

# Composition of Local Potential Functions with Reflection

Adam Stager and Herbert G. Tanner

**Abstract**—This paper suggests reflections can be practically useful if they are included in planning for collision capable robot platforms. By modifying a proven strategy for navigation with reflections we maintain global convergence results and reach the goal in less time. An algorithm for identifying reflection surfaces for a given cell decomposition is reported. Baseline and reflected scenarios are compared for two different cell decompositions. *Omnipuck*, a reflection capable omnidirectional robot meant to store and release impact energy, is used to obtain experimental results and draw conclusions for future work.

## I. INTRODUCTION

Conventional planning and navigation strategies have used geometric representations of boundaries to define collision free paths with convergence guarantees [1]. These strategies can be restrictive, using conservative bounds on inputs and velocities in order to ensure collision free trajectories. While they often work well, for systems called to operate near their actuation limit and at maximum velocity [2], sometimes under significant uncertainty or with minimal control authority, a designer may have to consider the possibility of the robot interacting with workspace boundaries.

Small and inexpensive robot platforms can be deployed in large numbers and serve a multitude of purposes; for instance, they can quickly scan dangerous or otherwise not easily accessible regions that would otherwise require significant time and/or resources. Yet these robots present unique challenges in robot navigation and control because they are susceptible to environmental disturbances. Sometimes, manufacturing variability at those scales can trigger stochastic behavior [3] to be observed in the robots' motion. When such robots are commanded to travel at high speeds, they may lack the control authority necessary to follow precisely a given trajectory. As a result, and despite the potential benefits, the unpredictability of these platforms' motion behavior can preclude their deployment. This paper suggests a way or reclaiming some of that application potential.

Feedback helps to reduce the effect of uncertainty at small scales [4]. Potential fields yield feedback strategies that guide a robot toward a goal and can be practical when a robot is not capable of precisely tracking a desired trajectory [4]. Potential field methods can also guarantee convergence to some target location without forcing the robot along a particular path or trajectory. One approach avoids obstacles using vortex fields to achieve convergence [5]. Yet generating provably convergent potential fields over the whole robot workspace comes with its own analytical and application

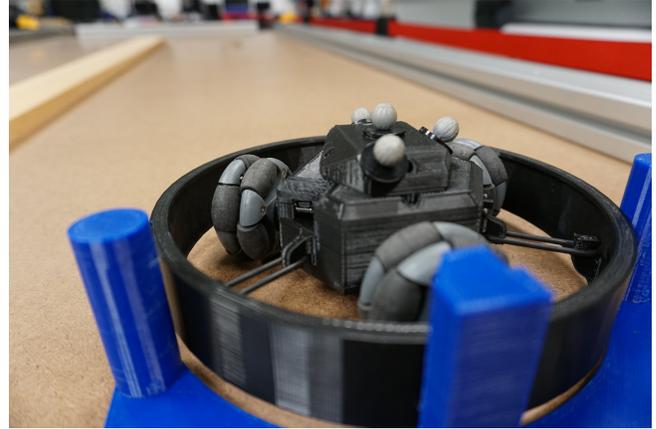


Fig. 1: The 3D printed Omnipuck is robust to collisions due to a protective, collision energy restoring ring around its body. Here it is positioned in a 3D printed jig for precise positioning of the initial position before each experiment.

challenges, which have been partially circumvented through the utilization of local potential functions. These local potential fields are defined over a cell decomposition of the workspace, and implement “flow-through” control policies in a sequential composition fashion [6]–[8]. The benefit of such policies is that it obviates the need for careful parameter tuning; however, special consideration must be taken to avoid collisions due to large spectral variations at the vertices of the environment, occasionally forcing a conservative bound on the velocity of the robot. For robots with minimal control authority, or having to operate close to their actuator limits, such restrictions can be problematic.

Yet robot-boundary interactions are not always detrimental. Flying insects, for example, have shown spectacular robustness to collisions [9]. Aerial robots have been designed to be robust to collisions and recent work has shown that small robots can navigate an environment more effectively given some collision resistance [10], [11]. Similarly, robots on the ground have been shown experimentally to benefit from boundary interactions [12], [13], even using them to rapidly change their direction of motion [14]. One platform, referred to as *Omnipuck*, includes a reflection ring allowing it to redirect collision energy, benefiting from collisions in certain situations [15]. Several models for reflecting have been developed to capture velocity, position, and uncertainties that result in collision behaviors [15], [16] but the effect of collision behaviors in high level planning has not been explored. For a robot capable of non-terminal physical interactions with its workspace boundaries, this paper reports

benefits in the time to reach a desired configuration and the plurality of paths that can be followed to this end.

