Receding Horizon Model-Predictive Control for Mobile Robot Navigation of Intricate Paths

Thomas M. Howard, Colin J. Green, and Alonzo Kelly

Abstract As mobile robots venture into more complex environments, more arbitrary feasible state-space trajectories and paths are required to move safely and efficiently. The capacity to effectively navigate such paths in the face of disturbances and changes in mobility can mean the difference between mission failure and success. This paper describes a technique for model predictive control of a mobile robot that utilizes the structure of a regional motion plan to effectively search the local continuum for an improved solution. The contribution, the receding horizon model-predictive control algorithm, specifically addresses the problem of path following and obstacle avoidance through cusps, turn-in-place, and multi-point turn maneuvers in environments where terrain shape and vehicle mobility effects are non-negligible. The technique is formulated as an optimal controller that utilizes a model-predictive trajectory generator to determine parameterized control inputs that navigate general mobile robots safely through the environment. Experimental results are presented for a a six-wheeled skid-steered field robot in natural terrain.

1 Introduction

Mobile robot navigation is the challenge of selecting intelligent actions from the continuum of possible actions which move towards some goal while considering limited perceptual information, computational resources, and planning time. It also often viewed as the combined problem of path following and obstacle avoidance in system architectures. Regional motion planning is the problem of planning beyond the perceptive (sensor) horizon. As regional motion planning algorithms generate increasingly sophisticated paths in challenging environments, the drawbacks of modern techniques become evident.

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3980, USA, email: {thoward,cjgreen,alonzo}@ri.cmu.edu

1.1 Motivation

As navigators must maneuver through intricate motion plans composed of cusps, turn-in-place, and multi-turn maneuvers, the geometric singularities of these inflection points become problematic. Consider the situation shown in Figure 1. In this example, the mobile robot has been driven from the reference trajectory by disturbances which may include errors in modeling dynamics, terramechanical properties, and mobility.

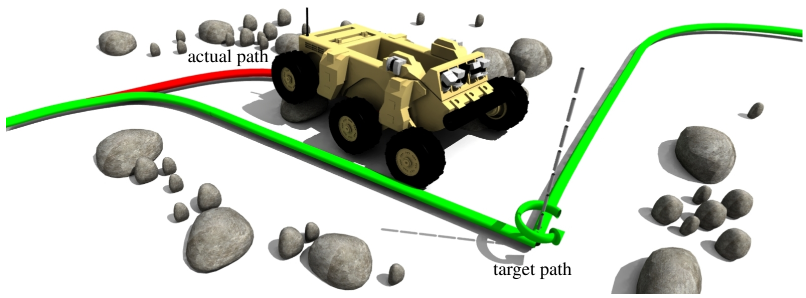


Fig. 1 A vehicle attempting to to follow the target path is subject to disturbances.

The popular class of "pursuit" algorithms [1] will round path corners, avoid cusps, and fail utterly for turn-in-place maaneuvers where the pursuit point becomes undefined. In context where such intricate maneuvers where generated by a path planner in order to avoid obstacles, a pursuit planner is inadequate. Sampling based obstacle avoidance techniques [6] sometimes fail for intricate path navigation because they cannot search the entire input or state space necessary to find the continuum solution.

A state-space trajectory is typically defined as a vector valued function of monotonic time (t). There are however circumstances where time is replaced by potenitally nonmonotonic functions of distance (s) or angle (ψ) to form a path. Path representations are used in order to achieve behaviors that allows velocity to remain unspecified. A "cusp" is a point in a trajectory where linear velocity changes sign. While cusps are discontinuous in path curvature, they are not discontinuous in state space trajectories and are perfectly feasible motions. Furthermore, while the concept of "forward on a path" is not well-defined for cusps (and likewise for point turns) whereas "forwards in time" always has meaning. The capacity of a state space trajectory representation to remove discontinuities and permit a forward horizon to be defined are the basis of our preference for this representation.

