Adaptive Obstacle Representations for Dynamical Navigation

Eric Aaron
Dept. of Mathematics & Computer Science
Wesleyan University
eaaron@wesleyan.edu

Juan Pablo Mendoza
Robotics Institute
Carnegie Mellon University
jpmendoza@ri.cmu.edu

Foster Nichols
Dept. of Mathematics & Computer Science
Wesleyan University
fnichols@wesleyan.edu

Abstract

This paper suggests and supports a design idea for improving dynamical navigation: adding an intermediary, adaptive obstacle representation level between perception and repeller representations. We illustrate our idea with our specific example of an adaptive obstacle representation level, which cleanly integrates into multiple existing navigation systems, treating each perceived obstacle entity as a locally sensitive, obstacle-valued function that returns an obstacle representation upon which steering and obstacle avoidance are based. Moreover, other elements of the navigation systems remain unaltered, thus preserving and extending original design virtues such as behavioral flexibility, computational efficiency, and dynamic responsiveness. Extensive simulations, validated with tests of real robots, demonstrate that our new representations compare favorably to previously employed representations on measures of effectiveness within a tested scenario, robustness over varying scenarios and ranges of parameter values, and computational efficiency.

Introduction

This paper presents and supports a novel design idea for improving dynamical systems-based robot navigation: adding an intermediary level of data representation, an adaptive obstacle representation layer separate from perception and the repeller representations upon which steering and obstacle avoidance are based. In particular, we present a geometrically sensitive *dynamic tangent* obstacle representation that exploits the reactivity inherent in dynamical navigation, treating perceived obstacle entities as *obstacle-valued functions* for steering calculations. Although our adaptive obstacle representation does not address all of the difficulties noted with such force-based navigation (Koren and Borenstein 1991), we demonstrate that for a family of dynamical navigation methods, it can significantly improve the effectiveness of navigation in a given scenario, the robustness of navigation over varying scenarios and ranges of parameter values, and computational efficiency. Moreover, it cleanly integrates into existing systems without requiring additional changes, thus preserving design strengths such as behavioral flexibility, computational efficiency, and responsiveness in dynamic environments.

In conventional dynamical navigation, angular repellers and attractors steer robots away from obstacles and toward targets. The underlying mathematics (e.g., in (Schöner, Dose, and Engels 1995; Large, Christensen, and Bajcsy 1999)), however, requires that the mathematical repeller representations be composed only of circles, regardless of the shapes of actual *obstacle entities*. (We consider only 2-dimensional navigation in this paper.) Many environments, though, contain both circular and non-circular obstacles, such as the office hallways navigated by a service robot. Some entities in the hallways (e.g., people, carts) may have roughly circular xy-projections and thus be naturally represented by the angular repulsion in conventional dynamical navigation (Figure 1). Non-circular obstacles such as hallway walls, however, can be more difficult to represent.

Some previous approaches to navigation in such environments have involved careful tuning of parameter values (see (Schöner, Dose, and Engels 1995; Large, Christensen, and Bajcsy 1999) for more information) and approximating obstacle entities with circles: walls were covered by small circles derived from discrete sampling of the walls; polygonal obstacles were approximated by collections of circles; etc. With too many small circles in approximations, however, steering calculations are inefficient. In addition, with circles of varying sizes or other geometric heterogeneity in the environment, robustness over parameter settings can fail: The parameters for effective navigation in one part of a space, for example, may be ineffective elsewhere, preventing success on complex tasks.

Our dynamic tangent (*DT*, for short) representations address these difficulties by improving geometric sensitivity, adding an adaptive layer of abstraction between obstacle entities and repeller representations: For each robot R, at each navigation timestep, each obstacle entity is treated as an obstacle-valued function, which returns a single-circle obstacle representation (retaining the connection to conventional approaches, in contrast to (Aaron and Mendoza 2011)). As an example, a wall may be represented by a circle touching the wall at the closest point to R on the wall (point p_m in Figure 2); the size and position of the circle vary with R during navigation, supporting obstacle avoidance. In this paper, after summarizing the conventional dynamical navigation framework that our DT representations enhance, we describe the construction of DT representa-

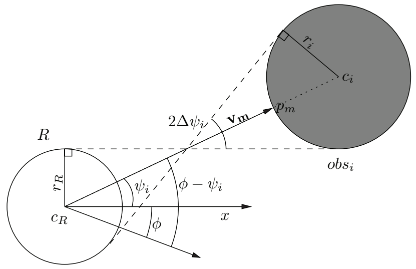

Figure 1: Obstacle avoidance in dynamical navigation, with robot R, obstacle obs_i, and other points, angles, and lines as labeled. When robot heading angle ϕ is not outside of the range delimited by the dotted lines—that is, when some point on R is not headed outside of every point on obs_i—R is steered outside of that range, avoiding a collision.

