



Reachability Analysis of Mobile Robot Trajectories in Polygons with SpaceEx

Alli Nilles

December 4, 2017

Blind, Bouncing Robots¹

Model the robot as a point moving **in straight lines** in the plane, “bouncing” off the boundary at a **fixed angle** θ from the normal:

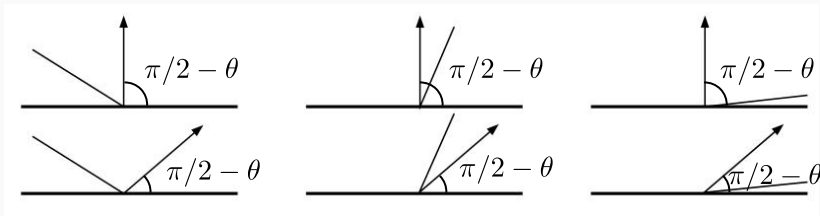


Figure 1. A point robot moving in the plane. The top row shows bounces at zero degrees from the normal. The second row shows bounces at 50 degrees clockwise from normal.

¹(Erickson and LaValle 2013)

Research Questions

- What kind of tasks are robots with extremely simple control laws capable of performing?
- Will the robot become “trapped” in a certain part of the environment? Or a certain motion pattern?

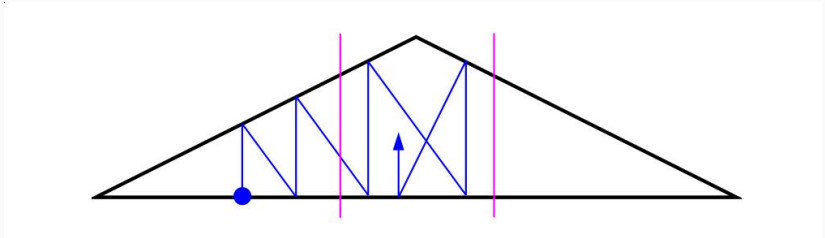


Figure 2. In this environment, bouncing at the normal, the robot will become trapped in the area between the purple lines.

These questions are related:

- if robot will get stuck in small part of state space under a constant control input, can use this for localization
- if robot will get stuck in a “big” part of state space, can use this for monitoring / coverage tasks

- Assume we know environment exactly

- Assume we know environment exactly
- Can implement on a roomba with bump sensor and IR prox detector²

²(Lewis and O'Kane 2013), IJRR

Implementation

- Assume we know environment exactly
- Can implement on a roomba with bump sensor and IR prox detector²
- “Collisions” can be virtual - for example, robot stops when it is collinear with two landmarks, and rotates until one landmark is at a certain heading

²(Lewis and O’Kane 2013), IJRR

Implementation

- Assume we know environment exactly
- Can implement on a roomba with bump sensor and IR prox detector²
- “Collisions” can be virtual - for example, robot stops when it is collinear with two landmarks, and rotates until one landmark is at a certain heading
- Also useful model of very small “robots” or microorganisms,³ or robots in low-bandwidth environments

²(Lewis and O’Kane 2013), IJRR

³(Spagnolie et al. 2017)

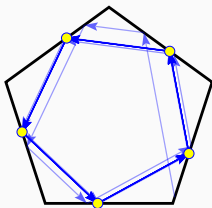
NASA's Mars Roomba Begins Mission To Clean Dust From Planet's Surface



According to NASA, the Mars Roomba's edge-cleaning mode will allow the vehicle to scour even the crevices where mountains meet the planet's surface.

