Reactive control in environments with hard and soft hazards

Reactive control in environments with hard and soft hazards

Sisir Karumanchi, Karl Iagnemma

***Abstract*— In this paper we present a generalization of reactive obstacle avoidance algorithms for mobile robots operating among soft hazards such as off-road slopes and deformable terrain. A new hazard avoidance scheme generalizes constraint based reactive algorithms [1], [2] from hard to soft hazards. Reactive controllers operate by directly parameterizing the closed-loop dynamics of the system with respect to the environment the robot is operating in. Traditionally, reactive controllers are parameterized by weighting virtual attraction and repulsion forces from goals and obstacles [3], [4]. One pitfall of such parameterizations is sensitivity of the tuning parameters to the operating environment. A reactive controller tuned in one set of conditions is not applicable in another (e.g. a different density of obstacles). The algorithm presented in this paper has two key properties which are significant i) Parameterization is environment independent. ii) It can deal with non-binary environments that contain soft hazards.**

I. Introduction

Despite many advances in deliberative planning algorithms, reactive controllers have the ability to quickly generate paths that reach a goal and satisfy kinodynamic constraints. The resulting paths are sub-optimal and there is no guarantee of finding a solution as the controller can get stuck in local minima. However, reactive controllers can still play an critical role in mobile robot decision making. A hierarchy of a reactive layer within a deliberative layer can ensure robustness by allowing the system to respond to changes in the environment that were not taken into account in the deliberative layer. This view can be supported by multiple architectures in the Urban challenge that used a local reactive layer that reasoned from a pre-defined selection of trajectories (tentacles) [5], [6]. In complex off-road terrain with soft hazards (beyond binary obstacle/non-obstacle classification; e.g. terrain slopes, deformable terrain), such pre-defined set of trajectories can benefit from adapting to the terrain conditions in real-time [7]. Optimization based trajectory adaptation can be time consuming [7], and hence forward simulations from a reactive controller can serve as an alternative. A formal definition of soft hazards is given below:

Definition 1: Soft hazards are environment conditions that impose additional differential constraints to the vehicle beyond those arising from its dynamics.

Soft hazards require the vehicle to adapt its behaviour to successfully negotiate them. For example, a mobile robot needs to slow down and change operating gear while going down hill to prevent the risk of excessive slippage and loss of control.

The following list mentions some scenarios where reactive controllers can be of value in generalizing decision making algorithms to non-binary environments with soft hazards:

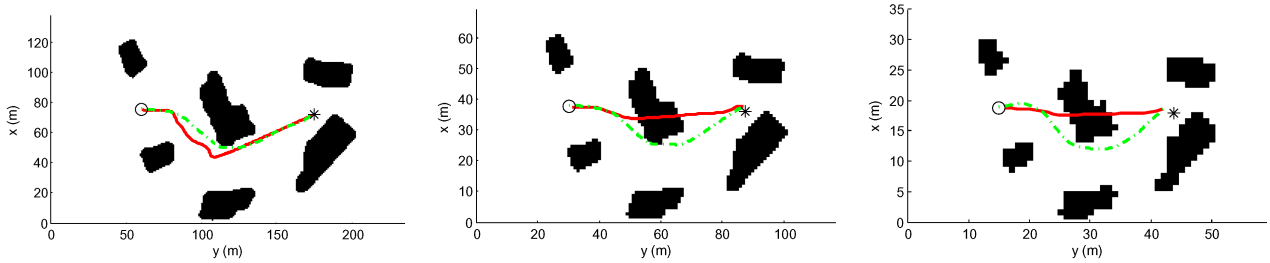
- Fast hazard avoidance that guarantees *motion safety* in hard and soft hazards.
- Fast forward predictions of closed loop behavior to create *terrain–adaptive tentacles* [5].
- Fast *visibility check for non-binary environments* in multi-query planners such as Probabilistic Road Maps (PRMs).
- Fast trajectory generation to be used as edges for planning in a *terrain adaptive state lattice*.

