Terrain Aware Inversion of Predictive Models for Planetary Rovers

Alonzo Kelly, Tom Howard

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA, USA

alonzo@cmu.edu, thoward@cs.cmu.edu

***Abstract* - Planetary rovers will continue to evolve in the direction of enhanced mobility and more challenging terrain. Once the mechatronics are in place, it becomes the job of software to exploit the mobility of the platform to the maximum degree possible while minimizing exposure to risks. Local motion planners for outdoor offroad terrain typically use relatively high fidelity models of vehicle motion to correctly predict the consequences of candidate actions. More correct predictions leads to more intelligent behavior, more effective science, and reduced mobility risk. Our formulation is based on an efficient inversion of the equations of motion to compute the precise controls necessary to achieve a desired position and orientation while following the contours of the terrain under arbitrary wheel terrain interactions. All higher level rover behaviors can benefit from such precision, terrain aware controls. Applications to instrument placement, wheel slip compensation, obstacle avoidance, and regional mobility planning will be presented.**

Index Terms* – *motion planning, planetary rover, mobile robot, trajectory generation

I. INTRODUCTION

Effective autonomous mobility in difficult terrain depends on a relatively high fidelity capacity to predict the consequences of candidate actions. Of course, such predictions depend on models of vehicle dynamics and adequate predictions of propulsive and steering forces generated by the terrain. While the associated terrain mechanical descriptors may not be entirely predictable, they can be predicted somewhat and, one way or another, every model of vehicle motion makes assumptions about how the terrain can be induced to propel the vehicle.

Given a capacity to predict the consequences of actions, the job of local motion planning becomes one of choosing one from a continuum of possible actions on a regular basis. This paper will introduce an approach to choosing actions which is based on a new capacity to invert the model of system dynamics in real time.

A. Vehicle Modeling

A somewhat general description of vehicle dynamics is a nonlinear vector differential equation of the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t) \tag{1}$$

The model may be kinematically driven with velocity inputs or dynamically driven with forces, as the case requires. Let the state vector $\mathbf{x}$ include at least the position (x,y,z) and orientation (ϕ,θ,ψ) of the vehicle expressed relative to a frame of reference fixed to the ground.

Very often, the propulsive forces generated by the vehicle are limited in magnitude and direction for several reasons including terrain limitations. Nonholonomic constraints (expressing the incapacity of wheels to move sideways) do not entirely apply under conditions of wheel slip but in either case, wheeled vehicles remain underactuated whether the wheels slip or not. Typically, the two degrees of freedom of linear velocity in the forward direction and angular velocity in the local tangent plane are actuated indirectly by adjusting wheel/track speeds, steer angles etc. This means one rate degree of freedom out of the remaining three $(\dot{x}, \dot{y}, \dot{\psi})$ is lost.

The impact of these facts on planning and control is that even the simplest useful models of mobility are underactuated differential equations. Furthermore, terrain shape must be known to predict motion because steering takes place in the instantaneous terrain tangent plane. Intuitively, the same steering signal executed on flat and then on nonflat terrain will drive a vehicle to two different places.

B. Problem Statement

In many contexts, the pose of the vehicle (x, y, ψ), not its rate, is the variable of interest and this problem of precision pose control is the one we address here. The task of driving a fork truck to pick up a pallet (Figure 1) illustrates the problem well.

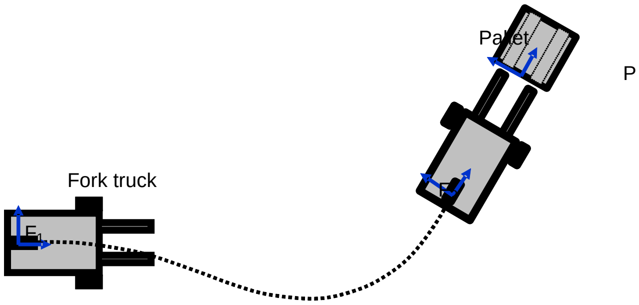

Fig. 1. In order to pick up a pallet, a fork truck must achieve a fairly precise target posture characterized by constraints on terminal position, heading, and curvature.

Vehicle mobility constraints assert that the pallet cannot be approached by moving sideways. Therefore, consider solving the problem in reverse by imagining a vehicle at the terminal state which is moving backward toward the start. Consider turn radius and its rate to be limited. It becomes intuitively clear that the initial maneuver in the forward direction must be a turn to the right although the pallet is initially to the left of the fork truck. If the fork truck simply drives left toward the pallet, it will not achieve the right heading. This difficulty arises in part because of underactuation. It is not possible to drive toward the goal and then rotate to the correct heading at the last minute because heading cannot be changed quickly or independently from position.

