Module C: Reactive Methods 2

Part 1. Gap Following Methods

Part 2. Other Reactive Methods for Collision Avoidance

References:

https://f1tenth.org and slides of UPenn ESE 680

http://www.cs.cmu.edu/~choset/

Research papers as cited

Other reactive methods

- **References (review papers on reactive methods, available on Course Site):**
  - J.A. Tobaruela and A.O. Rodriguez, "Reactive navigation in extremely dense and highly intricate environments," PLOS One, https://doi.org/10.1371/journal.pone.0189008, pp1-51, Feb. 2017.
  - J. Borenstein and Y. Koren, "The Vector Field Histogram -Fast Obstacle Avoidance For Mobile Robots." IEEE J. Robotics Automation, vol 7, no. 3, pp. 278-288, June 1991.
- **Teaching website and slides:**
  - http://www.robotmotionplanning.org/TeachingRobotics.html , by Dr. Erion Plaku at Catholic University of America; Latest offering of CSC/EE 576 in spring 2015
  - CMU course website at https://www.cs.cmu.edu/~motionplanning/ by Howie Choset 2010
- **Other Reactive methods for Obstacle Avoidance**
  - Class D: The Bug and Tangent Bug Algorithms (Bug0, Bug1, Bug2 and RoverBug, and many more)
  - Class A: The Artificial Potential Field Algorithm or Vector Field Histogram;
  - Class B: The Curvature-Velocity Method, Dynamic Window Approach, Beam Curvature Method, etc