Traditional reactive controllers operate by directly parameterizing the closed-loop dynamics of the system with respect to the environment the robot is operating in. Trajectories to a fixed goal are generated from forward simulation of the closed-loop dynamics. Traditional parameterization occurs by weighting virtual attraction and repulsion forces from the goal and obstacles [3], [4]. One major pitfall of such parameterizations is sensitivity of tuning parameters to the operating environment. A reactive controller tuned in one set of conditions is not applicable in conditions with a different topological arrangement of hazards (e.g. a different density of obstacles) (see Figure 1).

There exist a different class of reactive algorithms (Vector Field Histogram (VFH) [1], Dynamic Window Approach (DWA) [2]) that avoid topology sensitive parameters by representing the world in terms of constraints. Instead of parameterizing summation of attraction and repulsion forces, these algorithms use desired control set-points as parameters. The problem is posed as search for an optimal set-point instead of a search for optimal attraction and repulsion weights.

Optimal set-points are chosen from strictly feasible trajectory/control space based on a given objective (minimum cost/distance/time). Given constraints imposed by the environment onto the vehicle, the feasible space is determined by filtering out infeasible trajectories or control values. By propagating the set-points through vehicle motion equations, kinodynamic constraints are taken into account by construction. Such methods have no environment-specific tuning and are well-posed as a constrained optimization problem. This paper presents a generalized reactive controller that retains all the properties of VFH and DWA while seamlessly dealing with hard and soft hazards. This algorithm is referred to as *Generalized Hazard Negotiation (GHN)*.

Sisir Karumanchi and Karl Iagnemma are with the Robotic Mobility Group, Laboratory for Manufacturing and Productivity, Massachusetts Institute of Technology, USA. {sisir/kdi}@mit.edu



(a) Original scale (b) Half scale (c) Quarter scale

Fig. 1. Reactive controllers operating in the same test environment at three different scales. Solid red trajectory is the result of an attraction-repulsion reactive controller and the broken green trajectory is from the constraint-based reactive controller presented in this paper ($\circ$ indicates start location and $*$ is goal). When tuned at the original scale the former algorithm fails to successfully operate at other scales (1(b) and 1(c)). On the other hand, the constraint-based controller successfully avoids obstacles in all three scales as the tuning parameters are environment invariant.

Unlike VFH and DWA algorithms that were aimed at hard hazards, the GHN algorithm presented in this paper can reason about soft hazards. This is achieved by relying on a mobility space representation of the environment. Similar to velocity space reasoning in DWA, the GHN algorithm reasons in the space of velocity *limits*. Hard and soft hazards are represented seamlessly as different degrees of velocity constraints (speed limits). A recently developed morphological operation known as *mobility erosion* [8] forms an essential ingredient in determining topologically consistent velocity constraints. Erosion ensures motion safety for hard and soft hazards by taking into account vehicle size, reachability due to momentum, actuator limits, system latency and position uncertainty.

This paper is organized as follows. Section II discusses the different components of the GHN methodology and outlines it properties and limitations. In Section III, the utility of the proposed algorithm is analyzed in terms of path improvement in environments with soft hazards. Finally, concluding remarks are presented in Section IV.

II. Methodology

The methodology presented in this paper has two stages; First, state space constraints imposed by the environment are determined via sensor data analysis. Second, the constraints are used to choose desired control set-points, which are then propagated through low level controllers. The commands from the controller can be sent directly to the vehicle or to a vehicle motion model to generate closed loop simulations. The state space constraints are first determined from a mobility function and then processed with the mobility erosion operator. Given the eroded constraints, the chosen set-points are guaranteed to be kinodynamically feasible and safe. However the resulting trajectories are not guaranteed to be optimal and there is no guarantee of finding a solution.

A brief description of mobility erosion is given in the next subsection followed by discussion of the reactive controller.

