PLANNING ROBUST DYNAMIC TRANSITIONS FOR ENHANCED MOBILITY OF PLANETARY ROVERS

Timothy C. Bretl * and **Stephen Rock** *

* *Aerospace Robotics Lab, Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305*

Abstract: This paper considers the control of planetary rovers that use mobility strategies incorporating *aggressive maneuvers* such as jumping or hopping. Robust execution of an aggressive maneuver is difficult since success depends critically on the state of the system as it begins the maneuver. This paper presents an efficient algorithm for planning a robust maneuver based on the consideration of noise in the time at which the maneuver begins. A complete implementation of this algorithm is shown for a specific system in simulation.

Keywords: Robotics, Motion Planning, Robust Control, Hybrid Modes, Nonlinear Systems, Convex Optimization

1. INTRODUCTION

Mission goals for planetary robotics are becoming more ambitious in terms of both desired results and required autonomy. Surface mobility has been identified as one of the fundamental considerations in the design of planetary rovers that can meet these goals (Baumgartner, 2000).

Creating a highly mobile robot depends on both hardware design and the range of motions that can be handled safely by on-board motion planning and control algorithms. Model-based planning and control has been used (Iagnemma *et al.*, 1999) to enable wheeled rovers with rocker-bogie or active suspension to traverse rough terrain . Distributed control techniques have been applied (Pirjanian *et al.*, 2002) to allow coordinated groups of rovers to descend steep cliff faces.

To manage extremely long surface traversals, innovative new mobility strategies involving jumping or hopping robots are also being considered (Hale *et al.*, 2000). These strategies can be much more efficient than those based on wheeled or rocket-powered robots (Fiorini *et al.*, 1999). They can also provide an additional layer of mission robustness. If the robots are unable to move to a desired location because normal mobility strategies are insufficient or have failed, the more aggressive jumping or hopping strategies can be used.

However, more sophisticated control algorithms are required before mobility strategies involving jumping robots can be considered reliable. Nonperiodic jumping and hopping are examples of *aggressive maneuvers*, a class of motions that have proven difficult to plan and to execute. An aggressive maneuver is defined in general as a trajectory along which the system switches into and out of an aggressive operational mode such as free flight (Bretl and Rock, 2002). Within this operational mode, the ability of the robot to affect its trajectory is very low. Thus, the success of the maneuver depends critically on the state of the system as it enters the aggressive operational mode.

There are a variety of methods to handle systems that involve aggressive maneuvers. Methods that explicitly plan these maneuvers are application-

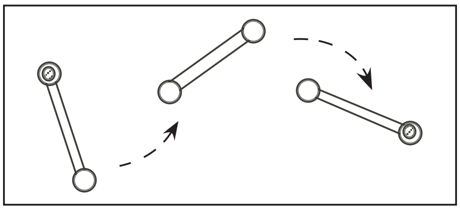

Fig. 1. A simple climbing robot, shown performing an aggressive maneuver. This maneuver is a jump between two pegs, consisting of a detach action, free flight, and an attach action.

specific, as in (Lynch and Mason, 1997) and (Gavrilets *et al.*, 2002). More general methods treat these types of systems using hybrid dynamical models. If a hybrid model has one of several very specific forms, it can be analyzed as in (Heemels *et al.*, 2001). Issues of controllability for specific applications such as periodic legged locomotion have been addressed (Goodwine and Burdick, 1997). However, these methods previously have not been extended to provide a tractable approach to planning and control synthesis for aggressive maneuvers.

This paper describes a general method for generating robust trajectories for aggressive maneuvers, where the level of robustness is guaranteed. It extends the work of (Bretl and Rock, 2002) to present a complete implementation of the planning algorithm for a specific system.

Section 2 presents the specific system to be examined in simulation and frames the problem of planning a single aggressive maneuver. Section 3 describes the implementation of a complete solution method for the example system and describes ways in which this method could be generalized for an arbitrary system. Section 4 discusses the results of applying the solution method to the example system. Finally, Section 5 presents possibilities for future work.

2. PROBLEM DEFINITION

Using a specific system as an example, this section identifies the critical problem in planning an aggressive maneuver. In particular, it identifies a criterion for determining the level of robustness of a given maneuver trajectory.

