Path Set Relaxation for Mobile Robot Navigation

Philipp Krüsi*, Mihail Pivtoraiko**, Alonzo Kelly**, Thomas M. Howard***, and Roland Siegwart*

*Autonomous Systems Lab, Institute for Robotics and Intelligent Systems, ETH Zurich
e-mail: kruesip,rsiegwart@ethz.ch

**Robotics Institue, Carnegie Mellon University
e-mail: mihail, alonzo@cmu.edu

***Mobility and Robotic Systems (347), Jet Propulsion Laboratory, California Institute of Technology
e-mail: Thomas.M.Howard@jpl.nasa.gov

Abstract

This paper addresses autonomous navigation and goal acquisition for mobile robots operating in difficult, cluttered environments. In particular a hierarchical approach to navigation is of interest, which subdivides the problem into global and local components. Local planners attempt to search the continuum of actions for a best (safest, efficient) route towards a goal. To achieve real-time performance, the search space is often sampled in low-dimensional action or state space. This paper explores a relaxation-based approach that optimizes the sampled action space for the perceived environment. The gradient-based approach minimizes the cost of each motion in the path set until convergence into a local optimum is reached. Simulation experiments show that relaxed arc sets offer better approximations of the acceptable path continuum and lead to safer navigation in rough terrain and dense obstacle fields. Additional experiments explore the benefits of sampled action spaces composed of higher-order action primitives (clothoids) and a graduated fidelity inspired lookahead technique.

1 Introduction

Mobile robot navigators attempt to traverse unknown or partially known terrain towards some goal. Some applications, particular planetary exploration, require autonomous navigation to operate efficiently as only periodic, high-latency communication with the robot is possible. Reliable and robust operation is also important as physical access to the platform is impractical (if not impossible) in such applications.

Computational resources of contemporary mobile robots are typically scarce. It is therefore common to subdivide the navigation problem into planning approximate routes to the goal (*global* planner) and controlling the robot to follow the global path and deviate around obstacles (*local* planner) [14, 15, 5, 6]. The global planner reasons over large distances and typically disregards the vehicle mobility constraints in order to reduce computational complexity. Such a planner may produce a path that is difficult or impossible to follow by the physical robot. It is therefore essential to design sophisticated local planners, which computes feasible motion commands while taking into account the information provided by the global planner and satisfying both mobility and environment constraints.

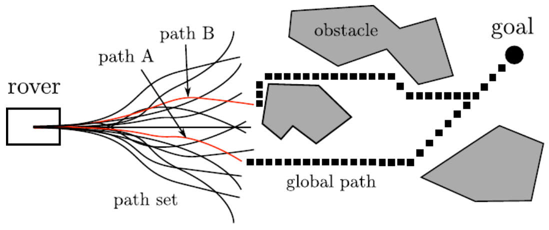


Figure 1. : A representation of common hierarchical techniques for mobile robot navigation. A local planner searches a path set for a feasible, safe maneuver towards the goal. A global planner searches in a two-dimensional state space (grid) to estimate the cost-to-goal from the end of each path in the path set.

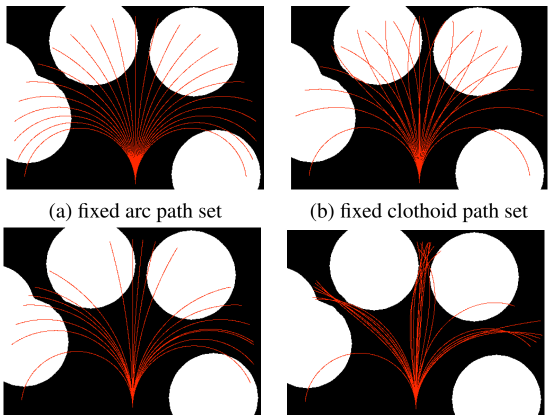
Navigation of mobile robots from the initial pose to the specified goal is typically performed by a series of motion steps. Each step consists of evaluating an array of motion alternatives, a *path set*, selecting the most attractive one and executing this motion. Figure 1 shows an example path set as well as two global paths from the endpoints of two paths in the path set.

Hierarchical navigators select actions by minimizing a cost function composed of a weighted sum of local and global path cost:

$$\min: \quad w_l c_{local} + w_g c_{global} \tag{1}$$

In Equation 1, c_{local} describes the goodness of a specific path based on local criteria such as obstacle avoidance and c_{global} rates the path from the perspective of the global planner. Goodness is itself a application-specific quantity that is often composed of energy, risk, utility, or some combination of the three.

One type of local planner searches a discrete set of constant curvature arcs in a path set [6]. While search in



(a) fixed arc path set (b) fixed clothoid path set

(c) relaxed arc path set (d) relaxed clothoid path set

Figure 2. : An example of several approaches to local planner path set generation. Fixed path sets of arcs and clothoids uniformly sample an action space to find a feasible route through the binary obstacle field. Relaxed path sets can optimize primitive motions to pass between obstacles and increase the number of safe candidate actions. Clothoids provide additional flexibility (S-turns) to pass through configurations of obstacles that even relaxed arcs cannot.

a one-dimensional path space is efficient, there are obstacle configurations where this approximation of the continuum may limit performance. For a number of environmental configurations, including narrow corridors, roads, and cluttered obstacle fields, a particular sampling of discrete arcs may artificially constraint the performance of the navigator.