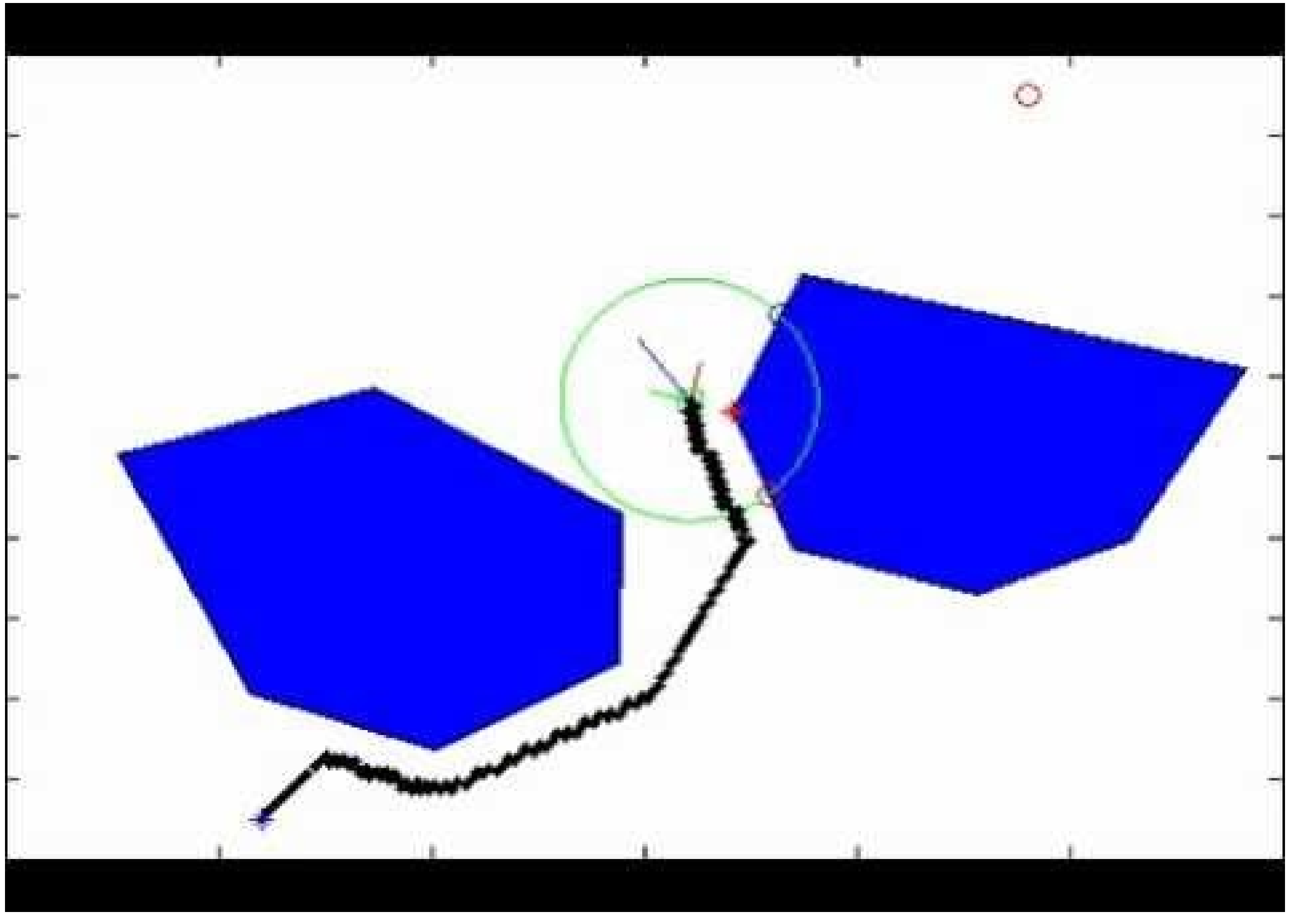
Reactive vs. Deliberative Paradigms

LEHIGH UNIVERSITY

- **Deliberative paradigm (N J Nilsson, "Shakey the Robot" 1984, SRI International)**
  - Focus heavily on reasoning and knowledge representation in the sense-plan-act cycle.
  - Use a global world model, plan actions and dispatch for execution; Computationally complex.
- **Reactive paradigm (Rodney A. Brooks 1986 Artificial Intelligence)**
  - Focus on sense-act cycle w/o heavy reasoning or knowledge representation;
  - Use robot's local world and quickly react to unpredictable environment,
  - Can be fast, low cost, but often suboptimal for navigation in complex environment;
- **Hybrid Methods (J.S. Albus and A.J. Barbera, 2005 Elsevier Annual Reviews in Control)**
  - Combine reasoning in deliberative paradigm and responsiveness in reactive paradigm, global/partial plan is input to reactive layer, deliberative layer may interrupt reactive layer if needed
  - Go with a "plan, sense-act" cycle: comma means the two are in parallel
  - Compromise between global objectives and local constraints is not always easy to find. Tradeoffs have to be empirically tuned or learned (AI comes to play in this area) – high computational cost

The Bug Algorithms

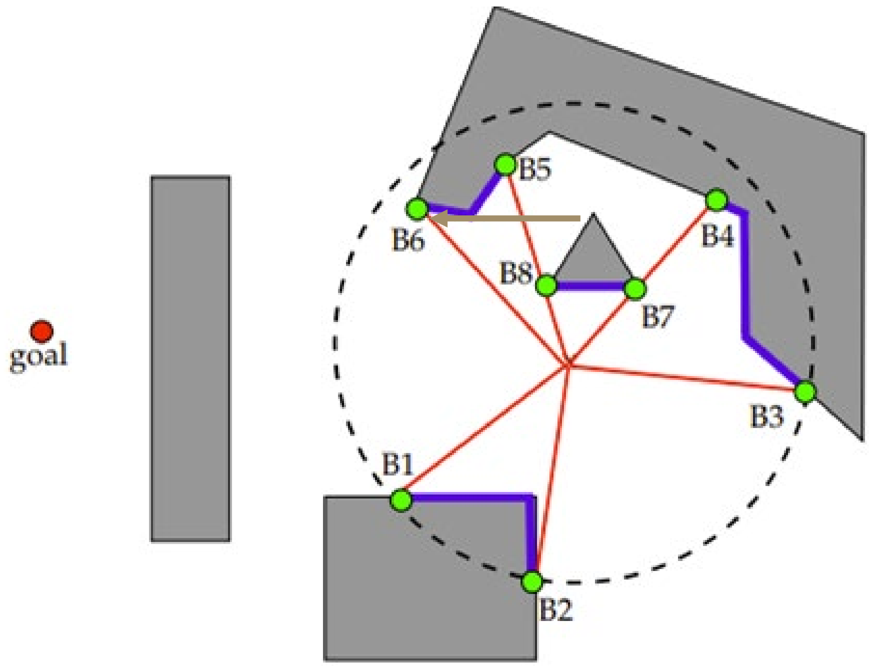
- No global model of the world, *i.e* all obstacles are unknown prior
- Only assumes *local* knowledge of environment & a ***global goal*** (specified by heading, distance)
- Original bug algorithm variations *(bug0, bug1)* dealt with planning based on tactile sensing (short range ultrasonic, bump sensors) published around 1986
- **TangentBug**, a variation that uses range measurements from LiDAR



