# State Space Sampling of Feasible Motions for High Performance Mobile Robot Navigation in Highly Constrained Environments

Thomas Howard, Colin Green, Alonso Kelly

# State Space Sampling of Feasible Motions for High Performance Mobile Robot Navigation in Highly Constrained Environments

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

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

**Abstract.** Sampling in the space of controls or actions is a well-established method for ensuring feasible local motion plans. However, as mobile robots advance in performance and competence in complex outdoor environments, this classical motion planning technique ceases to be effective. When environmental constraints severely limit the space of acceptable motions or when global motion planning expresses strong preferences, a state space sampling strategy is more effective. While this has been clear for some time, the practical question is how to achieve it while also satisfying the severe constraints of vehicle dynamic feasibility. This paper presents an effective algorithm for state space sampling based on a model-based trajectory generation approach. This method enables high-speed navigation in highly constrained and/or partially known environments such as trails, roadways, and dense off-road obstacle fields.

## 1 Introduction

Outdoor mobile robot navigation is a challenging problem because environments are often complex and only partially known, dynamics can be difficult to predict accurately, and both planning time and computational resources are limited.

The dynamics of a vehicle can be modeled by a nonlinear differential equation of the form:

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

where **u** is called the input or control vector and **x** is the state vector and both are time-varying points in input and state spaces respectively. The complexity of such accurate models of mobility combined with the scale of outdoor mobile robot navigation leads to a difficult tradeoff between the computational demands of perceptive intelligence at the local level and deliberative intelligence at the global level. It is difficult to be both smart and fast when computation is limited.

A common approach to this problem is to use a multi-level motion planning strategy to generate behavior that is both intelligent and responsive. A large-scale motion plan is generated infrequently based on simplified dynamic models and coarse representations of the environment. This global planner is deliberative and it understands the gross topology of the environment. A finer-scale plan is generated more frequently by a local planner that utilizes higher fidelity dynamic models and finer resolution representations of the environment. The local planner provides obstacle avoidance and ensures dynamic feasibility in the near-term while the global planner provides high-level guidance (e.g. waypoint navigation). This paper deals with generating local motion planning search spaces which satisfy feasibility and environmental constraints while exploiting global guidance.

### 1.1 Motivation

This paper will address a difficult issue that arises in the context of the above two-tiered architecture, and indeed in differentially constrained motion planning in general. The formulation of the local planning problem involves constraints and utilities that are most conveniently expressed in two spaces:

- State Space: Those arising from the environment
- Control Space: Those arising from vehicle mobility

Traditional approaches to local motion planning involve searching alternatives expressed in control space because such alternatives are inherently feasible. Feasibility matters because commanding infeasible actions will lead to collisions or inability to execute other critical maneuvers. When most feasible motions are likely to satisfy environmental constraints, control space sampling is an effective approach. However, if the environment imposes severe limits on acceptable motions, this traditional approach does not work well.

Global guidance is expressed fundamentally in terms of a utility or constraint field over state space. For example, a global path is specified and minimum deviation is desirable, or a road lane is specified and its edges cannot be breached except in lane change maneuvers. Alternatively, a navigation function might associate with every point in state space its expected cost to the goal and its gradient specifies the preferred vehicle orientation at that point. Such guidance is highly valuable because the global path is less likely to contain obstacles, it is often an optimal global solution based on a coarse but informative map, and it may pass through narrow safe regions whose traversal is critical to reaching the goal.

Figure 1 illustrates how sampling in control space is a very poor approach under such conditions whereas state space sampling is ideal.

