Constrained Optimization Path Following of Wheeled Robots in Natural Terrain

Thomas M. Howard, Ross A. Knepper, and Alonzo Kelly

The Robotics Institute,
Carnegie Mellon University,
5000 Forbes Avenue, Pittsburgh, PA 15213, USA
{thoward,rak,alonzo}@ri.cmu.edu

Abstract. A smooth-primitive constrained-optimization-based path-tracking algorithm for mobile robots that compensates for rough terrain, predictable vehicle dynamics, and vehicle mobility constraints has been developed, implemented, and tested on the DARPA LAGR platform. Traditional methods for the geometric path following control problem involve trying to meet position constraints at fixed or velocity dependent look-ahead distances using arcs. We have reformulated the problem as an optimal control problem, using a trajectory generator that can meet arbitrary boundary state constraints. The goal state along the target path is determined dynamically by minimizing a utility function based on corrective trajectory feasibility and cross-track error. A set of field tests compared the proposed method to an implementation of the pure pursuit algorithm and showed that the smooth corrective trajectory constrained optimization approach exhibited higher performance than pure pursuit by achieving rough four times lower average cross-track error and two times lower heading error.

1 Introduction

When autonomous vehicles move in complex outdoor environments, precision motion control becomes both more difficult and necessary. Modeling errors increase sufficiently in magnitude relative to those in structured indoor environments that an approach based on improving any amenable aspect of mobility models seems warranted. We present a model predictive, optimal control approach to trajectory following which relies on a capacity to model many aspects of rough terrain vehicle mobility.

1.1 Motivation

Future robotics missions of planetary exploration will require precision motion control for geologic experiments, instrument placement, and infrastructure construction for permanent colonies. Traditional methods of path following can fail in challenging outdoor environments because they rely on simple models of vehicle motion and do not fully consider the effects of rough terrain, dynamics, or mobility constraints. In most modern applications, mobile robot path trackers

O. Khatib, V. Kumar, and D. Rus (Eds.) Experimental Robotics, STAR 39, pp. 343–352, 2008.
springerlink.com

merely react to and try to compensate for disturbances, whereas human drivers typically can predict and adjust their controls to prevent path following errors. Path trackers that only react to disturbances have difficulty performing precision instrument placement problems, unless the mobile robot exhibits extremely high mobility to correct for the terminal state error. Precision control is also more necessary in complex environments due to increased prevalence and lethality of hazards. It is more challenging because perception technology presently falls short of adequate prediction of terrain material (traction and compressibility) properties that truly do determine the magnitudes of the external forces that in turn determine how the vehicle responds to its control inputs. Vehicle actuator and body dynamics can also play a significant role in mapping those forces onto vehicle motion. On the other hand, these effects are not entirely unknown. Even without detailed knowledge of terrain shear strength and friction, it is possible compute estimates of wheel interactions that are better than no model at all. Steering dynamics and terrain following are also highly predictable in many cases. To the degree that any controller can predict the mean behavior of these pseudo-random processes, it can reduce the magnitude of the model disturbances and improve path following performance.

1.2 Related Work

Geometric path tracker algorithms for outdoor mobile robots have been part of robotic architectures since the very beginning. Pure pursuit [1, 2] remains one of the simplest and most often applied algorithms for solving the geometric path-tracking problem today, although many variations of the algorithm exist. In [4], the path follower calculates a control based on a combined pure pursuit / PI controller. Recent work applied to the Rocky series rover platforms at JPL in rough terrain incorporates into the controller the effects of the observed slip rate on the heading [3]. A feed forward approach to minimize total path following error was presented in [9], which increased the look-ahead distance in proportion to the heading error and incorporated a dynamic vehicle simulator. Early work in mobile robot control addressing the use of higher-order primitives to meet position, heading, and continuity constraints is discussed in [6, 7]. [6] addressed the need to bound velocity and acceleration on the corrective controls to meet vehicle limitations. A robust control tracking method for differentially steered mobile robots was discussed and simulated in [11], where a simplified vehicle dynamics model was used in the control loop to improve tracking performance. All of these methods have advantages and disadvantages. The choice of the look-ahead distance in the pure pursuit algorithm can lead to large tracking errors (look-ahead distance too large) or instability (look-ahead distance too small) if improperly tuned. Typically, path trackers modify the look-ahead distance by scaling it proportionally with velocity [10]. The feed forward approach can improve the stability of the algorithm, but requires a simulator and only searches a subset of feasible motions. The PID controller is more difficult to tune and incurs large heading error when the target path changes abruptly, but is stable. Our approach differs from the prior art in several ways. First, we have leveraged recent work in model-predictive real-time trajectory generation to determine smoother corrective trajectories. By using these more general methods for trajectory generation, we can incorporate predictive models of propulsion, suspension, and dynamics into the controls. Second, we have developed and implemented a constrained-optimization approach for on-the-fly computation of an optimal corrective trajectory.