2.1 Example System

The example system studied in this paper is a robotic system that *depends* on jumping in order to navigate through its environment. This system is a simple "climbing robot," shown in Figure 1. The robot moves in a vertical plane and consists of a single rigid bar, the endpoints of which can attach to, detach from, and exert a torque on pegs scattered throughout its environment.

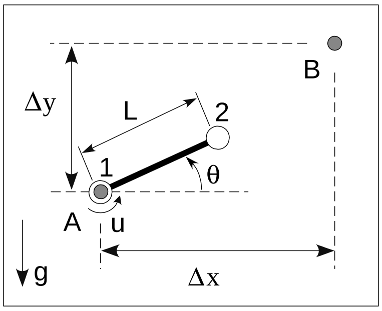

Fig. 2. Problem setup for the climbing robot attempting to jump between Pegs A and B.

Although this system and its environment are highly idealized, the planning and control techniques developed in this paper can be applied directly to more complicated real designs. As long as the contact dynamics can be modeled as discrete then the techniques can be applied to situations with three-dimensional motion and high-DOF manipulation.

2.2 Problem Setup

Assume, as shown in Figure 2, that the robot, which consists of a massless link of length L between two equal masses m, is attempting to jump from Peg A to Peg B. To simplify analysis in this particular example, it will detach End 1 from Peg A and, after a period of free flight, attach End 2 to Peg B. In general, it could attach either end to Peg B.

The robot's trajectory is not controllable after it detaches from the first peg, so the success of a jump depends only on the state $(\theta_0, \dot{\theta}_0)$ of the robot at the time it detaches. Given these initial conditions and the location (x_A, y_A) of Peg A, the position $(x(t), y(t))$ of End 2 of the robot for any future time t is expressed as follows:

$$\begin{aligned} x(t) &= x_A + \frac{L}{2}\cos(\theta_0) + \\ &\quad + \frac{L}{2}\left(\cos(\theta_0 + \dot{\theta}_0 t) - \dot{\theta}_0 t \sin(\theta_0)\right) \quad (1) \\ y(t) &= y_A + \frac{L}{2}\sin(\theta_0) - \frac{gt^2}{2} + \\ &\quad + \frac{L}{2}\left(\sin(\theta_0 + \dot{\theta}_0 t) + \dot{\theta}_0 t \cos(\theta_0)\right) \quad (2) \end{aligned}$$

For the peg geometry $(\Delta x, \Delta y)$ shown in Figure 2, the distance of End 2 from Peg B at time t can be expressed as the function

$$D\left(\theta_0, \dot{\theta}_0, t\right) = \left\| \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} - \begin{bmatrix} x(t) - x_A \\ y(t) - y_A \end{bmatrix} \right\|_2 \quad (3)$$

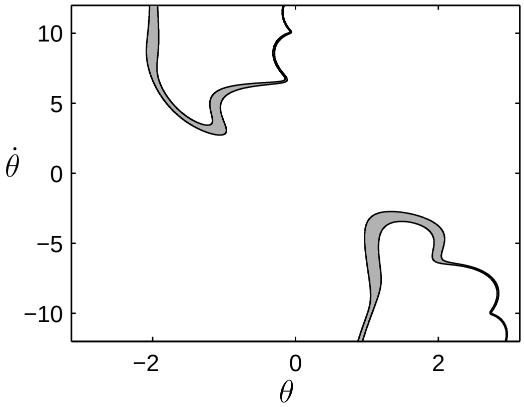

Fig. 3. Goal preimage in the range $\dot{\theta}_0 \in [-12, 12]$ for a single jump of the climbing robot. The region is the shaded area only.

The minimum distance between End 2 and Peg B over all future time is given by

$$D_{\min}\left(\theta_0, \dot{\theta}_0\right) = \min_t D\left(\theta_0, \dot{\theta}_0, t\right) \tag{4}$$

Assume that End 2 of the robot can attach to Peg B if it is within some radius ϵ of the peg. Then for a jump from Peg A at a state $(\theta_0, \dot{\theta}_0)$ to be successful, the following equation must be satisfied:

$$D_{\min}\left(\theta_0, \dot{\theta}_0\right) \leq \epsilon \tag{5}$$

2.3 Goal Preimage

The set of solutions to Equation 5 forms a region in the space of initial conditions. In the absence of noise, this region is the *preimage* of the jump goal, since the robot can detach from Peg A at a state corresponding to any point in this region to arrive safely at Peg B.