A. Mobility Erosion

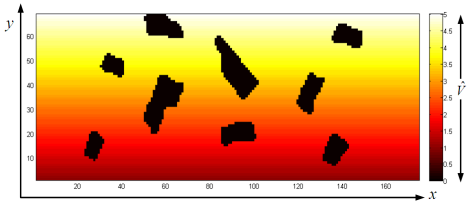
Planning algorithms and reactive controllers often consider the vehicle to be a point mass in the configuration space. In order to account for the shape and size of the vehicle in the physical space, traditional path planning uses the *Minkowski sum* operator [9] to grow binary obstacles. These *enlarged* obstacles enable the decision making algorithms to ignore vehicle size and treat it as a point mass. However, the Minkowski sum is only applicable for hard hazards. The mobility erosion operation generalizes obstacle growing to soft hazards by taking into account both vehicle size as well as momentum.

Motion safety for hard hazards implies that position constraints need to be enforced during state transitions. Similarly, motion safety for soft hazards requires that proprioceptive constraints are adhered to (bounds on maximum slip, skid, minimum traction coefficients etc.). In order to adjudge soft hazard safety in state space, the fixed constraints in proprioception space are transformed into state space (position, heading, speed and acceleration) using an inverse model which is learned offline [8], [10]. The transformed proprioceptive constraints are captured in an instantaneous mobility function which forms an input into the erosion operation. The mobility function thus associates exteroceptive terrain features with speed limits.

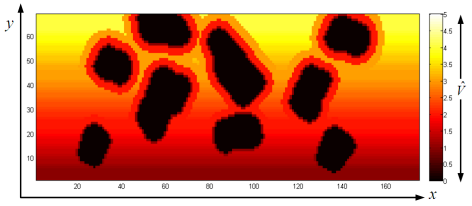
In this paper, we assume that an instantaneous mobility function is given *a priori*. In the case of hard hazards, the function would be a discrete 0-1 loss function (0 for a zero speed limit and 1 for maximum speed limit). More complex functions can be learned from experimental data either given expert demonstrations or from proprioceptive feedback as demonstrated in [10]. The complexity of the mobility function seamlessly generalizes motion safety from hard to soft hazards.

Mobility erosion is a *maxmin* formulation as shown in Equation (1). It searches for the mobility value (m) that maximizes the *minimum mobility in the neighborhood* without exceeding the value given by the instantaneous mobility function. Erosion is a local operation and fits in with the computational requirements for a reactive controller.

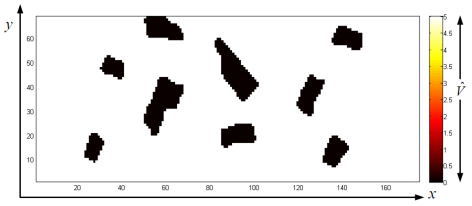
Equation (1) is a 'convolution like' operation performed on two sets i) a set of states arranged in a topology (S) that represents the environment (task space) and ii) a set of offsets in the topology called the structuring element (SE) that specifies the neighborhood to be consider when processing the environment (S) in a state by state manner. Initially, each



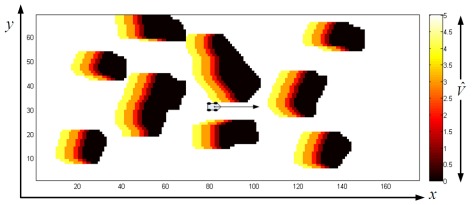
(a) Instantaneous mobility with hard and soft hazards



(b) Erosion on 2(a) using a circular structuring element



(c) Instantaneous Mobility (units: m/s)



(d) Erosion in $\{x, y, \theta\} : \theta = 0^\circ$

Fig. 2. (b) shows isotropic mobility erosion on an topology with hard and soft hazards (shown in (a)) using a circular structuring element. (d) shows the result of anisotropic mobility erosion in $\{x, y, theta\}$ using a rectangular structuring element for a sample orientation (the assumed vehicle heading is indicated by an arrow). (Units: x/y-m; Color scale: white = $5m/s$ and black = $0m/s$)

state s in the topology (S) is associated with a speed limit using the given mobility function based on its exteroceptive properties (e) (e.g. terrain color) $\{I(s) = f(e(s))\}$. Equation (1) is then used to shrink (erode away) high speed areas due to their proximity to low speed areas. The extent of erosion is specified by the size and shape of the structuring element (SE) and the mobility values in the neighborhood.