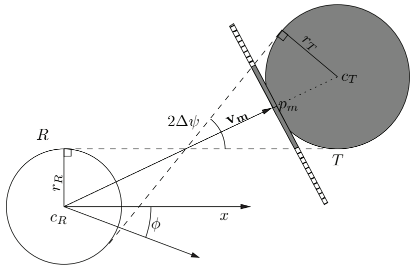

Figure 2: An example *dynamic tangent* wall representation, with robot R and obstacle representation T of the wall. By varying the size and location of T as R navigates—but with T always centered around $\mathbf{v_m}$ and touching the wall at p_m—T enables the mathematics of avoiding circular obstacles (Figure 1) to also steer R around the wall.

tions and summarize our extensive tests of *DT navigation* (i.e., steered solely by DT representations), showing that DT representations compare favorably to other representations in target-reaching effectiveness, robustness of effectiveness over varying scenarios and parameter settings, and computational efficiency. The primary contributions of this paper are:

- The design approach based on an additional, adaptive obstacle representation layer: Instead of the conventional design with only perception and repeller representation levels, we propose a three-level design. This layer leaves other design elements unaffected, thus preserving previous design strengths while improving navigation in unpredictable or geometrically heterogeneous environments.
- The particular dynamic tangent-based obstacle-valued function employed in our examples, which improves navigation in a range of tested environments. Note that other intermediary levels could also be employed in the general three-level design approach noted above; the specific details of DT navigation are not necessarily general or inherent to all possible adaptive obstacle representations.

Our extensive simulations show the kinds of substantial benefits that arise from adaptive DT representations; our tests with real robots validate these results and illuminate practical benefits of DT representations in implementation.

Dynamical Navigation

This section summarizes dynamical navigation from (Large, Christensen, and Bajcsy 1999), which extended (Schöner and Dose 1992; Schöner, Dose, and Engels 1995) with additional behavioral structure. Our adaptive obstacle representation layer leaves all of these foundations unchanged, simply building upon them for more effective navigation.

In dynamical navigation, artificial repulsion and attraction fields steer robot headings; like related force-based methods (e.g., (Khatib 1986) and many followers; see also (Koren and Borenstein 1991)), it is adaptive in dynamic environments. In this paper, consistent with (Schöner and Dose 1992), we presume constant velocity and discuss only heading angle for obstacle avoidance. To briefly summarize the system, the evolution of heading angle ϕ during navigation is determined by angular repellers and attractors in a dynamical system

$$\dot{\phi} = |w_{tar}| f_{tar} + |w_{obs}| f_{obs} + \textit{noise}, \qquad (1)$$

where $\dot{\phi}$ is the time derivative of ϕ, functions f_{tar} and f_{obs} represent targets and obstacles, respectively—the contributions of attractors and repellers to steering—and w_{tar} and w_{obs} are weight functions for each term. (The *noise* term prevents undesired fixed points in the dynamics.) A target is represented by a simple sine function, $f_{tar} = -a\sin(\phi - \psi_{tar})$, inducing a clockwise change in heading when ϕ is counter-clockwise of the target (and symmetrically, counter-clockwise when ϕ is clockwise), thus acting as an attractor.

Obstacle functions are more complicated, encoding *windowed repulsion* scaled by distance, so that repellers do not affect collision-free paths and nearby repellers are stronger than distant ones. For an obstacle obs_i, its angular repeller in DT navigation is the product of three functions:

$$\begin{aligned} R_i &= \frac{\phi - \psi_i}{\Delta\psi_i}\, e^{1-\left|\frac{\phi-\psi_i}{\Delta\psi_i}\right|} && (2)\\ W_i &= \tfrac{\tanh(h_1(\cos(\phi-\psi_i)-\cos(\Delta\psi_i+\sigma)))+1}{2} && (3)\\ D_i &= e^{-\frac{d_m}{d_0}}. && (4) \end{aligned}$$