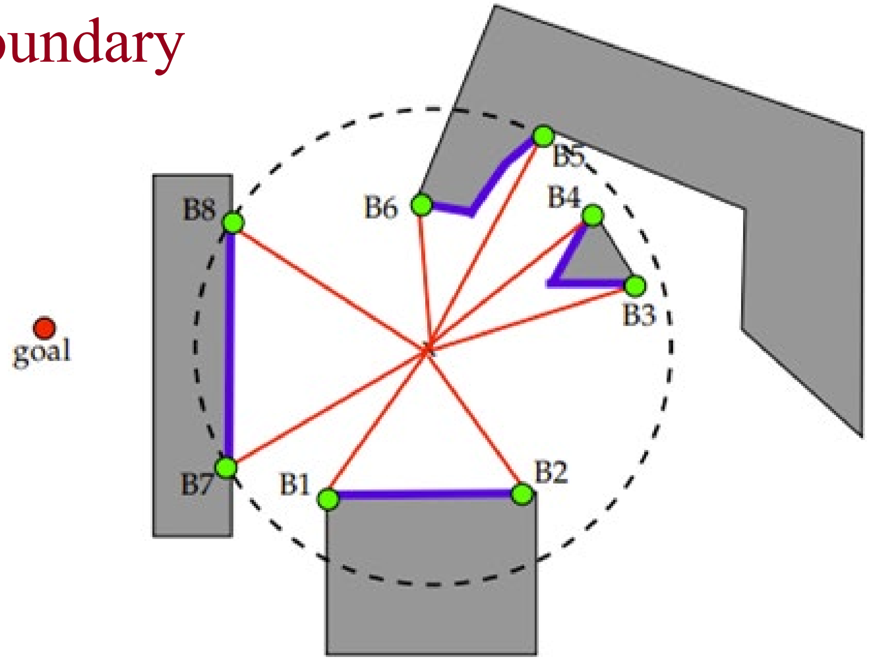
https://www.youtube.com/watch?v=wuTrGFAw3ew

Tangent Bug Algorithm

- Use range sensor measurements to compute endpoints of continuous segments on obstacle boundaries.
- Algorithm currently thinks it has unobstructed way to goal



- Algorithm now sees that it cannot go straight to the goal. What can it do?
- Choose point B_i that minimizes heuristic d(x, B_i) + d(B_i, ***Goal***). If this distance starts increasing, algorithm starts to follow some boundary



Disadvantages of Bug Algorithms

LEHIGH UNIVERSITY

- Prone to taking long trajectories towards goal, occasionally gets too close
- Distance heuristic $d(x, \boldsymbol{B_i}) + d(\boldsymbol{B_i}, \boldsymbol{Goal})$ requires knowledge about distance to goal, which isn't necessarily easy to get (likely requires some beacon setup)
- Even if we let the goal be some local point in the LiDAR scan, we still need another heuristic to figure out which goal point this should be.