$$E(s) = \max_{m \in [0, I(s)]} \left\{ \min_{s' \in \{SE(m)\}} \{m, I(s + s')\} \right\} \qquad (1)$$

The size and shape of the structuring element is variable and is given by the vehicle size plus worst-case stopping distance. More formally the structuring element represents the minimal *reachability* space of the vehicle for a look-ahead time given by the worst case stopping distance. The worst-case stopping distance changes according to different environmental conditions. Given an initial estimate of maximum speed limit defined for that environmental condition (instantaneous mobility value) the worst-case stopping distance can be determined. This makes the structuring element a function of the maximum speed limit defined for a given environmental condition, e.g. for a point vehicle with initial speed limit v, assuming second-order dynamics the worst-case stopping distance is given as $\frac{v^2}{2a_{\max}}$, where $a_{\max}$ is the maximum possible deceleration that can be achieved in the given terrain. Reachability is also affected by system latency (δ) which adds an increment to the stopping distance ($v \times \delta$). In addition to reachability, localization uncertainty can be taken into account in the structuring element by dilating it with the 2-sigma uncertainty ellipse .

Figures 2(b) shows a sample application of isotropic mobility erosion with a circular structuring element in a topology with discrete obstacles and varying mobility conditions. Isotropic erosion ignores vehicle heading. Further resolution in orientation can be obtained with anisotropic erosion using a rectangular structuring element. If the structuring element is orientation sensitive, one can generate a set of mobility maps for different values of orientation. Figure 2(d) shows sample result of anisotropic erosion on a topology with hard hazards shown in Figure 2(c) using a rectangular structuring element. Figure 2(d) is one slice from a set of eight orientations from the 3D space of position (x,y) and heading. Orientation sensitivity is a key requirement for choosing a valid orientation set-point for the reactive controller presented in this work .

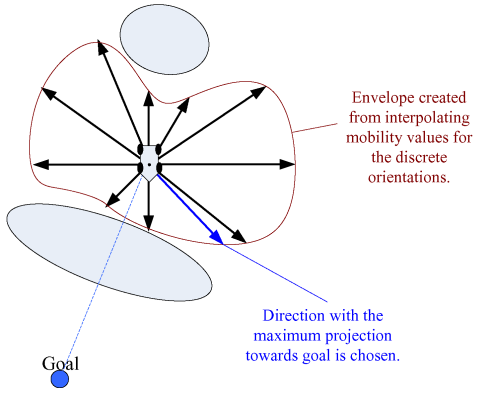


Fig. 3. Choosing a desired orientation: A mobility envelope is created (red) by fitting the discrete mobility values (speed limits) for different orientations with a spline. The black vectors indicate magnitude and orientation of the velocity limits determined from erosion. The direction with maximum mobility projection to goal is choosen.

B. Generalized Hazard Negotiation (GHN)

As presented in the previous section, constraints imposed by the environment in position and orientation can be represented as a set of orientation specific speed limits derived from the erosion operator. The erosion operation is local and can be applied to the neighborhood within the sensor range in real-time. Given the speed limits, the GHN algorithm then determines heading and speed set-points via Equation 2.

$$\begin{aligned} \theta^* &= \arg\max_{\theta} \{M(\theta)\cos(\Delta\theta_{goal})\} \\ v^* &= M(\theta_c) \end{aligned} \quad (2)$$

Where $M(.)$ is the mobility envelope, $\Delta\theta_{goal}$ is the heading change to goal and θ_c is the current vehicle heading.

The set-point selection process given in Equation 2 is as follows:

1) At any given position the current set of mobility values for all orientations are used to create an mobility envelope ($M(.)$; see Figure 3).
2) An orientation with the maximum mobility projection to the goal is chosen as the desired orientation set-point (θ^*). This is a greedy decision that chooses the direction with the best possible distance gain to goal.
3) The mobility value (speed limit) for the current vehicle orientation is chosen as the desired speed set-point (v^*). The eroded speed limit represents the maximum possible velocity which is considered safe.

