Reading report
Motion planning in dynamic environment using velocity obstacle[1]

Dharmin Bakaraniya

May 14, 2018

Abstract

This paper presents a method for robot motion planning in dynamic environments. It consists of selecting avoidance maneuvers to avoid static and moving obstacles in the velocity space, based on the current positions and velocities of the robot and obstacles. It is a first- order method, since it does not integrate velocities to yield positions as functions of time.

The avoidance maneuvers are generated by selecting robot ve- locities outside of the velocity obstacles, which represent the set of robot velocities that would result in a collision with a given obstacle that moves at a given velocity, at some future time. To ensure that the avoidance maneuver is dynamically feasible, the set of avoidance velocities is intersected with the set of admissible velocities, defined by the robots acceleration constraints. Computing new avoidance maneuvers at regular time intervals accounts for general obstacle trajectories.

The trajectory from start to goal is computed by searching a tree of feasible avoidance maneuvers, computed at discrete time intervals. An exhaustive search of the tree yields near-optimal trajectories that either minimize distance or motion time. A heuristic search of the tree is applicable to on-line planning. The method is demonstrated for point and disk robots among static and moving obstacles, and for an automated vehicle in an intelligent vehicle highway system scenario.

1 Summary

- Motion planning in dynamic environment is considerably more difficult that static motion planning because it needs to solve path planning and velocity planning at the same time.

- Motion planning in static environment is guaranteed to come up with a solution if one exists but in dynamic environment the solution is ‘intractable’.

- "The velocity obstacle is a first-order approximation of the robots velocities that would cause a collision with an obstacle at some future time, within a given time horizon." [1]

- Velocity obstacle (VO) is an extention of configuration space obstacle for a dynamic environment.

- For any given obstacle B_1 and moving object A, there will be a collision if $\mathbf{v}_{A,B_1}$ lies between the two tangents of B_1 (λ_f and λ_r).

- The circle around which the tangents are to be drawn depend on the velocity of A. This makes a lot more sense because if A was moving very slowly then the circle around B_1 would be very small which means a small set of very specific movement of A will actually result into collision.

- The area between these tangents is called collision cone.

- Velocity obstacle is addition (Minkowski vector sum) of collision cone with the velocity $\mathbf{v}_{B_1}$
$$VO = CC_{A,B_1} + \mathbf{v}_{B_1}$$

- For stationary obstacle, $\mathbf{v}_{B_1} = 0$ and thus VO will not be translated.

- VO for multiple obstacle can be combined
$$VO = \bigcup_{i=1}^{m} VO_i$$

- VO can be calculated periodically to fight against variable velocity.

- To avoid obstacle, the robot calculates it reachable velocities by considering the acceleration that it can achieve and the time till next interval.

- The authors state that the dynamic constraints are more restrictive than nonholonomic kinematic constraints when the robt is in motion.

- There can be at most 3 regions that the robot can be depending on its own velocity and the velocity of the obstacle namely *front*, *rear* and *diverging*. All these regions are basically maneuvers that the robot can take to avoid colliding with obstacle.

- For multiple obstacles, the authors use the notation of S_{string}, where *string* is ordered string of maneuvers (f,r,d) that the robot can take. For example, S_{fr} means the maneuver that will enable the robot to avoid first obstacle by going in *front* of it and avoid the second obstacle by going in *rear* of it.

- From this reachable velocities, the robot can then calculate reachable avoidance velocities (RAV) by substracting VO from reachable velocities.

- To calculate avoidance trajectories, authors suggest two approaches.

  1. Global search over all feasible meaneuvers at regular interval.
     - Global search is performed by creating a search tree.
     - To reduce the complexity, the authors suggest to use grids for each RAV.
     - Each depth level in the tree represents time.
  2. Heuristic search for online application when the trajectories of obstacles are not known before hand. There are 2 hueristic proposed by the authors
     (a) TG (to goal): Selects highest avoidance velocity in direction of goal
     (b) MV (maximum velocity): Selects maximum velocity towards the goal within α variance.
     (c) ST (structure): Selects velocity according to obstacle's perceived risk.

- The heuristic approaches is a function which prioritises
  - **Primary goal**: survival of robot
  - **Secondary goal**: reaching the target, minimizing travel time and selecting desired trajectory structure

- The heuristic approaches could be combined or switched depending on the situation.

- An interesting fact is that if the time interval between 2 calculations is set very high, then there might not be any solutions as the whole RAV would be covered by VO.

- Experiments on real robot was perfromed on the highway exit ramp problem. Here obstacles are traveling at constant speed in a direction in their lanes. The robot has to reach an exit from the highway while avoiding the obstacles.

- First global search was performed with interval of 1 second and discretising the RAV to 12 points on average. The tree was expanded to depth of 5. Using iterative deepening algorithm, motion time was 3.5 seconds using S_{ff}.

- TG took 6.07 seconds using S_{rr}

- MV took 3.56 seconds using S_{ff}

- Using both MV and TG strategies, the robot took 5.31 seconds and used the maneuver S_{fr}.

2 Deficits and contributions

- There were no experiments performed on a real robot to test the theory. All the experiments that the authors show were performed in computer simulation of circular objects.

- There was no data given on how much actual processing time each iteration takes in TG or MV. Time taken was only given for global search.
- The velocity of the obstacles around the robot has to be explicilty provided to the robot.
- Only circular objects are considered. The authors defend their position by saying that any shape can be approximated by a set of circles.
- The advantages of this approach are
  1. 'It permits efficient geometric representation of potential avoidance maneuvers of the moving obstacle'
  2. No limit on number of obstacle to be avoided.
  3. No seperate representation for stationary and moving obstacle.
  4. Considers robot dynamics and actuator constraints.

References

[1] Fiorini, Paolo, and Zvi Shiller. "Motion planning in dynamic environment using velocity obstacle". International Journal of Robotics Research, 1998.