Figure 3 shows a portion of this preimage for the climbing robot for the peg geometry $(\Delta x, \Delta y) = (2\sqrt{2}, -2\sqrt{2})$ and $L = 1$. The width of this region is determined by the variable ϵ. The set of solutions for $\epsilon = 0$, resulting in free-flight trajectories that place End 2 exactly on Peg B, is a curve running through the region.

Equations 1-5 can be modified to account for possible noise or control inputs during free-flight. In general, noise will tend to shrink the preimage, while control inputs will tend to expand it.

A feasible trajectory for an aggressive maneuver is any trajectory that ends in a command to detach from Peg A at a state inside the goal preimage. Any planning technique can be used to select this nominal detach state, or nominal transition point. Typically there will be considerations such as obstacle avoidance or global path optimization that result in the selection of a particular detach state. However, it is possible that the selection is arbitrary, because characterizing the entire goal preimage is computationally intensive.

2.4 Robustness Criterion

Several types of uncertainty arise in the execution of an aggressive maneuver. The shape of the goal preimage accounts for uncertainty after the detach action. It is assumed that good control is available for tracking trajectories before the detach action, so uncertainty before the detach action is assumed to be small.

Uncertainty *during* the detach action can be modeled as uncertainty in the exact time at which the detach action occurs. A detach action, although considered to be a discrete event, actually has associated continuous dynamics which are difficult to quantify. At some point these dynamics will result in a transition from the operational mode in which the robot is hanging on Peg A to the free-flight operational mode. The state at which this transition is made is the state that must be inside the goal preimage for a jump to be successful. So the detach action should be regarded as occuring some uncertain amount of time after the detach command is given.

In addition, assume that the detach command is independent of the continuous inputs and trajectory before the detach action occurs. So between the detach command and the transition to a free-flight operational mode, the effect of the continuous torque input does not change. Then for a jump to be considered robust, the continuous trajectory through the nominal detach state must remain within the goal preimage for at least as long as the range over which the time-of-transition is expected to vary.

3. PROBLEM SOLUTION

The generation of a trajectory for an aggressive maneuver that satisfies the robustness criterion developed in Section 2 was implemented in four steps. First, a nominal transition point was chosen. Next, a portion of the goal preimage was generated inside a local region about the transition point. Then, an optimization problem was solved to generate the portion of the trajectory just prior to the transition. Finally, a global plan from arbitrary initial conditions was generated to reach the beginning of the trajectory leading to the transition.

The selection of a nominal transition point, as mentioned in Section 2.3, involves higher level considerations which are beyond the scope of this paper. For this reason, a transition point was arbitrarily chosen using a random search technique.

The generation of a global plan is a well-understood problem. The technique used in this paper was feedback control of system energy, which is a practical approach for the climbing robot system. Details of this approach are given by (Astrom and Furata, 1996).

This section focuses on the two other parts of the solution method, the generation of a local goal preimage and the generation of an optimal trajectory leading to the transition. These algorithms take the nominal transition point as input and return a trajectory that begins at some point x_0 inside the local goal preimage and ends in a transition. The global planning method is then applied to reach x_0 from arbitrary initial conditions.

3.1 Local Goal Preimage

In planning the trajectory leading to a detach action it will be assumed that the goal preimage is convex. This assumption does not hold for the region shown in Figure 3 and does not hold in general. However, this assumption holds within some local neighborhood of a nominal transition point, which is all that needs to be considered for local modification of the continuous trajectory. In addition, this assumption allows the use of the highly tractable planning method to be presented in this section, the solution to which provides a good sub-optimal policy. This solution could be used as a starting point for optimization within the full nonconvex region. Therefore, a convex subset of the true goal preimage will be used for synthesis.

