Motion planning: Decoupled motion planning algorithm for high DOF robot with topological re-planner

Final report

Sergey Zolotykh
WPI
szolotykh@wpi.edu

***Abstract*—Motion planning in high dimensions is computationally hard problem, one way is to decrease difficulty of the problem is to decouple it. This project suggest decoupling algorithm were first planning path of the robot tip and then for each part of the path planning robot configuration. The algorithm should computationally less expensive than classical motion planning algorithm for high DOF robot arm.**

I. Introduction

Motion planning for high degrees of freedom system is difficult problem which is require sophisticated approaches and computational powerful equipment, since search problem in high dimensions has large branching factor. In order to simplify the high DOF problem it is possible to divide the problem on several low dimensional problems. This method called decoupling and actively applies for some classes of motion planning problems such as multiple robot motion planning or humanoid robots motion planning where planning happens separately for upper and lower parts of the robot body.

Motion planner usually use direct search methods and searches C-space with some commonly used algorithms such as A* for lower dimensional C-space or RRT (Rapidly exploring random tree) for high dimensional problem, or their modifications.

Some of the search algorithms taking to account topological features of the work space. By extending C-space with topology class. Topology class is a parameter which is show how path relates to the objects in work space. This kind of search allows to find different fusible path in work space as well as in C-space.

II. Related Work

Decoupling method is often applies for high dimensional problem on order to divide complicated problem on several simpler lower dimensional problems. This method applies to multi-robot motion planning problem [4]. Decoupling multi-robots motion planning algorithms divide multi-robot motion planning problem on several small problems for each individual robot. This allows to minimize computational time and even reduce it to real time. In work [4] the multi-robot planner first plan path for single robot with out knowledge about other robots, then plan path for for i first robots, but reduce first $i-1$ robots to 1 DOF robots. This allows to algorithm reduce C-space on each planning step to $n+i-1$, where n is number degrees of freedom of current robot. This algorithm have balance between computational speed and reliability.

III. Problem Statement

THe main goal of this project was creating fast and reliable motion planner for high degrees of freedom robotics systems on example of 10 DOF robotics arm. Smaller computation time comes with division of the problem on several simpler problems in other words applying decoupling method to the problem, but from other hand this method make system less reliable and unstable. In order to increase reliability and find balance between computation time and reliability the re-planning methods can be applying to the problem to compensate not compliances of decoupling methods. In this work we trying to develop decoupled motion planning algorithm for high dimensional robots with topological re-planner.

IV. Methodology

The main idea of the algorithm is to divide motion planning process of robotics arm on two separate phases. In a first phase the calculate the path for end effector of the robotics arm from start position to the goal position. During the second phase find configuration of the arm for each point on the path of end effector. In order if motion planner is fail during first or second phases, the algorithm may launch the topological re-planner to find another fusible path for end effector throw obstacles.

A. Calculation of topological class

Topological class shows how many times end effector went around obstacle while moving from initial position to the end position. In order to calculate topological class of the path in environment with single obstacle we have to summarize all angles between vector from center of the obstacle to previous position of end effector $u2$ and vector from center of obstacle to current position of end effector $u1$ (see figure 1).

$$class = \sum_{i}^{N} direction * \theta_i \tag{1}$$

The angle between two vectors can calculated by equation 2 which use dot product of two vectors. Results of this equation is always positive which is means that it shows only absolute value of $theta$ and does not show the direction of change.

$$\cos(\theta_i) = \frac{u1 \cdot u2}{\|u1\| \, \|u2\|} \tag{2}$$

In order to calculate direction of movements of end effector with respect to obstacle we use the cross product equation of two vectors to obtain equation 3 for direction. This equation return positive value for counterclockwise direction and negative value for clockwise direction.

$$direction = u1_x * u2_y - u1_y * u2_x \tag{3}$$

B. First phase: End effector path planner

First phase of motion planner starts from discretization of work space and creating occupation grid. Size of grid cells should be chosen depending on problem and constrains parameters such as computation time or available failure rate. The choice of grid size affect not only on end effector path planning phase, but also affect on arm configuration planning phase.

For actual path search algorithm was chosen A* augmented to find shortest path with different topology. This A* searches in $2+n$ dimensional space where n is number of obstacles in environment.

$$[x, y, TopologicalClass] \tag{4}$$

The A* use quotidian heuristic (Eq. 5) and taking to account only difference between coordinates.

$$heuristic = \Delta x^2 + \Delta y^2 \tag{5}$$

Also A* algorithm implemented for this project use 8-connected approach to generate new children nodes.