The remaining question is whether one can explicitly include boundary interactions in planning algorithms in order to realize some of their practical benefits for improving convergence time. A hybrid system allowing for impact-triggered switching on boundaries can give rise to a mathematical framework that allows for global convergence guarantees under high-speed conditions for low control authority robotic vehicles. With this overall objective, this paper is organized as follows. Section II describes the development of a reflected hybrid control policy, Section III describes a systematic way of identifying reflecting boundaries, Section IV compares experimental results from baseline and reflected navigation strategies, and is followed by concluding remarks in Section V. The goal of this work is to suggest a new paradigm for high level planning and navigation that allows faster and possibly more efficient navigation by leveraging workspace boundary interactions for robots that can afford them.

## II. PRIMITIVE MOTION PLANNING POLICIES

In this section we describe a “flow-through” strategy for driving single integrator dynamics through polytope facets and into a desired region, using sequences of primitive local potential functions [8]. Typically, such methods guarantee convergence to a local outlet zone within a cell; here, however, we consider reflection zones as switching boundaries, contact with which triggers a reversal on the direction of the field in the polytope.

Consider a cluttered workspace, the free-space portion of which is decomposed into a finite union of convex, disjoint polytope cells, and denote such a cell  $\mathcal{P}$ .<sup>1</sup> For simplicity of exposition, we restrict the description in  $\mathbb{R}^2$ , where cells are simply convex polygons each defined by a set of half-plane constraints. Specifically, an  $m$  sided polytope  $\mathcal{P}$  can be defined by the midpoints of each side  $p$  and the set of corresponding inward normal vectors  $\mathbf{n}$

$$(p_i, \mathbf{n}_i) \quad | \quad i = 1, \dots, m \quad (1)$$

For a complete derivation of the constraints for convex polytopes in higher dimensional spaces, see [17]. The adjacency of cells induces a graph, searching through which by any of the available methods, can yield a sequence of cells that lead to a desired configuration [1]. Then, each intermediate cell in this sequence can be assigned an inlet and outlet zone; the inlet connecting it to its predecessor and the outlet to its successor in the sequence. A locally convergent control policy driving the system through these zones is guaranteed to drive any initial configuration from its current cell  $\mathcal{P}$  to its adjacent target cell  $\mathcal{P}_t$ .

Several strategies have been proposed for determining such a locally convergent control policy [6], [7]; it turns out that in experimental implementation, the one affording the

<sup>1</sup>One might choose any of the established methods for decomposing the workspace [1], although it is worth noting that certain cell configurations may provide more desirable reflection opportunities.

highest speeds without collisions was one that utilizes fields that steer toward straight lines or circular arcs within each cell (cf. [18]). Stitching together such basic primitive vector fields within a sequential composition framework afforded simplicity, reliability in terms of collision avoidance, and increased robot speeds compared to alternative flow-through methods.

Letting  $(x_0, y_0)$  denote the midpoint of the inlet zone,  $(dX, dY)$  the unit vector along the line joining  $(x_0, y_0)$  to the midpoint of the outlet zone, and using  $p \in \mathbb{R}$  as a tuning parameter, the directed vector fields of the type of Fig. 2a can be described mathematically as

$$\dot{x} = dX + p(x_0 - x - [(x_0 - x)dX + (y_0 - y)dY]dX) \quad (2a)$$

$$\dot{y} = dY + p(y_0 - y - [(x_0 - x)dX + (y_0 - y)dY]dY) \quad (2b)$$

The benefit of such a potential field is its ability to direct the motion of the system toward a particular point rather than toward the entire boundary of an outlet zone. This will prove useful later for reflections but is also practical because it tends to keep the robot away from cell boundaries.

In a similar way, directed circular arc fields (Fig. 2b) can be described in terms of a center  $(x_c, y_c)$  and a radius  $r$  as

$$\dot{x} = r(y - y_c)c_w - 4p(x - x_c)[(x - x_c)^2 + (y - y_c)^2 - r^2] \quad (3a)$$

$$\dot{y} = -r(x - x_c)c_w - 4p(y - y_c)[(x - x_c)^2 + (y - y_c)^2 - r^2] \quad (3b)$$

As before, the tuning parameter  $p$  adjusts how fast the system converges to the prescribed geometric primitive. A directional parameter  $c_w = 1$  if the field is directed in the clockwise direction and is  $c_w = -1$  otherwise.

