Determining stable trajectories of blind, bouncing robots

Alexandra Nilles

Abstract

Mobile robots with limited sensing often navigate by travelling in straight lines until encountering an obstacle. We consider a robot which can align itself to a certain fixed angle relative to the environment boundary and drive in straight lines in the free space.

Previous work introduced an algorithm that, given a bounce angle and an environment, identifies parts of the boundary which will no longer be visited after a bounded number of bounces or distance travelled. **awkward sentence, fix**

However, if the robot dynamics included stable cycles of length longer than two bounces, the previous algorithm would not terminate. In this paper, we analytically determine the location and stability of such *limit cycles* for regular polygons, and regular polygons under affine transformations. This analysis leads to simple, open-loop control schemes that allow either predictable and stable "patrolling" dynamics, or chaotic, ergodic "exploratory" dynamics. The results are useful for controlling simple mobile robots with minimal sensing and actuation, navigating in spaces with known geometry.

Introduction

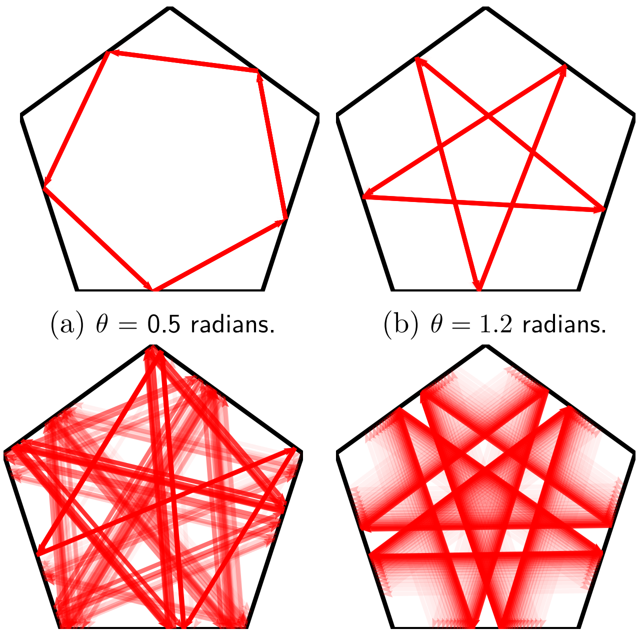
Consider the dynamics of a simple mobile robot, such as a robotic vacuum with a contact or proximity sensor, as it navigates a room. More and more, such robots are able to compute high-resolution estimates of their workspace and their position in it, using techniques such as SLAM and multi-sensor fusion. However, for some tasks such as patrolling or covering a workspace, we can treat the robot as a dynamical system independent from specific hardware implementation, and analyze the resulting dynamical system to find predictable motion trajectories.

However, dynamical analysis can reveal control schemes which take advantage of large regions of stability to produce robust open-loop control schemes. This paper contributes an analysis of the dynamics of robots which move in straight lines in the plane, and can perform a controllable rotation when they encounter an obstruction. In environments with at least some symmetry or regularity - as in most human-designed environments - the dynamics of such robots reveal robust motion strategies with guaranteed long-term dynamics, such as limit cycles (good for predictable patrolling robots), or chaotic dynamics (good for hard-to-predict exploration or coverage of a space). For regular polygons and regular polygons under affine transformation, we solve analytically for the range of bounce angles guaranteed to result in different types of dynamics, such as stable orbits or ergodic motion.

The resulting control schemes have analytic bounds on their stability, and in practice are robust to actuator model errors.

There is also related work on the combinatorial complexity of the region touched by specular bouncing ("visibility with reflection") in simple polygons [1].

Such a model fits mobile robots such as differential drive robots with bump sensors and as few as one single-point infrared range sensors [2].



(a) $\theta = 0.5$ radians.

(b) $\theta = 1.2$ radians.

(c) $\theta = 1.65$ radians.

(d) $\theta = 1.9$ radians.

Figure 1. Bounce trajectories (150 bounces) in a regular pentagon.

Model Definition

A point robot moves in a bounded subset of the plane P, defined by continuous boundary $\delta P : \mathbb{S}^1 \to \mathbb{R}^2$. For most of this discussion, the environment will be a simple polygon, so δP is piecewise linear, with n vertices $(p_0, p_1, \ldots, p_{n-1})$ connected by straight edges.

The robot drives in a straight line until encountering δP. It then rotates until its heading is at an angle θ clockwise of the inward-facing boundary normal, where $-\pi/2 < \theta < \pi/2$. Then the robot sets off in a straight line again. The map created between points on δP is $B_\theta : \delta P \to \delta P$, defining our dynamical system. When the map is iterated k times, we write B_θ^k. This map is not well defined on vertices of δP, but since the number of points sending the robot to a vertex is a measure-zero set, we will not consider such trajectories.

We recall the observations of prior work [3], such as the following:

Observation 1: Let e_1 and e_2 be parallel edges of δP. Let $p_1 \in e_1$ and $p_2 \in e_2$ be points on the boundary of P. If $B_\theta(p_1) = p_2$, then $B_\theta(p_2) = p_1$.

