2011 IEEE International Conference on Robotics and Automation
Shanghai International Conference Center
May 9-13, 2011, Shanghai, China

A Compact Representation of Locally-Shortest Paths and its Application to a Human-Robot Interface

Abdullah Akce and Timothy Bretl

Abstract— **The space of all possible paths through a finite-dimensional configuration space is infinite-dimensional. Nevertheless, paths taken by "real" robotic systems often cluster on a finite-dimensional manifold that is embedded in this infinite-dimensional space and that is governed by a principle of optimality. We take advantage of this property to enable a human user to efficiently specify a desired path for a robot moving through a planar workspace with polygonal obstacles using a sequence of noisy binary inputs, as might be derived from a brain-machine interface. First, we show that the space of all such paths having length that is bounded and locally minimal is homeomorphic to the unit disk. Second, we note that any path mapped to the interior of this disk is a subset of some other path mapped to its boundary. Third, we provide an optimal communication protocol by which the user can, with vanishing error probability, select a point on this boundary. Finally, we validate our approach in preliminary experiments with human subjects.**

I. INTRODUCTION

Our work is motivated by the design of brain-machine interface systems [1]. These systems translate measurements of brain activity into commands for a prosthetic device, effectively allowing people to control robots just by thinking. Traditional applications include text entry and point-to-point cursor movement but a growing array of future applications include the control of artificial limbs [2], humanoid robots [3], and wheelchairs [4], [5]. In each case, the role of the interface is to facilitate quick and reliable communication of *intent*, i.e., a description of what the user wants the robot to do. Because of the inherent uncertainty in measurement and interpretation of brain activity, this process requires a compact representation of possible intent that lends itself to statistical inference.

As one example, in previous work we enabled a human pilot to fly an unmanned aircraft with input only from an electroencephalograph (EEG), which was used to distinguish between left- and right-hand motor imagery in the brain [6]. These correlates of motor intent came at a low rate and with a low signal-to-noise ratio, e.g., a single bit at 1 Hz with a 10% crossover probability. It was not appropriate to map each input to a control signal as would have been done with a more traditional interface like a joystick. Instead, it took an entire sequence of inputs, prompted by feedback from annotated onboard video, to tell the aircraft what to do. In this case, "what to do" meant a desired path to be followed by an autopilot. A key question, therefore, was how to represent this path.

A. Akce is with the Department of Computer Science and T. Bretl is with the Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, IL, 61801, USA {aakce2,tbretl}@illinois.edu

The reason this question is hard to answer is that the space of all possible paths through a finite-dimensional configuration space is infinite-dimensional. In other words, there exists no finite set of parameters describing elements of this space. We are required to make a choice, and whether we decide to use a parametric representation (e.g., in terms of motion primitives [7]–[10] or a non-parametric representation (e.g., in terms of a gaussian process [11]), we still lose the ability—in practice—to describe arbitrary paths. What this means is that, in the context of a particular application, different representations may lead to very different levels of performance.

In [6], we made a heuristic choice, using an ordered symbolic language to represent paths of piecewise-constant curvature. This choice allowed us to design an interface that was optimal in the sense that it allowed users to specify paths in our language with vanishing error probability in the number of binary inputs. However, this choice restricted the paths that could be flown by the aircraft and made it hard to incorporate certain types of statistical information, for example about how path likelihood varies in the presence of obstacles.

In this paper we suggest one way to make the choice of representation more systematic. We proceed from the observation that, although the space of all paths is infinite-dimensional, paths taken by robots of interest often cluster on a finite-dimensional manifold that is embedded in this space and that is governed by a principle of optimality. This principle of optimality can be used to generate a compact representation of all paths likely to be seen in the context of a particular application.

To make things concrete, in this paper we will consider a point robot moving through a planar workspace with polygonal obstacles. The configuration space of this robot is $\mathcal{Q} = \mathbb{R}^2$. The free space, consisting of all configurations that do not place the robot in collision with obstacles, is $\mathcal{Q}_{\text{free}} \subset \mathcal{Q}$. A path is a continuous function $f\colon [0,1] \to \mathcal{Q}$. We call this path feasible if its range lies entirely in $\mathcal{Q}_{\text{free}}$, i.e. if $f(t) \in \mathcal{Q}_{\text{free}}$ for all $t \in [0,1]$. Given some initial configuration $q_{\text{initial}} \in \mathcal{Q}_{\text{free}}$, we denote the space of all possible paths for which $f(0) = q_{\text{initial}}$ by $C_{\mathcal{Q}}[0,1]$. We call $C_{\mathcal{Q}}[0,1]$ the *solution space*, inspired by the notion that every $f \in C_{\mathcal{Q}}[0,1]$ is a solution to some classical path planning problem [12]–[14]. To generate a representation of the solution space, we will assume that human users choose paths that avoid obstacles and that are locally shortest.