Both types of potential fields are visualized as streamlines in Fig. 2. In general, although robot trajectories produced by such a sequential composition strategy are continuous, they are not necessarily differentiable at the transitions where fast changes in the gradient can occur. If smoothness is desired, the parameter  $p$  in each cell can be tuned to give smooth field lines on the boundaries. Here we maintain a constant  $p = 0.3$  over each linear vector field and  $p = 0.03$  over semi-circular fields.

One benefit of using potential fields to direct the robot to the goal is their robustness to disturbances. This is particularly useful for small and inexpensive robots because they are prone to uncertainties in their motion. Although uncertainties may push the robot into boundaries, or even in reverse over the inlet and back into the previous cell, a non-zero drift toward a sequence of cell outlets maintains convergence to the goal with probability one as long as there is no bound on time.

## III. REFLECTIONS

To introduce reflecting boundaries, referred to as *reflection zones*, we add a switching state which is activated at the moment of a collision and results in a redirection of the potential field in  $\mathcal{P}$ . After defining a sequence of cells guaranteeing global convergence to the outlet zone without contact with boundaries, a strategy for identifying reflection

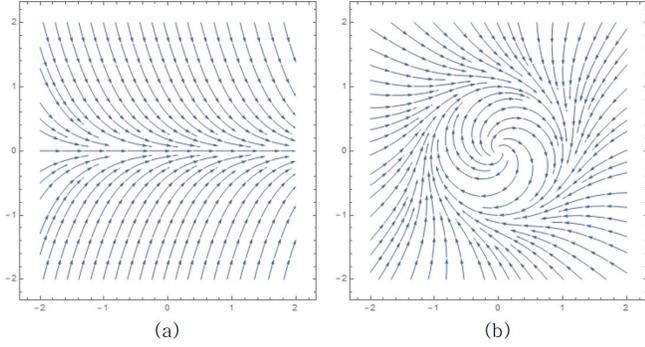


Fig. 2: Line (a) and arc (b) vector fields pictured with parameters  $p = 2.0$  and  $p = 0.2$  respectively.

zone candidates determines which cells are likely to benefit from reflecting behavior. This ordering is necessary so that angles between inlet and outlet zones can be defined. Once a reflection candidate is identified the minimum angle between inlet and outlet normal vectors and the remaining boundaries determines the reflection zone. After reflecting from a boundary with the induced reflection field, the potential field switches with the original potential leaving from the outlet zone, thus preserving convergence guarantees from each cell.

#### A. Reflection Candidates

With a set of adjacent cells and locally converging field policies inside them, we proceed by defining a condition for adding a reflection to a nominal cell sequence while maintaining global convergence to the desired cell and configuration. Intuitively a reflection can act as a way of quickly changing direction, especially for actuation-limited systems where it can take a measurable amount of time to counteract opposing momentum. For each cell  $\mathcal{P}$  we define  $p_{in}$ ,  $\mathbf{n}_{in}$ , the midpoint and inward normal vector for the inlet region, and  $p_{out}$ ,  $\mathbf{n}_{out}$  the midpoint and inward normal vector on the outlet region, respectively. Define angle  $\alpha = \cos^{-1}(\mathbf{n}_{in} \cdot \mathbf{n}_{out})$ . Then for any  $\mathcal{P}$  with  $\alpha \leq \pi/2$  a reflection inducing a vector field switching will be introduced. This strategy for determining reflection candidates filters out cells where the inlet and outlet zones are closely aligned.

#### B. Selecting Reflection Zones

For any cell  $\mathcal{P}$  selected for inclusion of a reflection zone, we define a set

$$\mathcal{S} = \{(p_i, \mathbf{n}_i) \mid i \in \{1, \dots, m\}\} \setminus \{(p_{in}, \mathbf{n}_{in}), (p_{out}, \mathbf{n}_{out})\}$$

Figure 3 demonstrates one such example where  $i = 1$ . Convexity of  $\mathcal{P}$  guarantees any two midpoints from the facets of  $\mathcal{P}$  are simply connected. The reflection field will be a directed linear vector field, streaming from  $p_{in}$  to  $p_i \in \mathcal{S}$ . To select the endpoint in (2) first define  $\mathbf{v}_{in}(p)$  as the unit vector from  $p_{in}$  to a midpoint  $p_i \in \mathcal{S}$ ; define similarly  $\mathbf{v}_{out}(p)$  as the unit vector from  $p_{out}$  to that same  $p_i \in \mathcal{S}$ . Then select the  $p \in \mathcal{S}$  that minimizes

$$\min_{p \in \mathcal{S}} (\mathbf{n}_{in} \cdot \mathbf{v}_{in}(p))^2 + (\mathbf{n}_{out} \cdot \mathbf{v}_{out}(p))^2 \quad (4)$$