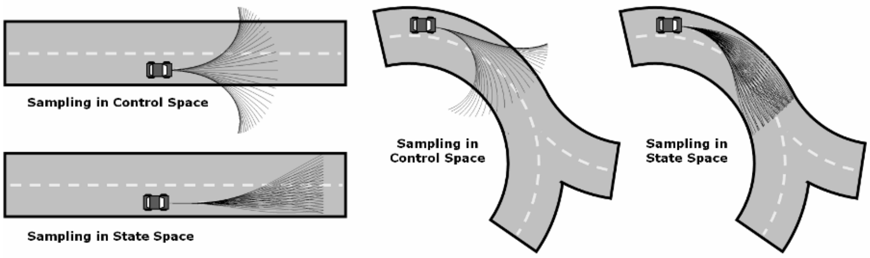


Fig. 1: Motion Planning in Constrained Environments. Search spaces generated by sampling in control space vs. state space are shown in environments that are highly constrained (e.g. road networks). Virtually all options generated by sampling in control space leave the lane or are oriented to do so eventually, whereas those generated by sampling in state space remain within the road network.

A second consideration is path sampling efficiency. All approaches to planning fundamentally consider a finite number of alternatives and pick the best one, but some sets of alternatives are better than others are. Separation matters because nearly identical paths will likely intersect the same obstacles and waste computation. We define a well-separated set of trajectories to be one that covers a majority of the state space with a minimum of overlap. When a global path is specified, it would be useful to control the distribution of trajectory end states to ensure that they are both near and oriented along the global path.

For example, consider the search space illustrated in Figure 2. It was generated by uniformly sampling in control space for different initial conditions. Notice that the trajectories are denser in the direction opposite to the initial curvature ($\kappa_0$) because the vehicle's maximum turning rate ($d\kappa/dt$) leads to the same output for several distinct inputs. Sampling in state space permits direct control over the spacing of the endpoints of these trajectories.

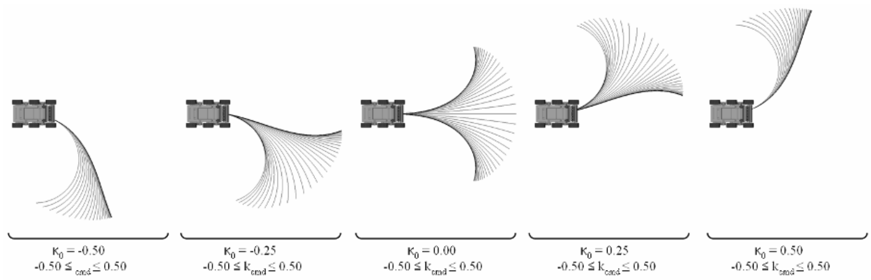


Fig. 2: Irregular Mapping from Control Space to State Space. Accurate dynamic simulations of a set of vehicle controls (constant curvature arcs uniformly sampled between ±0.50 radians/meter) are shown for different initial vehicle curvatures. The responses to the controls are not uniformly separated despite the uniform separation of the controls.

### 1.2 Related Work

Some of the earliest work in outdoor model-based mobile robot motion planning appears in [1], where the local motion planning search space is generated by sampling in the control space of curvature. Each control is passed through a vehicle dynamics model to estimate the response of the vehicle to the control. The shape of the response is highly dependent on the vehicle model and the initial vehicle state (curvatures and velocities).

Similar approaches have been adapted in a variety of other mobile robots [5]. The method presented in [2] samples trajectories around the arc that reacquires a target path, where “nudges” and “swerves” represent small and large lateral offsets (adjustments of the solution trajectory) respectively. Another method for generating local motion planning search space is egographs [7]. This approach generates a well-separated dynamically feasible search space for a limited set of initial vehicle states offline.

### 1.3 Technical Approach

An effective local planning search space would ideally be optimal, efficient, and robust. The search space would be optimal if it could maximally exploit global guidance, efficient if it could control path separation, and robust if it searched only feasible motions. Recent advances in real-time model-based trajectory generation have provided the capability to make progress towards achieving these goals. In [3] a general method is presented that computes control inputs that satisfy a pair of boundary states subject to the vehicle dynamics model, and in [4] we apply it to the problem of path following in the absence of obstacles.