Reference: I. Kamon, E. Rimon, and E. Rivlin, "TangetBug: A Range-Sensor-Based Navigation Algorithm," SAGE Journal, 1998. https://doi.org/10.1177/027836499801700903

Also see video at http://www.robotmotionplanning.org/TeachingRobotics.html and

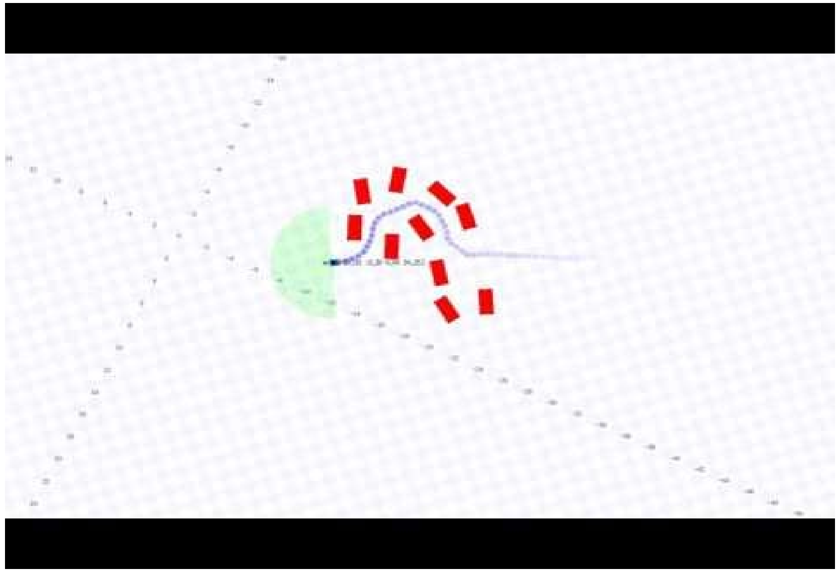
slides at https://www.cs.cmu.edu/~motionplanning/lecture/Chap2-Bug-Alg_howie.pdf

The Artificial Potential Field Algorithm

- Think of placing an electric/magnetic field over the environment
- Robot is at q_{start}, making its way to q_{goal}
- Attach positive charges to obstacle boundaries, which generates a repulsive field; Attach a negative charge to your goal, which generates an attractive field.
- This yields a potential field expressed as