Lemma 2: Let e_1 and e_2 be non-parallel edges of δP. Let q be the point where the infinite extensions of e_1 and e_2 intersect. Let ϕ ibe the interior angle of the extensions of e_1 and e_2. Let $p_1 \in e_1$

be a point, and suppose that $B_\theta(p_1) \in e_2$. If $B_\theta^2(p_1) \in e_1$, then there exists some c, where $c > 1$, dependent only on θ and ϕ such that $cd(p_1, q) = d(B_\theta^2(p_1), q)$.

The intuition behind the observation is that the robot will have a period two cycle between parallel edges; and will continuously move "outward" from corners (even if the edges do not actually meet in a corner).

Prior work includes as algorithm to identify distance- and link-unbounded segments on arbitrary polygons: regions of the boundary where the robot may bounce an arbitrary distance or an arbitrary number of times. However, the algorithm will not terminate when such regions are points on the polygon boundary, such as in Figures 1a and 1b. This is akin to saying the dynamical system induced by B_θ and the initial condition of the robot has a stable limit cycle, where $B_\theta = B_\theta^k$ for some k.

Regular Polygons

As observed in [3], in an equilateral triangle when θ is $\pi/2 - \epsilon$ for small ϵ, the resulting trajectory is an inscribed equilateral triangle. In this case, the boundary-classifying algorithm [3] does not terminate, since the *distance unbounded* region of the boundary is infinitesimally small (the attractor is one dimensional). It would be useful to identify such cases - where the attractor is a set of points, not intervals, on δP. Then the algorithm in [3] can be used only for maps with attractors that are segments on δP, for which the algorithm is guaranteed to terminate.

Cycles as Fixed Points of Bounce Map

For example, consider a mapping function $f_\theta : \delta P \to \delta P$ that constrains the robot to bounce counterclockwise in a regular n-gon with side length l, striking each edge sequentially, such as in Figure 1a. Imagine the robot begins at a point that is some distance x away from the nearest clockwise vertex.

Then, using the triangle formed by two adjacent edges and the robot's trajectory between them, we are able to solve for the point where the robot collides with the next wall, $f_\theta(x)$. ϕ is the interior angle of the regular polygon, $(n-2)\pi/n$.

make fig?

By the law of sines:

$$\frac{f_\theta(x)}{\sin(\pi/2 + \theta)} = \frac{l - x}{\sin(\pi - (\pi/2 + \theta) - \phi)}$$

$$f_\theta(x) = \frac{(l - x)\sin(\pi/2 + \theta)}{\sin(\pi/2 - \theta - \phi)} = c(l - x)$$

By iteration, we find that the fixed point of this mapping function is:

$$f_\theta^\infty = \sum_{i=1}^{\infty}(-l)(-c)^i = l + \sum_{i=0}^{\infty}(-l)(-c)^i$$

The sum is geometric, and finite when $|c| < 1$. If this condition holds, then the fixed point becomes:

$$f_\theta^\infty = l + \frac{-l}{1+c} = \frac{lc}{1+c}$$

So we would expect the trajectory of a robot with bounce angle θ satisfying $|c| < 1$ to converge to a limit cycle in the shape of an inscribed n-gon, with collision points at distance $(lc)/(1+c)$ from the nearest vertex in the clockwise direction.

Implications for Uncertainty in Actuation

If a stable orbit is found in a given environment, we can then use the bounds on the stability derived above to determine a range of angles that will result in periodic orbits. For example, in a regular pentagon

Generalization

Note on Simulation

The figures and experimental simulations for this paper were computed using a program written in Haskell and relying heavily on the excellent *Diagrams* library [4]. **numerical precision**

The simulator is also quite general, and could be of use to those studying classical billiards, or variants such as pinball billiards. It is also capable of simulating random bounces, or random noise on top of a deterministic bouncing law. Code is open source and on GitHub.

Extension to Other Polygons

In non-regular polygons, we cannot solve for orbits as the fixed point of one mapping function. Yet, limit cycles still exist in polygons with enough symmetry, as seen in Figure 2.

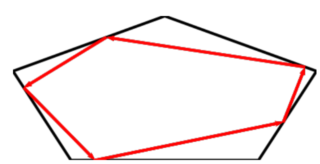


Figure 2. Stable limit cycles exist in polygons with fewer symmetries than regular polygons.

Conclusion and Discussion

[1] B. Aronov, A. R. Davis, T. K. Dey, S. P. Pal, and D. C. Prasad, "Visibility with multiple reflections," in *Algorithm theory — swat'96: 5th scandinavian workshop on algorithm theory*

reykjavík, iceland, july 3–5, 1996 proceedings, R. Karlsson and A. Lingas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 284–295.

[2] J. S. Lewis and J. M. O'Kane, "Planning for provably reliable navigation using an unreliable, nearly sensorless robot," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1339–1354, September 2013.

[3] L. H. Erickson and S. M. LaValle, "Toward the design and analysis of blind, bouncing robots," in *IEEE international conference on robotics and automation*, 2013.

[4] B. Yorgey, *Diagrams*. 2017.