A reference trajectory is the state space trajectory ($\mathbf{x}(t)$)[1] provided by a regional motion planner (or other form of global guidance). The reference inputs ($\acute{\mathbf{u}}(t)$) are the controls which cause the vehicle to follow the path perfectly in the absence of disturbances. In the presence of disturbances, the reference control signals that correspond to a disturbance free trajectory must be augmented by corrective controls to null the following error over some time horizon.

For effective intricate path navigation, a technique is needed which can exploit the structure of the target path to search in the local continuum for actions which minimize path deviation and avoid obstacles. This is the process of parameteric relaxation, the technique of rendering a functional on a few parameters in order to permit relaxation of a trajectory (for optimization purposes) by searching a small number of degrees of freedom.

1.2 Related Work

There has been substantial research in the problem of developing effective, efficient mobile robot navigators. Early path following controllers operate on the assumption of tracking a single lookahead point [4] and have been greatly extended in the literature. In [12], effective search spaces for navigation in roads and trails were produced by generating nudges and swerves to the motion that reacquires the center of the lane.

An alterative approach involves sampling in the input space of the vehicle. In [6], navigation search spaces were generated by sampling in the input space of curvature. This approach also estimated the response of each action through a predictive motion model subject to the initial state constraints to more accurately predict the consequences of the actions. Egographs [8] represent a technique for generating expressive navigation search spaces offline by precomputing layered trajectories for a discrete set of initial states. Precomputed arcs and point turns comprised the control primitive sets that were used to autonomously drive Spirit and Opportunity during the Mars Exploration Rovers (MER) mission [2]. Trajectory selection was based on a convolution on a cost or goodness map. This approach was an extension of Morphin, an arc-planner variant where terrain shape was considered in the trajectory selection process [11]. Another closely related algorithm is the one presented in [3], where the an arc-based search space is evaluated based on considering risk and interest.

Other techniques such as rapidly-exploring random trees (RRTs) [7] have been effectively used to generate search spaces around the mobile robot to navigate cluttered, difficult environments and generate sophisticated maneuvers including u-turns. [9] presents a reactive path following controller for a unicycle type mobile robot built with a Deformable Virtual Zone to navigate paths without the need for global path replanning.

[1] The state ($\mathbf{x}$) contains the vehicle position, orientation, velocity, time, distance, or any other quantity of interest

1.3 Discriminators

The main contribution of this work is the development of a receding horizon model-predictive controller (RHMPC) or trajectory follower that effectively navigates intricate paths in complex environments. The algorithm leverages the capacity to generate the reerence controls for a given reference trajectory. This capability exists because the path is feasible and was generated by a trajectory generator that understands the association between inputs and the corresponding state-space trajectory. Our paticular preference is parameterized controls, but the key issue is that the controls are known, however represented, that correspond exactly to the reference trajectory. Field experiment results are shown that demonstrate that the proposed method can effectively navigate intricate paths.

2 Technical Approach

This section describes the issues related to navigation of intricate paths generated by regional motion planners, the methods by which parameterized controls are generated, and the trajectory optimization techniques used to generate corrective actions. The trajectory follower is formulated as an optimal control problem:

$$\begin{aligned} \text{minimize:} \quad & \mathbf{J}(\mathbf{x},\mathbf{u},t) \\ \text{subject to:} \quad & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u}) \\ & \mathbf{x}(t_I) = \mathbf{x_I} \\ & \mathbf{u}(\mathbf{x}) \in \mathbf{U}(\mathbf{x}), \quad t \in [t_I, t_F] \end{aligned} \tag{1}$$

The problem is one of determining actions from a set of functions ($\mathbf{U}(\mathbf{x})$) to represent the control inputs ($\mathbf{u}(\mathbf{x})$) which, when subject to the predictive motion model ($\mathbf{f}(\mathbf{x},\mathbf{u})$), minimize a penalty function ($\mathbf{J}(\mathbf{x},\mathbf{u},t)$). An additional requirement for the trajectory follower is that the resulting control must defined for a specific period of time or distance. This allows the optimized path to be evaluated for hazards to ensure vehicle safety.