A general method for computing this convex subset is to first compute the complete goal preimage, then search for the largest ellipsoid that contains the nominal detach point and that is completely contained within the preimage. Level set methods are an appropriate way of calculating goal preimages for general systems (Mitchell and Tomlin, 2000).

In practice, it is often quicker to use a heuristic method such as that shown in Figures 4 and 5. First, a polygonal approximation to the local goal preimage is generated. Since the method in Figure 4 relies on only a small number of gradient calculations, it has worked better for this application than other methods such as active contour models (Kass *et al.*, 1987). Next, the polygonal approximation is truncated to a convex subset. This truncation is known as the “potato peeling” problem, for which an optimal solution exists (Chang and Yap, 1986). However, this solution is impractical

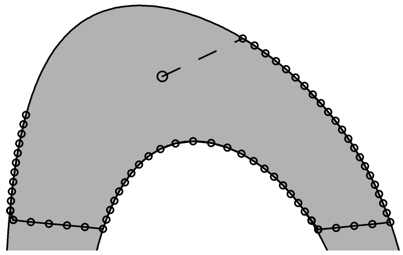

Fig. 4. Iterative generation of a polygonal approximation to the local goal preimage. At each iteration, a small step is taken in the approximate direction of the contour, then adjusted locally to find a new vertex on the boundary.

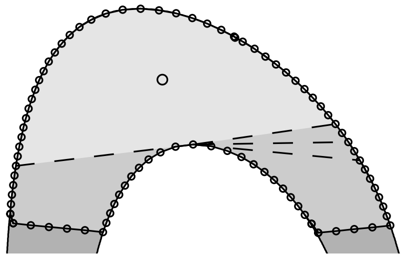

Fig. 5. Truncation of the polygonal approximation of the goal preimage to a convex subset. First, cuts are made along all edges adjacent to concave vertices. Then, vertices adjacent to any remaining concave vertices are removed in descending order of distance from the nominal detach point.

to implement, so the heuristic shown in Figure 5 was used.

Using the above techniques, a convex subset of the local goal preimage can be generated in near real-time (5-10 seconds on an 800 MHz processor.) This subset is polygonal, and can be expressed as a set of linear inequalities.

3.2 Local Trajectory Optimization

Assume that the time-of-transition is expected to vary over a time interval T. Also assume that the set of continuous dynamics before the discrete transition is given by

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{6}$$

Finally assume, as discussed in Section 3.1, that the goal preimage is convex and can be expressed by the set of linear inequalities

$$\mathbf{F}\mathbf{x}(t) \leq \mathbf{g} \tag{7}$$

Then the planning problem for the robustness criterion given in Section 2.4 is expressed as follows:

Problem 1. (Existence)

$$\begin{aligned} \text{Find} : &\ \mathbf{x}_0, \mathbf{u} \\ \text{SubjectTo} : &\ \mathbf{F}\mathbf{x}(t) \leq \mathbf{g}, 0 \leq t \leq T \\ &\ u_{\min} \leq u(t) \leq u_{\max}, 0 \leq t \leq T \end{aligned}$$

In general, the solution to Problem 1 is not unique. The problem statement can be extended to include multiple objectives. For example, let $(\mathbf{X}, \mathbf{U})_T$ be the set of all solutions $(\mathbf{x}_0, \mathbf{u})$ to Problem 1 for a given value of T. Then the solution to Problem 1 of minimum input norm for a given value of T can be found by solving

Problem 2. (Minimum Input Norm)

$$\begin{aligned} \text{Find} : &\ \text{argmin}_{\mathbf{x}_0, \mathbf{u}} \|\mathbf{u}\|_2 \\ \text{SubjectTo} : &\ (\mathbf{x}_0, \mathbf{u}) \in (\mathbf{X}, \mathbf{U})_T \end{aligned}$$

In addition, let $\Psi(T) : T \rightarrow (\mathbf{x}_0, \mathbf{u})$ be the solution to Problem 2, and define the *minimum-escape-time* function as

$$f(\mathbf{x}_0, \mathbf{u}) = \inf \{T : \Psi(T) \in (\mathbf{X}, \mathbf{U})_T\} \quad (8)$$

Then the solution to Problem 2 for the maximum possible value of T, the "most robust" solution of minimum input norm, can be found by solving

Problem 3. (Most Robust)

$$\text{Find} : \text{argmax}_{\mathbf{x}_0, \mathbf{u}} f(\mathbf{x}_0, \mathbf{u})$$

The most robust solution can tolerate the highest possible level of uncertainty in time-of-transition.