The mobility envelope and the projection metric used to choose the orientation set-point are environment independent and can seamlessly generalize between hard and soft hazards.

Equation 2 is a one-step greedy decision and does not take the full sensor range into account (as the mobility envelope from erosion only considers the minimal reachable space which is usually less than the sensing range). The complexity growth of using Equation 2 to simulate trajectories is linear in the distance to the goal ($O(d_{goal})$). Significant improvement can be achieved with look ahead by performing multiple forward simulations with Equation 2 towards a discrete set of sub-goals (SG) and choosing the time-optimal sub-goal to reach the goal. The increased look ahead provides better anticipation to avoid local minima and hazards, however the controller can still get stuck in local minima.

The look ahead process effectively generates local trajectories (tentacles) for the vehicle to choose at every time step. The sub-goal trajectory with the least time estimate to goal is chosen and only the first control input is sent to the vehicle (akin to receding horizon control). This requirement for multiple forward simulations increases computational complexity. The complexity now grows at a rate of $O(k \times L \times d_{goal})$ where L is the look ahead and k is the number of sub-goals. The dependence of complexity growth on look ahead can be controlled by sampling sub-goals at a fixed ratio of the look ahead distance ($w \times L$) instead of sampling at every time step. This reduces the complexity down to $O(\frac{k \times d_{goal}}{w})$.

The sub-goal and set-point selection process is represented by Equation 3.

$$\begin{aligned} s^* &= \arg\max_{s \in SG} \left\{ t_{c\_to\_s}(c,s) + \hat{t}_{s\_to\_g}(s,g) \right\} \\ \theta^* &= \arg\max_{\theta} \{M(\theta)\cos(\Delta\theta_s)\} \\ v^* &= M(\theta_c) \end{aligned} \quad (3)$$

Where s^* is the desired sub-goal set-point, c is the current position, g is goal, SG is a discrete set of sub-goals ($\{s_1, s_2, \cdots, s_n\}$), and $\Delta\theta_s$ is the heading change to sub-goal. The expected time to sub-goal ($t_{c\_to\_s}$) for a given trajectory from the current location (c) can be estimated by dividing the distance traveled with their respective mobility values. Similarly, the time estimate ($\hat{t}_{s\_to\_g}$) from the sub-goal (s) to the main goal (g) can be determined with forward simulation by assuming the region outside the sensing range has maximum mobility (maximum speed limit). The latter is akin to the usage of optimistic heuristics in A*/D* planners.

The sub-goals are sampled at the edge of the sensing range or at the distance to goal (which ever is smaller). Any sub-goal sampling pattern can be used as long as i) they are within the sensing range ii) moving to the sub-goal makes a positive distance gain to the main goal. The second requirement ensures that the distance to goal decreases over time and ensures stability by preventing oscillations. If no viable sub-goal exists the controller terminates. Note that the positive distance gain ensures stability in the Lyapunov sense but not asymptotic stability [9]. In the event where the goal is within the sensing range, the sub-goals are sampled with the goal distance as the radius. The controller can only reach the goal approximately; the forward simulation is terminated when a the simulation is within a small neighborhood of the desired goal.

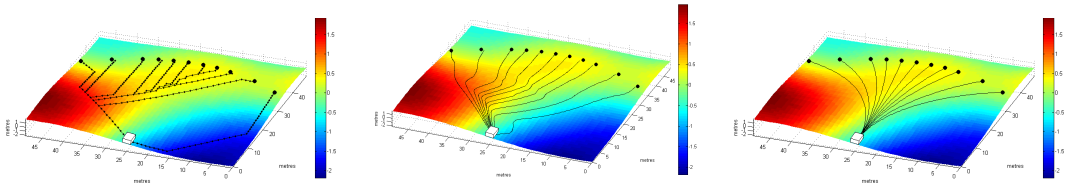
The controller will fail to find a trajectory when all the sub-goals choices lead to a local minima or are infeasible. In general, this happens when the profile/size of the obstacles is greater than its sensing horizon. Such a scenario can occur when the vehicle is stuck in non-convex obstacles (mazes, bug trap problem) where no sub-goals exist that make a steady positive distance gain to the goal. The latter is a non-minimum phase problem[1] which cannot be solved with reactive controllers as they have no sense of history. As a result, reactive controllers are most appropriate as a low-level subsystem inside a higher level planner and not as an isolated system.

