazine*, 29:53–73, 2009.

[Bollino *et al.*, 2007] K. P. Bollino, L. R. Lewis, P. Sekhavat, and I. M. Ross. Pseudospectral optimal control: A clear road for autonomous intelligent path planning. In *AIAA Infotech@Aerospace 2007 Conference and Exhibit*, Reston, VA, 7-10 May 2007. AIAA.

[Brown University, 2010] Brown University's repository for ROS tools and nodes, September 2010. http://code.google.com/p/brown-ros-pkg/.

[Ceriotti and McInnes, 2010] M. Ceriotti and C.R. McInnes. An earth pole-sitter using hybrid propulsion. In *AIAA/AAS Astrodynamics Specialist Conference 2010*, Toronto, Canada, 02-05 Aug 2010.

[Diankov and Kuffner, 2008] R. Diankov and J. Kuffner. OpenRAVE: A planning architecture for autonomous robotics. Technical Report. CMU-RI-TR-08-34, Robotics Institute, July 2008.

[Fox *et al.*, 1997] D. Fox, W. Burgard, and S. Thrun. The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33, 1997.

[Geraerts and Overmars, 2003] R. Geraerts and M.H. Overmars. A comparative study of probabilistic roadmap planners. *Algorithmic Foundations of Robotics V*, pages 43–58, 2003.

[Gong *et al.*, 2009] Qi Gong, L. R. Lewis, and I. M. Ross. Pseudospectral motion planning for autonomous vehicles. *Journal of Guidance, Control and Dynamics*, 32(3):1039–1045, May-June 2009.

[Hart *et al.*, 1968] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions of Systems, Science and Cybernetics*, 4(2):100–107, 1968.

[Howard and Kelly, 2007] T. M. Howard and A. Kelly. Optimal rough terrain trajectory generation for wheeled mobile robots. *International Journal of Robotics Research*, 26(2):141–166, 2007.

[Hurni *et al.*, 2010] M. A. Hurni, P. Skehavat, M. Karpenko, and I. M. Ross. A Pseudospectral Optimal Motion Planner for Autonomous Unmanned Vehicles. In *Proceedings of the IEEE American Control Conference*, pages 1591–1598, Baltimore, MD, USA, June 30 - July 02 2010. IEEE.

[Iagnemma *et al.*, 2008] K. Iagnemma, S. Shimoda, and Z. Shiller. Near-optimal navigation of high speed mobile robots on uneven terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4098–4103. IEEE, 2008.

[Karaman and Frazzoli, 2010] S. Karaman and E. Frazzoli. Incremental sampling-based algorithms for optimal motion planning. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[Kelly and Nagy, 2003] A. Kelly and B. Nagy. Reactive nonholonomic trajectory generation via parametric optimal control. *International Journal of Robotics Research*, 22(7-8):583–601, 2003.

[Korf, 2001] R. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial intelligence*, 129(2):97–109, 2001.

[LaValle and Kuffner, 2001] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. *International Journal of Robotics Research*, 20(5):378–400, May 2001.

[Likhachev and Ferguson, 2009] M. Likhachev and D. Ferguson. Planning long dynamically feasible maneuvers for autonomous vehicles. *International Journal of Robotics Research*, 28(8):993–945, 2009.

[Quigley *et al.*, 2009] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In *International Conference on Robotics and Automation*, 2009.

[Quinlan and Khatib, 1993] S. Quinlan and O. Khatib. Elastic bands: Connecting path planning and control. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 802–802. Citeseer, 1993.

[Shiller and Gwo, 1991] Zvi Shiller and Yu-Rwei Gwo. Dynamic motion planning of autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 7:241–249, 1991.

[Stentz, 1995] A. Stentz. The focussed D* algorithm for real-time replanning. In *International Joint Conference on Artificial Intelligence*, volume 14, pages 1652–1659. Citeseer, 1995.

[von Hundelshausen *et al.*, 2008] F. von Hundelshausen, M. Himmelsbach, F. Heck, A Mueller, and H-J Wuensche. Driving with tentacles: Integral structures for sensing and motion. *Journal of Field Robotics*, 25(9):640–673, September 2008.