Function R_i is an angular repeller with width $\Delta\psi_i$, centered around heading-angle value ψ_i (see Figure 1); windowing function W_i limits repulsion to significant effects only within $\Delta\psi_i$ (plus safety margin σ) from ψ_i; and scaling function D_i limits overall repulsion strength based on d_m, the minimum distance between the robot and the obstacle. (Designer-chosen constant d_0 is a scaling parameter for D_i.) Each $f_{obs_i} = R_i \cdot W_i \cdot D_i$, then, represents obs_i, and to control steering, individual contributions are summed to $f_{obs} = \sum_i f_{obs_i}$ and then combined with f_{tar} in the weighted sum in Equation 1. The weights themselves are determined by a system of competitive dynamics for reactive behavior selection; full details are available in (Large, Christensen, and Bajcsy 1999).

Dynamic Tangent Representations

In this section, we describe our dynamic tangent obstacle representations, providing a specific example of an adaptive obstacle representation layer for dynamical navigation.

Our DT approach unconventionally represents a non-circular wall entity by a conventional single circle T and repeller $R_i \cdot W_i \cdot D_i$; each T has an application-dependent angular range of repulsion and touches the entity at the nearest point to the robot. For a circular robot R with center c_R, at each timestep in navigation, the DT representation of a wall $\mathcal{W}$ is constructed as follows: Call p_m the *projection* of R on $\mathcal{W}$—the point on $\mathcal{W}$ of minimal distance from c_R (hence from the boundary of R)—with associated vector $\mathbf{v_m} = p_m - c_R$ as the shortest line from c_R to the wall; then, the obstacle representation of $\mathcal{W}$ is constructed as a circle T, oriented symmetrically around $\mathbf{v_m}$ and tangent to the perpendicular to $\mathbf{v_m}$ at point p_m (see Figure 3). To construct T, first find p_m and $\mathbf{v_m}$, which follow from elements in Figure 3. The remainder of our explanation is separated into two conceptually similar cases: $\mathbf{v_m} \perp \mathcal{W}$; or $\mathbf{v_m} \not\perp \mathcal{W}$.

For the intuition behind the construction, consider the default, non-boundary case, as in Figure 3a—$\mathbf{v_m} \perp \mathcal{W}$ and p_m is not "close" to either endpoint of $\mathcal{W}$ (where "close" will soon be clarified). In this case, the construction of T continues by identifying wall-endpoint $\mathcal{W}_R$ around which R is heading (which could be done by comparing cross products). Based on $\mathcal{W}_R$, and given an application-specific value for parameter D—a default value for how much of $\mathcal{W}$ is to be covered by repulsion—T is constructed so that its angular range of repulsion with respect to R covers the D units of $\mathcal{W}$ from p_m toward $\mathcal{W}_R$, and symmetrically, the additional D units from p_m toward the other endpoint.

To also cover boundary cases where $\mathbf{v_m} \perp \mathcal{W}$ but $|\mathcal{W}_R - p_m| < D$—i.e., where p_m is "close" to endpoint $\mathcal{W}_R$ (see Figure 3b)—employ the value $D_R = \min(D, |\mathcal{W}_R - p_m|)$ instead of D. Then, T can be similarly constructed so that its angular range covers the D_R units of the wall from p_m toward $\mathcal{W}_R$ and the symmetric D_R units along the line of the wall from p_m toward the other endpoint. The case where $\mathbf{v_m} \not\perp \mathcal{W}$ is conceptually similar (see Figure 3c): Here, p_m is always an endpoint of $\mathcal{W}$, and vector $\mathbf{v_m}$ and endpoint $\mathcal{W}_R$ and value D_R can again be found.

Demonstrations and Experiments

This section describes the tests and results that demonstrate the effects of our particular DT representations on dynamical navigation, thus also suggesting the broader value of adaptive obstacle representations to this family of navigation methods. To demonstrate the wide applicability of DT navigation, we ran more than a million simulations, along with hundreds of real robot tests using the iRobot Create platform. Robots were connected to consumer-grade portable computers, with slightly modified OpenCV image processing routines and a mast-mounted camera behind the robot to simplify 360-degree perception. For simulations, we created a simple Python- and OpenGL-based navigation simulator with straightforward idealized perception that blocked segments of walls occluded by other walls; simulated robots were circles of radius 0.1 in a 12×12 navigation space, and velocity was a constant distance per navigation step so heading angle ϕ was the only behavioral variable for navigation, as in (Schöner and Dose 1992).