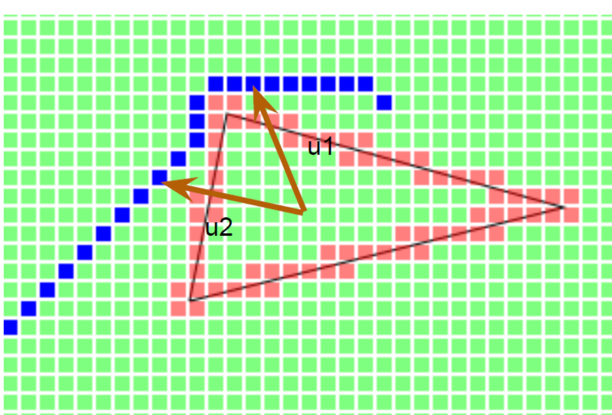

Fig. 1: Calculation of topology class

Figure 2 shows an example of path of topology (-1, 1) generated by augmented A*. It can be seem that path avoids second obstacle (bottom right) counterclockwise and first obstacle (top left) clockwise.

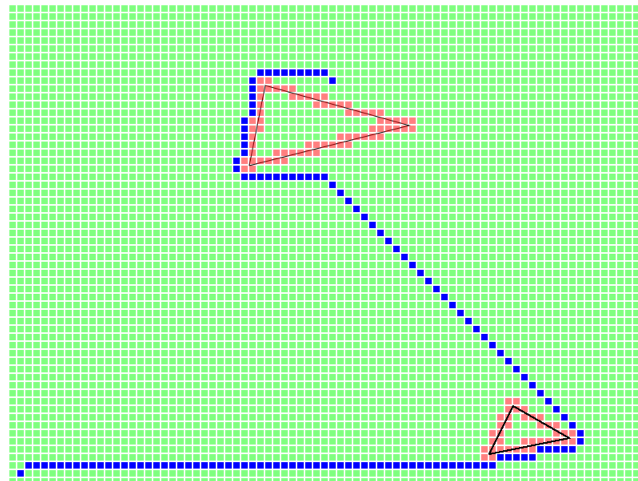

Fig. 2: Example of path generated by A*. Topological class of the path is (-1, 1)

C. Second Phase: Arm configuration planner

During the second phase planner looks for configurations of the arm which can place arm end effector to right point or right sector bounded by grid size on the path generated during first phase. In order to find single configuration the probabilistic search algorithm was implemented. It searches between randomly generated arm configurations for configuration which is closer to previous one, placing end effector to right place and collision free.

The configuration search algorithm generates set of randomly generated configurations using previous configuration as a base for all of them. First, it is choosing joints which value should be modify, by probabilistic weight of joint. Lower joints have smaller probabilistic weight, than upper ones which is means that the algorithm often choose upper joins of the arm, rather lowers. This allows to avoid large changes in configurations and generate configurations which is close to each other. Second, algorithm randomly choose values on what it will modify chosen joints.

D. Re-planning

There are several condition for re-planning. Re-planning may occur because the arm can not reach the point on generated path for end effector (invalid path condition), or configuration planner can not find fusible configuration for arm for some position of end effector (invalid configuration condition). Re-planner tries to find path for end effector with different topology to avoid felling condition.

V. Results/Example Problems

To test algorithm functionality, it performance and reliability the simulation was created and several test was launched.

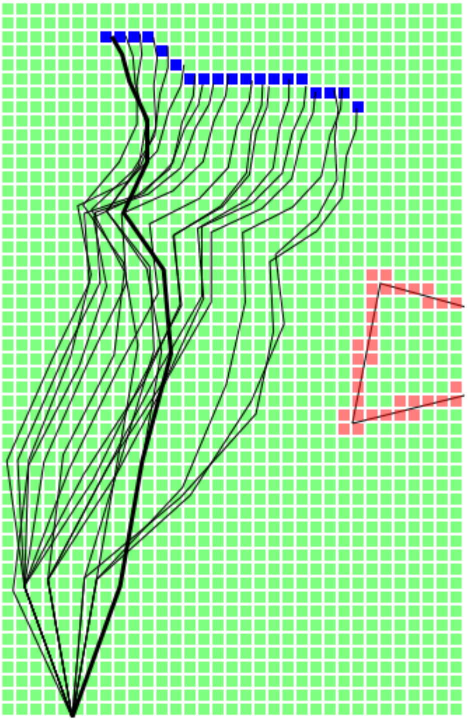

Fig. 3: Example of arm configurations search algorithm

The simulation is a set of functional scripts written on python performing different parts of algorithm such as workspace discretization, path planning for end effector, and configuration search. Those scripts produce json output files which can be used as a input to next stage scripts. For visualization of output simulation data and environment was written web application on JavaScript. This application takes as a input json files produced by each simulation script and visualize their data no HTML canvas. The visualized result of simulation you can see on figure 4 and 5.