The state of the system at any time step t is

$$\mathbf{x}(t) = \mathbf{A}^t \mathbf{x}_0 + \sum_{k=0}^{t-1} \mathbf{A}^{(t-1)-k} \mathbf{B} u(k) \quad (9)$$

So for a fixed value of T, the constraints on the domain of Problem 1 are linear in the variables $\mathbf{x}_0$ and $\mathbf{u}$. Thus, the problem is convex and can be solved as a linear program. The objective function of Problem 2 is a quadratic function of $\mathbf{u}$ while the constraints are identical to those of Problem 1, so Problem 2 is also convex and can be solved as a quadratic program.

Further, sublevel sets of $-f(\mathbf{x}_0, \mathbf{u})$ are convex, so $f(\mathbf{x}_0, \mathbf{u})$ is quasiconcave. Thus, Problem 3 is a quasiconcave optimization problem. This type of problem can be solved efficiently using bisection around a convex sub-problem (Boyd and Barratt, 1991; Boyd *et al.*, 1994).

Although the planning method thus far assumes linear dynamics, it extends naturally to nonlinear dynamics. It has been assumed that the goal

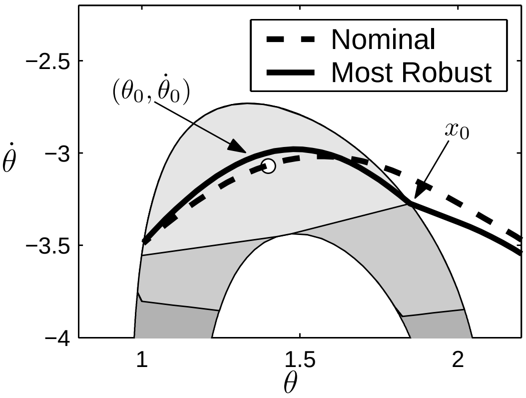

Fig. 6. Most-robust trajectory through a convex subset of the local goal preimage.

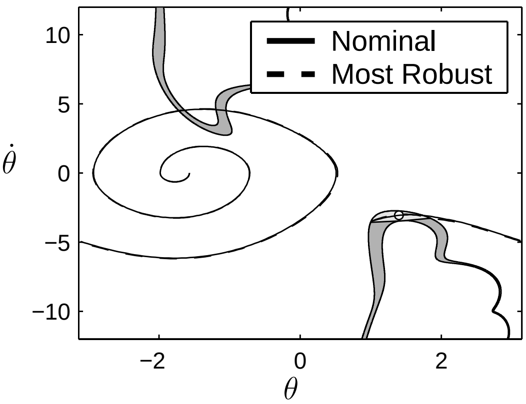

Fig. 7. Global most-robust trajectory.

preimage has been truncated to a small local convex subset about the nominal transition point. Therefore, it can be expected that a linearization about this point will be very accurate throughout the subset. As a result, the solution to the planning problem applied to the linearization will be a very good approximation of the optimal solution of the nonlinear planning problem within the convex subset. For details of this linearization, see (Bretl and Rock, 2002).

4. RESULTS

This section shows the results of applying the planning method presented in this paper to the climbing robot system.

Figure 6 shows the *most robust* solution of the planning problem for the nominally desired transition point $(\theta_0, \dot{\theta}_0) = (1.4, -3.1)$. This solution is the trajectory that remains in the convex subset of the local goal preimage as long as possible, in this case for 0.27 seconds. Figure 7 shows the full global trajectory associated with the most robust solution, starting from $(\theta, \dot{\theta}) = (-1.57, 0)$.

In addition, Figures 6 and 7 show a feasible nominal trajectory for the same jump generated using only the energy feedback control method described in Section 3. This trajectory remains within the goal preimage for 0.23 seconds, which is nearly optimal. In fact, over a range of nominally desired transition points, the trajectories generated using the robust synthesis method remained within the goal preimage only 30% longer than those generated using the energy feedback control method.