$$U(q) = U_{att}(q) + U_{rep}(q)$$

- This 3D potential function gives us a local 'landscape' at each time stamp



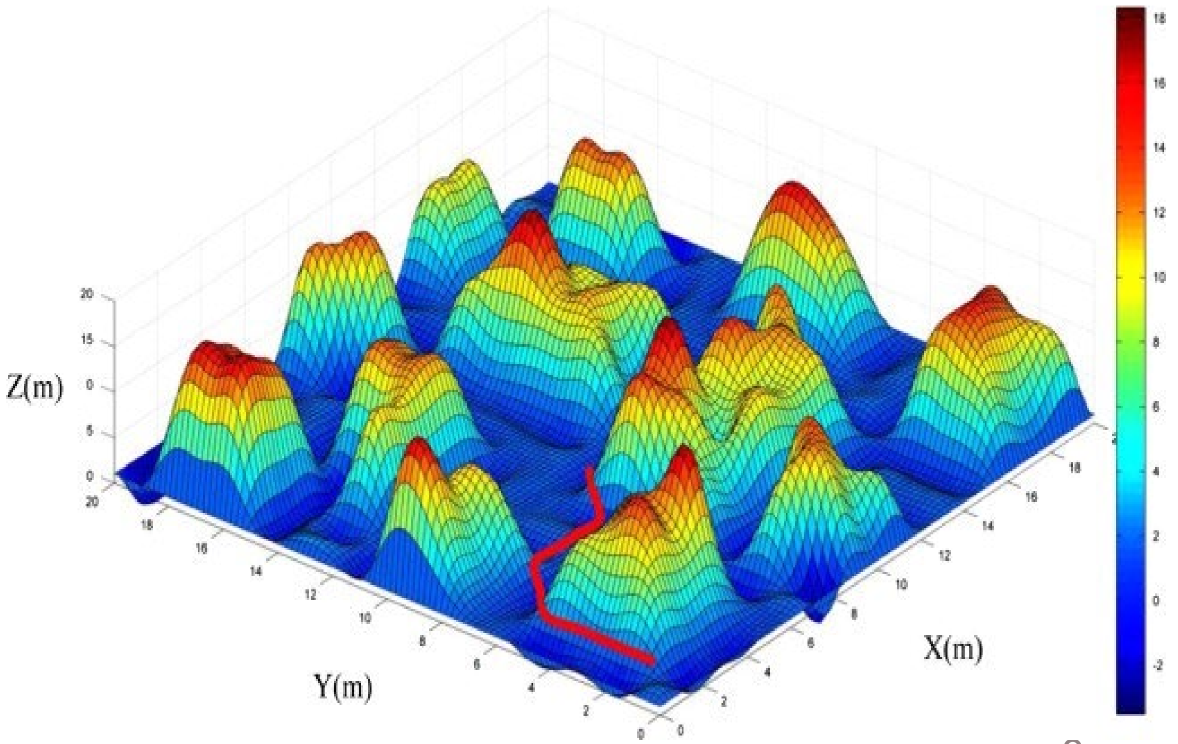
https://www.youtube.com/watch?v=UVUTcZisA94

Also an interesting lecture recording by C.J. Taylor: https://www.youtube.com/watch?v=MQjeqvbzhGQ