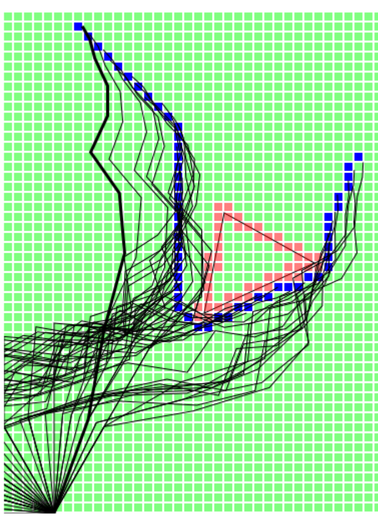

Fig. 4: First topological test simulation results

As was mentioned, several test was performed in order to test algorithm functionality. The environment for all test stays the same, but planning conditions verify.

First was tested general functionality of algorithm how it can calculate paths for end effector and find arm configurations. The results of this test is shown on figure 4 and 5.

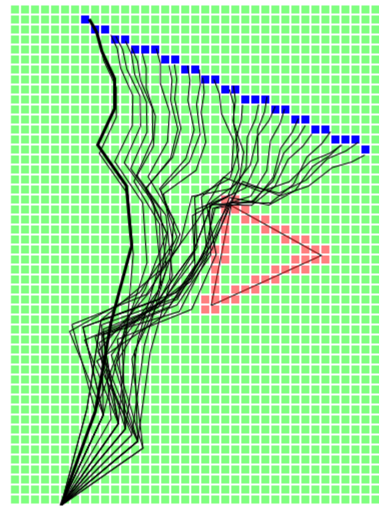

Fig. 5: Second topological test simulation results

Second test performed to test computational time and reliability of algorithm. it shows that algorithm can compute 53 arm configurations for path shown on figure 6 in less then 10 seconds in average, with more then half runs in between 2 and 3 second. Which is really fast result for 10 DOF arm and algorithm written on Python. This test also shows the computation time can be very large, more then 60 seconds. This happens because algorithm uses random search to find configurations.

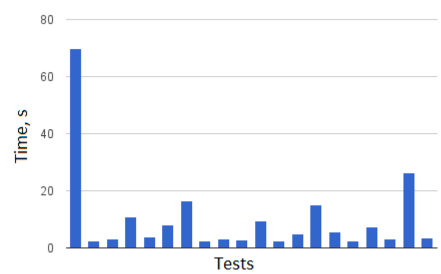

Fig. 6: Path configuration search time

VI. Conclusion

The algorithm introduced in this project shows reliability and high performance for motion planning of high degrees of freedom robotics arm. This was shown on 10 DOF robotics arm in 2D. Same algorithm can applied to different high DOF motion planning problems with minor changes for specifics of particular problem. The algorithm proving possibility of applying decoupling concept to robotics arms motion planning and may applying for real application only with specific additions for application.

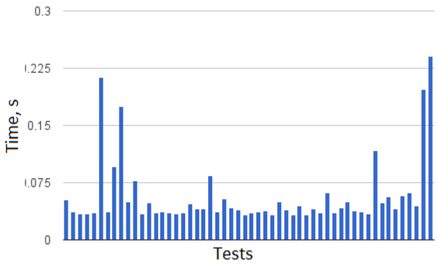

Fig. 7: Time of search single configuration in path

VII. Acknowledgements

Motion planning class lectures and assigments made large inpact on this project. Thank you professor and classmates for help and inspiration.

References

[1] LaValle, Steven M. ”Rapidly-Exploring Random Trees A New Tool for Path Planning.” (1998).

[2] Kavraki, Lydia E., et al. ”Probabilistic roadmaps for path planning in high-dimensional configuration spaces.” Robotics and Automation, IEEE Transactions on 12.4 (1996):566-580.

[3] Brock, Oliver, and Oussama Khatib. ”Real-time re-planning in high-dimensional configuration spaces using sets of homotopic paths.” Robotics and Automation, 2000. Proceedings. ICRA’00. IEEE International Conference on. Vol. 1. IEEE, 2000.

[4] Saha, Mitul, and Pekka Isto. ”Multi-robot motion planning by incremental coordination.” Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on. IEEE, 2006.

[5] Bhattacharya, Subhrajit, Maxim Likhachev, and Vijay Kumar. ”Topological constraints in search-based robot path planning.” Autonomous Robots 33.3 (2012): 273-290.