2.1 Control Parameterization

One of the most difficult problems in motion planning involves reducing the continuum of actions to a manageable space to search. The trajectory following technique that we present uses the reference controls, which may be only piecewise continuous, as the initial guess. First, the target path is broken into the primitives used by the motion planner as shown in Figure 2a. For each action, there exists a set of controls that, when applied to the system, produce a path segment of a certain shape.

Each set of parameterized controls ($\mathbf{u}(\mathbf{p},\mathbf{x})$) have freedoms which are represented by the vector $\mathbf{p_i}$.

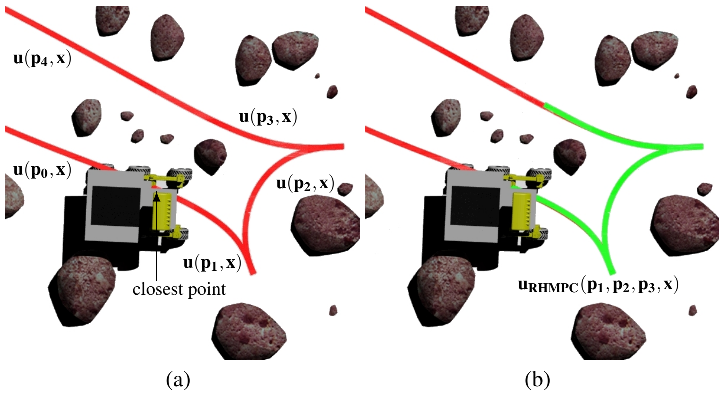


Fig. 2 The initial set of controls for the trajectory follower is determined by segmenting a portion of the regional motion plan at the point closest to the vehicle.

The initial guess for the parameterized control inputs ($\mathbf{u_{RHMPC}}$) is defined by the sequence of path segments between the closest point on the path and predefined fixed control horizon (Figure 2b). In this example, the free parameters of the receding horizon model-predictive controller ($\mathbf{p_{RHMPC}}$) are defined by a concatenation of free parameters in the control inputs:

$$\mathbf{p_{RHMPC}} = \left[\, \mathbf{p_1} \;\; \mathbf{p_2} \;\; \mathbf{p_3} \,\right]^T \tag{2}$$

2.2 Path Deviation Optimal Control

Once the parameterization for the control inputs have been found, it becomes important to modify the inputs to compensate for disturbances and errors in the motion model. This technique seeks to minimize a cost function ($\mathbf{J}$) by modifying a set of control inputs.

$$\mathbf{J}(\mathbf{x},\mathbf{u},t) = \Phi(\mathbf{x}(t_I),t_I,\mathbf{x}(t_F),t_F) + \int_{t_I}^{t_F} \mathscr{L}(\mathbf{x}(t),\mathbf{u}(\mathbf{p},\mathbf{x}),t)dt \tag{3}$$

The initial corrective action is evaluated through the predictive motion model subject to the initial state constraints to obtain an estimate of the cost function (Figure 3). If gradient of the cost function with respect to the parameterized control input freedom exceeds a threshold, the algorithm adjusts the control inputs to minimize the integrated penalty function ($\mathscr{L}$). The parameterized freedoms are modified iteratively through standard optimization techniques (gradient descent, Newton's method, etc.) because the gradient of the cost function is determined numerically:

$$\mathbf{p}_{\mathbf{RHMPC}\,i} = \mathbf{p}_{\mathbf{RHMPC}\,i-1} - \alpha \nabla \mathbf{J}(\mathbf{x}, \mathbf{u}), \quad i \geq 1 \tag{4}$$