Proceeding from this assumption, in Section II we show that the set of all feasible paths in $C_{\mathcal{Q}}[0,1]$ having length that is bounded and locally minimal is homeomorphic to the unit disk. We also note that any path mapped to the interior of this disk is a subset of some other path mapped to its boundary. Then, in Section III we provide an optimal communication protocol by which the user can, with vanishing error probability, select a point on this boundary using a sequence of noisy binary inputs. We also show the resulting interface and validate our approach in preliminary experiments. Finally, in Section IV we briefly discuss the broader context of our work and present opportunities for future inquiry.

II. A COMPACT REPRESENTATION OF LOCALLY-SHORTEST PATHS

In this section, we show by direct construction that the set of all feasible paths in $C_{\mathcal{Q}}[0,1]$ having length that is bounded and locally minimal is homeomorphic to the unit disk. First, we review the structure of shortest paths in a planar workspace with polygonal obstacles—commonly known as a polygonal domain—and describe efficient ways to compute this structure [14]–[16]. Then, we apply these results to define a smooth bijection between the set of locally-shortest paths with bounded length and the set of points in a closed unit disk.

A. Visibility and the Structure of Shortest Paths

We call a configuration free if the robot at this configuration is not in contact with an obstacle, and call it semi-free if the robot touches an obstacle. We define $\mathcal{Q}_{\text{free}}$ as the set of free and semi-free configurations. If $\mathcal{Q}_{\text{free}}$ is a simple polygon, as it is in the absence of obstacles, there is a unique shortest path between any two configurations. If $\mathcal{Q}_{\text{free}}$ is a polygonal domain with h obstacles, the number of locally-shortest paths can be exponential in h. This is because obstacles induce a combinatorial structure to the paths. We say that two paths π and π' are homotopic if we can continuously deform one to another. The homotopy defines an equivalence relation and this relation divides paths into homotopy classes. For each homotopy class, there is a unique shortest path. We say that a path is globally-shortest if it has the shortest length among all locally-shortest paths.

The shortest paths in a polygonal domain P has a very special structure. We say that two points p, q in $\mathcal{Q}_{\text{free}}$ are visible if there exist a line segment between p and q in $\mathcal{Q}_{\text{free}}$. A vertex v of P is reflex if the interior angle between its two incidents edges is greater than π. Let V_R be the set of reflex vertices in P. The shortest path from q_{initial} to q_{final} is always a polygonal line in C_{free} with vertices chosen from the set $V = V_R \cup \{q_{\text{initial}}, q_{\text{final}}\}$. Its first edge is a line segment from q_{initial} to a reflex vertex and its last edge is a line segment from a reflex vertex to q_{final}. All intermediate edges are line segments between the visible reflex vertices, and they either lie in the boundary of P (a boundary edge) or they are tangents to P at both endpoints (a bitangent edge). The graph containing the boundary and bitangent edges of P between the visible reflex vertices is called a shortest-path roadmap G, or a reduced visibility graph. This graph representation allows us to obtain globally-shortest paths from q_{initial} to q_{final} efficiently. We extend G to contain edges from $\{q_{\text{initial}}, q_{\text{final}}\}$ to the reflex vertices that are visible, and then we do a graph search from q_{initial} to q_{final} in the resulting graph. If multiple globally-shortest path queries from a fixed q_{initial} are going to be performed, we can construct a shortest-path map using the continuous Dijkstra method. This map is a planar decomposition of $\mathcal{Q}_{\text{free}}$ into cells such that all globally-shortest paths ending in the same cell are identical except at their last polygonal segment. Once constructed, this map representation allows us to obtain the globally-shortest paths from q_{initial} without searching a graph. Another version of the shortest path problem is to find the locally-shortest path in a given homotopy class. The homotopy class can be expressed as a sequence of triangles in a triangulation of P. In this case, the shortest path can be computed efficiently using the funnel algorithm. See [14]–[16] for details.