Two potential avenues for improving the performance of path set based navigators include relaxation and modified sampling techniques (Figure 2). If cost-gradients can be extracted from small perturbations of the sampled motions, paths can be optimized for a particular local planning problem to improve the path set quality. A second approach moves beyond arcs to higher-order primitives (e.g. clothoids) to more effectively represent the reachable state space.

This paper continues with a review of the prior art in mobile robot navigation in Section 2. Section 3 discusses the advantages and drawbacks of several types of path parameterization. Section 4 outlines path set relaxation of mobile robot navigation applied to arc-based and clothoid-based path sets. Section 5 presents several experiments to address the performance benefits of relaxation and path sets using different motion primitives. The paper concludes with a discussion of the results and prior work in Sections 6 and 7.

2 Related Work

Outdoor robot navigation and obstacle avoidance has received a great deal of attention over the last several decades. Hans Moravec's work on the Stanford Cart was a particularly notable early effort in this domain [10]. The Cart utilized a variant of a *bug* algorithm [9] for driving toward a specified goal position, while avoiding previously unknown obstacles, detected using stereo vision. The Autonomous Land Vehicle (ALV) [2] was also capable of autonomous navigation. It traversed partially unknown outdoor environments by attempting to follow global guidance toward the goal, via pre-defined waypoints, utilizing a local control architecture that was responsible for generating immediate actions of the robot. That architecture was developed as a set of low-level behaviors that reacted to the sensed environment. The behaviors in this set received votes depending on certain detected features, and the winning behavior took control of the vehicle until the next round of voting was conducted. A sense-plan-act loop implemented in this manner inspired a great deal of modern work in this domain.

Stentz and Hebert [16] extended the ideas behind the ALV and Moravec's work, and proposed an autonomous navigation system that joined a behavior-based local control architecture with a motion planner capable of generating higher-level, global, guidance toward the goal. The former component was implemented as a set of constant-curvature arc primitives, where an element was suitably chosen for execution based on the environment. The latter was D*, a deterministic graph search algorithm, operating on a grid representation of the environment, capable of incorporating updated environment information via reusing previous computation [17]. This system was further refined in follow-up work [5, 15]. A system that similarly features the local and global planning components was successfully demonstrated for exploration of Mars [14]. Similar solutions are employed for autonomous robot navigation to this day [6]. In this paper we follow the same approach, yet extend it by considering clothoid motion primitives, endowed with relaxation capability. Other work has demonstrated navigation search spaces with alternative representations including polynomial curvature splines [3]. The search space construction technique in [3] also biases the sampling of the motions in the path set based on the estimated global path cost and environmental constraints.

Improving the quality of robot motions is often posed as an optimization problem. Much work was done in modifying motions without regard to obstacles [8, 7]. More recently, obstacle-aware path optimization methods were developed, both for parametric [4] and non-parametric [12, 1, 13] path representations. Our approach is most closely related to the former. Its contribution is in leveraging path optimization methods in the aforementioned two-tier autonomous navigation architecture in order to improve the local motion alternatives of the robot.

3 Motion Representation

The parameterization of the motion primitives considered by the local planner is an important factor for navigation. Such parameterization represents an approximation to the continuum of all the motions that are executable by the robot. On the basis of Taylor's theorem, any feasible vehicle motion can be parametrized as a curvature polynomial in arc length s:

$$\kappa(s) = k_0 + k_1 s + \ldots + k_n s^n \tag{2}$$

Accordingly, a path is entirely defined by its length L and the parameter vector $\vec{k}$:

$$\vec{k} = [k_0, k_1, \ldots, k_n]^T \in \mathbb{R}^{n+1}$$

An arc-shaped path is described by a single parameter, its curvature:

$$\kappa(s) = k_0, \quad s \in [0, L] \tag{3}$$

Arc-based motion planners have been used by a wide range of outdoor mobile robots [14]. The simplicity of arcs is beneficial in many aspects, but simultaneously limits the flexibility of robot motions. The set of poses that can be reached by following an arc for a fixed distance is given by a single curve in the configuration space (x, y, θ): for every accessible position (x, y) there is exactly one associated heading value θ and a unique path.

Introducing an additional parameter leads to paths whose curvature changes linearly with curve length. Accordingly, the shape of a clothoid of length L is described by the parameter vector $\vec{k} = [k_0, k_1]^T$, which defines the curvature of the path:

$$\kappa(s) = k_0 + k_1 s, \quad s \in [0, L] \tag{4}$$

The advantages of clothoids compared to arcs are twofold. First, the reachable subset of the configuration space for a fixed-length path is no longer a curve, but a three-dimensional surface. This yields greater flexibility: the position of the path endpoint is no longer related to a particular heading angle. Additionally, clothoids feature a larger variety of shapes compared to arcs. This may be highly advantageous for obstacle avoidance, since a specific position (x, y) can be reached by various different paths. The drawback of clothoids (or any higher-order polynomials than arcs) is the mobility system requirement that curvature may vary while the mobile robot is travelling. From these considerations we conclude that planner performance can be improved for many mobile robots simply by using a path set composed of clothoids instead of arcs. This specific conjecture is experimentally studied later in Sections 5.2 and 5.3.

