Experimental Study of Curvature-based Control Laws for Obstacle Avoidance

Fumin Zhang, Alan O'Connor, Derek Luebke and P. S. Krishnaprasad

Institute for Systems Research & Department of Electrical and Computer Engineering
University of Maryland at College Park
College Park, MD 20742
{fuminz, ocon, derek, krishna}@isr.umd.edu

Abstract— A novel curvature-based steering control law is introduced to produce the obstacle avoidance behavior for unicycle type robots traveling (flying) at constant speed. Different methods of curvature estimation from noisy range data are explored and compared via experiments. Performance of the obstacle avoidance algorithm is investigated through simulations and experiments.

I. INTRODUCTION

In an unknown environment, a moving robot has to be able to avoid collision in order to survive. Currently, when a map is not provided, there are two major categories of obstacle avoidance algorithms and control laws. The first category is potential function (or navigation function) based algorithms as introduced in [7], [10], [12] and various other papers. A robot follows the gradient vector field generated by some artificial potential function assigned to the workspace. Detection of new obstacles will cause a recalculation of the potential function. This approach is computationally efficient and easy to implement. Furthermore, with appropriate dynamic model, the algorithm produces an explicit control law for the robot. But it has two drawbacks. First, the potential function will almost certainly have critical points other than the goal. Hence the robot might be trapped near such a critical point. Second, the path of the robot may be hard to compute and predict because of the complexity of the potential functions. The second category of algorithms is based on local path planning, a typical case is the "bug" family originating from the work of Lumelsky and Stepanov[8], where a robot is instructed to follow boundary curves of obstacles. It is proved that by following the boundary curves of the obstacles and leaving the boundary when a certain condition is satisfied, a robot is guaranteed to reach the goal. Some interesting results on the optimality aspects of local path planning is addressed in [13]. The path-planning algorithms always produce well-defined paths. However, it does not tell the robot how to follow such a path. One has to solve the inverse dynamics problem to find an applicable control law. This process is difficult and a solution is not guaranteed.

Because it is very difficult to avoid the unwanted critical points of the navigation functions, a recent development in this category is to keep the robot moving at a constant speed with steering control. With the constant speed assumption, such a control law has a gyroscopic nature [16]. The authors of [14], [15] and [18] give empirical ways to design such a vector field. However, the methods are still in their preliminary stages and are only applied to a limited number of special cases. In [11], the authors proposed a navigation steering law with obstacles treated via potential functions on the space of steering angles. This paper has inspired the group led by Warren to investigate the steering behavior of human beings in a non-static environment [3]. Some of the latest interesting results are published in [4]. This approach is promising when the obstacle is relatively small compared to the aperture of the range sensors.

In a two dimensional space, the boundary curves of the obstacles provide cues for the design of steering vector fields. An oriented boundary curve is determined by its curvature function in the canonical Frenet-Serret frame [9] if one knows the position of one point on the curve. The curvature function can be measured from the range data obtained by sensors such as sonar or laser range finder (ladar). By using such curvature measurements, we have given a systematic way to produce gyroscopic steering vector fields which guarantee avoiding obstacles and reach of the goal. The mathematical results are presented in [17]. We take advantage of new developments in shape theory [19] and formation control [6]. The essence is that we not only produce a justifiable control law, but also predict well-defined paths for the robot.

The performance of our control laws depends on curvature estimates obtained by a robot as a result of analyzing its sensory information. Because of the derivative nature of curvature, even when a high accuracy range sensor such as a laser range finder is employed, the curvature estimates still contain significant noise. Therefore, it is

important for us to study all possible ways of getting smooth curvature estimates and compare the performance of different methods. We tested two appealing methods. The first is proposed and analyzed in [2] for biological data, where the classical curvature estimate is employed with an abundant set of empirical rules to improve the accuracy. The second method is examined in [1], where a geometric curvature estimate is established. We made a slight extension to this method to improve its accuracy. The performance of these two methods are compared via experiments.

Our algorithms are implemented on a Pioneer II mobile robot with laser range finder. The feedback control law is implemented in the form of an MDLe atom [5]. We carried out a series of experiments to test the performance of the obstacle avoidance and boundary following ability of the control law. Selected results are presented in section IV.