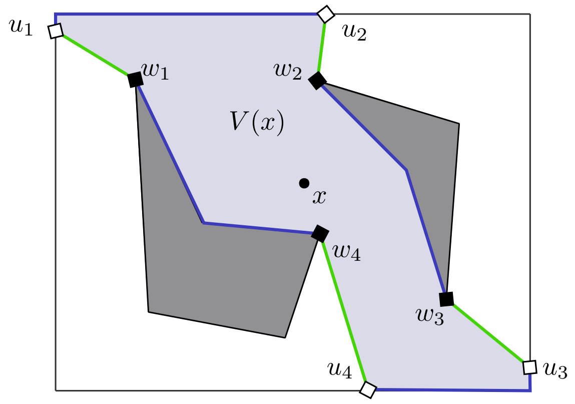

Fig. 1. The visibility polygon of a point x, denoted by $V(x)$, is shown in light gray. The way points from x are marked with black diamonds, and the extension points are marked with empty diamonds. The extension edges are shown in green, and the boundary edges are shown in blue.

For a polygonal domain P, the region of P visible from a source point x is called the visibility polygon of x, and denoted by $V(x)$, see Fig. 1. The visibility polygon can be computed in time $O(n \log n)$ if P has n edges. We refer a line segment in $V(x)$ that crosses the interior of P as an extension edge. For each extension edge, we refer its vertex that is closest to x as a way point and its other vertex as an extension point. The way points are vertices through which shortest paths originated from x can cross. We refer to a polygonal chain between two extension edges in the boundary of $V(x)$ as a boundary edge. This allows us to represent the boundary of $V(x)$ as a sequence of extension and boundary edges.

Our representation for locally-shortest paths is most closely related to gap navigation trees [17]. It is constructed from the sequence of critical events occurring in the robot's visibility region as the robot moves in the environment. This tree structure can be used to solve several visibility-based tasks including locally-optimal navigation.

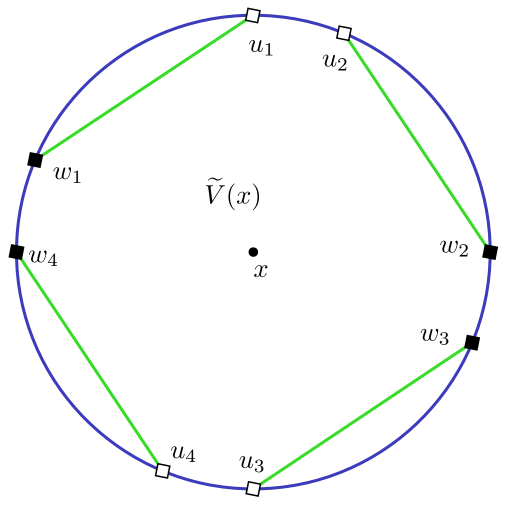

Fig. 2. Embedding the visibility polygon $V(x)$ of Figure 1 to a region (cell) of the unit disk, denoted by $\widetilde{V}(x)$. The boundary edges of $V(x)$ become circular arcs along the disk boundary and the extension edges of $V(x)$ become the chords of the unit disk.

B. *Locally-Shortest Paths as Points in the Unit Disk*

We construct a homeomorphism between the set of all locally-shortest paths (with bounded length) originating from $x \in \mathcal{Q}_{\text{free}}$, denoted by $\Pi(x)$, and the (closed) unit disk, denoted by D^1. We define $\Pi_b(x) \subset \Pi(x)$ to be the set of locally-shortest paths in $\Pi(x)$ that terminate at a semi-free configuration (a boundary point). Our objective is to construct a homeomorphism $f : \Pi(x) \to D^1$ such that the restriction of f to $\Pi_b(x)$ is the boundary of the unit disk, i.e. the unit circle S^1.

We first map the locally-shortest paths ending in the visibility polygon of x, $V(x)$, to a closed planar region (cell) of D^1, denoted by $\widetilde{V}(x)$. The cell $\widetilde{V}(x)$ is bounded by a set of chords in D^1 and a set of circular arcs along the disk boundary. We have a chord for each extension edge in $V(x)$, and we have a circular arc for each boundary edge in $V(x)$. The cyclic ordering of the chords and circular arcs in $\widetilde{V}(x)$ matches to the cyclic ordering of the edges of $V(x)$. An embedding of $V(x)$ of Figure 1 to $\widetilde{V}(x)$ is shown in Figure 2. The shortest paths ending at the boundary edges of $V(x)$ map to the points along the disk boundary, and the shortest paths ending at the extension edges of $V(x)$ map to the points on the chords.