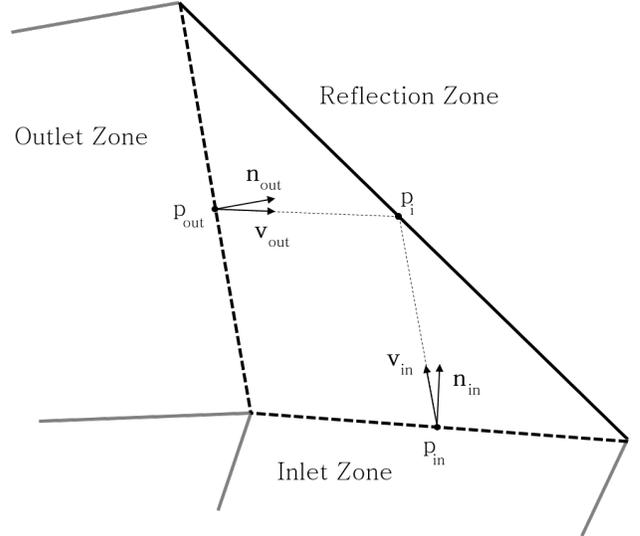


Fig. 3: A simple scenario for defining and selecting the reflection zone in cell  $\mathcal{P}$ .

The solution to this minimization problem is not necessarily unique for symmetric cell geometries. In such cases, one can select the point that minimizes  $\mathbf{n}_{in} \cdot \mathbf{v}_{in}(p)$ . For example in the case of an isosceles right triangle cell,  $\min_{p \in \mathcal{S}} = 0$  with  $\mathbf{v}_{in} = \mathbf{n}_{in}$ ,  $\mathbf{v}_{out} = \mathbf{n}_{out}$ , and the midpoint of the hypotenuse as the endpoint in (2).

This approach favors a reflection surface that requires minimal change in direction for a robot entering normal to the inlet zone and exiting normal to the outlet zone. Constraining the problem to search over the midpoints of cell edges makes the solution more tractable, however this formulation can be extended by allowing points  $p_{in}$ ,  $p_{out}$ , and  $p \in \mathcal{S}$  to slide along their respective cell edges. This extension essentially takes the form of a (constrained) continuous variation of the optimization problem in (4). Indeed, by relaxing the assumption of side midpoints as the only potential entries and exits from a polytope, one can further reduce the directional variation and take better advantage of reflection as a means of instantaneous velocity reversal to speed up steering and navigation. This extension falls beyond the scope of this particular paper and is part of ongoing work. Here, the objective is to experimentally demonstrate the beneficial nature of boundary interactions and take the first step into integrating such behaviors in a motion planning controller.

## IV. EXPERIMENTAL RESULTS

To experimentally demonstrate the proposed benefits of including collisions in planning, a collision resistant Omnipuck robot is tested in two distinct vector field configurations shown in Fig. 4. The first configuration was constructed entirely from cells with linearly directed vector field lines whereas the second configuration contains both linear and semi circular field lines. The cell decomposition varies slightly to accommodate flow through conditions for guaranteed convergence to the outlet zone. Here the method

