

Summary of reading reports for best approaches in “Robot motion planning in dynamic environment”

Dharmin Bakaraniya

June 23, 2018

1 Velocity based

1.1 Dynamic window based approach to mobile robot motion control in the presence of moving obstacles[1]

- As the name states, this approach is an extension of DWA[2].
- It is similar to the global DWA[3], but instead of combining DWA with NF1 algorithm, they have combined DWA with FD* algorithm[4] for global path planning.
- The authors considers the obstacles (moving or static) as (a group of) occupied cells rather than an actual object with some convex shape.
- Moving obstacles are detected as moving occupied cells. The planner projects the trajectory of these moving cells based on their velocity and direction.
- If the trajectory of robot is intersecting with the trajectories of any cells at time t, then the occupancy grid is updated for FD* algorithm. The FD* algorithm will see the location of collision as occupied and the actual location of the occupied cell is seen as free.
- On top of this, the authors have created a measure to prevent the robot from oscillating and safety cost mask for computational efficiency.
- Their experiments are very simplistic but the approach seems quite robust.
- Works for robot of any shape with any type of kinematic and dynamic constraints.
- Does not assume the shape of the obstacle.
- Assumes velocity of obstacles to remain same for few iterations.
- Real time performance but computationally expensive.
- Does not address errors in perception and control.

1.2 Efficient and safe on-line motion planning in dynamic environments[5]

- This approach combines NLVO[6] (which is based on Velocity obstacle[7]) and Inevitable collision state (ICS)[8].
- The authors show that choosing safe time horizon before actual collision is very important to ensure complete safety. After this time has passed, the robot will have only 2 options, either to stop moving or to pass the obstacle.
- The authors state that their approach is for local planning while a global planner will provide intermediate checkpoints to reach goal position. The main goal of the proposed local planner is to avoid ICS at all costs. It's secondary goal is to reach target position.
- At each time interval, NLVO are calculated and *attainable Cartesian velocity* (ACV) are generated based on dynamic constraints. Tree search and cost function are applied to find near optimal trajectories.
- The experiment to test this approach was done in a simulation where a point sized robot successfully avoided 70 moving obstacles in a dense area and reached the target in near optimal time. The authors reassure that point mass robot is just used from simplicity and it is not a limitation.
- Works with obstacles having non linear velocity.
- Guarantees to never enter ICS.
- Tested only for point mass holonomic robot.
- The approach only solves local navigation, global planner is still needed.
- Does not address errors in perception and control (eventhough ICS approach[8] considered error in preception).

1.3 Motion planning in dynamic environments using the velocity space[9]

- This approach is an extension of velocity obstacle[7].
- The approach maps all the information (goal position, sensor readings, current velocity, etc) to velocity space.
- Similar to velocity obstacle, *dynamic object velocity* is defined. This is a 3 dimensional surface describing not only which velocity will result in collision but also when it will happen. This surface is defined in *Dynamic velocity space* (DVS) which is velocity space with time as its third dimension.
- A window is considered around current velocity while choosing motion command. The size of this window is dependent on dynamic constraints of the robot.
- The experiments show a car like robot navigating among 4 moving obstacles along a minimum time trajectory in a simulation.

- Works for robot of any shape with any kind of kinematic and dynamic constraints (as long as they can move straight and move in a circle).
- Everything has to be mapped to velocity space.
- Not tested for complex dynamic environment with large number of obstacles.
- Does not address errors in perception and control.

2 Roadmap based

2.1 Randomized kinodynamic motion planning with moving obstacles[10]

- A set of randomly chosen intermediate locations in configuration space are chosen. For each of these configurations, if there is no collision, then they are added to a tree as *milestones*.
- A path from start to goal is then calculated based on the above generated tree.
- The authors state that the way in which sampling is performed, determines how fast the solution is found.
- The authors propose a near optimal sampling solution. They also provide a mathematical proof for how their algorithm is guaranteed to find solution with increasing number of milestones. They also provide some techniques to increase the performance of the planner.
- The experiments are performed for multiple types of robot in simulation and one on a real holonomic robot.
- Works for robot of any shape with any type of kinematic constraints.
- Works with obstacles having non linear velocity and randomly updated heading.
- Possibility of not finding solution even if it exists.
- Does not provide optimal path to reach target position.
- Does not address errors in perception and control.

3 Sensor based

3.1 Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios[11]

- The authors propose a collision avoidance motion planner for dynamic environment based on situated activity paradigm. Here the actions are chosen based on a predefined set of situations.
- The authors describe their approach consisting of 5 situations with their 5 corresponding actions.

- Through experiments performed on a real holonomic circular robot, they showed that their approach is very robust in highly cluttered and dense environment and even in local minima problem.
- The approach only solves local navigation, global planner is still needed.
- Applicable only on holonomic robot. Not tested for dynamic environments.
- Does not address errors in perception and control.

References

- [1] M. Seder and I. Petrovic, "Dynamic window based approach to mobile robot motion control in the presence of moving obstacles," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1986–1991, IEEE, 2007.
- [2] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [3] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, pp. 341–346, IEEE, 1999.
- [4] A. Stentz, "The focused d* algorithm for real-time replanning," in *International joint conference on Artificial Intelligence*, vol. 95, pp. 1652–1659, 1995.
- [5] O. Gal, Z. Shiller, and E. Rimon, "Efficient and safe on-line motion planning in dynamic environments," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 88–93, IEEE, 2009.
- [6] Z. Shiller, F. Large, and S. Sekhavat, "Motion planning in dynamic environments: Obstacles moving along arbitrary trajectories," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, pp. 3716–3721, IEEE, 2001.
- [7] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
- [8] T. Fraichard and H. Asama, "Inevitable collision states a step towards safer robots?," *Advanced Robotics*, vol. 18, no. 10, pp. 1001–1024, 2004.
- [9] E. Owen and L. Montano, "Motion planning in dynamic environments using the velocity space," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2833–2838, Aug 2005.
- [10] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.
- [11] J. Minguez and L. Montano, "Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 45–59, 2004.