In tests, our default value for parameter D was $D = 4r_R$, chosen after verifying the effect of D on navigation: In tests, when a robot R reached a path along a wall (Figure 4), the distance d_m from the wall to R is, as expected, related directly to D and radius r_R. The size-dependent value $D = 4r_R$ results in an individualized safety margin of roughly $d_m = 1.7r_R$ between walls and robots; in the future, it seems this relationship between d_m and D could be a basis for adaptive, dynamical formation control applications (cf. (Bicho and Monteiro 2003)).

Basic Testing for Effectiveness

Before testing real robots, basic tests in the simulation scenarios of Figure 6 established DT effectiveness; most of these tests are also among the tests in the next subsection of this paper. In each scenario, robots started at 100 random positions on the left sides of their worlds, with effectiveness measured by how many reached all targets for that scenario.

Results: Canyons and Hallways In the Canyon, Canyon2, and Octagon scenarios, requiring navigation around and into a convex shape, all 100 simulated DT navigation robots reached the target. In the Hallways scenario, with 3×2-sized office-obstacles (the inner rectangles), robots navigated to a sequence of five target locations (Figure 5), requiring extensive navigation and turning. DT performance was again perfect in both the Hallways scenario and the Hallways2 variant, taking smooth, efficient paths; additionally, DT performance was perfect in another Hallways variant, with two additional stationary obstacles of radius 0.3 in hallways. The Polygons scenario was more challenging, however, testing DT navigation to five target locations (similar to those in Hallways) around several polygons and a moving wall, rotating in the center of the space; default DT navigation was 92% effective, and as noted in the next subsection of the paper, DT performance was perfect with some non-default parameter values.

Results: Simulated Inaccurate Sensors As a further test, in a single-wall scenario similar to Figure 6a, we simulated DT navigation with randomly inaccurate sensors: every 30 timesteps, a random-length wall segment would disappear, or the distance to a random wall segment would be misperceived by up to 0.2 units, lasting for 5 timesteps; the length or endpoints of the wall could also be mis-perceived by some context-dependent amount. DT navigation performance was perfect in these simulations, showing adaptivity that is also beneficial for real robots with noisy sensors.

Comparing Obstacle Representations

To compare DT representations to other obstacle representations for dynamical navigation, including ones previously used in (Schöner and Dose 1992; Large, Christensen, and Bajcsy 1999), we tested navigation in the Canyon, Canyon2, Hallways, Hallways2, and Polygons simulation scenarios

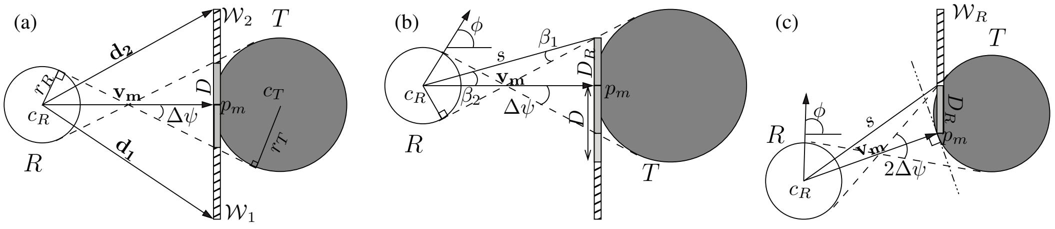

Figure 3: Constructing DT representation T, in different cases of robot R with respect to a wall. Different details are highlighted in each sub-figure.

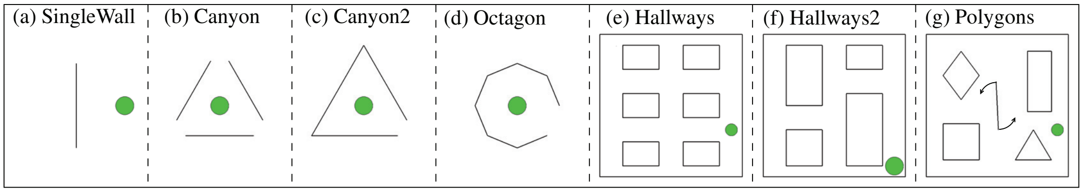

Figure 6: Scenarios for simulations, and their names in this paper. In each, only walls (obstacle entities, not obstacle representations) and a target are shown.