4 Path Relaxation

The principle of path relaxation can be applied to arcs, clothoids, or any arbitrary motion primitive. Based on the assumption of planar environments we use a cost map ($c_{map}(x, y)$) to assign a certain cost to every point (x, y) in the plane. The objective function for relaxation is the cost of a path, which we define as the integral of the cost map along the path as follows,

$$c(\vec{k}) := \int_0^L c_{map}(x_p(\vec{k}, s), y_p(\vec{k}, s))\, ds \tag{5}$$

where x_p and y_p describe the course of the path in the world coordinate frame. The parameters $\vec{k}$ cannot take arbitrary values, since curvature is typically limited to a maximum value $\kappa_{max} > 0$ (potentially large) by system constraints. Point-turn motions (infinite curvature) are treated similarly, as discussed in later in the section. We define the set $S^{n+1} \subseteq \mathbb{R}^{n+1}$ of all feasible paths as follows.

$$S^{n+1} = \{\vec{k} \in \mathbb{R}^{n+1} \mid |\kappa(s)| \leq \kappa_{max}\ \forall s \in [0, L]\} \tag{6}$$

Given the above definitions, path relaxation can be formulated as a constrained optimization problem:

$$\begin{aligned} \text{min:} \quad & c(\vec{k}) \\ \text{s.t.}\ & \vec{k} \in S_r^{n+1} \end{aligned} \tag{7}$$

where $S_r^{n+1} \subseteq S^{n+1}$ is a subset of feasible paths containing the initial path. Typically relaxation is applied to a set of N motion primitives $\{\vec{k}_1^0, \ldots, \vec{k}_N^0\}$, and the optimization of each path is constrained to a certain region S_r^{n+1} around its initial parameters, such that every point in S^{n+1} is covered by at least one of these regions. This guarantees that the entire range of feasible motions is searched for local minima of cost, and ensures that a certain selection of motions is maintained for subsequent selection via Equation 1 – an attractive property from the point of view of completeness of the local planner.

In general, the integral (5) cannot be computed in closed form. The cost of a path $\vec{k}$ of length L is therefore approximated by the sum of N_c cost measurements along the path, denoted by $\hat{c}(\vec{k})$:

$$c(\vec{k}) \approx \hat{c}(\vec{k}) := \Delta s \sum_{i=0}^{N_c - 1} c_{map}(x_p(\vec{k}, i\Delta s), y_p(\vec{k}, i\Delta s)) \tag{8}$$

Thereby, the resolution is given by $\Delta s = L/(N_c - 1)$. For the purpose of path relaxation, the cost map $c_{map}(x, y)$ must be defined in such a way that the objective function $\hat{c}(\vec{k})$ allows computation of a well-behaved numerical gradient driving the paths away from obstacles (Section 5.1.1).

4.1 Arcs

The most basic version of path relaxation consists of varying the curvature of an arc (3), such that its cost induced by the underlying map is minimized. According to (6), the set of feasible arcs is given by

$$S^1 = \{k_0 \in \mathbb{R} \mid -\kappa_{max} \leq k_0 \leq \kappa_{max}\}\ .$$

Consequently, the curvature range for relaxation of a single arc k_0^0 is an interval S_r^1:

$$S_r^1 = \{k_0 \in S^1 \mid k_0^0 - \Delta\kappa^- \leq k_0 \leq k_0^0 + \Delta\kappa^+\} \subseteq S^1$$

The relaxation algorithm for constant curvature arcs, presented in Algorithm 1, computes the optimal curvature in a given range S_r^1 using gradient descent with a fixed step size $\delta\kappa$, which is reduced in the vicinity of a minimum in

Algorithm 1: RelaxArc($k_0^0, \Delta\kappa^-, \Delta\kappa^+$)

```
Require: $\delta\kappa > 0$, $\delta\kappa_{min} > 0$
Input: Initial curvature and range for relaxation
Output: Curvature of relaxed arc
$\hat{c}_0 \leftarrow \hat{c}(k_0^0)$, $\hat{c}^+ \leftarrow \hat{c}(k_0^0 + \delta\kappa_{min})$, $\hat{c}^- \leftarrow \hat{c}(k_0^0 - \delta\kappa_{min})$
if $\hat{c}^- < \hat{c}_0$ and $\hat{c}^- < \hat{c}^+$ then
    $\delta\kappa \leftarrow -\delta\kappa$
else if $\hat{c}^- \geq \hat{c}_0$ and $\hat{c}^+ \geq \hat{c}_0$ then
    return $k_0^0$
$k_{new}, k_{old}, k_0^* \leftarrow k_0^0$
$\hat{c}_{new}, \hat{c}_{old} \leftarrow \hat{c}_0$
$coll \leftarrow$ CheckCollision($k_0^0$)
while $|\delta\kappa| \geq \delta\kappa_{min}$ do
    while $\hat{c}_{new} \leq \hat{c}_{old}$ do
        $\hat{c}_{old} \leftarrow \hat{c}_{new}$
        $k_{new} \leftarrow k_{new} + \delta\kappa$
        $coll_{old} \leftarrow coll$
        $coll \leftarrow$ CheckCollision($k_{new}$)
        if $coll_{old} = false$ and $coll = true$ then
            return $k_{old}$
        $k_{old} \leftarrow k_{new}$
        $\hat{c}_{new} \leftarrow \hat{c}(k_{new})$
        if $\hat{c}_{new} \leq \hat{c}_{old}$ then
            if $k_{new} > k_0^0 + \Delta\kappa^+ + \delta\kappa$ then
                return $k_0^0 + \Delta\kappa^+$
            if $k_{new} < k_0^0 - \Delta\kappa^- + \delta\kappa$ then
                return $k_0^0 - \Delta\kappa^-$
            if $\hat{c}_{new} = \hat{c}_{old}$ then
                return $k_{new} - \delta\kappa$
            $k_0^* \leftarrow k_{new}$
    if $k_{new} - \delta\kappa = k_0^0$ then
        $k_{new} \leftarrow k_0^0$
    else
        $k_{new} \leftarrow k_{new} - 2\delta\kappa$
    $\hat{c}_{old}, \hat{c}_{new} \leftarrow \hat{c}(k_{new})$
    $\delta\kappa \leftarrow \delta\kappa/2$
return $k_0^*$
```