II. CURVATURE-BASED CONTROL LAWS FOR OBSTACLE AVOIDANCE

The controlled motion of a unicycle type robot is modeled as:

$$\begin{align}
\dot{x} &= v\cos(\theta) \\
\dot{y} &= v\sin(\theta) \\
\dot{\theta} &= u \,,
\end{align} \tag{1}$$

where $(x,y) \in \mathscr{R}^2$ is the location of the robot in the plane and θ determines its heading angle in a laboratory coordinate system. The control v is speed and u the steering control. This model is a good approximation of the planar motion of a variety of aerial or ground robots. As in the case of unmanned aerial vehicles, one often prefers to set v as a constant and only control u. *We make the assumption that v is always set to unit speed.* Therefore, our goal is to design a control law for u to achieve obstacle avoidance.

We note that under the unit speed assumption, equation (1) can be viewed as representing the motion of a charged particle in a magnetic field of strength proportional to u.

A. Introduction to the control law

Robots are usually equipped with range sensors, typically sonar or laser range finder. Usually, range sensors are able to scan a cone centered around the heading direction of the robot. We assume we have sufficiently many sensor rays. When a sensor ray detects an obstacle, i.e. the ray intersects the (oriented) boundary curve of the obstacle, then the tangent vector and the *plane curvature* [9] at the point of intersection can be estimated by the methods of section III. Let α denote the angle measured counterclockwise from the heading direction of the robot to a selected sensor ray. If this ray detects a point on the boundary curve of the obstacle, then we denote by ϕ_α the

angle made by the vector tangent to the boundary curve at that point, with respect to the laboratory frame. Further, we let r_α be the corresponding range measurement and κ_α be the curvature of the boundary curve at the detected point. When $\alpha = 0$, all measurements are associated to the central ray of the sensor (laser range finder).

In [17], we showed that a robot not only can avoid an obstacle but also can follow the boundary curve of that obstacle by utilizing measurements obtained from two sensor rays, one at $\alpha = 0$, the other at $\alpha = -\pi/2$. One of the technical details is that, when either of the sensor rays fails to detect a boundary curve (say due to being out of range), then we simply assign $\phi_\alpha = \theta$, $\kappa_\alpha = 0$ and $r_\alpha = r_{max}$ when $\alpha = 0$ and $r_\alpha = r_c$ when $\alpha = -\pi/2$, where r_{max} is the maximal range for a sensor ray and r_c is the desired proximity to the detected boundary curve during wall-following. We could interpret these assignments as determining virtual boundary curves with tangent vectors aligned to the heading direction of the robot. These virtual boundary curves have zero curvature at the virtual point of intersection at distance r_{max} or r_c as the case may be.

The control laws derived from [17] are based on a control Lyapunov function

$$V = 4\sin^2\left(\frac{\theta - \phi_0}{4}\right) + 4\sin^2\left(\frac{\phi_0 - \phi_{-\frac{\pi}{2}}}{4}\right) + g(r_{-\frac{\pi}{2}}), \tag{2}$$

where we define

$$g(z) = \int_0^z f(\sigma)d\sigma \,. \tag{3}$$

In this equation, the function $f(\cdot)$ is defined as a smooth function satisfying

$$\begin{align}
f(z) &= 0 \ \text{when} \ z = r_c \\
f(z) &< 0 \ \text{when} \ 0 \le z < r_c \\
f(z) &> 0 \ \text{when} \ z > r_c \,.
\end{align} \tag{4}$$

An example is $f(z) = (z - r_c)^3$ and $g(z) = \frac{1}{4}(z - r_c)^4$.

The intuition behind the use of this particular function V is that if a feedback control law is such that $\dot{V} \le 0$ along closed loop trajectories, then a boundary following behavior would result as the limiting behavior (motion state) corresponding to $\theta = \phi_0 = \phi_{-\frac{\pi}{2}}$ and $r_{-\frac{\pi}{2}} = r_c$. Details of the corresponding mathematical arguments will appear in [17]. In the setting of finite range sensors, the role of initial condition is important.