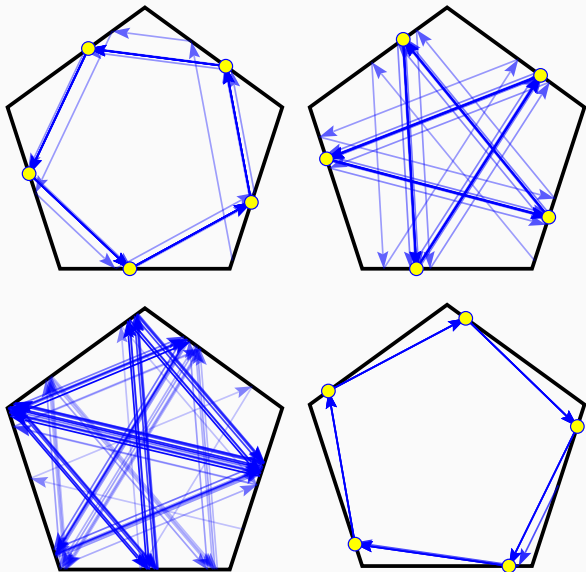
Discovery Through Simulation

- Haskell with *Diagrams* library (Yorgey 2012)
- fixed-angle bouncing, specular bouncing, add noise
- render diagrams from simulations automatically⁴

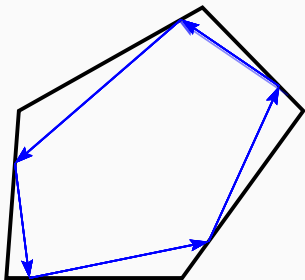


⁴<https://github.com/alexandroid000/bounce>

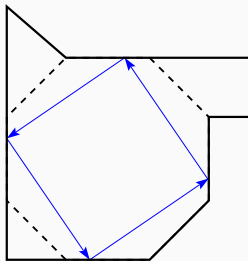
Simulation Results



Other Polygons



(a) A stable orbit in a sheared pentagon.



(b) A stable orbit in a nonconvex environment.

Figure 4. Stable orbits also exist in non-regular polygons.

What We Want

- check if cycle exists where theory says it does for convex polygons

What We Want

- check if cycle exists where theory says it does for convex polygons
- do experiments with nonconvex polygons to develop theory there

What We Want

- check if cycle exists where theory says it does for convex polygons
- do experiments with nonconvex polygons to develop theory there
- minimize simulation / discretization / floating point artifacts

What We Want

- check if cycle exists where theory says it does for convex polygons
- do experiments with nonconvex polygons to develop theory there
- minimize simulation / discretization / floating point artifacts
- extend to multiple robots and nondeterministic bouncing

What We Want

- check if cycle exists where theory says it does for convex polygons
- do experiments with nonconvex polygons to develop theory there
- minimize simulation / discretization / floating point artifacts
- extend to multiple robots and nondeterministic bouncing

What We Want

- check if cycle exists where theory says it does for convex polygons
- do experiments with nonconvex polygons to develop theory there
- minimize simulation / discretization / floating point artifacts
- extend to multiple robots and nondeterministic bouncing

Tools from this class especially help with the last two

Results

```
loc1 = Location 1 "interior"
      "-500.0 &lt;= x &amp;&amp;
      x &lt;= 0.0 &amp;&amp;
      0 &lt;= y &amp;&amp;
      y &lt;= 500"
      "x'==vx &amp; y'==vy"

square_ha :: HA
square_ha = HA { name = "test"
                , params = mkParams $ mkPoly sq
                , locations = [loc1]
                , transitions = mkTs (mkPoly sq) (0 @@ rad)
                }
```