1.3 Problem Statement

For a robot with perfect perception, modeling, and control, a path follower would not be required. In reality however, path followers for mobile robots are required to account for unmodeled vehicle dynamics (e.g. wheel slip). We seek to develop a general path follower that can account for predictable errors due to rough terrain and observed vehicle dynamics, and which can automatically select the optimal corrective trajectory from the continuum.

2 Technical Approach

Recent work in continuous primitive trajectory generation for arbitrary vehicle models [5] has improved the capacity to generate corrective paths that meet general position, heading, and curvature constraints in rough terrain. The algorithm gains its generality by relying on numerically linearizing and inverting forward models of propulsion, suspension, and motion. This approach can accommodate such effects as rough terrain, actuator dynamics, wheel slip, and any other somewhat predictable effects of interest. It can also accommodate boundary and internal constraints while optimizing an objective function. Such a function might for example involve such criteria as obstacle avoidance, cost, risk, time, or energy consumption in any combination. The proposed path tracking method relies on the continuous primitive trajectory generator to plan corrective trajectories back onto the path. This is the similar to the pure pursuit method, except that higher order boundary state constraints have been imposed and a more expressive corrective trajectory is required. Pure pursuit requires position constraints, while the proposed method can meet arbitrary posture constraints (Figure 1).

One problem with current tracking algorithms is the choice of the look-ahead distance. When the look-ahead distance is too large, a path follower will tend to cut corners, while a short look-ahead distance often results in instability. In effect, the choice of target range, whether performed a priori or online, is inherently an optimization problem that searches over possible target ranges in order to minimize cross track error while providing smooth, stable control. During each cycle, the proposed path follower determines corrective trajectories to a set of goal states along the target path. The optimum that is selected by minimizing a utility function that penalizes cross track error and high curvatures. Formally, this is a constrained optimization problem, where the free variable is look-ahead distance, the constraints require that the terminal state lie on the target path,

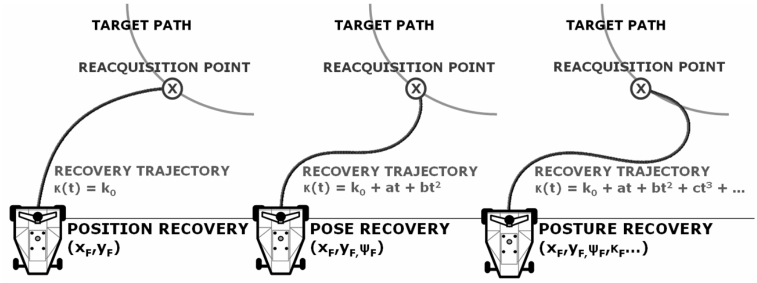

Fig. 1. Corrective Trajectory Continuity. This figure demonstrates several different trajectory generation boundary conditions that lead to increasingly higher levels of continuity for corrective trajectories. The corrective trajectory becomes increasingly complicated at the benefit of higher terminal state continuity of the target path.

and we look to find the optimal trajectory based on minimizing some cost (J) along the path:

$$J[x(t), u(t)] = \phi[x(t_f), t_f)] + \int_{t_0}^{t_f} Y[x(t), u(t), t]dt \tag{1}$$

Typically the integrand (Y) will penalize cross-track error, high angular velocity, and/or heading error while the state penalty function (ϕ) will penalize the amount of time required to complete the corrective trajectory. Two methods for solving for a nearly optimal corrective trajectory are discrete search and local gradient methods. A discrete search method will explicitly evaluate a series of corrective trajectories spanning the parameter space and will select the one with the lowest cost. Gradient methods will perform local minimization by estimating the rate of change of the cost function with respect to the free variables in the system. The gradient method provides the capability to find a local optimal corrective trajectory but can miss the global optimum if several minima exist along the cost function and the initial guess is far from the global optimum. Discrete search methods will find a globally optimal solution, but only to the extent that the search space is discretized. Figure 2 shows an example of corrective trajectory discrete search for a single free parameter (look-ahead distance).

Our path following algorithm relies on two nested control loops, a planning loop and an execution loop. The planning loop determines the proper control based on the current state and a set of goal states along the path (Figure 2) while the execution loop commands the desired linear and angular velocities according to the determined curvature and linear velocity functions.

