Landmark-Based Navigation

In $\mathbb{R}^2$, a finite set $L \subseteq \mathbb{R}^2$ of n labeled landmarks.

A pointrobot moves in $\mathbb{R}^2$ ($C = \mathbb{R}^2$, ~~$C = \mathbb{R}^2 \times S^1$~~)

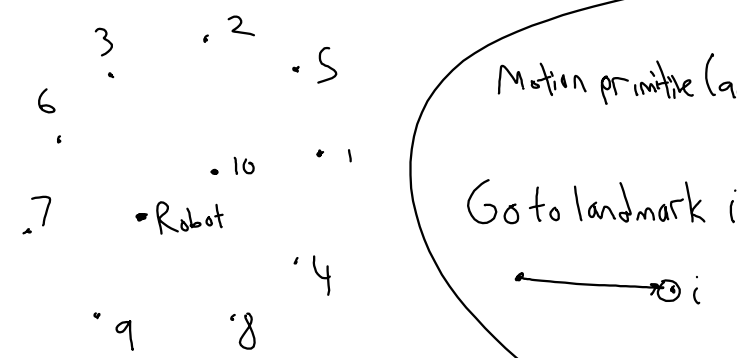
$X = \mathbb{R}^{2n+2}$

Landmark positions are unknown

Robot's position unknown

Sensor: Cyclic permutation of landmarks

$h: X \to Y$



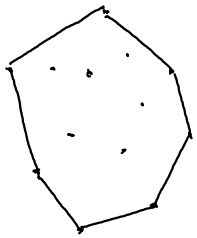
Motion primitive (action)

Go to landmark i

~~$y = h(x) = (1,10,5,2,3,6,7,9,8,4)$~~

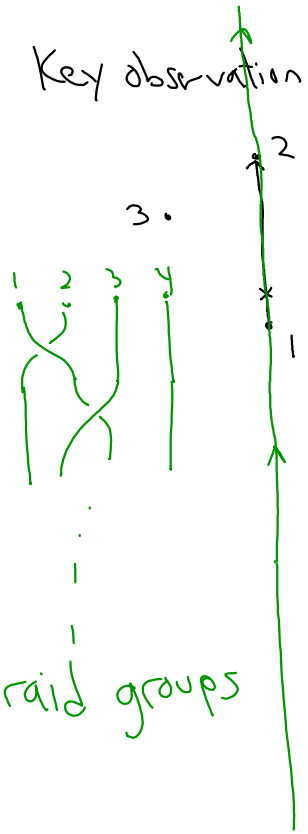
cyclic

Task: Determine which landmarks are on the convex hull boundary of L.



More generally, for any $L' \subseteq L$, which landmarks are in the hull of L'?

Key observation



braid groups

Go to 2 (from 1)

Infer that 3 is to the left

Simple strategy: Travel between all pairs of landmarks $O(n^2)$ time

$O(n \lg n)$ - Tovar, Freda, LaValle Cont. Math.

A Sensor-Centric Theory of Computation

Jason O'Kane. PhD Thesis. July 2007
"On Comparing the Power of Robots"

Recall from theory of computation

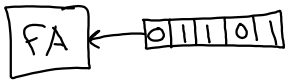
A problem is encoded as a set of ~~str~~ strings ($\in \Sigma^*$) called L, a language

A machine should accept strings in L and reject all others.

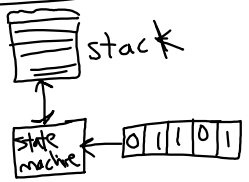
decision

Machines

FA - finite automaton



PDA



pushdown automaton

TM - Turing machine

Regarding a problem (~~to~~ language):

1) Decidability (solvable?)
2) Complexity (if yes, how costly?)

Establishing equivalences of machines

- Can solve the same problems
- Ex Adding more tapes to a TM

Compare the power of machines to solve problems

- Some machines can solve more than others

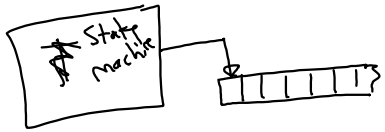
FA $\prec$ PDA $\prec$ TM

Ex Palindromes

redivider
sarppvaknuppias

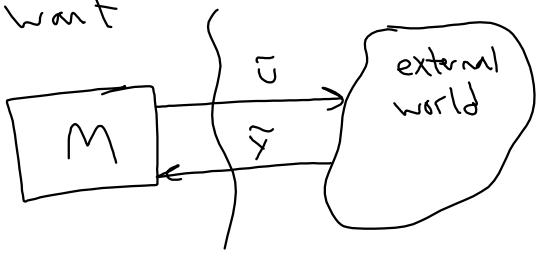
We want the same kinds of concepts for $\underbrace{\text{sensing/actuation}}_{\text{robotic}}$ systems

Instead of



TM

We want



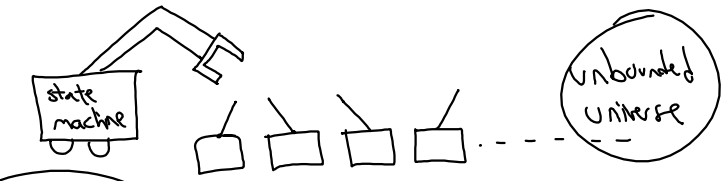
$\Sigma^* = (\varepsilon, 0, 1, 00, 01, 10, 11, 000, 001, \ldots)$

$L = (\varepsilon, \ , 1, \ , 01, \ , \ , 000, \ , \ldots)$

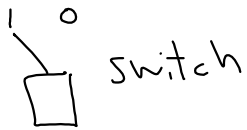
$0.1010100 10 \ldots$

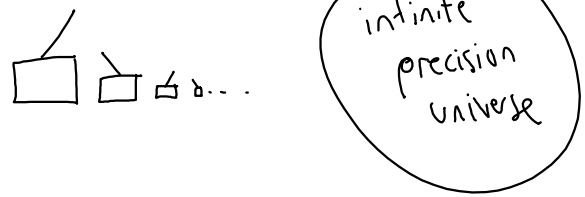
1D point robot with perfect sensing

Can a robotic system simulate a TM?



John Reif
Power of Frictional Mechanical Systems





Adapt existing theory of computation?

- Measured uncertainty? Precision? Physics?
- Too biased toward writing everything down in advance → motion planning (Piano movers' problem)
- Arbitrary discretizations that obscure the true complexity

On-line algorithms