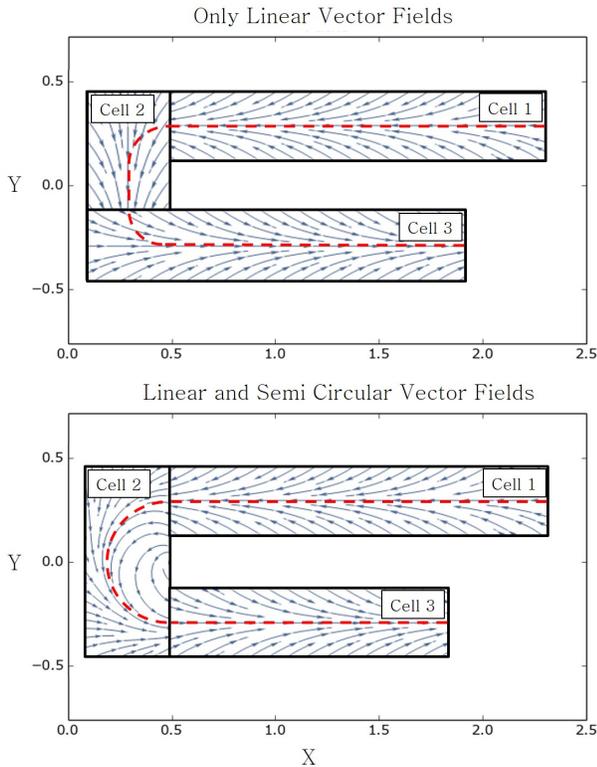


Fig. 4: Black outlines separate cells in two baseline scenarios, red dashed lines denote the desired trajectory assuming unbounded acceleration to follow vector field lines.

of decomposing the workspace is not optimized; instead there are two examples to illustrate differences between arc/linear fields and also give two situations for reflection. Any of the cell decomposition methods [1] can be used over the workspace, but optimizing the cell geometry while retaining local cell convexity would likely increase the opportunity for reflections. Three experiments are conducted for each field configuration:

- a constrained baseline case where a speed limiting constraint reduces control input to minimize boundary interactions, reduce overshoot, and force more close following of the desired field lines,
- an unconstrained baseline where the control input is set to a high constant value allowing for high speed motion of the robot and poor field line tracking, and finally
- experiments where switching reflection fields are added as described in Section III.

#### A. Experimental setup

A Vicon Motion Capture system tracks the position of the Omnipuck which is processed through a laptop with commands sent at 20Hz based on the current vector field. The robot is placed in a 3D printed jig, fixed to a smooth medium-density fiberboard (MDF) tabletop. Twenty five trials are taken for each scenario.

#### B. Constrained baseline

This scenario is implemented and tested in both vector fields of Fig. 4, where the magnitude of the field was scaled empirically to the highest level that prevented collisions while still producing effective motion towards the goal. These experiments act as a “control” because typically such constraints on velocity would be incorporated into a planner that is designed to avoid boundary interactions.

For robots with minimal control authority, however, such planning directives can be detrimental. Omnipuck, specifically, uses low torque, high speed motors and has a relatively large dead-band; the motors will not readily produce forward motion when the control input is too low. Still, the baseline trajectories in the linear field (Fig. 5a) are being tracked reasonably well despite undesirable low-speed boundary collisions occurring after transitioning between Cell 1 and Cell 2. Figure 5b shows a more dramatic boundary interaction in the scenario where a semi-circular vector field was meant to control the robot along a smooth arcing path. In these experiments the robot was able to counteract its inertia on only a few occasions, collided with the wall at generally low speed, and continued to follow the field after such events. Yet, these boundary interactions slowed its forward velocity significantly, resulting in slower overall convergence time compared to the linear baseline experiments.

#### C. Unconstrained baseline

In this case too, boundary interactions are undesirable, or at least unplanned. While maintaining an identical control sequence, now the speed is set to maximum. This second set of experiments explores how planning without boundary considerations can effect the robot’s trajectory and overall convergence time in high-speed regimes. In the scenario with linear field lines of Fig. 5c one observes a clear overshoot and collision as the robot transitions from Cell 1 to Cell 2 with more momentum than a braking control input can overcome. The reflection ring surrounding Omnipuck protects the robot from damage, but the post-impact velocity vector combined with the application of the control input causes subsequent collisions on the opposing boundary. In the configuration of Fig. 5d the control input produced by the semi-circular vector field in Cell 2, drives instead the robot back into the boundary, colliding three times in Cell 2 before entering Cell 3. In both scenarios overshoot due to higher overall speed and momentum cause poor tracking of the field lines, however significantly faster convergence is observed due to non-terminal collisions and global convergence guaranteed by the sequence of locally convergent potential fields.

#### D. Reflections

A set of experiments with boundary reflections implements a switching vector field as described in Section III. Now collisions are leveraged allowing faster navigation. In both scenarios a reflection surface is identified on the leftmost boundary of Cell 2, and due to the  $\alpha = \pi/2$  angle between the inward and outward normal vectors of Cell 3, an additional collision is forced (in the linear configuration only).

Experimental Setup	Avg Time (s)	Time Variance
Constrained Baseline (linear)	8.27	0.503
Unconstrained Baseline (linear)	5.07	0.063
Reflection (linear)	4.82	0.038
Constrained Baseline (linear/arc)	11.68	2.099
Unconstrained Baseline (linear/arc)	6.05	0.126
Reflection (linear/arc)	5.43	0.229

TABLE I: Results indicate fastest convergence when the planner accounts for reflections.