The task of controlling vehicle pose is that of inverting the system dynamics differential equation to produce the controls which will achieve a goal terminal state. Since the terminal state is an integral of the dynamics, the problem is to select a function $u(t)$ which, when placed inside this dynamics integral, will produce the desired terminal state:

$$\mathbf{x}_f = \int_0^{t_f} \mathbf{f}(\mathbf{x}, \mathbf{u}, t)\, dt \qquad (2)$$

There will usually be an entire continuum of functions $u(t)$ which could solve the problem but only one is needed. Intuitively, a slight change to the beginning of a solution can be compensated by another later while still achieving the desired terminal state. A human driver somehow solves this problem with little effort but techniques to solve it computationally are far from immediately obvious.

C. *Prior Work*

Trajectory generation has ancient roots in the theory of differential equations, the calculus of variations, and Lagrange's variational approaches to dynamics. From basic mathematics, techniques are available to both optimize an objective functional over an unknown curve and to require that boundary conditions on the solution be satisfied. Some of the earliest work related to trajectory generation of direct relevance to robotics is work on curves of minimal length under constraints on curvature [1]. Kanayama's original work on clothoids introduced the idea of using continuous piecewise linear curvature curves for robot trajectory generation [2]. Some work has concentrated on the problem of producing sequences of simple primitives which are optimal overall [3].

In [4], a holonomic geometric path is found in an obstacle field and path segments are smoothed using optimal control. A near real-time optimal control trajectory generator is presented in [5], which solves eleven first-order differential equations subject to the state constraints. Work on the problem of trajectory generation in arbitrary terrain is rare. In [6] a planner is presented which initially assumes flat terrain but then trims the results to accommodate rough terrain.

Our own work on this topic in recent years is summarized in [7][8]. Our most recent approach, applicable to this paper, is characterized by the use of numerical methods for all aspects of the algorithm – including dynamics integration, linearization, and inversion. We express an unknown control input in terms of unknown parameters. In doing so, we convert an original optimal control problem into one of nonlinear programming and solve for the unknown parameters using established techniques.

II. IMPLEMENTATION

This section presents the overall the software implementation.

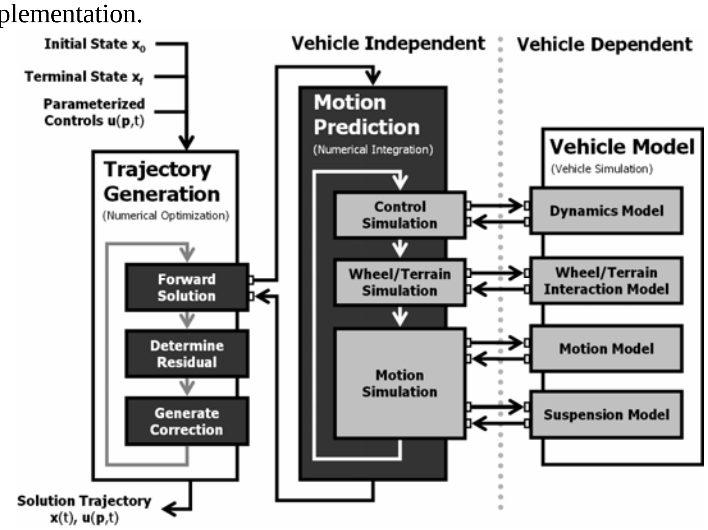

Fig.2. The system architecture distinguishes three main elements, only one of which is vehicle dependent. Each represents a computational loop where the innermost loop is on the right side.

A. *Architecture*

A three level, three loop architecture is used as illustrated in Figure 2. The highest level loop (Trajectory Generation - left side) inverts the dynamics integral. The medium level loop (Motion Prediction – center) integrates the equations of motion. The lowest loop (Vehicle Model – right side) is the differential equation itself. These elements are distinguished in software for reasons of generality. While the vehicle model is clearly vehicle dependent, the solution for integration of the differential equation and the solution of the nonlinear parametric terminal state equations are entirely vehicle independent.

B. *Parameterized Controls*

A parameterized control is a mapping from a parameter vector (a few numbers) onto a continuous function. The fact that the parameters express a relatively small number of degrees of freedom is key to their capacity to reduce computation. Figure 3 illustrates a typical case: a trapezoidal linear velocity.