order to increase accuracy. We determine the direction for curvature change based on the gradient in the cost function $\hat{c}(\cdot)$ at the initial curvature value, and repeatedly modify the curvature by $\delta\kappa$. This is carried out as long as the cost of the arc decreases in every step *and* the curvature stays inside S_r^1. If the cost starts to increase inside S_r^1, a local minimum has been crossed. In this case, curvature is set to the value of two steps before and relaxation resumes with a reduced step size. This procedure is repeated until the step size falls below a certain minimum value $\delta\kappa_{min}$. Using a fixed step size is advantageous for our application. It guarantees that the algorithm does not escape the given region for relaxation.

Some mobile robots have the ability to perform point turns. Slightly modified, Algorithm 1 can be used to find the optimal heading change for such a maneuver. The application of the algorithm to this problem is straightforward: the only modification concerns the parameter that is varied, where curvature is replaced by heading.

4.2 Clothoids

In order to represent a feasible motion, a clothoid of length L, as defined in (4), must satisfy

$$|\kappa(s)| = |k_0 + k_1 s| \leq \kappa_{max} \quad \forall s \in [0, L] \ .$$

Due to the linearity of $\kappa(s)$, it is sufficient to constrain the curvature at the beginning and the end of the path. Consequently, the set of feasible clothoids, S^2, is given by

$$S^2 = \{[k_0, k_1]^T \in \mathbb{R}^2 \mid |k_0| \leq \kappa_{max}, \ |k_0 + k_1 L| \leq \kappa_{max}\} \ ,$$

which represents the area of a parallelogram in the parameter space (k_0, k_1), as illustrated in Figure 3 (left).

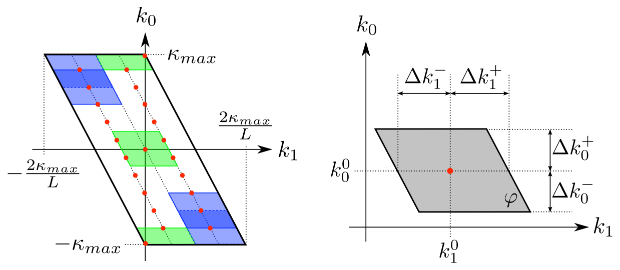


Figure 3. : Left: parallelogram of all feasible clothoids of length L, S^2. Right: region for relaxation of a single clothoid, S_r^2.

The region for relaxation of a single clothoid, denoted by S_r^2, is a subset of S^2 containing the initial clothoid. When relaxation is applied to a set of N clothoids of length L, each initial clothoid appears as a point inside the parallelogram S^2. Since it is desirable that the the entire area of S^2 is covered by the relaxation region of at least one clothoid in the set, a good choice for S_r^2 is a small parallelogram with the same skew angle φ as the parallelogram S^2, as shown in Figure 3 on the right.

Algorithm 2 for clothoid relaxation represents a variant of gradient descent. Given the initial clothoid parameters $[k_0^0, k_1^0]^T$ and the parallelogram S_r^2 defining the boundaries for relaxation, k_0 and k_1 are iteratively modified by certain step sizes δk_0 and δk_1until either a local minimum of the cost function $\hat{c}(\cdot)$ is found or the boundary of S_r^2 is reached. Standard gradient descent employs step sizes proportional to the gradient components. However, this may yield excessively large step sizes where gradients are high. This is especially undesirable in our case, since the minimization is constrained to a relatively small region. We therefore introduce upper and lower bounds for the absolute values of the step sizes, denoted by $(\delta k_0^l, \delta k_0^u)$ and $(\delta k_1^l, \delta k_1^u)$, respectively:

$$\delta k_i^l \leq |\delta k_i| \leq \delta k_i^u, \quad i \in \{0, 1\}$$

Given the components of the numerical gradient of $\hat{c}(\cdot)$, denoted by dk_0 and dk_1, the respective step sizes are computed via a Gaussian mapping as follows (σ is a design