This paper improves on [4] by generating a set of feasible actions by sampling in the surrounding state space. By using the model-based trajectory generation algorithm to generate motions between the current vehicle state and a set of terminal states on the boundary of the local motion planning search space, the state space sampling technique is superior in its efficiency (mean separation of trajectories) and robustness to initial conditions than its control space sampling counterparts. The approach in this paper differs from all of the prior work in its capacity to generate more expressive local motion planning search spaces.

## 2 Adaptive Search Spaces

The algorithm in [3] creates an opportunity to produce end state sample distributions that satisfy environmental, separation, and path following constraints while also being able to produce feasible motions that achieve these states. Section 2.1 outlines our general approach to structuring the search space adaptation algorithm while Section 2.2 discusses tuning the feasible set shape parameters for real vehicle applications.

### 2.1 Adaptive Search Space Design

A general description of state space sampling techniques is to generate a set of actions by solving for trajectories between n boundary state pairs ($\mathbf{x_p}$). The first state in each pair is the initial or current state of the vehicle ($\mathbf{x_i}$) while the second state is the terminal state ($\mathbf{x_f}$) which is reached at the end of each trajectory.

$$\mathbf{x_p} = \begin{bmatrix} \mathbf{x}_{i,0} & \mathbf{x}_{i,1} & \cdots & \mathbf{x}_{i,0} \\ \mathbf{x}_{f,0} & \mathbf{x}_{f,1} & \cdots & \mathbf{x}_{f,n} \end{bmatrix} \text{ where } \mathbf{x} = \begin{bmatrix} x & y & \psi & k & v & \ldots \end{bmatrix}^T \quad (2)$$

Note that this definition is independent of the method of generating the search space. We will utilize the model-based trajectory generation algorithm presented in [3] because of its ability to generate feasible actions between boundary state pairs in real-time, although other techniques could be substituted. Our approach is based on designing rules and parameters that define the shape of the outline of the trajetories and adjust the terminal states based on global guidance and initial vehicle state information.

For example, consider the example search spaces generated using this approach shown in Figure 3. Here, terminal state positions are selected uniformly at a horizon (constant radius) of 5.0 meters from the center of the robot between ±45 degrees from the forward central axis of the vehicle. Three headings at each terminal state position are considered, one aligned with the ray cast from the vehicle to the terminal state position and two others offset at a terminal heading adjustment angle (±45 degrees).

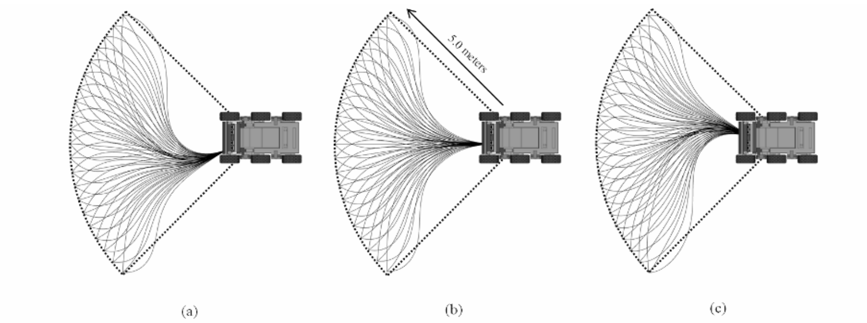


Fig. 3: Uniform Terminal State Sampling for Adaptive Search Space Generation with Varied Initial States. The ability to control the shape of the search space for three initial states: (a) turning left, (b) straight, and (c) turning right. Notice the sampling is more uniformly distributed than in Figure 2.

The result of this approach is a set of sophisticated maneuvers that are roughly equidistant, making this an efficient (non-redundant) search space. Notice that even in the face of varied initial vehicle states, the search space shape adapts to the dynamic constraints of the vehicle. The trajectories reach the same set of well-separated terminal states, in contrast to the results of control space sampling techniques (Figure 2).