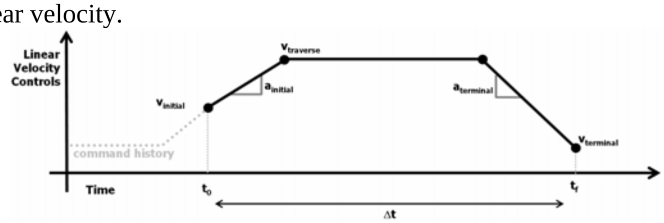

Fig.3. Two examples of parameterized controls. Any parameterization creates a mapping from a few numbers (the parameters) onto a continuous function.

The corresponding parameter vector consists of initial, final and traverse velocities, and the accelerations between them.

C. *Trajectory Generation*

The numerical method applied to find the parameter values is Newton's method. The terminal state in (2) is considered to be a nonlinear function of the parameters and an initial guess for the solution parameters is refined by repeated solution of a locally linear approximation. Since the partial derivatives of dynamics integral cannot generally be found analytically, estimates of the derivatives are computed using finite difference approximations.

D. *Motion Prediction*

Any method for integrating a differential equation is potentially applicable at this step. Although Runge-Kutta may be more efficient, we have been able to use Euler's method of integration so far to determine the change in vehicle state over a small time step (Δt):

$$\mathbf{x}(t+\Delta t) = \mathbf{x}(t) + \dot{\mathbf{x}}(\mathbf{x},\mathbf{u},t)\Delta t \qquad (3)$$

We enforce terrain contact constraints explicitly at each time step by computationally allowing the vehicle to settle on the terrain to achieve minimum error between wheel contact points and terrain elevations. This procedure also produces suspension deflections as a byproduct in quasi-static cases.

E. *Vehicle Model*

Our models in most cases are kinematically actuated – driven by velocities rather than forces. Such models are sufficiently accurate for our purposes but much faster to compute than doubly integrated, dynamically actuated models. Velocities requested by the autonomy layer are mapped onto desired wheel speeds. Effects such as wheel slip are incorporated at this stage by expressing their dependence on speed and slope to produce the response wheel speeds. These responses are mapped onto response velocities of the entire system.

F. *Runtime*

Runtime depends somewhat on terrain roughness. Over a large number of test cases, a factor of 3 separates the easy cases from the difficult ones. As illustrated in Figure 4, most of this is due to an increase in iterations.

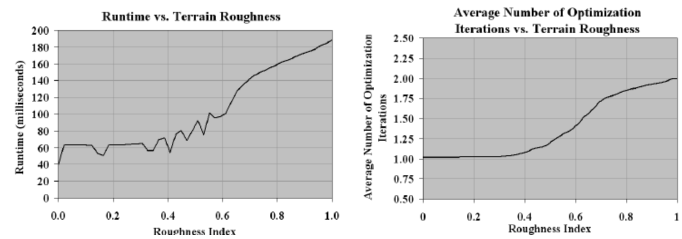

Fig.4. Algorithm Runtime. These results are for several hundred highly complicated test cases run on a 1.8 GHz laptop computer.

These results are for a highly complicated 12 actuator vehicle model including wheel slip and very rough terrain. Trajectory generation queries for simpler vehicle models on flat terrain are solved in under a millisecond.

III. APPLICATIONS

The algorithm has been refined on numerous mobile robot research programs over the last decade and it has recently been adapted to planetary rovers on the Mars Technology program. Some of the potential applications for planetary rovers are described below.

A. *Basic Pose Control and Instrument Placement*

The problem of placing a science instrument on a rock is equivalent to the forktruck problem described in Figure 1. The vehicle must compute in real-time a trajectory which achieves the correct terminal position, heading and curvature. With reference to Figure 1, the problem is to determine a feasible motion from frame F1 to frame F2 given a measurement of the relationship between frame F1 and frame P.

B. *Off Road Path Following*

One class of path following algorithms is based on the concept of a corrective trajectory terminating at a reacquisition point somewhere forward on the path. We have implemented an adaptive version of such an algorithm as illustrated in Figure 5. A search is conducted along the target path to determine the reacquisition trajectory which provides best tradeoff between the aggressiveness of the maneuver and the integral of crosstrack error. Each point of reacquisition has the correct heading and curvature and the optimal control formulation adjusts lookahead automatically as speed, path curvature, and crosstrack error vary over time.