In summary, the GHN algorithm aims to reactively reduce the distance to the goal while conforming to the environment imposed constraints. Instability is avoided by restricting the movement to ensure a net decrease in distance to goal over time. This is done with two rules i) only sub-goals with a positive distance gain to goal are considered ii) The mobility projection metric in turn ensures that orientations with a positive distance gain to a sub-goal are chosen as set-points. Finally, the derived set-points from Equation 3 are propagated through low level steering and speed PID controllers and a vehicle motion model to ensure kinodynamic feasibility of the generated trajectory by construction. The only tuning parameters in the reactive controller are in the low level PID controllers and they are independent of the environment.

III. Results

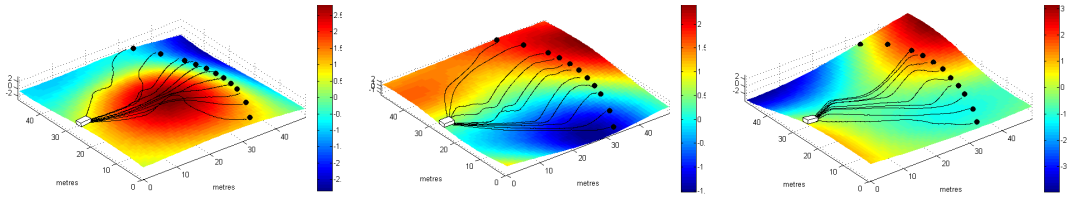
In this section, simulation results are presented that compare the performance (average velocity and time-to-goal) of

[1] In a non-minimum phase problem, the distance to goal has to increase before decreasing.



(a) A* paths (b) GHN trajectories (c) Constant curvature trajectories

Fig. 4. Experimental Context 1: Trajectory generation for a fixed set of goals in a sample terrain surface (50m × 50m). The different trajectories to a fixed set of goals were used to compare the performance of GHN algorithm against optimal path planning using A* and fixed constant curvature trajectories that were determined geometrically. The white object represents the vehicle bounding box (3m × 2m × 1.5m) and it shows the scale of the environment. The color axis denotes elevation and is in meters.



(a) (b) (c)

Fig. 5. Experimental Context 2: GHN trajectories for a subset of randomly sampled terrain surfaces (elevation maps). The terrain surfaces were sampled from a Gaussian process with a squared exponential covariance function [11]. Note that, no trajectories were generated to some goals in (c) where GHN trajectory generation failed (the direct positive distance gain path to these goals involved steep slopes with zero mobility). The color axis denotes elevation and is in meters.

terrain adaptive local trajectories using the GHN algorithm against optimal path planning using A*[2] and sub-optimal constant curvature trajectories. The results demonstrate the utility of reactive controllers in generating terrain adaptive trajectories that on average are an improvement over constant curvature trajectories in terms of time to goal and average velocity. Since velocity is varied according to the environment specific speed limits (from mobility maps), a higher average velocity indicates that a route with higher mobility was chosen on average. Trajectories were generated for a fixed set of goals (11 goals that are 40m away from a fixed starting position) in a environment with varying slopes as a means to represent soft hazards. A look-ahead of 30m was used in the GHN algorithm as it represents the nominal sensing range for elevation assessment from scanning LIDAR sensors. This experimental context for a sample environment is shown in Figure 4. The trajectory generation using the three different techniques was repeated for 500 randomly generated 50m×50m terrain surfaces. A sample set of these terrain surfaces and their corresponding GHN trajectories are shown in Figure 5.

