Kinodynamic Region Rapidly-exploring Random Trees (KRRRTs)

Brandon Martinez, Eli Zamora, Jory Denny, Read Sandstrom, and Nancy M. Amato

Parasol Lab, Department of Computer Science and Engineering, Texas A&M University,
College Station, TX, USA, b.m_755@hotmail.com, ezamora1222@yahoo.com
(jdenny, readamus, amato)@cse.tamu.edu

Abstract*— Kinodynamic motion planning is the problem of finding a collision-free path for a robot under constraints e.g., velocities and accelerations. State of the art techniques rely on sampling-based planning which samples and connects configurations until a valid path is found. Many sampling-based planners have been developed for non-holonomic problems e.g., Kinodynamic Rapidly-exploring Random Trees (KRRT). Although KRRT's computational efficiency is competent, it lacks human problem solving skills. Human-guided planning combines automated planning and human intuition, allowing greater problem complexity and computational efficiency. A recent human-guided planner, Region Steering, allows sampling to bias or avoid user specified workspace regions, but was designed for holonomic bodies only. In this work, we extend Region Steering to handle such problems with dynamic constraints. Our novel method, Kinodynamic Region RRT (KRRRT), allows the user to specify attract or repel regions that the planner will concentrate or avoid planning, then applies KRRT's motion constraints to the problem. In our experimental results, we found that utilizing regions made our planner up to six times faster than KRRT.

I. Introduction

In today's society, there is a large emphasis on technology, especially computers and their capabilites to develop software that could potentially shape the course of our future. Self-driving vehicles could be waiting for us in the future, but before this can happen, a great amount of work has to be put towards making this possible. One key aspect of making a robot autonomous is motion planning. Motion planning is the problem of finding a collision-free path for a robot from a start to goal configuration.

Because exact solutions are computationally intractable, *Sampling-Based Planning* is the state of the art solution. [7], [8] Sampling-based planning generates a roadmap, which is then used to extract a valid path to the goal.

Since vehicles cannot always move in straight lines, we use a diffrent sampling based planner called *Kinodynamic Rapidly-exploring Random Trees (KRRTs)*. [8] This type of planner takes into account motion constraints such as velocity and acceleration. Since these forces are applied to the robot, the points it samples are connected by curved edges. Problems that require these type of constraints are known as non-holonomic. Although KRRT has shown competent computational efficiency, its lack of problem solving skills makes fairly easy problems, intractable.

Human interaction with a planner would make extremely difficult problems, less difficult to solve. Human-guided planning combines automated planning and human intuition, allowing the planner to solve complex problems faster. [1], [5] A recent human-guided planner, Region Steering, allows the user to specify attract or repel regions, helping the planner concentrate its planning to those specific attract regions. [3]In this case, the human guides the planner to where it most needs to focus. Although this method showed promising results, it is unclear how to extend to non-holonomic problems.

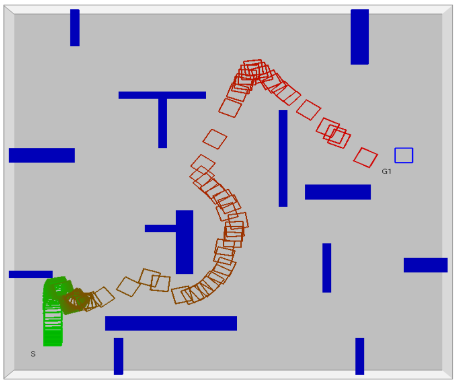

Fig. 1. 2D Environment path take by robot after user input

In this work, we extend Region Steering so that it can be applied to non-holonomic problems. We present a method, Kinodynamic Region RRT (KRRRT), that allows the user to specify regions, while applying non-holonomic constraints to the robot. Using attract regions, the planner will concentrate its planning there, reducing sampling time. Our method would be ideal in problems that would take an automated planner a sizeable amount of time. During our tests, we tested for computational speed against KRRT, and had favorable results.

II. Related Work

In this section, we address related work that is most relevant with our proposed planner.

A. Preliminaries

Kinodynamic motion planning is the problem of finding a collision free path for a robot with motion constraints e.g., velocities and accelerations. The robot placement can be described by a point in the *state-space*. As the planner plans a path for the robot, the edges created between points curve depending on the value of the motion constraint. These constraints make KRRT a difficult problem, since inevitable collision is greater in a planner whose velocity increases.

B. *Sampling-Based Motion Planning*

Sampling-Based motion planning is the process of randomly sampling configurations in the workspace. Depending on where the configurations are sampled, the planner proceeds to identify them as valid, or invalid. Finally, the planner attempts to find a valid path from the start to the goal by connecting valid points that are near each other.

