# Determining stable trajectories of blind, bouncing robots

Alexandra Nilles, Steve LaValle

## 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 $\theta$, relative to the environment boundary, and drive in straight lines in the free space. Previous work introduced an algorithm that classifies parts of the boundary of polygonal environments that are reachable after the robot has traveled a certain distance or bounced a certain number of times. 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 transformations. This analysis leads to simple, open-loop control schemes that allow either predictable and stable "patrolling" dynamics, or ergodic "exploratory" dynamics. The results are useful for controlling simple mobile robots with minimal sensing and actuation, 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 implementations.

Such analysis can reveal ways to create simple control schemes which take advantage of large regions of stability, resulting in robust and predictable dynamics. This paper analyzes the dynamics of robots which move in straight lines in the plane, and can perform a controllable rotation relative to obstructions they encounter. In environments with some symmetry or regularity - as in most human-designed environments - we find control schemes which result in 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 transformations, we solve for the range of bounce angles guaranteed to result in different types of dynamics.

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

This dynamical system is similar to classical billiards [1], where the incident angle is always related to the outgoing angle by $\theta_i = -\theta_o$ (*specular* bouncing, see Figure 1). *Pinball billiards* is the case where the agent is deflected toward the normal with each bounce, $\gamma\theta_{inc} = -\theta_{out}$ for $0 \leq \gamma \leq 1$ [2]. If the agent bounces at the normal vector each time, this is called *slap billiards*, a model closely related to our dynamical system. However, we are unsure if our model preserves the structure necessary to use dominated splitting to analyze the structure of attractors. **todo: become sure either way**

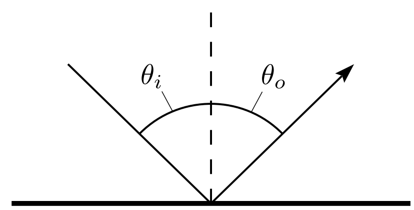



**Figure 1.** Specular bouncing occurs when the incoming angle of the agent ($\theta_i$) is equal to the outgoing angle ($\theta_o$)

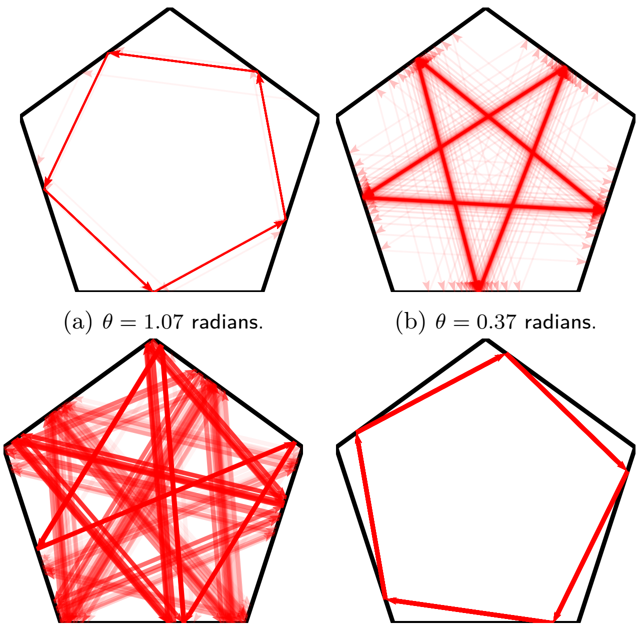
We are inspired by work on map dynamics in polygons such as [3] and [4], and it is possible that similar techniques from projective geometry could be applied to this dynamical system. Also related is work on the combinatorial complexity of the region touched by specular bouncing ("visibility with reflection") in simple polygons [5]. **todo how related**

In robotics, our model is inspired by systems such as differential drive mobile robots with bump sensors and as few as one single-point infrared range sensors [6], which are able to execute the action of aligning to a specified angle to a wall, and travelling in straight lines.

## Model Definition

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

The robot drives in a straight line until encountering $\delta P$. It then rotates until its heading is at an angle $\theta$ 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 $\delta P$ is



(a) $\theta = 1.07$ radians. (b) $\theta = 0.37$ radians.

(c) $\theta = -0.08$ radians. (d) $\theta = -1.4$ radians.

**Figure 2.** Limiting behavior of bounce trajectories (150 bounces) in a regular pentagon. Older bounces become 1% more transparent with each bounce.

$B_\theta : \delta P \to \delta P$, defining our dynamical system. When the map is iterated $k$ times, we write $B_\theta^k$. This map is not well defined on vertices of $\delta P$, 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 [7], such as the following:

*Observation 1:* Let $e_1$ and $e_2$ be parallel edges of $\delta P$. 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 $\delta P$. Let $q$ be the point where the infinite extensions of $e_1$ and $e_2$ intersect. Let $\phi$ be 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 $\theta$ and $\phi$ such that $cd(p_1, q) = d(B_\theta^2(p_1), q)$.