The algorithm then iterates over the way points of $V(x)$. Let $V_{x\to w_i}$ be the set of points that become newly visible by going from x to a way point w_i. All shortest paths ending at a configuration $q \in V_{x\to w_i}$ contain the straight segment from x to w_i in their prefix. $V_{x\to w_i}$ can be computed by cutting the visibility polygon of w_i, $V(w_i)$, by the extension edge from w_i in $V(x)$. The embedding of $V_{x\to w_i}$ is the cell adjacent to the chord in $\widetilde{V}(x)$ that corresponds to the extension edge from w_i. We denote this cell by $\widetilde{V}_{x\to w_i}$. Its boundary, like $\widetilde{V}(x)$, consists of a circular arc for each boundary edge, and a chord for each extension edge. As the algorithm moves to a new way point, a new set of points become visible and

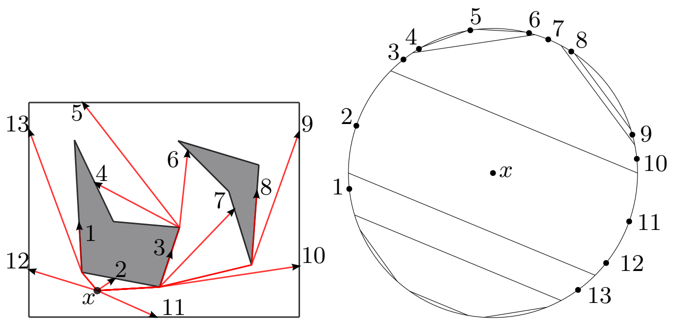

Fig. 4. A set of locally-shortest paths starting at x and ending at the boundary is shown on the left, and their corresponding location in the boundary S^1 of the unit disk is shown on the right.

we map them to the cells that are adjacent to the chords corresponding to extension edges of the previous visibility region. This process is illustrated in Figure 3.

We can map the interior of $V_{p\to q}$ to the interior of $\widetilde{V}_{p\to q}$ by doing a triangulation of $V_{p\to q}$ and matching triangles in $V_{p\to q}$ to the cells of $\widetilde{V}_{p\to q}$ that are either bounded by three chords or bounded by two chords and a circular arc. Any point in the interior of $\widetilde{V}_{p\to q}$ represent a path $\pi \in \Pi(x)$ that terminate in the interior of $V_{p\to q}$. Likewise, any point in the boundary of $\widetilde{V}_{p\to q}$ represents a path $\pi \in \Pi_b(x)$ that terminates in the boundary of $V_{p\to q}$. Figure 4 shows a set of paths in $\Pi_b(x)$ and their mapping to points in S^1.

We can induce an ordering between the paths of $\Pi_b(x)$ by considering their representation in the unit disk, as defined by the mapping $f_b : \Pi_b \to S^1$. Given two paths π_1 and π_2 in $\Pi_b(x)$, we say that π_1 is ordered to the left of π_2, $\pi_1 < \pi_2$, if and only if the clockwise angle from $f_b(\pi_1)$ to $f_b(\pi_2)$ is smaller than the counterclockwise angle from $f_b(\pi_1)$ to $f_b(\pi_2)$ (see Fig. 4). We use this property in the next section to obtain an efficient scheme for choosing a particular path π in $\Pi_b(x)$.

III. APPLICATION TO A HUMAN-ROBOT INTERFACE

In this section, we present an interface that enables a human user to efficiently specify a desired path for a robot moving through a planar workspace with polygonal obstacles using a sequence of noisy binary inputs, as might be derived from a brain-machine interface. First, we use the compact representation of paths from Section II to formulate the problem of interface design as a communication problem. Then, we apply tools from feedback information theory to derive an optimal communication protocol. Finally, we test a preliminary implementation of the resulting interface in simulation (also see the attached video).

A. *Interface Design as a Communication Problem*

Our goal is to design an interface that allows a human user to specify a desired path with a sequence of noisy binary inputs. We may cast this as a communication problem by making the following three choices: (1) to use locally-shortest paths ending at the boundary as desired paths for the mobile robot; (2) to use a classifier to obtain binary ("left"