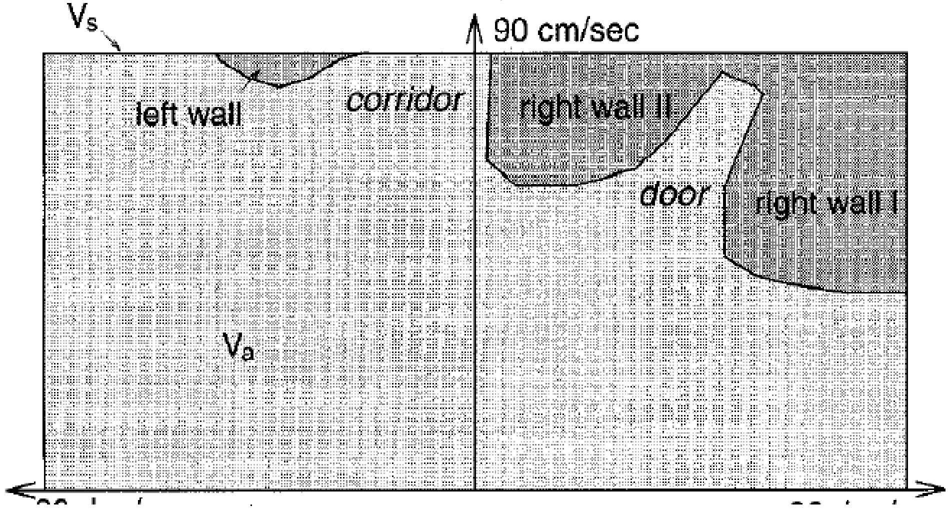
APF With Gradient Descent

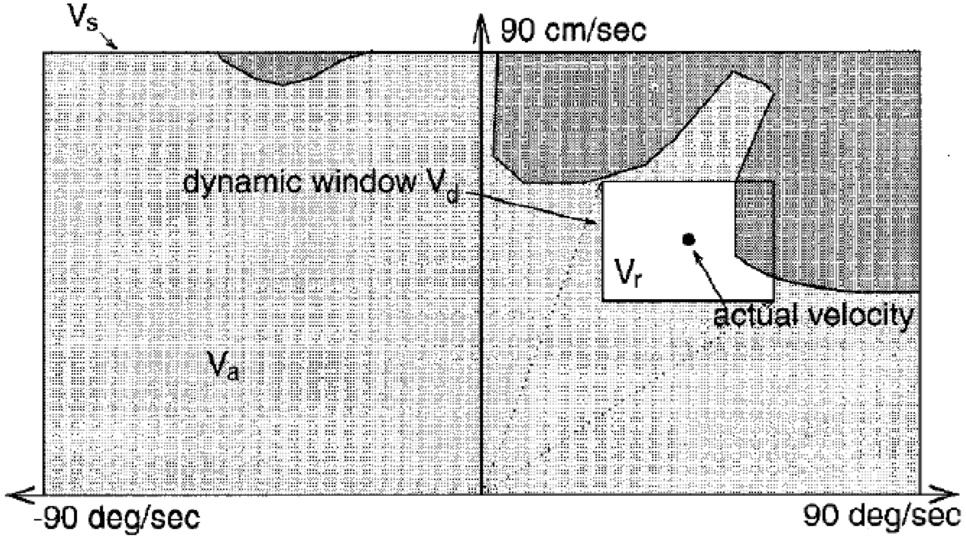
- Robot can move towards goal via ***gradient descent algorithm.***
- Analogous to marble rolls to the lower ground (see CJ Taylor's lecture video: https://www.youtube.com/watch?v=MQjeqvbzhGQ)
- Disadvantages:
  - Can stuck at LOCAL MINIMA with Gradient Descent!
  - If use a global map → *harmonic potential fields*
  - have to repeatedly compute gradient descent at each time step. For small local scans, use ***brushfire algorithm***
  - Once again, how to choose the goal point?

– $q(0)=q_{start}$
– $i = 0$
– while $\nabla U(q(i)) \neq 0$ do
  - $q(i+1) = q(i) - \alpha(i) \nabla U(q(i))$
  - $i=i+1$