3 Experiments and Experimental Results

We have conducted a set of experiments with our path-tracking algorithm on the differential-drive DARPA LAGR (Learning Applied to Ground Robotics) mobile robot platform. The constrained optimization path-tracker performance has been

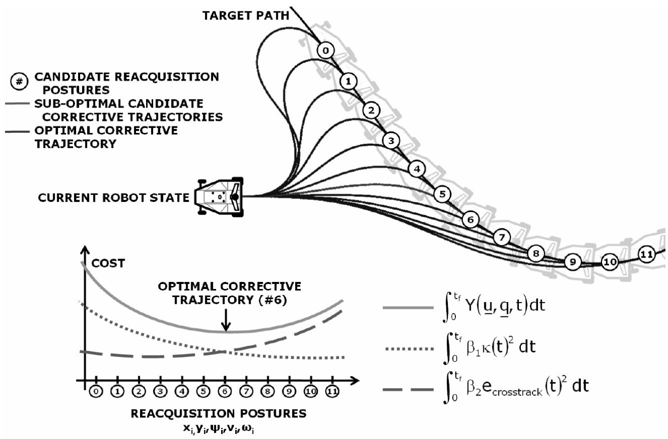

Fig. 2. Constrained Optimization Formulation Corrective Trajectory Determination. The look-ahead distance problem simplifies to a constrained optimization problem. The trajectory generator produces a search space consisting of smooth corrective trajectories at waypoints along the geometric path. In general, we want to minimize a utility function (J) over the search space. In this example, we look to minimize the integral of the weighted sum of squared curvature (smoothness) and the cross track (following) error along each corrective trajectory. After searching the span of possible solutions, the algorithm selects an optimal corrective trajectory that minimizes a weighted balance between curvature and cross track error.

compared to a baseline pure pursuit path tracker. The experiments developed for this test follow previous work [8] in evaluating path tracker performance by looking at paths with discontinuous heading, constant curvature arcs, and discontinuous curvature (Figure 3). We performed the tests with a fixed target speed of 0.5 m/sec along the path since many pure pursuit implementations now scale look-ahead with velocity. The cost function for the constrained optimization path follower was a weighted sum of the squared curvature along the path (favoring smooth paths) and the time it would take to complete the corrective trajectory (favoring short paths). The tests were conducted in a grass field outdoors where the terrain is uniformly bumpy and where wheel slip is generally unpredictable. This locale allows us to measure the resilience of these methods to unknown vehicle dynamics. Sections 3.1 through 3.3 will exhibit comparisons of the most significant runs of each test.

3.1 Discontinuous Heading Tests (Square) Experimental Results

The discontinuous heading test was designed to see how each tracking algorithm handled sharp angles in the pre-computed path. After tuning the performance of the pure pursuit path tracker, we measured roughly equal performance on the 3.0 meter square between the two methods (Figure 4). Some crosstrack and

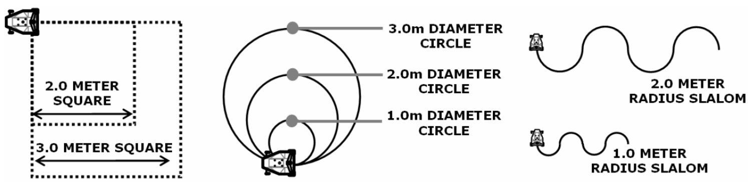

Fig. 3. Path Tracking Experiments. Our experiments involve testing the path tracker's ability to follow a discontinuous heading (square) path, a constant curvature arc (circle) path, and a slalom (discontinuous curvature) path.

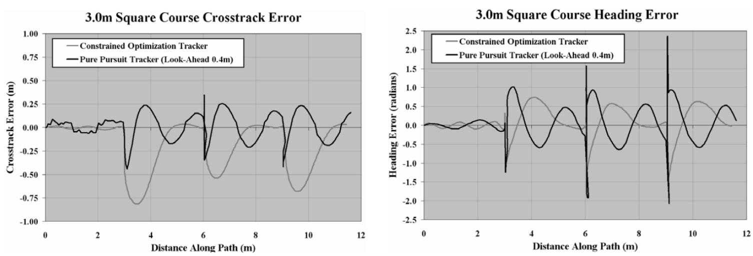

Fig. 4. 3.0m Square Path Tracking Experiments. These plots show the cross-track and heading errors measured for the pure pursuit and constrained optimization path tracking algorithms. Although the pure pursuit algorithm exhibited lower overall cross-track error, it had to stop and correct large heading errors three times along the path.

heading error is expected as it is dynamically infeasible for this vehicle to follow a square path at a constant speed.