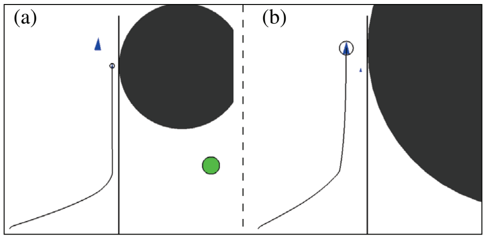

Figure 4: Two different-sized robots, sizes $r_R = 0.2$ and 0.6, reaching parallel paths along a wall, each from a setting of $D = 4r_R$. Figures show the trajectory and the DT-circle representing the wall for each. The lighter circle on the right of Figure (a) shows the target location for both robots.

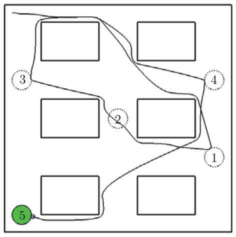

Figure 5: Target locations in the Hallways scenario. Also showing is an example trajectory of a robot steered solely by DT wall representations.

from Figure 6, and related scenarios for real robots. In each scenario, we compared the effectiveness of DT navigation to navigation based on the conventional, non-adaptive representations in Figure 7: a *Multi-circle* approach (*MC*), in which visible portions of walls were represented by coverings of small circles; a *Bounded, visible* (*BV*) approach, representing visible portions of walls by bounding circles; a *Bounded, pre-determined* (*BP*) approach, representing polygons by pre-determined bounding circles; and an *Inscribed, pre-determined* (*IP*) approach, representing polygons by pre-determined inscribed circles. For meaningful performance in the BV, BP, and IP approaches, walls surrounding a space were represented by large tangent circles. Below, we present results of tests of real robots and a very brief summary of Canyon2 and Polygons simulations; other results were qualitatively similar.

In each simulation scenario, for each obstacle representation approach, we tested varying settings for four parameters: distance scaling d_0, sampled at 0.1 over range $[0, 2]$ units; angular safety margin σ, sampled at 0.1 over range $[0, 1]$ radians; attractor amplitude a, tested at values 1 and 3; and the radius *mcSize* of circles employed in the MC approach, tested at $\frac{r_R}{2}$, r_R, $2r_R$, and $4r_R$. (See Equations 2–4 for d_0, σ, and a in context.) In this section, we discuss only results that illustrate the best performance for each obstacle representation; relatedly, when preliminary tests indicated that some parameter values would not add to our discussion—for instance, when MC representations were clearly ineffective for *mcSize* greater than r_R—we did not exhaustively test those values. Then, for each parameter setting, we simulated 100 robots from random starting positions to the target locations for a scenario (described in the previous subsection), recording the number that successfully reached all targets. We compared obstacle representations on three measures: effectiveness, measured by how many of the 100 simulated robots reached their targets; robustness of navigation effectiveness over different parameter values and scenarios; and computational efficiency of each obsta-

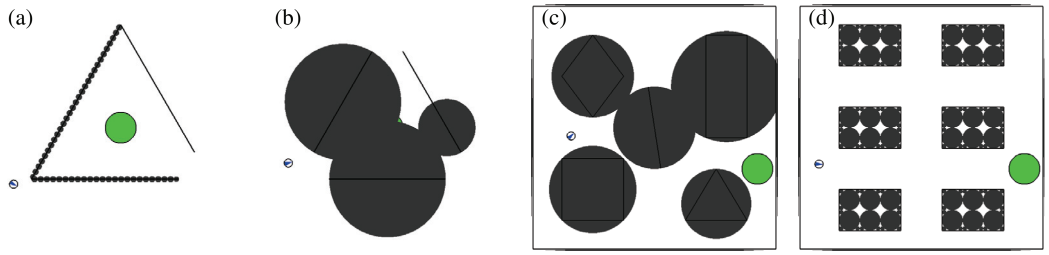

Figure 7: Different approaches to wall representation: (a) Multi-circle (abbreviated *MC*); (b) Bounding, visible (*BV*); (c) Bounding, pre-computed (*BP*); (d) Inscribed, pre-computed (*IP*). In each, dark circles are obstacle representations; also shown is a robot (small, partly filled circle) and a target.