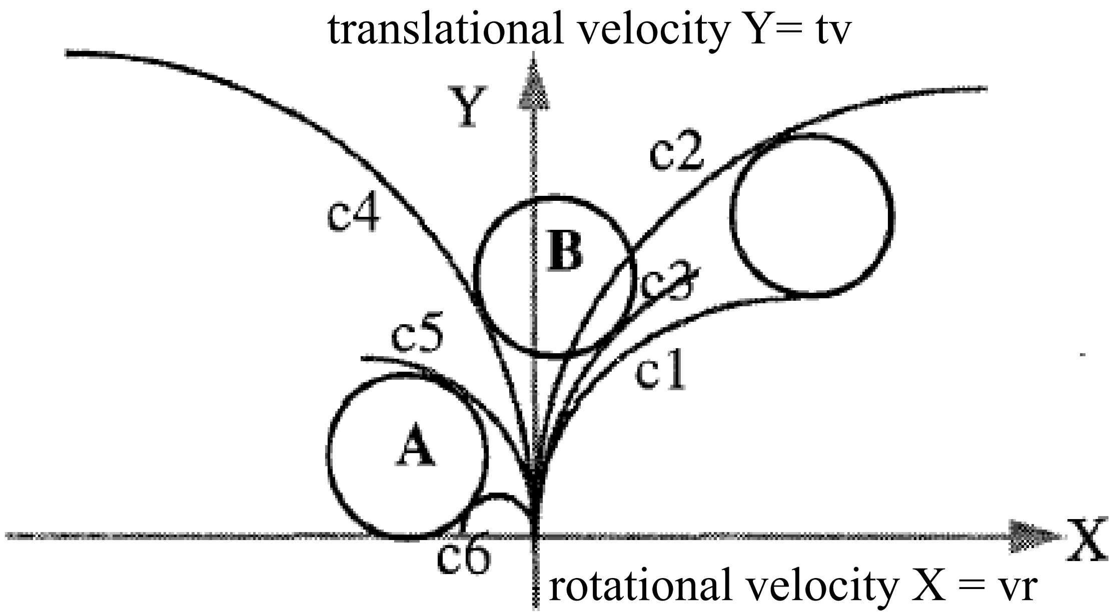


Dynamic Window Approach & Curvature Velocity Algorithm





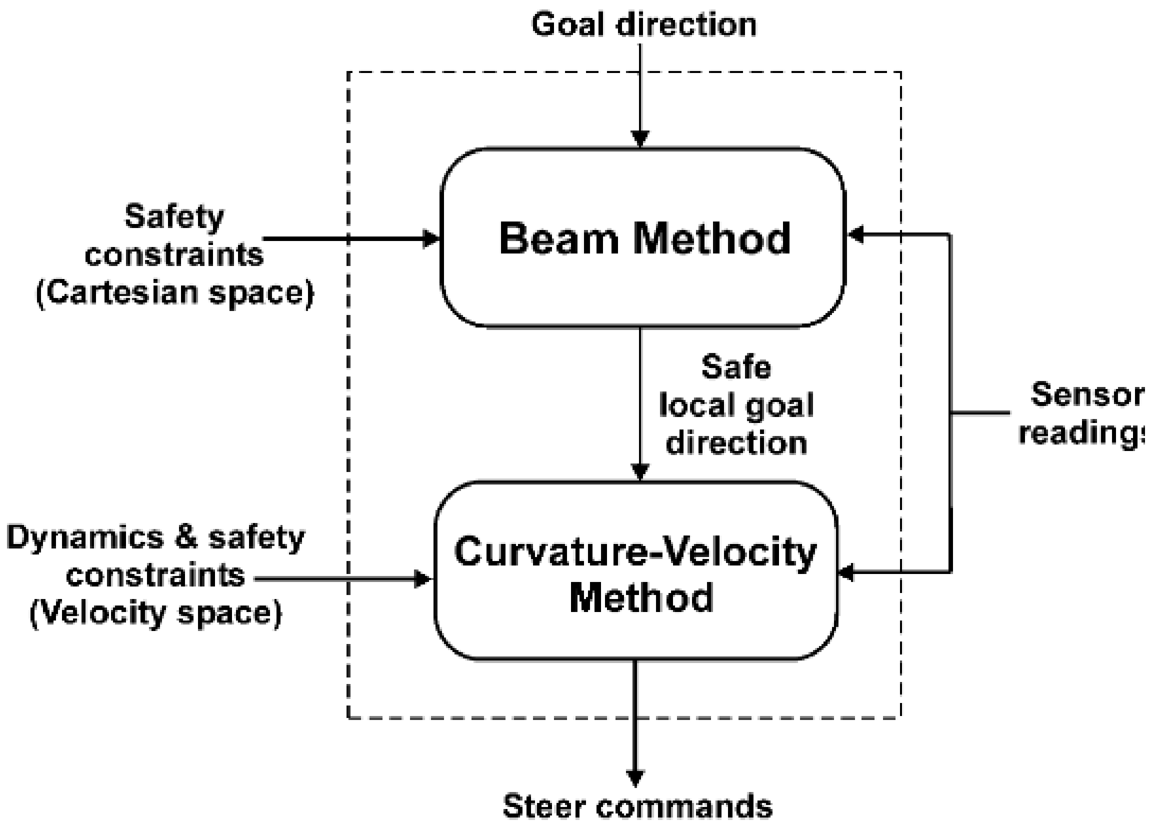
- Reactive method to avoid local obstacles, where tangent curvatures are formulated in the velocity space around the obstacles