Algorithm 2: RelaxClothoid(k_0^0, k_1^0, S_r^2)

```
Require: $0 < \epsilon \ll 1,\ 0 < h \ll 1,$
         $0 < \delta k_0^l < \delta k_0^u,\ 0 < \delta k_1^l < \delta k_1^u$
Input: Initial parameters and range for relaxation
Output: Parameters of relaxed clothoid

$coll \leftarrow true$
$dk_0, dk_1 \leftarrow 0$
$\delta k_0, \delta k_1 \leftarrow 1$
$k_0^* \leftarrow k_0^0,\ k_1^* \leftarrow k_1^0$
while $|\delta k_0| > \epsilon$ or $|\delta k_1| > \epsilon$ do
    $coll_{old} \leftarrow coll$
    $coll \leftarrow$ CheckCollision($k_0^*, k_1^*$)
    if $coll_{old} = false$ and $coll = true$ then
        return $[k_{0,old}^*, k_{1,old}^*]$
    $dk_{0,old} \leftarrow dk_0,\ dk_{1,old} \leftarrow dk_1$
    $dk_0 \leftarrow (\hat{c}(k_0^* + h, k_1^*) - \hat{c}(k_0^*, k_1^*))/h$
    $dk_1 \leftarrow (\hat{c}(k_0^*, k_1^* + h) - \hat{c}(k_0^*, k_1^*))/h$
    if $dk_0 \cdot dk_{0,old} < 0$ then
        $\delta k_0^u \leftarrow \delta k_0^u/2,\ \delta k_0^l \leftarrow \delta k_0^l/2$
    if $dk_1 \cdot dk_{1,old} < 0$ then
        $\delta k_1^u \leftarrow \delta k_1^u/2,\ \delta k_1^l \leftarrow \delta k_1^l/2$
    $\delta k_0 \leftarrow \mathrm{sgn}(dk_0) \cdot (\delta k_0^u - (\delta k_0^u - \delta k_0^l) \cdot \exp(-\frac{dk_0^2}{2\sigma^2}))$
    $\delta k_1 \leftarrow \mathrm{sgn}(dk_1) \cdot (\delta k_1^u - (\delta k_1^u - \delta k_1^l) \cdot \exp(-\frac{dk_1^2}{2\sigma^2}))$
    $k_{0,old}^* \leftarrow k_0^*,\ k_{1,old}^* \leftarrow k_1^*$
    $k_0^* \leftarrow k_0^* - \delta k_0,\ k_1^* \leftarrow k_1^* - \delta k_1$
    if $[k_0, k_1]^T \notin S_r^2$ then
        $\delta k_0^u \leftarrow \delta k_0^u/2,\ \delta k_0^l \leftarrow \delta k_0^l/2$
        $\delta k_1^u \leftarrow \delta k_1^u/2,\ \delta k_1^l \leftarrow \delta k_1^l/2$
        $k_0^* \leftarrow k_{0,old}^*,\ k_1^* \leftarrow k_{1,old}^*$
return $[k_0^*, k_1^*]^T$
```

parameter):

$$\delta k_i = \mathrm{sgn}(dk_i)(\delta k_i^u - (\delta k_i^u - \delta k_i^l)e^{-\frac{dk_i^2}{2\sigma^2}}), \quad i \in \{0, 1\}$$

The upper and lower bounds for the step sizes are reduced to one half of their current values when either the corresponding component of the gradient changes its direction (crossing of a minimum) or the boundaries of S_r^2 are passed, in which case additionally k_0 and k_1 are set to their previous values. This enables both accurate solutions and reasonably fast convergence.

5 Experimental Results

This section presents simulation experiments of the presented techniques for path set relaxation and varying motion parameterization. Section 5.1 discusses implementation of the cost map, motion alternatives, relaxation, and navigation lookahead. Sections 5.2 and 5.3 compare the performance of four path set alternatives composed of fixed arcs, relaxed arcs, fixed clothoids, and relaxed clothoids.

5.1 Experimental Setup

5.1.1 Cost Map Representation

In the simulation experiments the environment is represented by a two-dimensional grid-based map. Every map cell covers an area of 20×20 cm and has an integer cost value in the range $[0, c_{max}]$, with $c_{max} = 255$. Circular obstacles are used to represent impassable areas. In order to provide a reasonable gradient in the cost function $\hat{c}(\cdot)$, such an obstacle appears in the map as an approximation of a two-dimensional Gaussian function $G(r)$, defined as follows.

$$G(r) = c_{max} \cdot e^{-\frac{r^2}{2\sigma^2}}$$

The cost at a point (x, y) in the plane, denoted by $c_{map}(x, y)$, is computed using bilinear interpolation between the values of the four closest map cells. Interpolation is necessary for the computation of numerical gradients of $\hat{c}(\cdot)$.

5.1.2 Motion Alternatives and Relaxation

In the experiments local planners applied with $N = 23$ motion alternatives (either arcs or clothoids) of equal length 3 m and a maximum curvature of 1 m^{-1}.

The arc-based path sets are designed such that the samples are evenly distributed between the maximum and minimum curvatures ($\pm\kappa_{max}$). Path relaxation is performed by relaxing each arc within the curvature range between its two neighbors in the initial set.