We note that in the control Lyapunov function, the quantities $(\theta - \phi_0)$, $(\phi_0 - \phi_{-\frac{\pi}{2}})$ and $r_{-\frac{\pi}{2}}$ are independent of the choice of laboratory frame. They can be measured in a coordinate frame attached to the robot. Therefore the function V and the derived control law u are also independent of the choice of laboratory frame.

B. Simulation results

Our control law is first tested in simulations in Matlab and we show results as in Figure 1, 2 and 3. In every simulation we assign a one obstacle environment. The first obstacle has an elliptical boundary. The second obstacle has a convex polygonal boundary and the third has a non-convex polygonal boundary. In all three cases the robot successfully avoids the obstacles and follows the boundary curves.
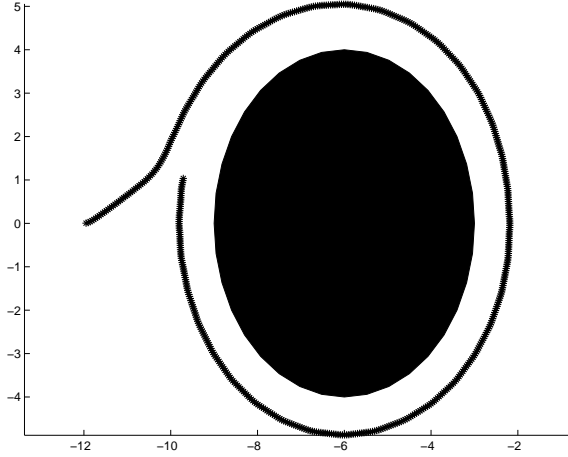

Fig. 1. Matlab simulation of following the boundary of an elliptical obstacle. The robot is instructed to keep unit distance from the obstacle boundary to its right.
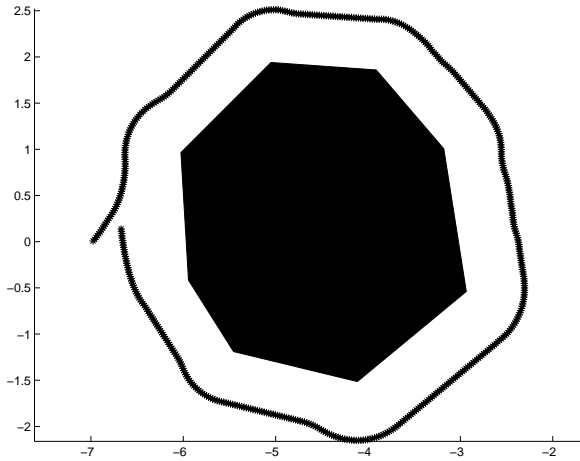

Fig. 2. Matlab simulation of following the boundary of a polygonal obstacle. The robot is instructed to keep 0.5 unit distance from the obstacle boundary to its right.

III. CURVATURE ESTIMATION FROM LADAR MEASUREMENTS

Because curvature is important to our obstacle avoidance control law, obtaining an accurate estimate of cur-
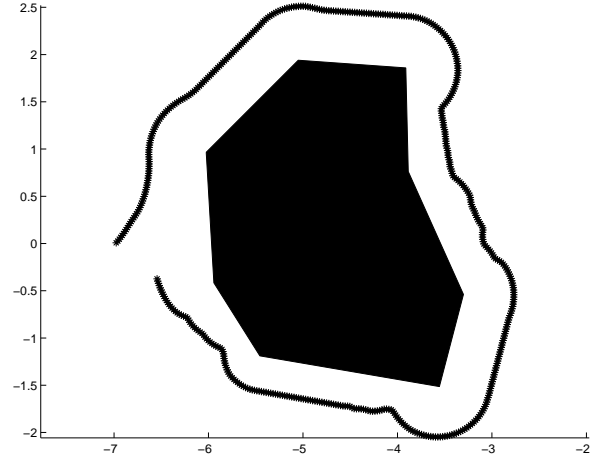

Fig. 3. Matlab simulation of following the boundary of a polygonal obstacle. The robot is instructed to keep 0.5 unit distance from the obstacle boundary to its right. In all three figures the robot is initially heading on a collision course towards the obstacle.

vature is fundamental. However, curvature estimation involves the computation of derivatives. Such processes are prone to noise and small measurement errors. Filtering and averaging techniques need to be applied.