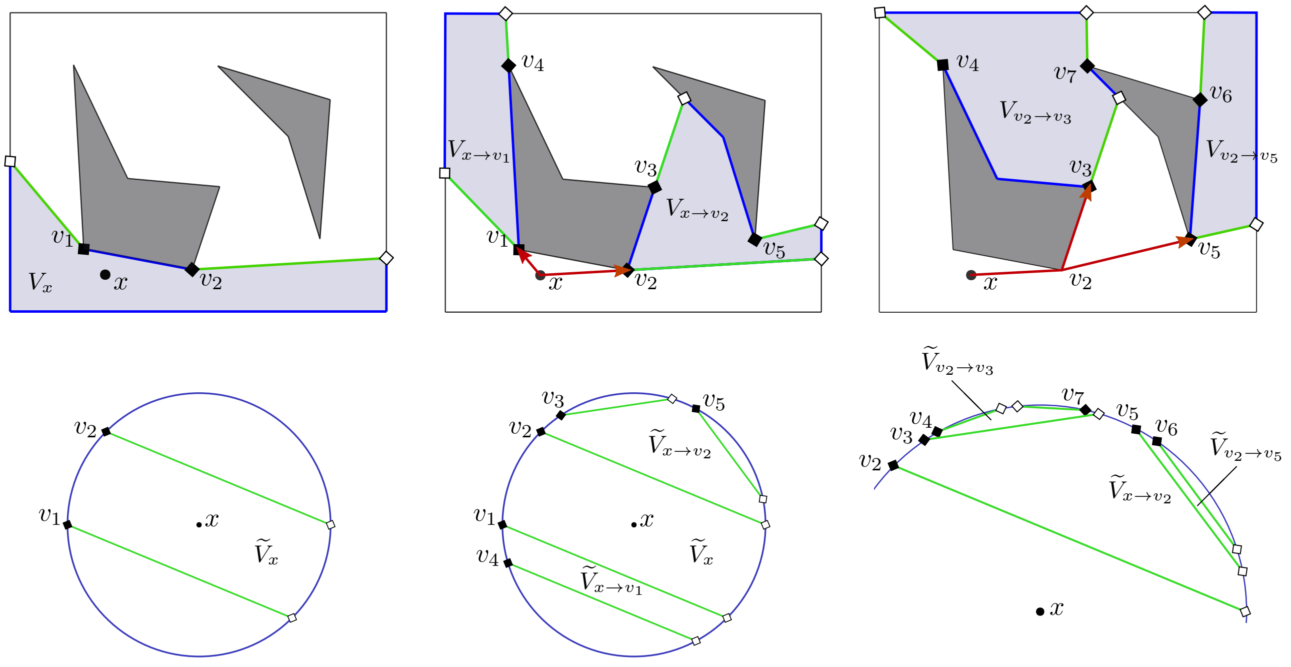

Fig. 3. Three iterations of constructing a unit disk representation of locally-shortest paths. The top row shows the set of points that become newly visible as the algorithm iterates over the way points. The bottom row shows the embeddings of these visibility regions in the unit disk. V_x is the visibility polygon of x, and $V_{p\to q}$ is the set of points that become newly visible by moving from p to q. Their embeddings are $\widetilde{V}_x$ and $\widetilde{V}_{p\to q}$, respectively.

or "right") commands from the user asynchronously; and (3) to use a graphical display to provide feedback to the user.

Our first choice (use locally-shortest paths to boundary) allows us to define the user's objective as specifying a real number in S^1. This choice does not restrict the possible paths for the mobile robot to only paths that end at the boundary. Assume that the user can stop the robot by using a different command or a different input mechanism. Then, the path followed by the mobile robot can be any locally-shortest path. Our second choice (use a classifier to obtain binary commands) allows us to model the noisy input source as a binary symmetric channel (BSC) with crossover probability ϵ. The input to this channel is $x_k \in \{0,1\}$, where we associate $x_k = 0$ with the command to "choose left" and $x_k = 1$ with the command to "choose right". The output of this channel is y_k, where $P(y_k|x_k) = \epsilon$ if $y_k \neq x_k$ and $P(y_k|x_k) = 1-\epsilon$ otherwise. Our third choice (use a graphical display) allows us to assume that the BSC can provide causal noiseless feedback, in this case expressed as a candidate path.

With this abstraction, we can reformulate the problem of interface design as the problem of constructing an optimal communication protocol for transmitting a message point $\theta^* \in S^1$. In the next section, we derive a communication protocol that is both optimal and easily implementable by a human user.

B. Optimal Communication Protocol

Our goal here is to ensure a vanishing probability of error in transmitting the desired path represented by a message point θ^* that is assumed to be drawn uniformly at random from S^1, the set of angles between $[0, 2\pi)$. In the presence of feedback, it is possible to significantly reduce the complexity of the protocol and significantly increase the rate at which error probability decreases (even though the capacity remains constant). This is achieved by a posterior matching scheme [18] that admits exponential decay in the probability of error in transmitting θ^* as we feed binary commands to the channel. It describes a recursive framework to construct optimal feedback communication protocols. For a binary symmetric channel, we can construct such a protocol as follows: Assume that after some time step k, the interface computed the posterior distribution $P_{\Theta|Y^k}(\theta|y_1\dots y_k)$. First, the interface finds the *median pair* $(\mu_k, \bar{\mu}_k)$, a pair of angles with the following property:

$$\bar{\mu}_k = (\mu_k + \pi) \mod (2\pi),$$
$$P_{\Theta|Y^k}(\mu_k \leq \Theta < \bar{\mu}_k) = P_{\Theta|Y^k}(\bar{\mu}_k \leq \Theta < \mu_k) = 0.5.$$