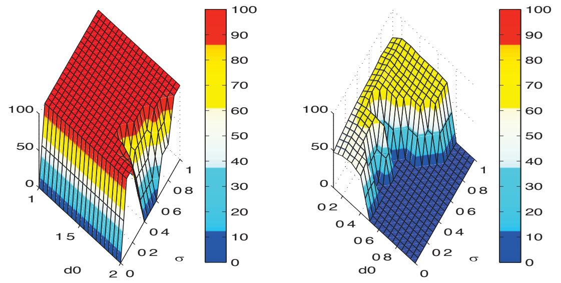

Figure 8: Results of simulations with DT (left) and MC (right) representations in the Canyon2 scenario, with $a = 1$ and *mcSize* of 0.05. Effectiveness values are between 0 and 100, over a range of σ and d_0 values. DT performance was robustly perfect; MC performance was less robust over parameter values and had peak effectiveness of 72. This is qualitatively similar to performances in other scenarios.

Table 1: Comparisons of computational efficiency, over different wall representations and navigation scenarios.

Milliseconds per Timestep					
Scenario	DT	MC	BV	BP	IP
Hallways	11.21	124.64	7.35	6.87	52.62
Polygons	10.12	126.66	12.48	9.24	39.96
Canyon	4.66	68.12	3.54	3.93	n/a
Canyon2	3.28	44.63	2.64	3.41	n/a

cle representation. We compared efficiency only among our simulations, each with the same general Python code base and run on a consumer-grade Intel Core 2 Duo computer.

Real robot tests were informed by simulations and run in different, similarly illustrative scenarios, as described below.

Results: Effectiveness and Robustness Figures 8–9 illustrate our data on effectiveness and robustness, showing how many robots succeeded for values of d_0 and σ in our scenarios. Figure 8 shows robust, perfect DT performance in the Canyon2 scenario, compared with less effective MC representations; in Canyon and both Hallways scenarios, DT performance was also perfect for a wide range of parameter values, and in the Polygons scenario (Figure 9), DT was the only approach to have robust effectiveness of 90 or greater or perfect performance for any parameter values. In contrast, BP and BV representations had no success for any parameters in Canyon2, and very little success in Polygons; IP representations were better but still significantly less effective than DT representations (Figure 9). Across scenarios, MC representations were substantially more effective than BP, BV, or IP representations—peak MC values were in the 60–90 range—with frequently inconsistent robustness, becoming ineffective for wide ranges of d_0 values. Consistently, DT representations were the most effective and robust; only MC representations were meaningfully comparable.

Because of this, we tested real robots with only DT and MC representations, and both did well in simple environments. In environments with significant heterogeneity, however—e.g., moving from wide, open corridors to blocked or narrow ones (with parameter $D = r_R$); or navigating around a moving obstacle before making a sharp turn (with $D = 2r_R$)—DT navigation substantially outperformed MC representations: In tests in these environments, with a small range of d_0, σ, a, and *mcSize* values, DT navigation succeeded in every test, whereas MC representations succeeded in less than 10% of nearly 150 tests. This is only a very brief summary, but the results clearly support the robustness and effectiveness results from our simulations.

The practical benefits of improved effectiveness and robustness were substantial. In environments with long walls, for instance, DT navigation frequently succeeded with greater repulsion (greater d_0 value) than MC navigation. Because of this, DT navigation was less vulnerable to variations in perception: In MC-based navigation, small difficulties with perception could readily lead to collisions. Relatedly, DT navigation succeeded at higher velocities than MC-based navigation did. Also, the increased robustness of DT navigation significantly improved procedural aspects of the testing process: With less sensitivity to small variations in parameter values, initial locations, or other factors, DT navigation supported replicable experimental successes far more than other representations did.