For the linear configuration, the second collision in Cell 2 was avoided more often allowing for a favorable transition into Cell 3. On the other hand, in the second configuration the collision vector prevented the semi-circular vector field from driving the system into the leftmost boundary allowing it to maintain its velocity as it moved through Cell 2. Overall, the effect of the additional (reflection inducing) vector fields was less dramatic than expected, however the change in control input over the sequence of cells produced an overall reduction in convergence time (Table 1) and higher velocities on average. The reflecting fields failed to direct the robot into a more oblique collision with the boundary and thus were not able to capture some of the hypothesized benefit. Yet this benefit is expected to be recovered through the process of optimizing the location of planned entry, exit, and collision points, along the lines outlined at the end of Section III.

#### E. Discussion

Existing potential field methods can provide convergence guarantees and are particularly useful for systems effected by uncertainty, but guaranteeing obstacle avoidance can require unrealistic control inputs, or prohibitively low velocities. Attempting to avoid collisions by reducing velocity led to high convergence times in baseline scenarios. By lifting the velocity constraint and allowing for uncontrolled collisions, overall faster convergence is observed. This naive approach resulted in undesirable collisions with the boundaries that can reduce the perceived benefit of collisions. Planning for reflections by identifying reflection zones and altering the vector fields to intentionally reflect results in more predictable behavior further reducing convergence time.

After observing these experiments there are several insights worth mentioning. First, the speed that a collision can be detected is an important factor. In this work we maintained a relatively modest 20Hz rate of control meaning a collision

could not switch the vector field state for almost  $\frac{1}{20}$  seconds (in the worst case). Practically this delay results in the robot continuing toward the wall after a collision, counteracting the restoring force generated by the reflection. Ideally one would increase controller rate as much as possible to reduce delays and capture the most restoring force from a given impact. Second, reflection distance plays a role in the effect a boundary impact can have on the overall trajectory. Haphazard reflections may carry a robot into a portion of the workspace that is restricted or somehow detrimental. Incorporation of an estimate on post-collision distance is a valuable addition that is likely to have important implications especially if reflections are to be considered over a global convergence time optimization. Finally, as is visually apparent by the overlaid paths in Figure 5 and statistics in Table 1, uncertainty plays a significant role especially in cases where robots are small scale or inexpensive. Impacts on obstacle vertices are a particular situation that should be avoided in planning because reflection direction in this scenario is unpredictable.

## V. CONCLUSIONS

Avoiding collisions between a mobile robot and its workspace boundaries at any cost may not always be the most efficient navigation strategy, especially if the robot can withstand impact, or it is considered dispensable. Recently, several reflecting behaviors have been characterized, and this paper indicates how they can be intentionally included as part of the overall motion planning strategy. It turns out that by leveraging boundary interactions, convergence time in robot navigation can be markedly improved. The paper adds to the arsenal of globally convergent feedback-based navigation methods with a sequential composition strategy that combines locally convergent linear and semi-circular vector fields. Future work looks toward tailoring cell decompositions for optimizing the benefits of boundary interactions in planning over the entire sequence of cells rather than considering each cell individually. Multi-robot interactions where intentional collisions between robots divert motion toward cooperatively beneficial behaviors or goals is an especially interesting extension of the ideas presented herein. In contrast to traditional motion planning methods, collision avoidance is not a high priority; global convergence with reduced navigation times is. A series of experimental results utilizing a specially-designed platform that can leverage boundary interactions through elastic impact behaviors, confirm the anticipated benefits in terms of convergence time.