The intuition behind these prior results 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 an 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 2a and 2b. This is akin to saying the dynamical system induced by $B_\theta$ and the initial condition of the robot has a stable limit cycle, where $B_\theta = B_\theta^k$ for some $k$.

For example, in an equilateral triangle when $\theta$ is $\pi/2 - \epsilon$ for small $\epsilon$, the resulting limit cycle is an inscribed equilateral triangle. In this case, the boundary-classifying algorithm of [7] 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 $\delta P$. Then the algorithm in [7] can be used only for maps with attractors that are segments on $\delta P$, for which the algorithm is guaranteed to terminate.

### Limit Cycles of Sequential Edge Bounce Map

**Lemma 1:** In every regular $n$-sided polygon with side length $l$, there exists a $\theta$ such that the bounce map results in a stable limit cycle of period $n$, which strikes the boundary at points that are distance $x$ from the nearest clockwise vertex, for all $x$ with $0 < x < l$ except $x = l/2$.

**Proof:** Take a regular $n$-gon with side length $l$, and boundary $\delta P$. Let the robot begin its trajectory at a point $p \in \delta P$ which is at a distance $x$ from the nearest vertex in the clockwise direction. We will begin by constraining the robot to bounce counterclockwise, at an angle $\theta$ such that it strikes the nearest adjacent edge, such as in Figure 2a.

Define a map $f_\theta : (0, l) \to (0, l)$ that takes $x$, the robot's distance from vertex $p_i$, and maps it to $f(x)$, the resulting distance from vertex $p_{i+1}$, after application of this constrained bounce map.

Then, using the triangle formed by two adjacent edges and the robot's trajectory between them, we can solve for $f(x)$. Let $\phi = (n-2)\pi/n$ be the vertex angle of the regular polygon.

By the law of sines:

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

$$f_\theta(x) = \frac{(l - x)\cos(\theta)}{\cos(\theta - \phi)} = c(l - x)$$

By iterating this map, we find that the fixed point 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:

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

So we would expect the trajectory of a robot with bounce angle $\theta$ 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

For each stable orbit in a given environment, we can use the bounds on $c$ to determine the range of angles

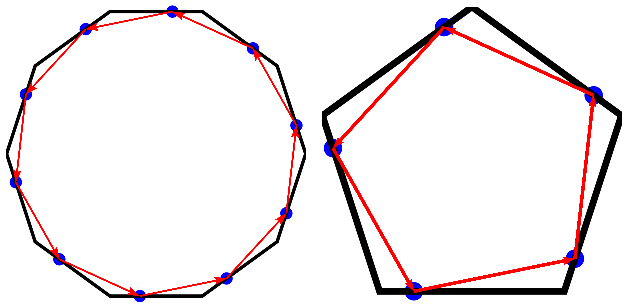


**Figure 3.** Predicted (collisions indicated by blue dots) and simulated limit cycles when bouncing to adjacent edge in regular polygons.

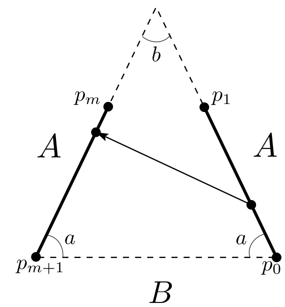



**Figure 4.** A bounce from edge $p_0p_1$ to edge $p_mp_{m+1}$. The other edges of the polygon are not drawn.

that will result in that orbit. The convergence condition $|c| < 1$ implies that a stable cycle will result for any $\theta$ within the bounds $\phi/2 < \theta < \pi/2$ or $-\pi/2 < \theta < -\phi/2$, which were confirmed through simulation. Thus, when designing a "patrolling" robot in an environment with regular polygonal geometry, a robot designed to bounce at an angle in the center of one of these ranges will be maximally robust to actuator or sensor errors. The resulting maximum allowable error, $\epsilon_{max}$, will be $\pm|(\pi - \phi)/2|$. Bounces with smaller errors will still result in stable orbits of the workspace.

However, actuator or sensor error may create orbits that impact the boundary at a different location than expected. If there is a constant error in the bounce angle, so that the effective bounce angle is $\theta + \epsilon$, with $\epsilon < \epsilon_{max}$, the resulting difference in the location of the collision point on each edge will be $\Delta_x = f^{\infty}_{\theta+\epsilon} - f^{\infty}_{\theta}$.