This says that μ_k and $\bar{\mu}_k$ are opposite angles in S^1, and the probability concentrated on the half circle from μ_k to $\bar{\mu}_k$ is equal to the probability concentrated on the opposite half circle from $\bar{\mu}_k$ to μ_k (see Fig. 5). The interface selects the median with higher posterior density as the estimate θ_k, and provides it (expressed as a locally-shortest path in the graphical display) as feedback to the user. Then, the user selects the next input x_{k+1} as 1 if $\theta^* \geq \theta_k$ or as 0 otherwise. Finally, if $y_{k+1} = 1$ (the case $y_{k+1} = 0$ is analogous), then the interface applies Bayes' rule to update the posterior distribution as

$$P_{\Theta|Y^{k+1}}\left(\theta|y_1\dots y_{k+1}\right) =$$
$$\eta \cdot \begin{cases} (1-\epsilon)\cdot P_{\Theta|Y^k}\left(\theta|y_1\dots y_k\right) & \text{if } \theta_k \leq \theta < \bar{\theta}_k \\ \epsilon \cdot P_{\Theta|Y^k}\left(\theta|y_1\dots y_k\right) & \text{otherwise,} \end{cases} \tag{1}$$

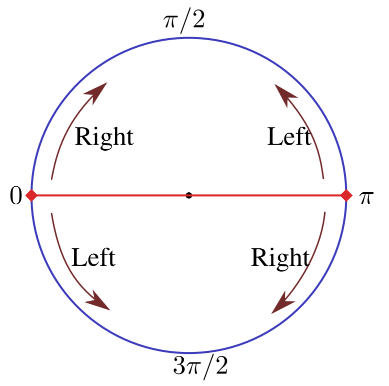

Fig. 5. The placement of $\theta \in S^1$ to the boundary of the unit disk. $(0, \pi)$ is a possible median pair for the uniform distribution over S^1. Each median divides S^1 into a left half-circle and a right half-circle.

where η is a normalizing constant. The new estimate θ_{k+1} is computed from the median pair of this distribution (see Fig. 6). Remarkably, this scheme is not only optimal but also easy for a human user—the "encoder"—to implement. Assume a graphical display shows the user the locally-shortest path γ_k represented by θ_k (a point on the boundary of our unit disk representation). Then, the user only has to decide if the desired path is ordered to the left ("choose left") or to the right ("choose right") of the candidate (see Fig. 6). In our preliminary interface, described in the next section, we use this scheme to allow users efficiently specify desired paths for a mobile robot.

C. Preliminary Implementation of the Resulting Interface

We built an interface based on simulation to validate our approach for specifying desired paths. We considered two applications. In the first application (static path selection), the user chooses a locally-shortest path $\pi^* \in \Pi_b(x)$ from a given starting point x . The interface displays its estimate of the user's desired path $\pi_k \in \Pi_b(x)$ in each step k. The user provides a "left" command if $\pi^* < \pi_k$, or a "right" command otherwise. In the second application (dynamic path selection), an omnidirectional robot navigates while the user is specifying a path $\pi^* \in \Pi_b(x)$. In this case, the interface operates as follows: First, it obtains the estimate path π_k from its observation of binary commands, like before. Second, it computes the locally-shortest path from the robot's current state q to the terminal point of π_k. Third, it displays this path to the user. In this case, the user's policy does not change. The ordering between the paths is still preserved (defined by the unit disk representation of $\Pi(x)$) and it is intuitive for the users to determine. This application is illustrated in Fig. 7.

We demonstrate the preliminary interface in the attached video. In the first part, we show the unit disk representation for a polygonal workspace. We illustrate how the paths change in the workspace as we move towards left and right on the boundary of the disk. In the second part, we demonstrate the static path selection task, where the user provides left or right inputs to specify a target path γ^* (shown in the display) for a stationary robot. During the execution of the task, the last input y_k provided by the user and the number of inputs obtained so far, k, are shown in the video.