We use a SICK LMS-200 laser range finder to obtain range measurements. The range-finder takes a 180 degree scan in the horizontal plane, measuring distance with an angular resolution of one degree. The range-finder has a range of 10m with relative error typically less than 0.8%. Notice that even such small relative error will still cause significant inaccuracy in the curvature estimates in the absense of filtering or averaging.

A. Curvature estimates

If a regular plane curve is parameterized as $\mathscr{C}(t)$ where t is not necessarily the arc-length, then the curvature can be calculated using the definition

$$|\kappa(t)| = \frac{\left\| \dot{\mathscr{C}}(t) \times \ddot{\mathscr{C}}(t) \right\|}{\left\| \dot{\mathscr{C}}(t) \right\|^3} . \tag{5}$$

(Recall that plane curvature is signed.) However, since this calculation requires the computation of the second derivative, it is sensitive to noise.

If the curve is parameterized by its arc-length then the curvature is

$$|\kappa(s)| = \left\| \frac{dT(s)}{ds} \right\| = \frac{d\theta}{ds} \tag{6}$$

where θ denote the direction of the tangent vector $T(s)$. Therefore, an estimate of the curvature is obtained by

$$|\widehat{\kappa}(s)| = \frac{\delta\theta}{\delta s} \tag{7}$$

Let P_{n-w}, P_n, P_{n+w} denote three points on a regular curve. Then

$$\delta\theta = arccos\left(\frac{< P_{n+w} - P_n, P_n - P_{n-w} >}{\left\| P_{n+w} - P_n \right\| \left\| P_n - P_{n-w} \right\|}\right) \quad (8)$$

and

$$\delta s = \left\| P_{n+w} - P_{n-w} \right\| \quad (9)$$

where w is a positive integer called the window size. This estimates is referred as the classical curvature estimate and is used in [2].

Another estimate of curvature is proposed in [1] as follows. Let $a = \left\| P_n - P_{n-w} \right\|$, $b = \left\| P_{n+w} - P_n \right\|$, $c = \left\| P_{n+w} - P_{n-w} \right\|$ and $s = (a+b+c)/2$. We draw the unique circle passing all three points. By applying Heron's formula, the curvature of such a circle is

$$|\widehat{\kappa}(s)| = 4\frac{\sqrt{s(s-a)(s-b)(s-c)}}{abc} . \quad (10)$$

In [1], it is proved that $\widehat{\kappa}$ is a good estimate of κ when the difference $(a-b)$ is sufficiently small. We refer to this estimates as the geometric estimate of curvature.

B. Noise reduction methods and comparasion

Both the classical and the geometric estimate of curvature are sensitive to noise in the range data. In order to reduce estimation error, a straightforward approach is to apply a Gaussian filter to the position data, or use a running average on the position data and the curvature estimates. The improvements depend too much on one's experience and the environment. Therefore, this class of methods is difficult to systemize.

In [2], various techniques used to filter curvature and torsion data obtained from three-dimensional trajectories of microorganisms are investigated. Empirically, the authors noticed that the ratio between the noise in the position at a point and the distance between points can be interpreted as the signal-to-noise ratio. In order to improve this ratio, one can increase the window size when calculating $\delta\theta$ and δs. However, the window size cannot be arbitrarily large, otherwise the estimate becomes severely biased. Therefore, one has to deal with the trade-off between the signal-noise ratio and the resolution by varying the window size. Because they are mainly concerned with off-line analysis of collected biological data, this trade-off is not too difficult to handle. However, it becomes much more tricky when real-time estimates have to be obtained for steering.

Meanwhile, the authors of [1] also noticed that a larger window size ought to be used. But they use a fixed window size with the geometric estimate. Here, we make a slight extension by using an averaged curvature estimate across varying window size. For example,

$$\bar{\kappa}(n) = \frac{1}{3}\sum_{w=7}^{9} \widehat{\kappa}(P_{n-w}, P_n, P_{n+w}) \quad (11)$$

where, with slight abuse of notation, $\widehat{\kappa}(P_{n-w}, P_n, P_{n+w})$ denotes the geometric estimate of curvature obtained at the nth point with window size w.