### 2.2 Utilizing Global Guidance Information

On the assumption than deviation from global guidance is less likely to lead the goal, a local motion planning search space will improved if it biases its search to be most consistent with global guidance. We typically use a global planner which continuously provides a navigation function (cost from any point to the goal) to the local planner. Given such information, it is better to sample terminal states at a higher density in lower global cost regions and at a lower density in higher cost regions, as shown in Figure 4. Some samples are retained in higher cost regions because the low cost regions produced by the global planner may not reflect actual dynamic constraints of the vehicle, and the global planner may not be able to react quickly to perceived obstacles.

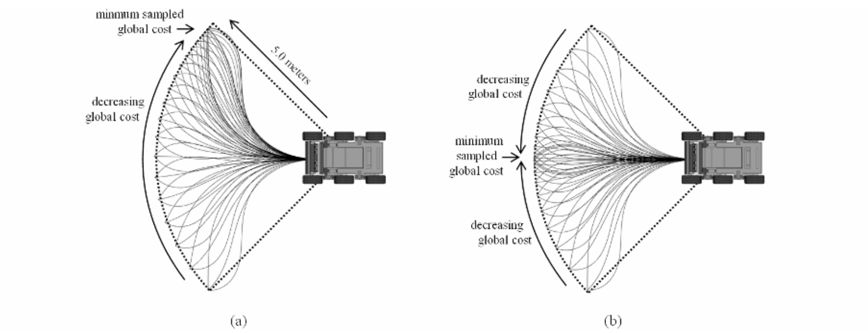


Fig. 4: Focused Terminal State Sampling for Adaptive Search Space Generation. The ability to exploit global guidance via state space sampling generates local motion planning search spaces that are denser in the direction of minimum global cost (and therefore more likely to be obstacle-free). Examples (a) and (b) shows the same setup from Figure 3 but focused in the direction of minimum global cost.

### 2.3 Model-Based Trajectory Generation

To generate each motion, the real-time model-based trajectory generation algorithm in [3] is called for each boundary state pair in $\mathbf{x_p}$. A model-based approach is used to determine the actual command required to complete the necessary motion. The vehicle model is important because the mobile robot will execute these commands directly. The model is itself a tradeoff between speed (dynamics are computed tens of thousands of times per second) and fidelity (how well it can accurately predict the response to the inputs). We have found first-order models of linear and angular velocity response to be an effective tradeoff that encodes the major constraints of motion feasibility.

In general, model-based trajectory generation techniques can be used to solve for motions for arbitrary vehicle state constraints (positions, headings, curvatures, rates of curvatures, velocities, etc…). Here, the trajectory generator is only required to meet terminal position and heading constraints since the vehicle will likely never execute the entire motions between replanning cycles.

In contrast to terminal state, the full initial vehicle state ($\mathbf{x}_i$) is necessary to initialize the vehicle dynamic model used by the algorithm.

### 2.4 Adaptive Search Envelope Determination

The envelope of the adaptive search space depends heavily of the dynamic limitations of the mobile robot. The horizon of the search space shape should adapt to the current speed because of braking distance and obstacle avoidance limitations. Likewise, the range of feasible terminal heading angles can vary heavily for vehicles with small curvature or angular acceleration limits.

An effective way to determine envelope is simply to exhaustively sample control space and record the extremes achieved in state space. This method would have to be employed off-line and it could not be used for vehicles whose models adjust over time. Another approach is to evaluate several aggressive maneuvers (such as max-turn right and left) to form rough bounds on reachable positions and headings at the horizon. This approach is simple, fast, and it can be used for adaptive vehicle models so we have preferred it over the off-line method.

## 3 Experiments & Experimental Results

In order to evaluate the performance of our approach, the adaptive search space algorithm presented in Section 2 has been tested in a series of simulation and field experiments. The simulation experiments are comparisons against arc-based (uniform sampling in control space) local motion planner search spaces in a series of randomized worlds. The field experiments consisted of a series of long distance missions. Sections 3.1, 3.2, 3.3, and 3.4 discuss the vehicle test platform, simulation setup, simulation results, and field experiment results and observations respectively.