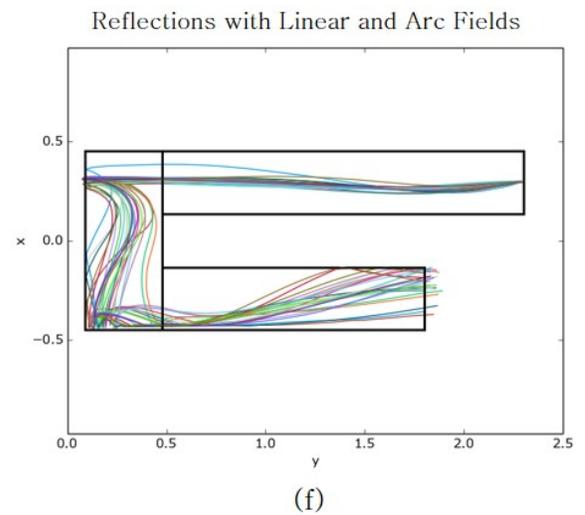
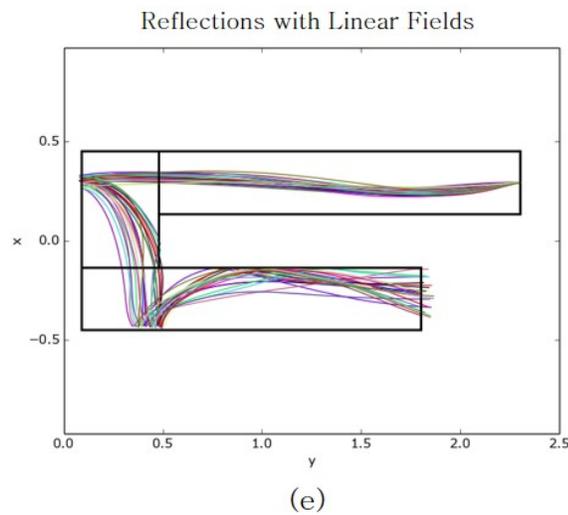
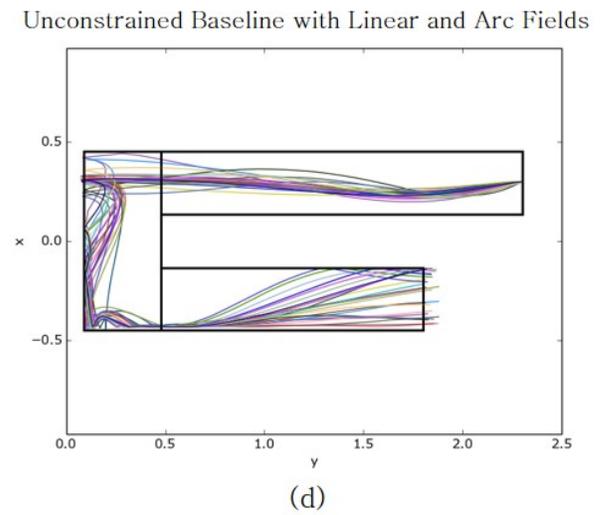
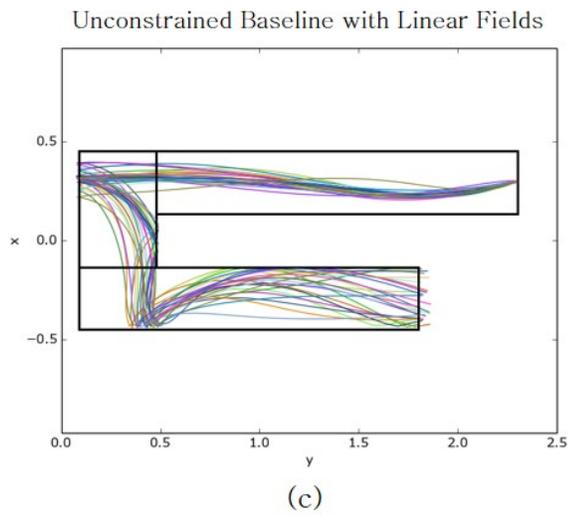
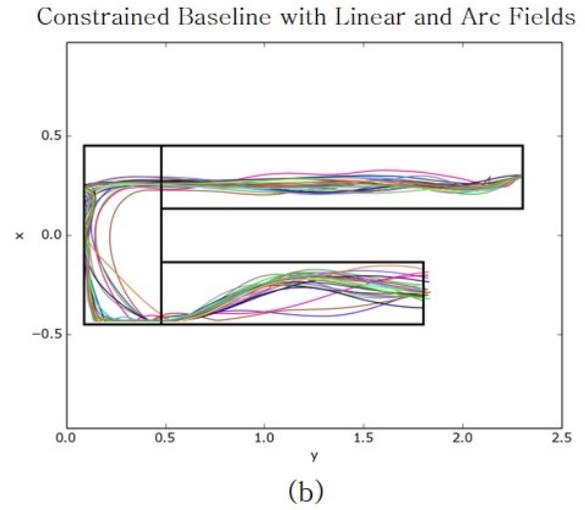
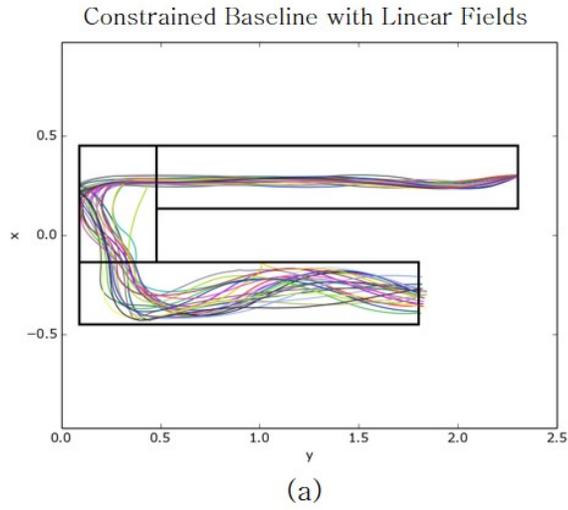