These different estimates are compared experimentally. We use a ladar to measure the curvature of two objects. One is a file cabinet which has flat sides and sharp corners, another is a trash-can with near cylinderical shape. Table 1 shows the averaged RMS error using different estimates on filtered or unfiltered data. We noticed that using larger window size eliminates the need for Gaussian filtering. Figure 4 and 5 plots the geometric estimates with regard to the scanning angles.

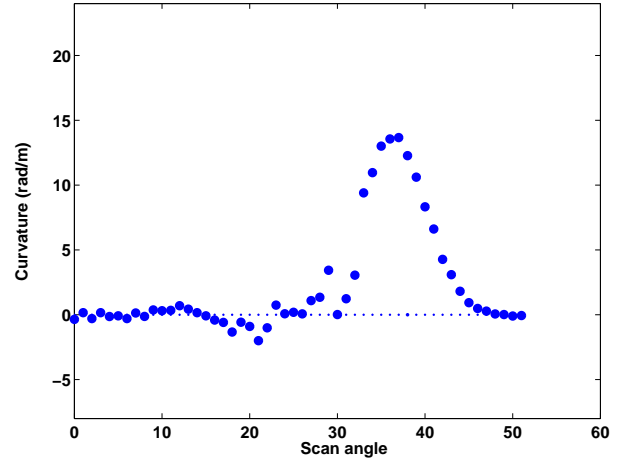| Averaged RMS Error of Curvature Estimates			
Estimates	Window Size	Trash-can	Cabinet
Geometric	1-3	15.47	13.47
Estimates	4-6	1.72	4.29
(Unfiltered)	7-9	0.71	3.43
Geometric	1-3	5.12	7.74
Estimates	4-6	1.23	4.40
(Filtered)	7-9	0.58	3.56
Classical	1	14.52	15.75
Estimates	5	1.87	5.46
(Unfiltered)	8	0.79	4.51
Classical	1	6.44	10.35
Estimates	5	0.79	5.53
(Filtered)	8	0.80	4.73

Table 1

Fig. 4. The geometric curvature estimates over varying window size $7-9$ of a cabinet with sharp corners. The x-axis represents the scanning angles of the ladar. The thinner dotted line is the zero curvature line. One can see clearly that the sharp angle of the cabinet caused a spike in the curvature estimates.

IV. EXPERIMENTAL RESULTS ON ROBOT EQUIPPED WITH MDLe

After testing and verifying different estimates in obtaining the curvature estimates, we decided to use the geometric estimate with varying window size $7-9$ to
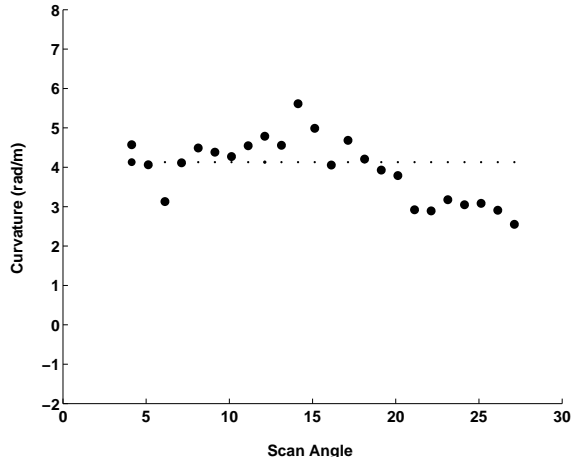
Fig. 5. The geometric curvature estimates over varying window size $7-9$ of a near cylinderical trash-can. The x-axis represents the scanning angles of the ladar. The thinner dotted line is the averaged constant curvature line. One can see that the trash-can is not a perfect cylinder.

Fig. 7. The curvature of the sidewall measured by the robot. The unit for the x-axis is meter and the unit for the y-axis is rad/meter. The data is still noisy but averaged near zero. It is easy to tell that there are four doors on the sidewall because of the spike pairs caused by the door jambs.

process online range data. The estimation algorithm is implemented as a module of the Motion Description Language system used to control a Pioneer II robot equipped with a laser range finder. The control law is implemented as an MDLe atom [5].