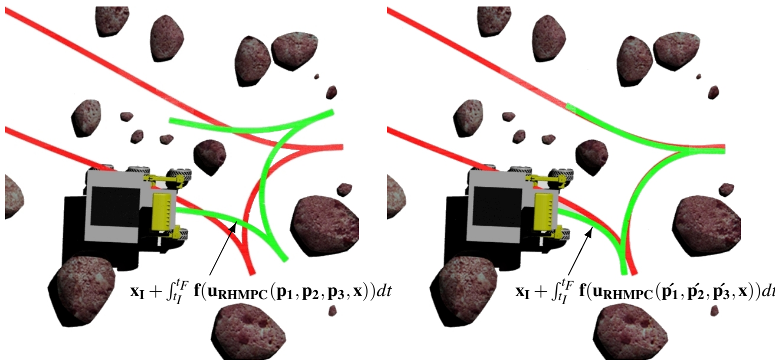


Fig. 3 A correction action is determined by optimizing the portion of the target path from the initial state of the vehicle.

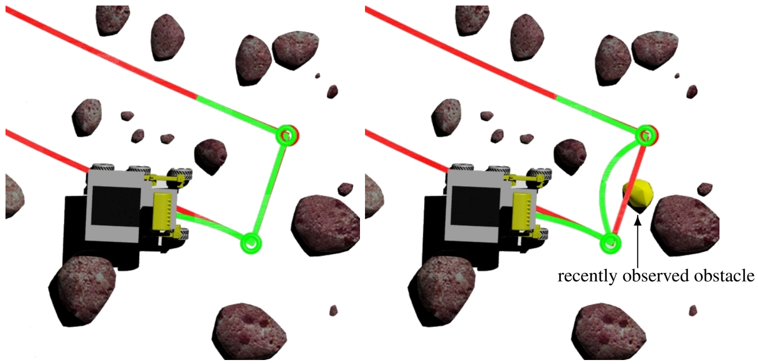
2.3 Integrating Observed Cost Information

There are several situations when a mobile robot should intentionally deviate from the target path. For one, static and dynamic obstacles can be observed before a regional motion planner has updated the path. Another reason could be the suboptimality of the target path itself, due to the discretized represention of the world in which the regional motion planner searches. One solution is to stop and request an refined or alternative plan. An alternative method is to simply deform the current action to provide a local obstacle avoidance maneuver.

The presented technique is naturally suited to deform the current action for local obstacle avoidance and path smoothing. The desired behaviors can be integrated by modifying the cost function to include a weighted penalty for obstacle cost:

$$\mathscr{L}(\mathbf{x}, \mathbf{u}, t) = w_1 \mathscr{L}_{deviation}(\mathbf{x}, \mathbf{u}, t) + w_2 \mathscr{L}_{cost}(\mathbf{x}, \mathbf{u}, t) \tag{5}$$

An illustration of this process is shown in Figure 4. In this example, new perception information places obstacles along the target path. Forward simulation of the corrective action shows that the vehicle will inevitably collide with these obstacles. Rather than compute an entirely new regional motion plan, an obstacle avoidance maneuver is generated by relaxing the path subject to this new weighted cost function.



(a) without observed obstacle (b) with observed obstacle

Fig. 4 Obstacle avoidance behaviors are integrated into the trajectory follower simply by modifying the utility function minimized in the trajectory relaxation.

3 Implementation

The regional motion planner used to generate feasible state-space trajectories for these experiments was a planner based on a graph connecting regularly arragned discrete nodes in state space [10]. This implementation of the motion planner operated on a 60m x 60m map centered on the vehicle. The lattice planner used a template consisting of forward, reverse, and turn-in-place trajectories with lengths varying from 3-9m each of which terminates on a discritized node in the state lattice. The lattice planner generates a target path by running A* on a graph with the set of edges which emanates from all nodes in the graph. The resulting target path is a series of sequential trajectories each of which is independently parameterized. Because the template contains forward, reverse, and turn-in-place actions, the resulting target path can have multiple cusps as well as turn-in-place actions embedded in it. These target paths were generated by the lattice planner at 2 Hz.