### 3.1 Vehicle Test Platform

The vehicle used for this series of simulation and field experiments was Crusher (Figure 5), a six-wheel skid-steered mobile robot that is the current platform for the DARPA UGVC-Perceptor Integrated (UPI) project. The simulation uses the same simplified dynamic model of the vehicle (which represents the system delays and the linear and angular acceleration limits) as the one used by the planning system, so execution of commands is nearly ideal. The fidelity of

Fig. 5: Crusher. The current platform of the DARPA UGCV-PerceptOR Integrated (UPI) program.

the model is reduced in the field experiments, as effects such as wheel slip and sliding are not predicted as well on the vehicle as they are in simulation.

### 3.2 Simulation Setup

To test the effectiveness of this adaptive search space on a statistically significant number of cases, a series of simulations were performed. The simulation consists of a rectangular vehicle driving through a randomly generated world while trying to reach the goal waypoint on the other side of the world. As the vehicle traverses the simulated world the global planner, Field D* [8], constantly provides the global path to goal and a field (navigation function) of path costs surrounding the vehicle. The cost computed from the convolution of the vehicle body along each trajectory, combined with the D* path cost at the end of that trajectory, comprises the score used to select the trajectory to follow. That trajectory is then followed during the next planning cycle, and a new trajectory selected at the end of that cycle. For simplicity, the planner is not allowed to perform any stopping or backup maneuvers and the vehicle speed is pre-computed based on obstacle cost and density.

The simulated worlds used are a set of three, connected real-valued obstacle fields filled with randomly positioned circular obstacles of different densities (a single example is shown in Figure 6). The target speeds in the high-, medium-, and low-density obstacle fields were 1.5 meters/second, 6.0 meters/second, and 9.0 meters/second respectively. The vehicle model in the simulator is the same vehicle model used by the trajectory generator (for the adaptive search space) and the vehicle motion simulator (for the arc-based search space). To ensure a fair comparison of search spaces in the test, both the adaptive and arc-based sets are comprised of the same number of trajectories (99). The arc-based set forward simulates 99 different constant curvatures whereas the adaptive set plans motions to three different headings at each of 33 different positions biased by the minimum D* path cost.

### 3.3 Simulation Results

One hundred simulated runs were performed for each of the local planner search spaces. For each pair of runs one simulated world is generated and both the arcs-based and adaptive search spaces were tested on that world. One of the representative simulation runs comparing the two approaches is shown in Figure 6. In all but one of the simulated worlds the adaptive search space outperformed the arc-based search space. On average, the overall path cost from the start to the end states was 24.8% lower for the adaptive search space, demonstrating that the improved flexibility and efficiency of the adaptive search space provides a performance advantage over constant curvature-control versions. In the best and worst cases, the adaptive search space had 79.5% lower and 20.0% higher overall path cost respectively. Lower overall path cost directly relates to probability of mission success as higher cost regions pose threats to vehicle survival.

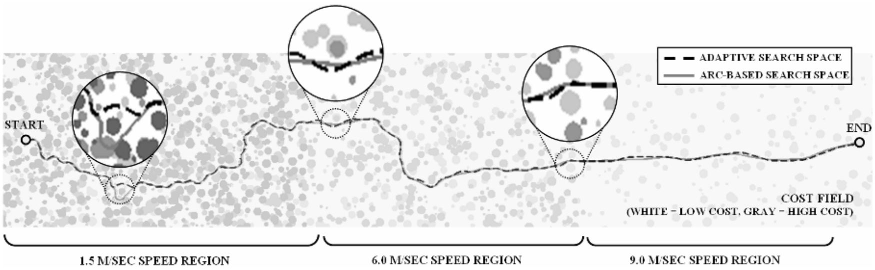


Fig. 6: Simulation of arc-based vs. adaptive search spaces. Each type of motion planning scheme (arc-based and adaptive search spaces) was evaluated in a representative simulated environment. In this particular example, the adaptive search space has a 43.0% lower cost along the path.