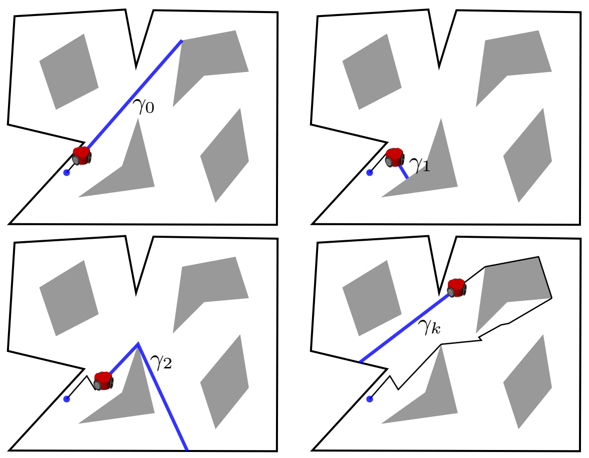

Fig. 7. The four snapshots illustrating the dynamic version of the interface, where the mobile robot navigates while the user is specifying their desired path. The candidate path (blue) and the actual path followed by the mobile robot (black) are shown. The user compares their desired path (not shown) to the candidate path and provides a "left" or "right" command. At first, the interface shows γ_0. Upon receiving a "right" command, it updates the candidate path to γ_1. In the next step, after a "left" command, the path becomes γ_2. The last frame shows an intermediate step.

After each input, the interface updates a posterior distribution over S^1 and computes the median θ_k. The value of θ_k and the shortest path γ_k that it represents are displayed in the video. As we receive inputs from the user, observe that γ_k converges to the target path γ^*, and the median θ_k converges to θ^*. In the third part, we demonstrate the dynamic path selection task, where the user navigates an omnidirectional robot amongst obstacles. In this case, the robot moves at a fixed speed along the path π_k corresponding to the current median θ_k. The interface displays shortest paths from its current state to the boundary point represented by θ. In this task, the user cannot specify the target path correctly. This is because the robot changes its course immediately if the line segment it is following changes after a user input. In order to avoid frequent changes in the course of the robot in practice, we may start the navigation task after observing a small number of inputs from the user.

IV. CONCLUSION

We presented a compact representation for locally-shortest paths and used this representation to enable a systematic way of designing human-robot interfaces for specifying a desired path with noisy and low-bandwidth input devices. First, we noted that the paths encountered in "real" robotics systems are often governed by a principle of optimality. This motivated us to restrict the set of all possible paths for the robot to the space of all locally-shortest paths from a starting point. Second, we constructed a homeomorphism between this finite-dimensional space and the unit disk. This representation allowed us to represent any locally-shortest path ending at the boundary of the workspace as a point in the unit circle. Third, we provided an optimal communication protocol that allowed the user to specify a point in S^1, and correspondingly a locally-shortest path, by doing a