The model-predictive trajectory generator [5] was used in both the lattice planner template generation and the path deviation optimal control. Actions in the motion template were either composed of second-order spline curvature functions parameterized by distance or constant angular velocity functions parameterized by heading. Generic spline classes defined by sequences of individual command profiles were used to optimize the shape of the trajectory. Each trajectory was estimated numerically by passing the actions through a predictive motion model designed for the

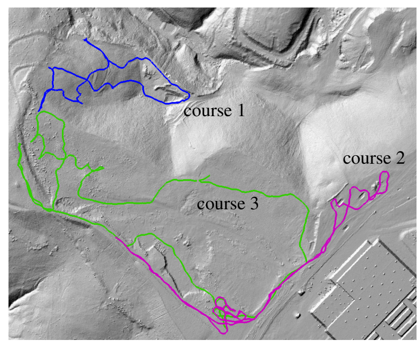
Crusher mobile robot. The corrective actions were generated by the receding horizon model-predictive controller at 20 Hz.

4 Experiments

A set of experiments were designed as a comparison between a navigator that used the presented trajectory follower and one that directly executed the regional motion plan. Both systems used the same version of a lattice planner that searches dynamically feasible actions which was specifically designed for the Crusher mobile robot's predictive motion model. Each field experiment was required to achieve a series of waypoints in an environment with updating perceptual information generated by an on-board perception system combined with limited overhead prior data.

The platform for the field experiments was Crusher (Figure 5(a)), a six-wheeled skid steered outdoor mobile robot. The multi-kilometer experiments were conducted at a test site in Pittsburgh, Pennsylvania with variable off-road terrain.

(a) Crusher, a six-wheeled skid-steered mobile robot



(b) The three courses at the Pittsburgh, Pennsylvania test site used for field experiments

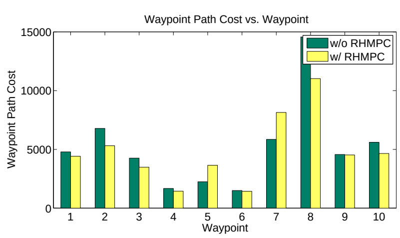
Fig. 5 The mobile robot and test environment for the trajectory follower field experiments.

The main metric used to measure success for the field experiments is integrated path cost, which is related to the risk, mobility, and traversability for a vehicle's configuration in the environment. While inherently unitless and scaleless, it provides the best metric for measuring performance because both the motion planner and the trajectory follower optimize this quantity.

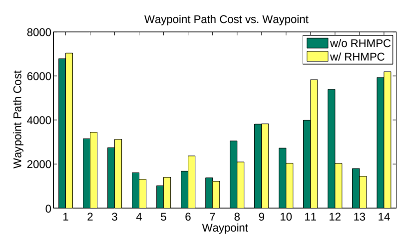
5 Results

This section presents the results of the three field experiments comparing the performance of the presented trajectory follower to a system directly executing the regional motion plan. Figures 6 show the integrated cost of each systems between each waypoint. It is useful to look at each waypoint-waypoint segment separately because each one can be considered to be an independent trial.

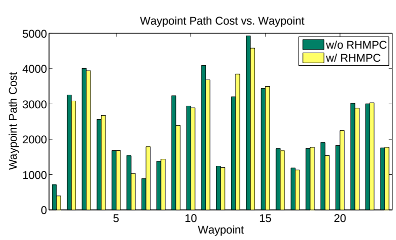
On average, the system using the trajectory follower slightly outperformed or achieved a similar level of performance of the alternative system. For portions of the course where disturbances relative to the predicted motion are uncommon or the local cost gradient was small near the disturbances very little improvement would be expected, with more improvement expected in cases where small system disturbances can quickly lead to significantly different path cost. Figure 6(d) shows the total integrated cost for each of the three field experiments. It is important to note that while the trajectory follower did not outperform the alternative system between every waypoint, it did improve the overall performance of the system by up to 7.2%. The variability in the results is expected because of the chaotic nature of outdoor mobile robots were any number of small changes can cause the robot to select a significantly different path.