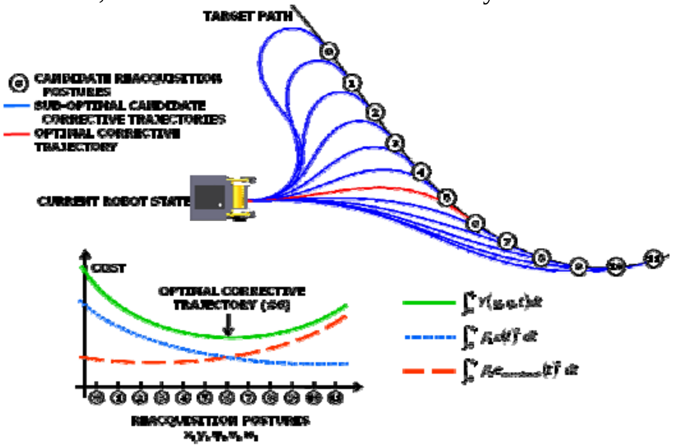

Fig.5. Adaptive Lookahead Path Following. A model predictive optimal controller chooses an acquisition point on the goal path which is the best tradeoff between aggressiveness and crosstrack error.

C. *Model Predictive Slip Compensation*

We have used our algorithm to develop mechanisms to predictively compensate for wheel slip. As Figure 6 shows, models of how wheel slip depends on slope can be inverted readily to cause the vehicle to approach a sloped goal on the high side in anticipation of sliding into it from above. Of course, such models will never be perfectly accurate but continuous compensation of this nature is likely to be far more effective than ignoring slip entirely.

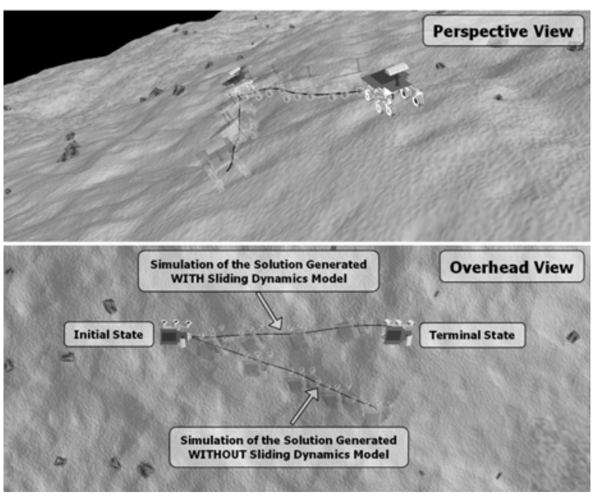

Fig.6. Predictive Slip Compensation. The vehicle model in this cases anticipates a sideslip velocity which is proportional to slope and forward velocity. The trajectory generator automatically compensates by approaching the goal from above.

D. *Control of Hyperactuated Systems*

An important goal of the trajectory generator is that of exploiting the full maneuverability of a vehicle with six steered and driven wheels. As shown in Figure 7, our algorithm can exploit the capacity of such a rover to drive sideways in order to orbit around a science target and deploy instruments most effectively.

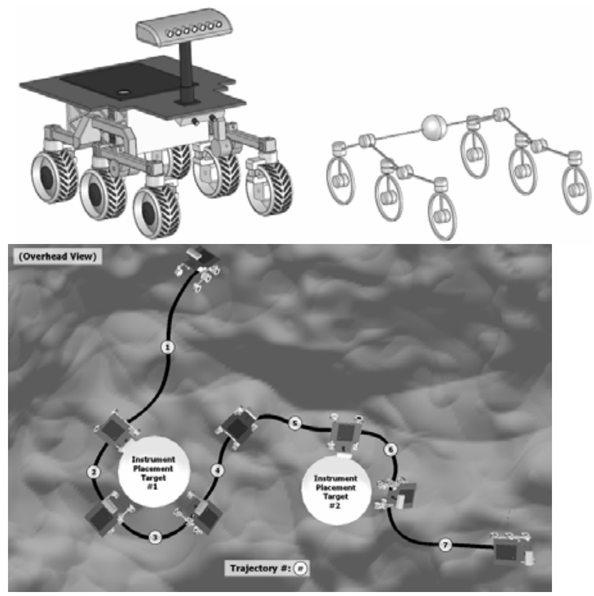

Fig.7. Hyperactuated System Control. The Rocky8 Mars rover prototype has six drive and steered wheels. Exploiting its capacity to move sideways leads to efficient instrument placement.

E. *Off-Road Hierarchical Control*

Motion planners used in off-road environments often follow a low fidelity global plan using a higher fidelity local planner which enforces dynamic feasibility constraints, avoids obstacles, and follows the global guidance. We have used the trajectory generator to sample uniformly in the workspace of the vehicle while terminating each candidate motion at the heading desired by global planning as illustrated in Figure 8.