Fig. 5: Experimental trajectories traveling through a sequence of directed vector fields from an initial position at the top right to a final outlet zone at the bottom right. Thick black lines denote cell boundaries. Linear fields only in (a),(c),(e) and linear with arc fields in (b),(d),(f).

## REFERENCES

- [1] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.
- [2] S. Karaman and E. Frazzoli, "High-speed motion with limited sensing range in a poisson forest," *Proceedings of the IEEE Conference on Decision and Control*, pp. 3735–3740, 2012.
- [3] K. Karydis, I. Poulakakis, J. Sun, and H. G. Tanner, "Probabilistically valid stochastic extensions of deterministic models for systems with uncertainty," *The International Journal of Robotics Research*, pp. 1278–1295, 2015.
- [4] K. Karydis, I. Poulakakis, and H. G. Tanner, "A Navigation and Control Strategy for Miniature Legged Robots," vol. 33, no. 1, pp. 214–219, 2017.
- [5] C. D. Medio and G. Oriolo, *Robot Obstacle Avoidance Using Vortex Fields*, S. Stifter and J. Lenarcic, Eds. Springer Verlag, 1991.
- [6] S. R. Lindemann and S. M. LaValle, "Smoothly blending vector fields for global robot navigation," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, vol. 2005, 2005, pp. 3553–3559.
- [7] D. Conner, A. Rizzi, and H. Choset, "Composition of local potential functions for global robot control and navigation," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, no. October, pp. 3546–3551, 2003.
- [8] R. Burridge, A. Rizzi, and D. Koditschek, "Sequential Composition of Dynamically Dexterous Robot Behaviors," *International Journal of Robotics Research*, vol. 18, no. 6, pp. 534–555, 1999.
- [9] A. Briod, "Observation of Insect Collisions," Ph.D. dissertation, Harvard University, 2008.
- [10] A. Briod, P. Kornatowski, A. Klaptocz, A. Garnier, M. Pagnamenta, J.-C. Zufferey, and D. Floreano, "Contact-based navigation for an autonomous flying robot," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3987–3992, 2013.
- [11] Y. Mulgaonkar, G. Cross, and V. Kumar, "Design of small, safe and robust quadrotor swarms," *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, pp. 2208–2215, 2015.
- [12] K. Karydis, D. Zarrouk, I. Poulakakis, R. S. Fearing, and H. G. Tanner, "Planning with the STAR(s)," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3033–3038.
- [13] V. J. Lumelski and A. A. Stepanov, "Dynamic Path Planning for a Mobile Automaton with Limited Information on the Environment," *Ieee Transactions on Automatic Control*, vol. 31, no. 11, pp. 1058–1063, 1986.
- [14] D. W. Haldane, M. M. Plecnik, J. K. Yim, and R. S. Fearing, "Robotic vertical jumping agility via series-elastic power modulation," *Science Robotics*, vol. 1, no. 1, p. eaag2048, 2016.
- [15] A. Stager and H. G. Tanner, "Mathematical Models for Physical Interactions of Robots in Planar Environments," in *Proceedings of the International Symposium on Experimental Robotics*, (to appear), 2018.
- [16] Y. Mulgaonkar, A. Makineni, L. Guerrero-Bonilla, and V. Kumar, "Robust Aerial Robot Swarms Without Collision Avoidance," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 596–603, 2018.
- [17] D. Conner, "Integrating Planning and Control for Constrained Dynamical Systems," Ph.D. dissertation, Carnegie Mellon University, 2007.
- [18] L. A. Valbuena Reyes and H. G. Tanner, "Flocking, Formation Control, and Path Following for a Group of Mobile Robots," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1268–1282, 2015.