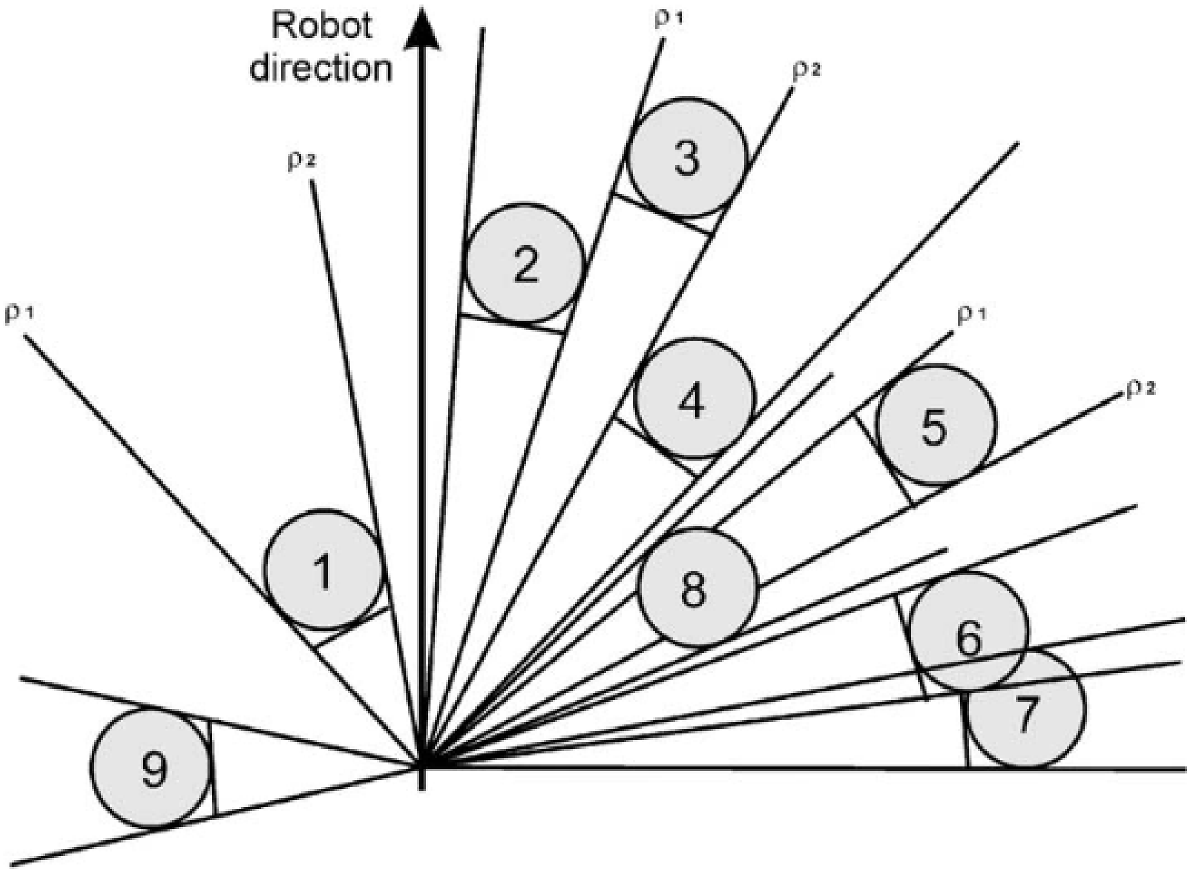


Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. IEEE Robotics Automation Magazine. 1997; 4(1):23±33.
Simmons R. The curvature-velocity method for local obstacle avoidance. IEEE Int'l Conf Robotics Automation. vol. 4; 1996. p. 3375±3382.

Beam Curvature Method

- Combines Beam Method with Curvature Velocity Method





Fernandez JL, Sanz R, Benayas JA, Dieguez AR. Improving collision avoidance for mobile robots in partially known environments: the beam curvature method. Robotics and Autonomous Systems. 2004; 46(4):205±219. https://doi.org/10.1016/j.robot.2004.02.004