3.2 Constant Curvature Arc (Circle) Experimental Results

The constant curvature (circle) tests were designed to see if each tracker was stable and/or subject to a constant cross-track error. Figure 5 demonstrates that the pure pursuit method was subject to constant offset errors whereas the constrained optimization path follower knew to turn harder and subsequently reacquired the target path.

3.3 Discontinuous Curvature (Slalom) Experimental Results

The discontinuous curvature (slalom) tests were designed to see how each tracking algorithm handled sudden changes in path curvature. Figure 6 shows that the constrained optimization path follower outperforms the tuned pure pursuit method. The performance gain comes from compensating for the heading and curvature state constraints in the planned trajectories that reacquire the target path.

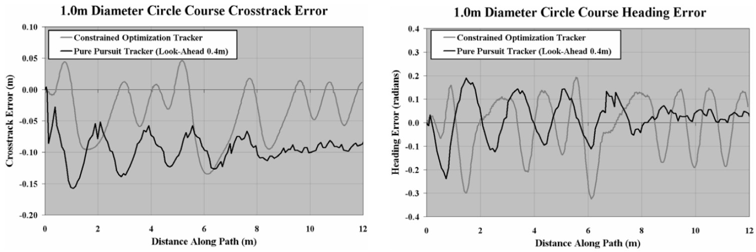

Fig. 5. 1.0m Diameter Circle Path Tracking Experiments. These plots show the cross-track and heading errors measured for the pure pursuit and constrained optimization path tracking algorithms on the 1.0m diameter circle test. The pure pursuit algorithm was consistently off by 10 cm.

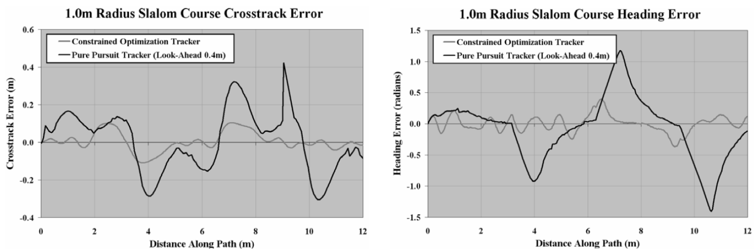

Fig. 6. 1.0m Radius Slalom Path Tracking Experiments. These plots show the cross-track and heading errors measured for the pure pursuit and constrained optimization path tracking algorithms on the 1.0m radius slalom test. The pure pursuit algorithm was not able to handle the curvature discontinuity effectively.

3.4 Additional Pure Pursuit Experimental Results

The performance of the pure pursuit path-tracking algorithm was measured using a variety of look-ahead distances tuned for the square (0.4m), circle (0.4m), and slalom (0.7m) tests (Figure 7) and compared against the performance of developed constrained optimization path follower. The plots show that properly tuned parameters for one test can exhibit poor path following performance on the others even at constant velocity whereas the constrained optimization path tracker used the same utility function to determine the optimal lookahead distance.

3.5 Dynamic Look-Ahead State Selection

The strength of the constrained optimization formulation of the path follower is that it dynamically selects the proper corrective trajectory based on evaluating the cost of a series of paths that reach different states along the path (Figure 8).

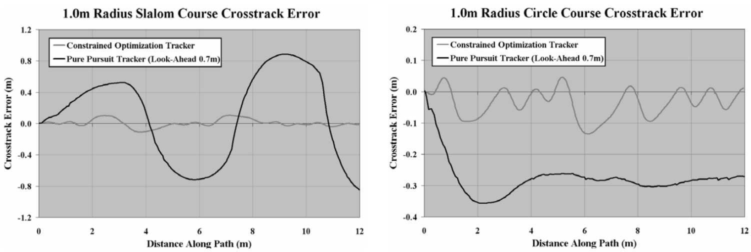

Fig. 7. Pure Pursuit Path Tracking Performance. These plots show how the tuned parameters for one test result in poor performance on the other tests. This demonstrates the strength of the constrained optimization approach, which takes into account the shape of the path when planning the corrective trajectory.

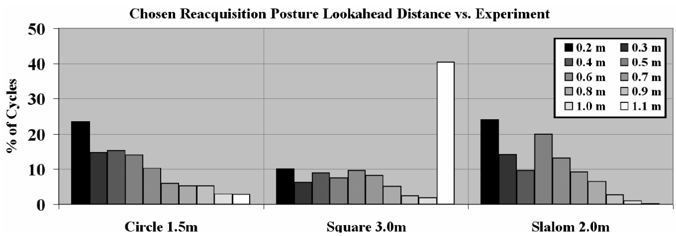

Fig. 8. Dynamic Look-Ahead Selection. The frequency is shown with which the constrained optimization path tracker selected each look-ahead distance for its terminal boundary state. In general, nearer targets are generally preferred. The high preference for the largest distance in the square test is caused by the lack of a cross-track error term in the utility function.