The terrain surfaces (elevation maps) were sampled from a Gaussian process with a squared exponential covariance function [11]. The hyperparameters (signal variance, length scale) of the covariance function were in turn sampled from one dimensional Gaussian distributions. The parameters of the Gaussian distributions[3] were chosen empirically so that the slopes of the sampled environment sufficiently covers the full mobility spectrum of typical ground vehicles (± 20 degrees in pitch and ± 15 degrees in roll).

Mobility maps were determined from the randomly generated elevation maps in three stages. First, pitch and roll slopes for eight different vehicle headings were determined using image processing based gradient filters. Second, a priori chosen mobility function was used to associate pitch and roll with an instantaneous speed limit. Third, the mobility erosion operation is performed on the set of instantaneous speed limits which form the input to the GHN algorithm, and the A* planner (as an negative cost map). It was shown in [10] that the mobility characteristics of unmanned ground vehicles in two dimensional slopes closely resembles a band pass filter in pitch and roll space. Therefore, a two dimensional butterworth function[4] in the space of pitch and roll was used as the mobility function[5].

TABLE I

COMPARISON OF A*, GHN AND THE FIXED CONSTANT CURVATURE TRAJECTORIES IN 500 RANDOMLY SAMPLED ENVIRONMENTS.

	A*	GHN	Fixed
avg. time to goal	16.24s	22.62s	31.19s
avg. velocity	3.41m/s	2.69m/s	2.08m/s
trajectory generation failure	0	8.55%	0

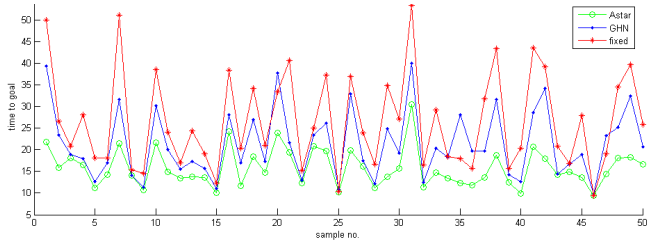
Forward simulations as part of the GHN algorithm were performed through a closed loop system of low-level controllers (steering and speed) and a vehicle motion model. An

[2]The A* planner uses the mobility maps as a negative cost representation to perform a grid search in x, y, θ. The paths are optimal in time-to-goal and average velocity since cumulative mobility is maximized. However, the grid search does not produce kinodynamically feasible paths. The A* results serve as a benchmark to assess the sub-optimality of the reactive controllers.

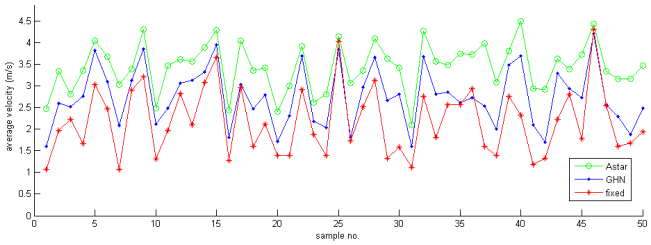
[3]Signal variance was sampled with mean = 2 and standard deviation=1; and the length scale was sampled with mean=25 and standard deviation=2.

[4]$G(\omega) = \sqrt{\frac{1}{1+\omega^{2n}}}; \omega = \sqrt{\left(\frac{pitch}{p\_cutoff}\right)^2 + \left(\frac{roll}{r\_cutoff}\right)^2}$

[5]The following parameters were used: $p\_cutoff = 8; r\_cutoff = 4; n = 2;$. The peak value of the mobility function was set to 5 m/s.



(a) Average time to goal performance of A*, GHN and constant curvature trajectories for 50 environments



(b) Average velocity performance of A*, GHN and constant curvature trajectories for 50 environments

Fig. 6. Average time to goal and velocity plots for a subset (50) of the 500 randomly sampled environments. x-axis is environment ID and y-axis indicates the performance metric. It can be seen that the blue trends (GHN) mostly lies between the red (constant curvature trajectories) and green trends (A* trajectories).

empirically tuned steering dynamics model presented in [4] for a four wheeled Ackermann steered all-terrain vehicle is used for simulations. The model parameterizes the steering dynamics as a second order system[6] with an actuator time delay of $0.2s$. Similarly, a proportional speed controller (gain $= 10$) and second-order dynamics were used to simulate velocity regulation. Finally, a kinematic motion model for a steered vehicle was used to couple the velocity and steering dynamics [12].