### 3.4 Field Experiment Results

Variants of the adaptive search space have been integrated and field tested on Crusher mobile robot (Figure 5). The algorithm demonstrated the capability to dodge obstacles and reacquire paths at speeds up to 12 meters/second in off-road environments. Figure 7 shows an example of the local planner utilizing the adaptive search space from field data. The minimum cost trajectory selected is a swerve maneuver that avoids the high cost regions in the front right and left of the robot. The new search space so consistently outperforms the older arc-based one that it has replaced it in all field tests.

## 4 Conclusions and Future Work

We have leveraged our own recent work in model-based trajectory generation to create a capacity to navigate effectively in difficult environments while preserving inherent feasibility of local motion plans. Key aspects of the technique include explicit computation of the shape of the feasible set, and sampling strategies that adjust the distribution of samples to exploit global guidance. The benefits of the adaptive search space have been demonstrated in simulation and confirmed in off-road navigation.

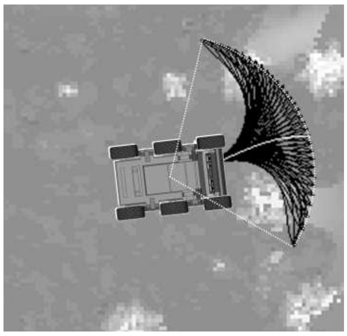

Fig. 7: Field Experiments of Dynamically Adaptive Search Spaces.

Current and future work of this algorithm include more extensive field testing and refinement of the search space design to more optimally represent the set of all feasible motions of the vehicle at different speeds. Also, a more gradual decay of the fidelity of the dynamic model (as opposed to the drop-off between the local and global motion planner) will lead to robots that make better local decisions based on future dynamic limitations.

## Acknowledgement

This work was sponsored by the Defense Advanced Research Projects Agency (DARPA) under contract “Unmanned Ground Combat Vehicle – PerceptOR Integration” (contract number MDA972-01-9-0005). The views and conclusions contained in this document 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] Kelly, A., Stentz, A. "Rough Terrain Autonomous Mobility - Part 2: An Active Vision, Predictive Control Approach ", Autonomous Robots, 5, 163-198, 1998.

[2] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Goffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., and Stang, P. (2005). Stanley: The Robot that Won the DARPA Grand Challenge. *Journal of Field Robotics*. 23(9), 661-692 (2006). 2006 Wiley Periodicals, Inc.

[3] Howard, T.M. and Kelly, A. (2006). Optimal Rough Terrain Trajectory Generation for Wheeled Mobile Robots. *International Journal of Robotics Research*, Vol. 26, No. 2, February, 2007, pp. 141-166

[4] Howard, T.M., Knepper, R.A., and Kelly, A. (2006). Constrained Optimization Path Following of Wheeled Mobile Robots in Natural Terrain. *International Symposium on Experimental Robotics 2006*, July, 2006.

[5] Wettergreen, D., Tompkins, P., Urmson, C., Wagner, M., and Whittaker, W. (2005). Sun-Synchronous Robotics Exploration: Technical Description and Field Experimentation. *International Journal of Robotics Research*. Vol. 24, No. 1, pp 3-30, January. 2005 Sage Publications.

[6] Green, C. and Kelly, A. (2006). Optimal Sampling in Space of Paths: Preliminary Results. Technical Report CMU-RI-TR-06-51. Robotics Institute, Carnegie Mellon University, November.

[7] Lacze,A., Moscovitz,Y., DeClaris, N. and Murphy, K. (1998). Path planning for autonomous vehicles driving over rough terrain. Proceedings of the 1998 IEEE ISIC/CIRA/ISAS Joint Conference.

[8] Ferguson, D. and Stentz, A. (2005). Field D*: An Interpolation-Based Path Planner and Replanner. Proceedings of the International Symposium on Robotics Research (ISRR), October, 2005.