## Generalization

**Theorem:** In every regular $n$-sided polygon with side length $l$, if $k|n$, there exists a stable periodic orbit with $k$ bounces where each collision with the polygon boundary is at a distance $x$ $(0 < x < l)$ from the nearest vertex in the clockwise direction. **todo: x as function of k**

**Proof:** Instead of bouncing between adjacent edges, we may ask what happens when the robot bounces between edge $p_0p_1$ and edge $p_mp_{m+1}$, "skipping" $m-1$ edges, such as in Figure 2b where the robot bounces off every other edge.

Let $m \leq \lfloor n/2 \rfloor$ (if this is not the case, reflect the polygon across the vertical center axis, solve, and reflect back).

Extend the line segments $p_0p_1$ and $p_mp_{m+1}$ to their point of intersection $q$, forming the triangle $p_0p_{m+1}q$. Let $a = \angle qp_{m+1}p_0 = \angle qp_0p_{m+1}$, by symmetry. Let $b = \angle p_{m+1}qp_0$. Let $A = |qp_{m+1}| = |qp_0|$ and $B = |p_{m+1}p_0|$. See Figure 4. Each of the sides of the polygon has length $l$, and the robot begins its trajectory at a point which is distance $x$ from $p_0$. We wish to find the resulting distance from point $p_m$, $f_{\theta,m}(x)$.

Then, by the law of sines, we have:

$$A = \frac{B\sin(a)}{\sin(b)}$$

We can then form the triangle from the points $p_0$, $p_{m+1}$, and the center of the regular $n$-gon. The distance from the center of a regular $n$-gon to any of its vertices is $\frac{l}{2\sin(\pi/n)}$. The angle subtended by the edges $p_0p_1$ through $p_mp_{m+1}$ is $2\pi(m+1)/n$. Thus we can solve for $B$:

$$B = \frac{l\sin(\pi(m+1)/n)}{\sin(\pi/n)}$$

The angle $a$ can be found by considering the polygon formed by edges $p_0p_1$ through $p_mp_{m+1}$, closed by edge $p_{m+1}p_0$. This polygon has $m+2$ vertices, so its angle sum is $m\pi$. $m$ of these vertices have the vertex angle of the regular $n$-gon, $(n-2)\pi/n$. The remaining two vertices have angle $a$. Therefore:

$2a + m(n-2)\pi/n = m\pi$

So $a = m\pi/n$.

And thus $A$:

$$A = \frac{l\sin(\frac{\pi(m+1)}{n})\sin(\frac{m\pi}{n})}{\sin(\frac{\pi}{n})\sin(\frac{\pi(n-2m)}{n})}$$

Then, using the triangle formed by the the bounce of the robot, and again the law of sines, we have:

$$\frac{A-x}{\sin(\theta - \pi/2 + (2\pi m)/n)} = \frac{A - l + f_{\theta,m}(x)}{sin(\pi/2 - \theta)}$$

Solved for $f_{\theta,m}$ and rewritten:

$$f_{\theta,m}(x) = \frac{(x-A)\cos(\theta)}{\cos(\theta + \pi(n-2m)/n)} + l - A$$

**Note:** When $m = 1$ (agent skips no edges while bouncing around polygon), $A$ reduces to $l$, and the expression for $f_{\theta,1}(x)$ reduces to $f_\theta(x)$ as previously derived.

**Note:** When $mk = n$, for some integer $k$, $A$ becomes $(-1)^k l$, and $f_{\theta,m} =$ **todo: finish this**

## 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 [8]. **todo: write up on 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.[1]

## Conclusion and Discussion

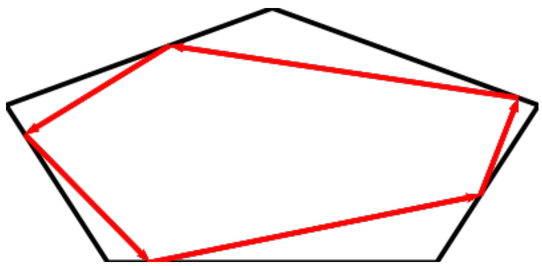


**Figure 5.** Stable limit cycles exist in polygons with fewer symmetries than regular polygons.

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 5.

## References


[1] S. Tabachnikov, *Geometry and billiards.* American Mathematical Society, 2005.

[2] R. Markarian, E. Pujals, and M. Sambarino, "Pinball billiards with dominated splitting," *Ergodic Theory and Dynamical Systems*, vol. 30, pp. 1757–1786, 2010.

[3] W. P. Hooper and R. E. Schwartz, "Billiards in nearly isosceles triangles," *Journal of Modern Dynamics*, vol. 3, no. 2, pp. 159–231, June 2013.

[4] R. Schwartz, "The pentagram map," *Experiment. Math.*, vol. 1, no. 1, pp. 71–81, 1992.

[5] 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.

[6] 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.

[7] 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.

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


[1] https://github.com/alexandroid000/bounce