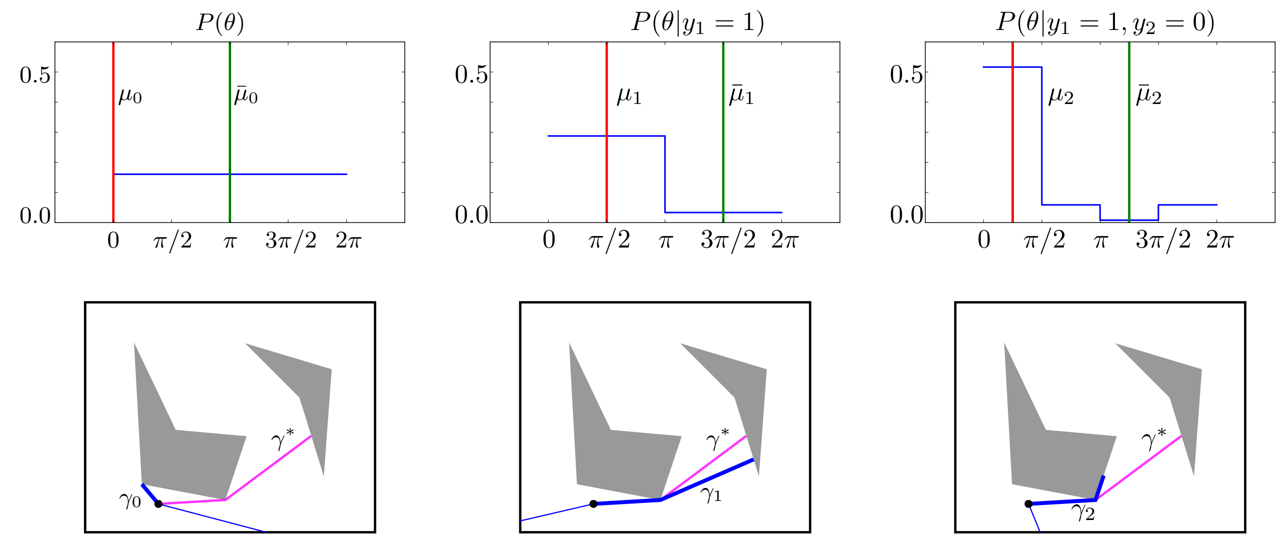

Fig. 6. The communication protocol between the user and the interface to a mobile robot for specifying a locally-shortest path. The user provides binary commands to specify the desired path γ^*. The interface maintains a probability distribution over S^1, and computes the median pair. The graphical display shows the locally-shortest path corresponding to the median with the higher probability density. In the initial step, shown in the first frame, the posterior is uniform and its chosen median μ_0 is displayed as the path γ_0. Upon observation of a "choose right" command, the likelihood of paths in the right half-circle of μ_0 are increased. As a result, the displayed path moves to the right of γ_0, as shown in the second frame. Similarly, the third frame shows the new posterior and the corresponding path after an observation of "choose left" command.

simple comparison between two paths. Experiments with a preliminary interface showed that this approach is feasible.

We believe that it is important to characterize the space of solutions to a motion planning problem. In the application we considered it was essential to obtain a compact representation of this solution space in order to specify an element of it efficiently. Although we considered locally-shortest paths here, our approach might be applicable in more general settings. In future work, we would like to consider geodesics in the presence of a cost function. One approach is to use potential field functions to make paths less likely to become close to obstacles or to make them arrive to possible goal locations quickly.

One of our current goals, as future work, is to use our approach in a brain-machine interface to enable the navigation of a mobile platform, or wheelchair amongst obstacles using only EEG as input.

V. ACKNOWLEDGMENTS

This research was supported by awards NSF-CNS-0931871, NSF-CMMI-0956362, and NSF-CMMI-0955088.

REFERENCES

[1] N. G. Hatsopoulos and J. P. Donoghue, "The science of neural interface systems," *Annual Review of Neuroscience*, vol. 32, no. 1, pp. 249–266, 2009.

[2] T. A. Kuiken, G. Li, B. A. Lock, R. D. Lipschutz, L. A. Miller, K. A. Stubblefield, and K. B. Englehart, "Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms," *JAMA*, vol. 301, no. 6, pp. 619–28, Feb 2009.

[3] C. J. Bell, P. Shenoy, R. Chalodhorn, and R. P. N. Rao, "Control of a humanoid robot by a noninvasive brain-computer interface in humans," *Journal of Neural Engineering*, vol. 5, no. 2, pp. 214–220, 2008.

[4] B. Rebsamen, E. Burdet, C. Guan, H. Zhang, C. L. Teo, Q. Zeng, C. Laugier, and M. H. Ang Jr., "Controlling a wheelchair indoors using thought," *Intelligent Systems, IEEE*, vol. 22, no. 2, pp. 18–24, 2007.

[5] I. Iturrate, J. Antelis, A. Kubler, and J. Minguez, "A noninvasive brain-actuated wheelchair based on a p300 neurophysiological protocol and automated navigation," *Robotics, IEEE Transactions on*, vol. 25, no. 3, pp. 614–627, June 2009.

[6] A. Akce, M. Johnson, and T. Bretl, "Remote teleoperation of an unmanned aircraft with a brain-machine interface: Theory and preliminary results," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, May 2010, pp. 5322 –5327.

[7] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Trans. Robot.*, vol. 25, no. 1, pp. 116–129, 2002.

[8] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics. Springer, 2005, vol. 15, pp. 365–374.

[9] K. Yamane, J. J. Kuffner, and J. K. Hodgins, "Synthesizing animations of human manipulation tasks," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 532–539, 2004.

[10] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 61–70, Mar. 2007.

[11] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, Mass.: MIT Press, 2006. [Online]. Available: http://www.loc.gov/catdir/toc/fy0614/2005053433.html

[12] J.-C. Latombe, *Robot Motion Planning*. Boston, MA: Kluwer Academic Publishers, 1991.

[13] H. Choset, K. Lynch, S. Hutchinson, G. Kanto, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, 2005.

[14] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at http://planning.cs.uiuc.edu/.

[15] J. S. B. Mitchell, "Shortest paths and networks," in *Handbook of Discrete and Computational Geometry, 2nd Ed.* Chapman and Hall/CRC Press, 2004, pp. 607–641.

[16] J. O'Rourke, "Visibility," in *Handbook of Discrete and Computational Geometry, 2nd Ed.* Chapman and Hall/CRC Press, 2004, pp. 643–663.

[17] B. Tovar, L. Guilamo, and S. M. LaValle, "Gap navigation trees: Minimal representation for visibility-based tasks," in *Proc. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2004.

[18] O. Shayevitz and M. Feder, "Optimal feedback communication via posterior matching," *Arxiv preprint 0909.4828*, 2009, submitted to IEEE Transactions on Information Theory.