Table I shows the comparison between optimal A* path planning, adaptive trajectories using GHN and fixed constant curvature trajectories. The table shows average of time to goal and velocity taken over all 500 environments. In each environment, 11 trajectories were generated to a set of goals. Figure 6 plots the average time to goal and average velocity trends for the first 50 of these environments.

It can be seen that the blue trend (GHN) in Figure 6 mostly lies between the red (const. curvature) and green trends (A*). This is evident for both average time to goal and average velocity. Optimal path planning using A* (green trend) provides the best performance both in terms of achieving a low time to goal and high average velocity. The adaptive GHN based trajectories (blue trend) demonstrate an improvement over fixed constant curvature trajectories (red trend), however they are not as good as the optimal paths. On average over all 500 environments GHN results in Table I demonstrate a 20% improvement in time to goal and 18% improvement in average velocity with respect to A* performance[7] over the fixed trajectories. However, GHN had a 8% trajectory generation failure (out of 11×500 attempts) due to local minima[8]. This can also be seen in Figure 5(c), where no trajectories were generated to some goals.

IV. Conclusions and Future Work

In conclusion, the GHN algorithm presented in this paper has three key properties which makes it novel: i) Environment assessment is decoupled from the parameterized closed-loop dynamics which results in environment-invariant parameters ii) Constraint based parameterization ensures motion safety for hard and soft hazards and generates kinodynamically feasible paths by construction iii) Finally, it is a local operation and has low computational complexity. However, the algorithm is sub-optimal and is best used in short term decision making.

The algorithm presented in this paper was a basic version and it assumed that the goal was specified in position space alone. No orientation constraint was specified at the goal. However, it is possible to extend the definition of goals to position and orientation by introducing inverse reachability constraints as ghost obstacles into the erosion framework.

Acknowledgment

This material is based upon work supported by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number W911NF-11-C-0101.

References

[1] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *Robotics and Automation, IEEE Transactions on*, vol. 7, no. 3, pp. 278–288, 1991.

[2] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[3] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 19, no. 5, pp. 1179–1187, 1989.

[4] B. Hamner, S. Singh, and S. Scherer, "Learning obstacle avoidance parameters from operator behavior," *Journal of Field Robotics*, vol. 23, no. 11-12, pp. 1037–1058, 2006.

[5] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *Journal of Field Robotics*, vol. 25, no. 9, pp. 640–673, 2008.

[6] J. Leonard *et al.*, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, vol. 25, no. 10, pp. 727–774, 2008.

[7] T. Howard and A. Kelly, "Terrain-adaptive generation of optimal continuous trajectories for mobile robots," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space 2005*, 2005.

[8] S. Karumanchi, "Off-road mobility analysis from proprioceptive feedback," Ph.D Thesis, The University of Sydney, 2010.

[9] S. LaValle, *Planning algorithms*. Cambridge Univ Pr, 2006.

[10] S. Karumanchi, T. Allen, T. Bailey, and S. Scheding, "Non-parametric learning to aid path planning over slopes," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 997–1018, 2010.

[11] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning* . The MIT Press, 2005.

[12] A. Kelly *et al.*, "Toward Reliable Off Road Autonomous Vehicles Operating in Challenging Environments," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 449–483, May 2006.

[6] Transfer function - $\left(1/258.7s^2 + 6.5789s + 1\right)$

[7] Improvement is assessed as percentage gain in the ratio of GHN and fixed trajectories performance metric's against A*'s performance metric.

[8] Since a set of goals are attempted, 8% failure does not mean that the vehicle gets stuck 8% of the time. The chance of trajectory generation failing to all goals is rare.