Figure 8 demonstrates that the dynamic look-ahead distance calculation is working in the path following experiments. In the circle and slalom courses where the path is generally smooth, the algorithm prefers shorter paths because they are generally smooth. In the square tests however, the discontinuity in heading causes very high curvature turns close in, so for the corners the algorithm selects longer, smoother corrective trajectories.

3.6 Experiment Summary

A dataset gathered consisting of 22,000 data points from 55 analyzed runs of the two algorithms showed that the constrained optimization path tracker average 9.47cm of cross-track error and 0.145 radians of heading error versus 39.14cm of cross-track error and 0.302 radians of heading error for the pure pursuit tests. The higher performance comes from dynamic look-ahead distance selection and satisfaction of additional state constraints (heading and curvature) when planning trajectories.

4 Conclusions and Future Work

In this paper, we have demonstrated the design and implementation of a control system based on a real-time trajectory generator that provides the predictive component in an optimal controller. The algorithm recomputes the look-ahead distance every cycle based on an explicit utility criterion and a model of how vehicle behavior changes with speed and terrain shape. This approach is not without limitations. Poor models of vehicle dynamics at high speeds can cause instability when executing these more complicated corrective maneuvers. Another drawback is the added complexity from computing trajectories in real-time. This approach is therefore suitable in applications where vehicle dynamics are reasonably predictable and the computational resources to generate trajectories online are available. Related future work will include unsupervised learning of the vehicle model to correct for unpredictable vehicle dynamics, constrained optimization of more than one variable (velocity, path tangents), utility functions that incorporate path costs for obstacle avoidance, and a fully unconstrained optimization approach to path following. The ability to search a set of trajectories related to some forward state along the path will allow for high-performance path tracking and obstacle avoidance to be achieved in a single algorithm.

Acknowledgement

This research was conducted at the Robotics Institute of Carnegie Mellon University under contract to NASA/JPL as part of the Mars Technology Program.

References

1. Amidi, O., Thorpe, C.: Integrated Mobile Robot Control. Technical Report CMU-RI-TR-90-17. Robotics Institute, Carnegie Mellon University (May 1990)
2. Coulter, R.: Implementation of the Pure Pursuit Path Tracking Algorithm. Technical Report CMU-RI-TR-92-01. Robotics Institute, Carnegie Mellon University (May 1992)
3. Hemlick, D.M., et al.: Path Following using Visual Odometry for a Mars Rover in High-Slip Environments. In: Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, USA (March 2004)
4. Hogg, R.W., et al.: Algorithms and Sensors for Small Robot Path Following. In: Proceedings of the IEEE International Conference on Robotics and Automation, Washington, D.C., USA (May 2002)
5. Howard, T.M., Kelly, A.: Trajectory Generation on Rough Terrain Considering Actuator Dynamics. In: Proceedings of the 5th International Conference on Field and Service Robotics 2005, Port Douglas, Australia (July 2005)
6. Kanayama, Y., et al.: A Stable Path Tracking Method for an Autonomous Mobile Robot. In: Proceedings of the IEEE International Conference on Robotics and Automation 1990, Cincinnati, Ohio (May 1990)
7. Nelson, W.: Continuous Steering-Function Control of Robot Carts. Proceedings of the IEEE Transactions on Industrial Electronics 36(3), 330–337 (1989)

8. Roth, S.A., Batavia, P.H.: Evaluating Path Tracker Performance for Outdoor Mobile Robots. In: Proceedings of Automation Technology for Off-Road Equipment 2002, Chicago, IL, USA (July 2002)
9. Singh, S., et al.: A System for Fast Navigation of Autonomous Vehicles. Technical Report CMU-RI-TR-91-20. Robotics Institute, Carnegie Mellon University (September 1991)
10. Urmson, C., et al.: High Speed Navigation of Unrehearsed Terrain: Red Team Technology for Grand Challenge 2004. Technical Report CMU-RI-TR-04-37. Robotics Institute, Carnegie Mellon University (June 2004)
11. Zhang, Y., et al.: Dynamic Model Based Robust Tracking Control of a Differentially Steered Wheeled Mobile Robot. In: Proceedings of the American Control Conference, Philadelphia, PA, USA (June 1998)