We tested the obstacle avoidance control law in the corridor of A.V. Williams building at University of Maryland. Such an environment has straight side-walls with *closed* doors. The results are shown in Figure 6 and 7.
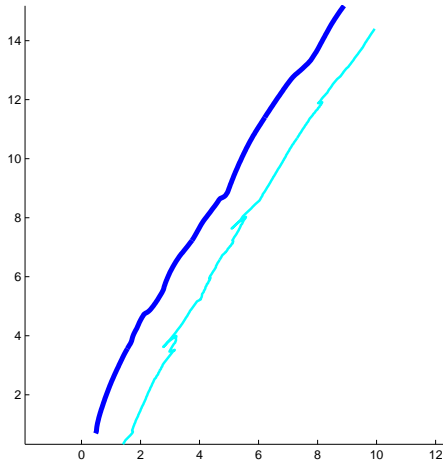

Fig. 6. The path of the robot in a hallway with closed doors. The unit for both the x-axis and the y-axis is meter. The thicker line on the left is the path. The thinner line on the right is the sidewall detected by the robot. Due to odometry error, the wall does not appear as a perfect straight line. Also, it is hard to tell where the doors are.

We then took the robot outside the building and tested our algorithm along a curved garden wall nearby the building. The results are shown in Figure 8 and 9.
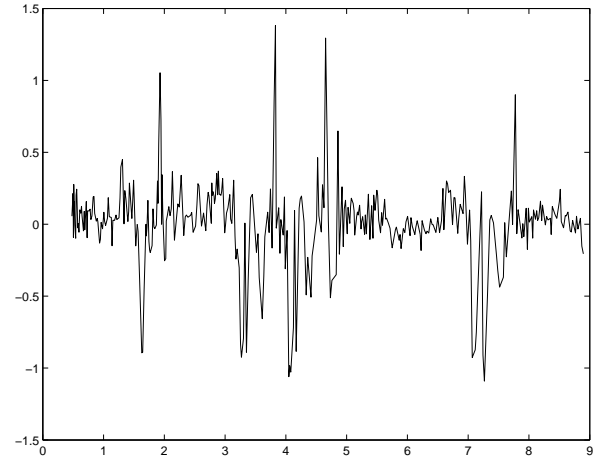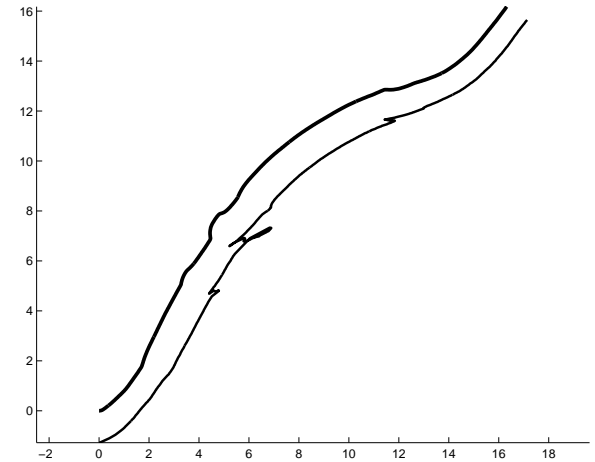

Fig. 8. The path of the robot following a curved garden wall. The units for both the x-axis and the y-axis is meter. The thicker line on the left is the path. The thinner line on the right is the garden wall detected by the robot. The small hump is caused by a pair of legs of someone sitting on the wall.