Results: Efficiency To compare computational efficiency across simulation scenarios and obstacle representations,

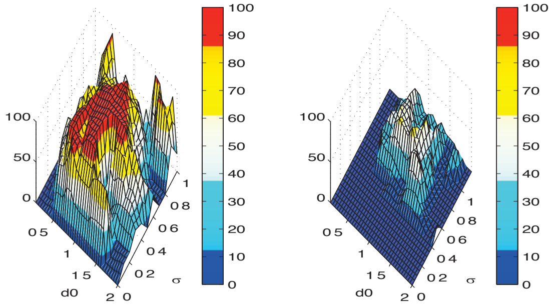

Figure 9: Results of simulations with DT (left) and IP (right) representations in the Polygons scenario, with $a = 1$. This is the only scenario in which DT did not achieve robust, perfectly effective performance; the restriction to constant velocity made it difficult for DT navigation to avoid the moving wall. Nonetheless, DT representations were the only option to achieve perfect performance for any parameter values, with robust effectiveness of 90 or greater. IP representations (with an MC representation of the moving wall) generally performed well compared to other representations in this scenario, with peak effectiveness near 60.

we profiled average computation time for navigation per timestep, over a complete run. Results in Table 1 show that efficiency correlates to the number of circles used in representations—more circles entail greater computational cost; in particular, DT representations are substantially more efficient than the only other option effective in our scenarios, MC representations. Real robot tests validated this conclusion: in MC-based navigation, when *mcSize* was small enough to be effective in narrow corridors, robots oscillated extensively, due to calculation costs; in contrast, DT navigation was smooth and effective, even in narrow passageways.

Conclusion

This paper suggests and supports a novel design idea for improving dynamical navigation: Adding an adaptive obstacle representation layer between perception and mathematical repeller representations can substantially improve performance, extending and preserving the original design virtues of the navigation system. To illustrate this idea and provide a specific example that applies to multiple navigation methods, we present a dynamic tangent-based adaptive obstacle representation level that treats perceived obstacle entities as locally sensitive obstacle-valued functions; we further demonstrate that DT representations improve upon other representations in measures of target-reaching effectiveness, robustness over varying scenarios and ranges of parameter settings, and computational efficiency. The greater robustness over parameter values is especially important for navigation in unpredictable or heterogeneous environments, improving the likelihood that parameter settings effective in one component sub-region of the environment will be effective in others, as well. The greater effectiveness and computational efficiency suggest that DT navigation might be especially apt for applications with limitations on available computing power (e.g., micro-robotics).

In general, adaptive obstacle representations can improve a variety of navigation methods, beyond the specific examples in this paper. Our particular DT representations, for instance, immediately fit with other steering-based navigation approaches, such as (Huang et al. 2006). Furthermore, the mathematical foundations of (Fajen et al. 2003) (and applications of it, such as (Hamner et al. 2008)) suggest that similar adaptive representations could also have benefits similar to those described in this paper. Even more broadly, adaptive obstacle representations can improve hybrid control or navigation systems, substantially strengthening the reactive levels at their foundations.

Acknowledgments

The authors thank Clare Bates Congdon, Zachary Dodds, and Jim Marshall for their advice and comments on this paper.

References

Aaron, E., and Mendoza, J. P. 2011. Dynamic obstacle representations for robot and virtual agent navigation. In *Proc. Canadian Conf. Artificial Intelligence*, 1–12.

Bicho, E., and Monteiro, S. 2003. Formation control for multiple mobile robots: A non-linear attractor dynamics approach. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 2016–2022.

Fajen, B.; Warren, W.; Temizer, S.; and Kaelbling, L. 2003. A dynamical model of visually-guided steering, obstacle avoidance, and route selection. *International Journal of Computer Vision* 54(1-3):13–34.

Hamner, B.; Singh, S.; Roth, S.; and Takahashi, T. 2008. An efficient system for combined route traversal and collision avoidance. *Auton. Robots* 24(4):365–385.

Huang, W.; Fajen, B.; Fink, J.; and Warren, W. 2006. Visual navigation and obstacle avoidance using a steering potential function. *Robotics and Autonomous Systems* 54(4):288–299.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research* 5(1):90–98.

Koren, Y., and Borenstein, J. 1991. Potential field methods and their inherent limitations for mobile robot navigation. In *IEEE Int. Conf. Robotics and Automation*, 1398–1404.

Large, E.; Christensen, H.; and Bajcsy, R. 1999. Scaling the dynamic approach to path planning and control: Competition among behavioral constraints. *International Journal of Robotics Research* 18(1):37–58.

Schöner, G., and Dose, M. 1992. A dynamical systems approach to task-level system integration used to plan and control autonomous vehicle motion. *Robotics and Autonomous Systems* 10(4):253–267.

Schöner, G.; Dose, M.; and Engels, C. 1995. Dynamics of behavior: Theory and applications for autonomous robot architectures. *Robotics and Autonomous Systems* 16(2-4):213–245.