Although this result indicates that energy feedback control is a good heuristic for generating robust transition trajectories for the climbing robot system, there are several other advantages of using the robust synthesis method. First, the robustness of the synthesis method is guaranteed. Second, it will be more difficult to find a controller heuristic for a general system.

Finally, the synthesis method allows the use of the null-space of the most robust solution to consider other optimization criteria. The fact that the two trajectories in Figure 6 are so similar is primarily because the null-space of the most robust solution was used in this example to minimize the input norm. A minimum-norm solution is the solution which is closest to a constant-energy trajectory, or in other words which is closest to the energy feedback control trajectory. It might be desirable instead to minimize the time of free-flight or to minimize the velocity of End 2 of the robot upon contact with Peg B. If these criteria are used in the formulation of the most robust solution, the resulting trajectory will be much different from the constant-energy solution.

5. CONCLUSION

This paper motivated the use of *aggressive maneuvers* in the creation of highly mobile planetary robots. It demonstrated that robustness to noise in the time of the transition to an aggressive operational mode is an appropriate design criterion for creating robust trajectories for these maneuvers, and presented a complete implementation of a corresponding planning algorithm for a specific robotic system.

The use of level set methods to calculate the complete goal preimage for a general system is an opportunity for future work. Also, the selection of nominal transition points based on higher-level considerations should be examined.

REFERENCES

Astrom, K. and K. Furata (1996). Swinging up a pendulum by energy control. *Proc. IFAC 13th World Congress.*

Baumgartner, E. (2000). In-situ exploration of mars using rover systems. *Proc. AIAA Space 2000 Conference.*

Boyd, S. and C. Barratt (1991). *Linear Controller Design - Limits of Performance.* Prentice-Hall.

Boyd, S., L. El Ghaoui, E. Feron and V. Balakrishnan (1994). *Linear Matrix Inequalities in System and Control Theory.* SIAM.

Bretl, T. and S. Rock (2002). Robust execution of aggressive maneuvers for planetary robotics. *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit.*

Chang, J.-S. and C. Yap (1986). A polynomial solution for the potato-peeling problem. *Discrete and Computational Geometry* **1**, 155–182.

Fiorini, P., S. Hayati, M. Heverly and J. Gensler (1999). A hopping robot for planetary exploration. *Proc. IEEE Aerospace Conference.*

Gavrilets, V., I. Martinos, B. Mettler and E. Feron (2002). Control logic for automated aerobatic flight of a miniature helicopter. *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit.*

Goodwine, B. and J. Burdick (1997). Gait controllability for legged robots. *Proc. IEEE International Conference on Robotics and Automation.*

Hale, E., N. Schara, J. Burdick and P. Fiorini (2000). A minimally actuated hopping rover for exploration of celestial bodies. *Proc. IEEE International Conference on Robotics and Automation.*

Heemels, W.P.M.H., B. De Schutter and A. Bemporad (2001). Equivalence of hybrid dynamical models. *Automatica* **37**(7), 1085–1091.

Iagnemma, K., F. Genot and S. Dubowsky (1999). Rapid physics-based rough-terrain rover planning with sensor and control uncertainty. *Proc. IEEE International Conference on Robotics and Automation* pp. 2286–2291.

Kass, M., A. Witkin and D. Terzopoulos (1987). Snakes: Active contour models. *Proc. First International Conference on Computer Vision.*

Lynch, K.M. and M.T. Mason (1997). Dynamic manipulation with a one joint robot. *Proc. IEEE International Conference on Robotics and Automation* pp. 359–366.

Mitchell, I. and C. Tomlin (2000). Level set methods for computation in hybrid systems. *Proc. Hybrid Systems: Computation and Control, LNCS 1790.*

Pirjanian, P., C. Leger, E. Mumm, B. Kennedy, M. Garrett, H. Aghazarian, P. Schenker and S. Farritor (2002). Distributed control for a modular, rconfigurable cliff robot. *Proc. IEEE International Conference on Robotics and Automation.*