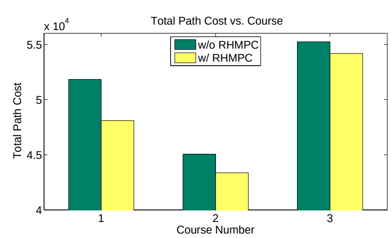
(a) Course 1 waypoint path cost



(b) Course 2 waypoint path cost



(c) Course 3 waypoint path cost



(d) Total path cost

Fig. 6 The waypoint and total path cost for a series of comparison runs on three courses

6 Conclusions

The receding horizon model-predictive control algorithm enables mobile robots to navigate intricate paths by utilizing the paths by relaxing parameterized controls that correspond exactly to the path shape. This technique enables several important behaviors, such as the ability to look through inflection points and turn-in-place actions in paths to better reason about the recovery trajectory. This technique makes it possible to intelligent search the local continuum for an action which minimizes path following error and/or avoids obstacles. It enables several important behaviors including the capacity to define a utility function in situations where pursuit planners fail and the ability to correctly follow path disctoninuities like cusps which are otherwise feasible motions. It also provides the capacity to generate feasible corrective actions for following error which are geometrically similar to the original path including discontinuities which are locally optimal in the function continuum of controls. Several multi-kilometer field experiments have shown that inclusion of the presented trajectory follower as a mobile robot navigator improves upon the metric that the regional motion planner attempts to optimize by searching in the local continuum.

7 Acknowledgments

This research was conducted at the Robotics Institute of Carnegie Mellon University under contract to NASA/JPL as part of the Mars Technology Program. Field experiments on the Crusher mobile robot platform were sponsored by the Defense Advanced Research Project Agency (DARPA) under contract Unmanned Ground Combat Vehicle PerceptOR Integration (contract number MDA972-01-9-0005). The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

1. O. Amidi. Integrated mobile robot control. Technical Report CMU-RI-TR-90-17, Carnegie Mellon, 1990.
2. J.J. Besiadecki, P.C. Leger, and M.W. Maimone. Tradeoffs between directed and autonomous driving on the mars exploration rovers. *International Journal of Robotics Research*, 26(91):91–104, January 2007.
3. D. Bonnafous, Sm. Lacroix, and Thierry Siméon. Motion generation for a rover on rough terrains. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 784–789, Maui and Hawaii and USA, October 2001.
4. R.C. Coulter. Implementation of the pure pursuit path tracking algortihm. Technical Report CMU-RI-TR-92-01, Carnegie Mellon, 1992.

5. T.M. Howard and A. Kelly. Optimal rough terrain trajectory generation for wheeled mobile robots. *International Journal of Robotics Research*, 26(2):141–166, February 2007.
6. A. Kelly and T. Stentz. Rough terrain autonomous mobility - part 2: An active vision and predictive control approach. *Autonomous Robots*, 5:163–198, 1998.
7. Y. Kuwata, A. Fiore, J. Teo, E. Frazzoli, and J.P. How. Motion planning for urban driving using rrt. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1681–1686, September 2008.
8. A. Lacze, Y. Moscovitz, N. DeClaris, and K. Murphy. Path planning for autonomous vehicles driving over rough terrain. In *Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference*, pages 50–55, Gaithersburg and MD and USA, September 1998.
9. L. Lapierre, R. Zapata, and P. Lepinay. Combined path-following and obstacle avoidance control of a wheeled robot. *International Journal of Robotics Research*, 26(4):361–375, April 2007.
10. M. Pivtoraiko, R. Knepper, and A. Kelly. Optimal, smooth, nonholonomic mobile robot motion planning in state lattices. Technical Report CMU-RI-TR-07-15, Carnegie Mellon, 2007.
11. R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadda, S. Koenig, G. Krishnaswamy, Y. Shinoda, W.L. Whittaker, and P. Klarer. Experience with rover navigation for lunar-like terrains. In *Proceedings of the 1995 IEEE Conference on Intelligent Robots and Systems*, volume 1, pages 441–446, August 1995.
12. S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Goffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, and P. Stang. Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.