The clothoid-based path sets consist of N points inside the parralelogram S^2 of all feasible motions. In order to assure that the mobile robot can move forward on a straight line as well as perform turns with maximum steering, we choose one clothoid with zero curvature and two with maximum curvature ($k_0 = \pm\kappa_{max}, k_1 = 0$). The remaining clothoids are equally distributed on the two lines $k_1(k_0) = (-k_0 \pm \kappa_{max}/2)/L$ inside the parallelogram. The dots in Figure 3 illustrate the clothoids used in the path set experiments. The illustration further shows the selected boundaries for relaxation in the form of small parallelograms. Note that every point in the space of feasible motion parameters is covered by the relaxation regions of at least two initial clothoids, except for the two small parallelograms at the top left and the bottom right.

5.1.3 Lookahead

As outlined in Section 1, using a global planner disregarding the vehicle constraints yields potentially infeasible paths. As a consequence, it may occur that there is no feasible forward motion, once the mobile robot has reached the endpoint of the path computed by the local planner, even though the global planner has found a path from this point to the goal. This is undesirable, since it requires a backward motion or a point turn in order to continue navigation. Figure 1 illustrates the problem: the global planner has computed a path from the endpoints of both paths A and B to the goal. However, if the mobile robot follows path B, it will end up facing an obstacle, unable to move in forward direction.

Graduated fidelity [11] is a technique used to decrease the complexity of motion planning search spaces over long distances to improve runtime performance. This idea can be applied to local planner path sets by elongating the end of each path segment with a straight line (Figure 4). At the point where the local planner path segment connects with the lookahead line, heading is preserved but curvature may be discontinuous. Likewise at the intersection of the lookahead line and the global planner, path position is continuous but heading is allowed to vary instantaneously. This technique gradually reduces the fidelity of paths in the path set to improve the likelihood of finding a safe, traversable path through the environment.

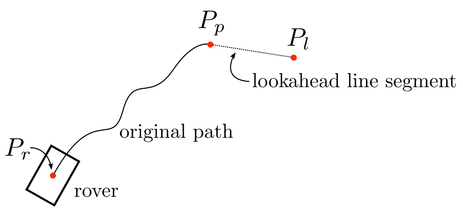


Figure 4. : Lookahead: the original path is elongated by a straight line segment for cost evaluation.

We denote the mobile robot's position by P_r, the endpoint of the actual path to be executed by P_p, the endpoint of the lookahead line segment by P_l and the path between any two of these points as drawn in Figure 4 by $\overline{P_iP_j}$. Given a set of motion alternatives, selection of the optimal one is performed as follows.

- If at least one path is free up to P_l, the path that minimizes $w_l \cdot c_{local}(\overline{P_rP_l}) + c_{global}(P_l)$ is chosen and executed up to P_p.
- If no path is free up to P_l, but there is at least one that is free up to P_p, the path that minimizes $w_l \cdot c_{local}(\overline{P_rP_p}) + c_{global}(P_p)$ is selected, and the mobile robot executes only one half of the path between P_r and P_p before replanning.
- If no path is free up to P_p, a backward motion or a point turn is chosen.

This lookahead technique assures a certain amount of free space in front of the mobile robot, once it has followed the selected path. This is highly beneficial, as it reduces the probability of the mobile robot being unable to move forward due to infeasibility of the global path. Most importantly, situations where it directly faces an obstacle are avoided.

5.2 Random Pose Experiments

In this set of experiments, we analyzed the planners' ability to find collision-free paths given arbitrary obstacle arrangements. To that purpose, we defined a large set of different robot poses, which were superimposed on maps consisting of randomly placed obstacles. For each pose and each planner we recorded whether at least one feasible path could be detected. Figure 5 shows the result of these experiments for arcs and clothoids of length $L = 3$ m in a map containing obstacles with radius $R = 0.8$ m. The plots indicate the percentage of all poses where the different planners were unable to find a feasible path. The experiment consisted of 8784 poses in total, represented on the left in Figure 5. Since the obstacles in the map were randomly distributed, this number includes poses where finding a collision-free path is trivial as well as poses where it is impossible. In these cases, the type of motion alternatives has no influence on the planner's ability to find an admissible path. The more interesting cases are therefore those where at least one, but not all of the planners succeeded or failed, which occured 513 times during the experiment. The chart on the right shows the failure rates regarding only these poses.

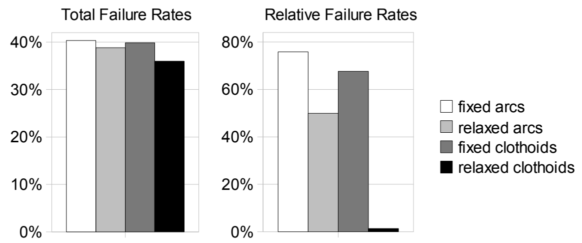


Figure 5. : Results of random pose experiments.

The results in Figure 5 demonstrate the benefits of path relaxation. Using relaxed arcs or clothoids instead of fixed paths improves performance in terms of failure rates. In particular, path sets composed of relaxed clothoids yield significantly better results than fixed arcs, fixed clothoids, or relaxed arcs. The advantage of using more expressive parameterizations for path set motions (fixed clothoids versus fixed arcs) is also evident.