Results

```
<?xml version="1.0" encoding="iso-8859-1"?>
<sspaceex xmlns="http://www-verimag.imag.fr/xml-namespaces/sspaceex" version="0.2" math="SpaceX">
  <component id="test">
    <param name="x" type="real" local="true" d1="1" d2="1" dynamics="any" />
    <param name="y" type="real" local="true" d1="1" d2="1" dynamics="any" />
    <param name="vx" type="real" local="true" d1="1" d2="1" dynamics="const" />
    <param name="vy" type="real" local="true" d1="1" d2="1" dynamics="const" />
    <param name="e1" type="label" local="false" />
    <param name="e2" type="label" local="false" />
    <param name="e3" type="label" local="false" />
    <param name="e4" type="label" local="false" />
    <location id="1" name="interior">
      <invariant>-500.0 &lt;= x &amp;&amp; x &lt;= 0.0 &amp;&amp; 0 &lt;= y &amp;&amp; y &lt;= 500</invariant>
      <flow>x'==vx &amp; y'==vy</flow>
    </location>
    <transition source="1" target="1" asap="true" >
      <label>e1</label>
      <guard>x - (0.0) &lt; (0.001) &amp;&amp; x - (0.0) &gt; -(0.001) &amp;&amp; (0.0) &lt;= y &amp;&amp; y &lt; (500.0)</guard>
      <assignment>vx := (-1.0) &amp; vy := (-0.000000000000000049789962505147994)</assignment>
    </transition>
    <transition source="1" target="1" asap="true" >
      <label>e2</label>
      <guard>y - (500.0) &lt; (0.001) &amp;&amp; y - (500.0) &gt; -(0.001) &amp;&amp; (-500.000000000000006) &lt;= x &amp;&amp; x &lt; (-0.000000000000000005551115123125783)</guard>
      <assignment>vx := (-0.00000000000000006123233995736766) &amp; vy := (-1.0)</assignment>
    </transition>
    <transition source="1" target="1" asap="true" >
      <label>e3</label>
      <guard>x - (-500.000000000000006) &lt; (0.001) &amp;&amp; x - (-500.000000000000006) &gt; -(0.001) &amp;&amp; (0.0) &lt;= y &amp;&amp; y &lt; (-0.0000000000000000001749191776789837)</guard>
      <assignment>vx := (1.0) &amp; vy := (-0.00000000000000001749191776789837)</assignment>
    </transition>
    <transition source="1" target="1" asap="true" >
      <label>e4</label>
      <guard>y - (0.0) &lt; (0.001) &amp;&amp; y - (0.0) &gt; -(0.001) &amp;&amp; (-500.000000000000001) &lt;= x &amp;&amp; x &lt; (0.0)</guard>
      <assignment>vx := (0.00000000000000006123233995736766) &amp; vy := (1.0)</assignment>
    </transition>
  </component>
</sspaceex>
```

- asap transitions not working for some reason: must specify invariant in location to keep robot from “escaping” the polygon

- asap transitions not working for some reason: must specify invariant in location to keep robot from “escaping” the polygon
- Also not sure how to specify nonconvex invariants

- asap transitions not working for some reason: must specify invariant in location to keep robot from “escaping” the polygon
- Also not sure how to specify nonconvex invariants
- Both supposed to be included in PHAVer on SpaceEx (Minopoli and Frehse 2014)

- asap transitions not working for some reason: must specify invariant in location to keep robot from “escaping” the polygon
- Also not sure how to specify nonconvex invariants
- Both supposed to be included in PHAVer on SpaceEx (Minopoli and Frehse 2014)

- asap transitions not working for some reason: must specify invariant in location to keep robot from “escaping” the polygon
- Also not sure how to specify nonconvex invariants
- Both supposed to be included in PHAVer on SpaceEx (Minopoli and Frehse 2014)
- Naive code generation: chunking strings together (only a few fields in xml file we need to change)

- asap transitions not working for some reason: must specify invariant in location to keep robot from “escaping” the polygon
- Also not sure how to specify nonconvex invariants
- Both supposed to be included in PHAVer on SpaceEx (Minopoli and Frehse 2014)
- Naive code generation: chunking strings together (only a few fields in xml file we need to change)
- Next step is to use Haskell XML Toolbox (HXT) to make this less janky

References

- Erickson, L. H., and S. M. LaValle. 2013. "Toward the Design and Analysis of Blind, Bouncing Robots." In *IEEE International Conference on Robotics and Automation*.
- Lewis, Jeremy S., and Jason M. O'Kane. 2013. "Planning for Provably Reliable Navigation Using an Unreliable, Nearly Sensorless Robot." *International Journal of Robotics Research* 32 (11): 1339–54.
- Minopoli, Stefano, and Goran Frehse. 2014. "Non-Convex Invariants and Urgency Conditions on Linear Hybrid Automata." In *International Conference on Formal Modeling and Analysis of Timed Systems*, 176–90. Springer.
- Spagnolie, Saverio E, Colin Wahl, Joseph Lukasik, and Jean-Luc Thiffeault. 2017. "Microorganism Billiards." *Physica D: Nonlinear Phenomena* 341. Elsevier: 33–44.
- Yorgey, Brent A. 2012. "Monoids: Theme and Variations (Functional Pearl)." In *Haskell*. ACM.