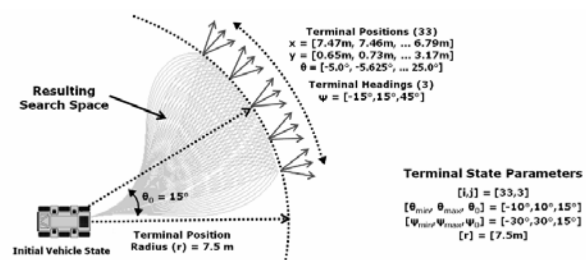

Fig.8. Motion Planning in Cluttered Offroad Environments. In this case, the search points at the ends of the trajectories derive their headings from the global plan.

F. Off-Road Global Search Space Design

The capacity to generate feasible motions to arbitrary reachable states makes it possible to construct search spaces for global planners which exhibit useful symmetries, encode only feasible motions, and are continuous where primitive motions join. The application of search algorithms to such a search space can produce complicated maneuvers like n-point turns automatically. Such a search space is illustrated in Figure 9.

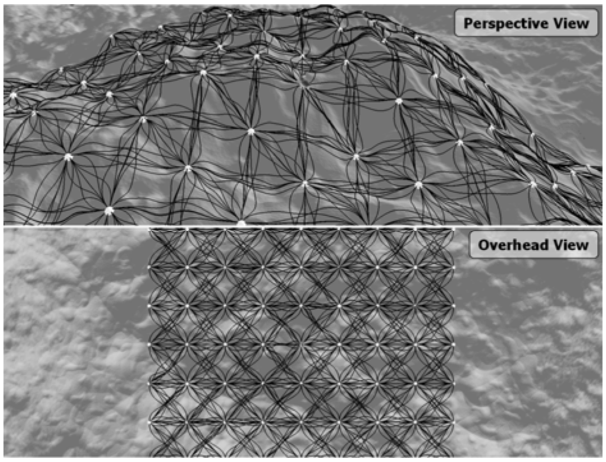

Fig.9. Feasible State Lattice for Global Planning. A regular lattice of states is laid out on terrain and the trajectory generator is used to connect them locally with feasible motions.

IV. Conclusion

Sampling and searching in the space of controls or actions is a well-established technique in motion planning for ensuring feasible local motion plans. It has been used almost universally in off road ground robotics for some time. The rationale for the use of this technique is that the resulting motions are at least dynamically feasible even if they do not quite take the robot where we would like it to go. A more compelling rationale is that better techniques – ones that produced feasible motions to precisely designated target states - did not exist. We have presented such a technique in this paper.

A capacity to efficiently produce a feasible motion to any reachable point in state space is enabling in several ways. It leads to robots that can interact with their environments in more purposeful and intelligent ways because getting to the precise somewhere where the job needs to be done becomes possible. It leads to robots whose precision understanding of their own mobility enables precision high speed maneuvering in challenging circumstances including those involving complicated terrain interactions, narrow corridors around a nominal path, and dense obstacle fields.

Acknowledgment

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] L.E. Dubins. On Curves of Minimal Length with a Constraint on Average Curvature and with Prescribed Initial and Terminal Positions and Tangents. American Journal of Mathematics, 79:497-516, 1957.

[2] Y. Kanayama and N. Miyake. Trajectory Generation for Mobile Robots. In Proc. of the Int. Symp. on Robotics Research, pages 16-23, Gouvieux, France, 1985.

[3] J.A. Reeds and L.A. Shepp. Optimal Paths for a Car that Goes Both Forward and Backward. Pacific Journal of Mathematics, 145(2):367-393, 1990.

[4] J.P. Laumond. Nonholonomic Motion Planning via Optimal Control. In the Proc. of the Workshop on Algorithmic Foundations of Robotics, pages 227-238, San Francisco, California, 1995.

[5] J. Reuter. Mobile Robot Trajectories with Continuously Differentiable Curvature: An Optimal Control Approach. in the Proc. of the IEEE/RSI Conference on Intelligent Robots and Systems, Victoria, British Columbia, October 1998.

[6] M. Cherif. Motion Planning for All-Terrain Vehicles: A Physical Modeling Approach for Coping with Dynamic and Contact Interaction Constraints. In the IEEE Transaction on Robotics and Automation, vol. 15, no. 2, pages 202-218, April 1999.

[7] Howard, T., Kelly, A., "Optimal Rough Terrain Trajectory Generation for Wheeled Mobile Robots", to appear The International Journal of Robotics Research, Vol. 26, No. 2, February 2007, pp. 141-166.

[8] Kelly, A., Nagy, B. "Reactive Nonholonomic Trajectory Generation via Parametric Optimal Control", The International Journal of Robotics Research. 2003; 22: 583-601.