V. SUMMARY AND FUTURE WORK

In this paper the simulation and experimental results for one of the curvature-based navigation control laws developed in [17] are presented. The results indicate that such curvature-based control laws are practical and produce expected behaviors. However, in order to deal with the complexity of the environments one has to devote more effort in solving problems such as the estimation of curvature from noisy range data. We will continue our experiments on other control laws in [17] that have been theoretically justified. At the same time, we will continue to investigate the techniques required to improve the performance of this class of control laws in practice.
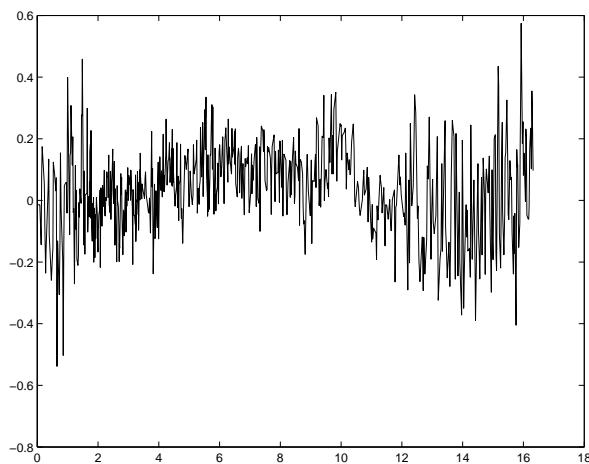
Fig. 9. The curvature of the garden wall measured by the robot. The unit for the *x*-axis is meter and the unit for the *y*-axis is rad/meter.

VI. Acknowledgment

The authors would like to thank Dr. Eric Justh for valuable comments and discussions about this paper.

VII. References

[1] E. Calabi, P. J. Olver, C. Shakiban, A. Tannenbaum, and S. Haker. Differential and numerically invariant signature curves applied to object recognition. *International Journal on Computer Vision*, 26:107–135, 1998.

[2] H. C. Crenshaw, C. N. Ciampaglio, and M. McHenry. Analysis of the three-dimensional trajectories of organisms: Estimates of velocity, curvature and torsion from positional information. *Journal of Experimental Biology*, 203:961–982, 2000.

[3] B. R. Fajen and W. H. Warren. The behavioral dynamics of steering, obstacle avoidance, and route selection. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):343–362, 2003.

[4] B. R. Fajen, W. H. Warren, S. Temizer, and L. P. Kaelbling. A dynamical model of visually-guided steering, obstacle avoidance, and route selection. *International Journal of Computer Vision*, 54:13–34, 2003.

[5] D. Hristu, P. S. Krishnaprasad, S. Andersson, F. Zhang, P. Sodre, and L. D'Anna. The mdle engine: a software tool for hybrid motion control. *ISR Technical Report TR2000-54, UMCP*, 2000.

[6] Eric Justh and P.S. Krishnaprasad. Equilibria and steering laws for planar formations. *To appear in System and Control Letters*, 2003.

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

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

[9] R. S. Millman and G. D. Parker. *Elements of Differential Geometry*. Prentice-Hall, 1977.

[10] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, 1992.

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

[12] R. Shahidi. Mobile robot navigation using potential functions. Master's thesis, Department of Electrical and Computer Engineering, University of Maryland, 1989.

[13] Z. Shiller. Online suboptimal obstacle avoidance. *The International Journal of Robotics Research*, 19(5):480–497, 2000.

[14] L. Singh, H. Stephanou, and J. Wen. Real-time robot motion control with circulatory fields. In *Proc. 1996 IEEE International Conference on Robotics and Automation*, pages 2737–2742, Minneapolis, Minnesota, 1996. IEEE.

[15] L. Singh, J. Wen, and H. Stephanou. Motion planning and dynamic control of a linked manipulator using modified magnetic fields. In *Proc. 1997 IEEE International Conference on Control Applications*, pages 9–15, Hartford, CT, 1997. IEEE.

[16] L.-S. Wang and P. S. Krishnaprasad. Gyroscopic control and stabilization. *Journal of Nonlinear Science*, 2:367–415, 1992.

[17] F. Zhang, E. Justh, and P. S. Krishnaprasad. Steering control, curvature and Lyapunov navigation. *Preprint*, 2003.

[18] Fumin Zhang. Geometric control of formations. *Ph.D. proposal, University of Maryland*, 2002.

[19] Fumin Zhang, Michael Goldgeier, and P.S. Krishnaprasad. Control of small formations using shape coordinates. In *Proc. of 2003 International Conference of Robotics and Automation*, pages 2510–2515, Taipei, Taiwan, 2003. IEEE.