5.3 Navigation Experiments

This section presents a second experiment quantifying the performance of navigators built with different path sets. These experiments simulate a mobile robot navigating to a goal position through an environment composed of randomly placed obstacles. The same set of experiments was undertaken for all four planners, and their performance was compared.

Cost maps with 1000×1000 cells were used, corresponding to a size of 200×200 m, and were populated by randomly distributed Gaussian obstacles of equal radius R. The difficulty of the terrain is determined by the obstacle density, which is defined as the percentage of the area occupied by obstacles. The experiments simulate perfect mobile robot path following as well as perception and localization. However, the perception horizon is limited to a radius of $R_{perc} = 10$ m around the mobile robot's position. For global planning, we use D* in an 8-connected grid, where the states correspond to the cells of the cost map and the cost to go from a cell to one of its neighbors is the weighted sum of the distance and the cost values of the two cells. The local planner evaluates a set of motion alternatives of length L by computing the local cost of each path according to (8). The path to be executed by the mobile robot is chosen according to (1), where global cost is defined as the D* value of the cell hit by the path endpoint. When the mobile robot has completed this motion it stops, the map is updated and the next motion is computed.

Start and goal are set to two opposite map corners. In order to keep the number of parameters affecting navigation performance low, we only allow the mobile robot to move forward and declare navigation as failed, as soon as it encounters a situation where it is unable to find any collision-free forward motion. Our primary measure for the quality of a certain local planner is therefore its failure rate in a series of navigation experiments with different maps. We also measured the average length of the path from the start to the goal. We did not discover any significant differences between the four planners, which reassures us that the proposed method does not have adverse effects with respect to arc-based local planning.

Figure 6 shows the influence of our lookahead technique on mobile robot navigation. These results have been obtained with a planner using fixed arcs of length $L = 3$ m in maps with obstacles of radius $R = 0.8$ m. The experiment included 50 different maps for each of the four obstacle density values. Path selection is performed as explained in Section 5.1.3, using a lookahead distance of 1 m.

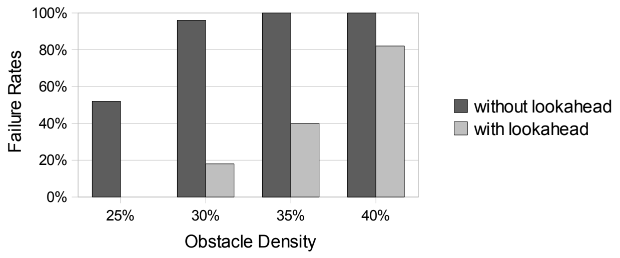


Figure 6. : Navigation using fixed arcs: influence of lookahead

The failure rates indicate the percentage of experiments where the mobile robot did not reach the goal. It can be seen that the probability of failure decreases when the motion planner uses lookahead. In experiments with different parameters (shorter motion alternatives or larger obstacles) lookahead had a similar effect. We conclude that lookahead is beneficial even for a robot that drives at very slow speed and stops after every motion step for replanning. In our experiments, increasing the lookahead distance to 1.5 or 2 m did not further improve the navigator performance. For the subsequent comparison of fixed and relaxed paths, we therefore use a lookahead distance of 1 m.

Figure 7 shows the results of our navigation experiments with the four different local planners. The indicated failure rates are based on 150 individual experiments for each planner and each obstacle radius (0.8 and 1.6 m), featuring three different obstacle densities (30, 35 and 40%, 50 maps each). Note that the only difference between the four planners is the set of available motion alternatives. In the case of fixed arcs and clothoids, this set is the same in every step, whereas in the other two cases the paths are adapted to nearby obstacles using our relaxation algorithms.

The results demonstrate the benefits of path relaxation for navigation. The failure rates are significantly lower when using a set of relaxed motions, especially in the case of clothoids. Furthermore, it can be seen that using a set of fixed clothoids instead of arcs decreases the probability of failure as well. The qualitative distribution of failure rates corresponds to the one of the random pose experiments (cf. Figure 5), which supports the initial hypothesis.

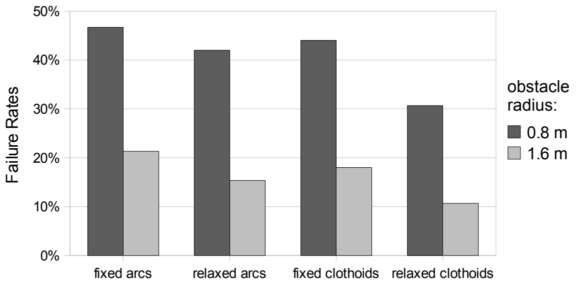


Figure 7. : Results of simulated navigation experiments for path sets composed of fixed arcs, relaxed arcs, fixed clothoids, and relaxed clothoids. Each bar is based on 150 experiments and indicates the percentage of cases where the goal has not been reached.

The presented results have been obtained using motion alternatives of length $L = 3$ m. We repeated the experiments with shorter paths ($L = 0.5$ m), which increased the failure rates due to short-sightedness of the planner. Relaxation was not able to improve the performance of the short actions. This can be explained by the fact that for $L = 0.5$ m a set of 23 fixed paths already provides a very dense sampling of the action space.

The cost of relaxation in terms of runtime mainly depends on the resolution Δs for path cost computation, as introduced in (8). Using small steps between cost measurement points on a path yields accurate results and helps avoiding relaxation onto obstacles, but increases the computational effort. The results presented above have been obtained using a resolution of $\Delta s = 1$ cm. As expected, each relaxation-based navigator required more time to evaluate the path sets. Compared to fixed arcs, runtime of the local planner in each step increased by approximately a factor 10 for arc relaxation and a factor 70 for clothoid relaxation. In experiments with reduced resolution of $\Delta s = 15$ cm, these factors could be reduced to around 1.5 and 7, respectively. Thereby, the results of relaxed clothoids were almost identical to the ones obtained with high resolution, and relaxed arcs performed only slightly worse than with more accurate relaxation.

6 Conclusion

This paper explored path set relaxation and parameterizations for mobile robot navigation. The experimental results demonstrate that relaxation improves performance of the navigator but also increase the computational complexity. In applications where safety is more important than speed, path set relaxation can be an effective approach to find the most efficient and minimum-risk routes through an environment. Path set relaxation can also be employed as it is needed, particularly when the fixed path set cannot identify a single safe route in the local planner. Furthermore, we have shown that a planner using a path set composed of fixed clothoids can outperform a planner with the same number of arcs. Finally, we have presented a planner lookahead technique for local planners based on the principles of graduated fidelity. The experiments have shown that path set lookahead is a beneficial step between local and global planners in hierarchical navigator architectures when global planners violate mobility constraints of the platform.

7 Future Work

On-demand or iterative path set relaxation is an important next step to decrease the computational requirements of the presented techniques. Path set relaxation might be a useful mode before human intervention where it can try to find a possible route through the environment. Additional quantitative experiments with different environments and path sets measuring the runtime, memory, and performance of the different navigators would provide better insight on the tradeoff between path set quality and real-time performance.

Another interesting area of research involves applying relaxation to more complex primitives than clothoids and other path set parameterizations. In particular these relaxation techniques could apply to path sets generated by sampling in the terminal state space of motions [3]. Rather than bias the sampling based on the global planner cost alone, each motion in the path set could be optimized based on the combined local/global path cost.

8 Acknowledgments

This research was conducted at the Robotics Institute of Carnegie Mellon University under contract to NASA/JPL as part of the Strategic University Research Partnership (SURP) program. The views and conclusions contained in this document are ours and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

References

[1] O. Brock and O. Khatib. Elastic strips: A framework for motion generation in human environments. *International Journal of Robotics Research*, 21(12):1031–1052, December 2002.

[2] M. Daily, J. Harris, D. Keirsey, D. Olin, D. Payton, K. Reiser, J. Rosenblatt, D. Tseng, and V. Wong. Autonomous cross-country navigation with the ALV. In *Proc. IEEE International Conference on Robotics and Automation*, pages 718–726 vol.2, 1988.

[3] T.M. Howard, C. Green, D. Ferguson, and A. Kelly. State space sampling of feasible motions for high performance mobile robot navigation in complex environments. *Journal of Field Robotics*, 25(6-7):325–345, June-July 2008.

[4] T.M. Howard, C. Green, and A. Kelly. Receding horizon model-predictive control for mobile robot navigation of intricate paths. In *Proceedings of the 7th International Conference on Field and Service Robotics*, July 2009.

[5] A. Kelly. *An Intelligent Predictive Control Approach to the High-Speed Cross-Country Autonomous Navigation Problem.* PhD thesis, Carnegie Mellon University, September 1995.

[6] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, P. Rander, S. Thayer, N. Vallidis, and R. Warner. Toward reliable off road autonomous vehicles operating in challenging environments. *Int. J. Rob. Res.*, 25(5-6):449–483, 2006.

[7] F. Lamiraux and J.-P. Laumond. Smooth motion planning for car-like vehicles. *IEEE Transactions on Robotics and Automation*, 17(4), 2001.

[8] J.-P. Laumond. Trajectories for mobile robots with kinematic and environment constraints. In *Proc. of International Conference on Intelligent Autonomous Systems*, page 346354, 1986.

[9] V.J. Lumelsky and A.A. Stepanov. Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. *Algorithmica*, 2:403–430, 1987.

[10] H. Moravec. *Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover.* PhD thesis, Stanford University, March 1980.

[11] M. Pivtoraiko and A. Kelly. Differentially constrained motion replanning using state lattices with graduated fidelity. In *Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems.*

[12] S. Quinlan. *Real-time modification of collision-free paths.* PhD thesis, Stanford University, December 1994.

[13] N. Ratliff, M. Zucker, J.A. Bagnell, and S. Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *ICRA*, 2009.

[14] M.W. Maimone S.B. Goldberg and L. Matthies. Stereo vision and rover navigation software for planetary exploration. In *IEEE Aerospace*, 2002.

[15] R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadlda, S. Koenig, G. Krishnaswamy, Y. Shinoda, W. Whittaker, and P. Klarer. Experience with rover navigation for lunar-like terrains. In *IROS*, 1995.

[16] A. Stentz and M. Hebert. A complete navigation system for goal acquisition in unknown environments. In rd, editor, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots'*, volume 1, pages 425–432, August 5–9, 1995.

[17] Anthony Stentz. The focussed D* algorithm for real-time replanning. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, August 1995.