C. *Kinodynamic RRT (KRRT).*

This planner's goal is satisfy both global obstacle constraints and local differential constaints. Global obstacle constraints are those that allow a collision free path. Local differential constraints are applied to non-holonomic robots. This planner inserts an initial state as a vertex. Then, it repeatedly selects a point at random and finds the nearest neighbor in the tree. It chooses a control that pulls the vertex towards the random point, then proceeds to insert a new edge and vertex. This generates a tree that rapidly explores the state space. Although this planner proved to be efficient, it lacked the problem solving skills that humans possess.

D. *Human-Guided Motion Planning.*

This sampling method was the most crucial in making our proposed planner work. Human-Guided planning consists of an active interaction between a user and an algorithm. The user anaylzes the workspace to determine a solution while the planner takes care of the high-precision computations. [4], [6] We used a method of Human-guided planning called *Region Steering*. This method allows the user to specify regions in the workspace that the planner would bias or avoid sampling in. Specifying attract regions helps concentrate the planner's resources to that certain area. Region steering provides live feedback to the user during the planning process. This allows the user to identify regions that help the planner where it most needs help. This method was developed for holonomic robots only, meaning that the robot had no acceleration, and could only move in straight lines.

III. Main method

Our method combines *Region Steering*, a holonomic technique, and *KRRT*, a non-holonomic planner, creating the first human-guided non-holonomic planner. Through *Collaborative Planning*, a closed-loop interaction between an operator and a robot, the user can continously input regions into the environment thus steering the robot in a desired direction. [3] As shown in Algorithm 1, our method begins by allowing the user to specify regions in a given environment prior to planning. Once the user begins the planning process:

- regions can be moved within the environment (dynamic regions)
- additional regions can be created (multiple regions)
- regions remain initial position (static regions)
- exisiting regions can be deleted

all to help the planning process. The planner then concentrates planning towards regions currently specified in the environment, and/or towards the goal.

Algorithm 1 KRRRT Algorithm

Input: Environment env, Region
Output: Roadmap $rdmp$

while $\neg!envsovled$ **do**
  $n \leftarrow$ RANDOMRATIO = RAND
  if $randomRatio < growthFocus$ **then**
    $dir \leftarrow GoalBiasedDirection$
  else
    $dir \leftarrow RegionDirection$
  $recent \leftarrow ExpandTree(dir)$
  if $recent!valid$ **then**
    $qNew \leftarrow Vertex(recent)$
  if $EvaluateGoal(recent)$ **then**
    return $envsolved$
return o

IV. Experimental Analysis

We compared our planner, Kinodynamic Region RRT, to the fully automated Kinodynamic RRT planner. We showcase how our technique leverages the input provided by the user to enhance roadmap construction time. We seek to show the sufficiency of our planner given the golobal and local constraints it faces.

A. *Setup*

The methods were all implemented in a C++ motion planning library developed in the Parasol Lab at Texas A&M University. It uses a distributed graph data structure from the Standard Template Adaptive Parallel Library (STAPL) [2], a C++ library designed for parallel computing. All experiments were run on Dell Optiplex 780 computers running Fedore 20 with Intel Core 2 Quad CPU 2.83 GHz processors with the GNU gcc compiler version 4.7.

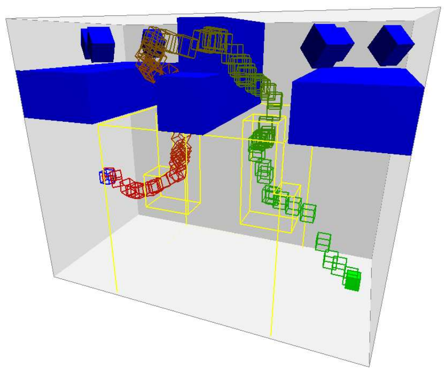

Fig. 2. 3D Environment path taken by robot after user input

For our experiments, roadmap construction terminates after solving a construction query or sampling a total of 15

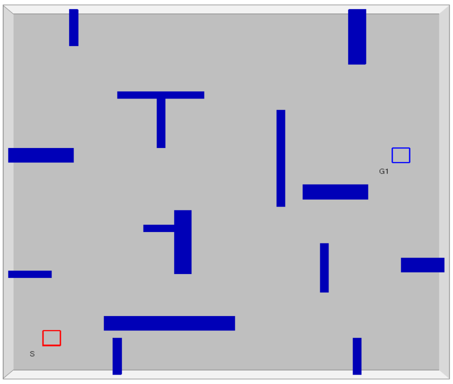

Fig. 3. 2D Environment

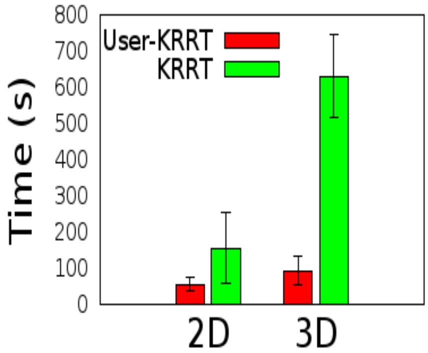

Fig. 5. Chart of average run time results

thousand nodes. We report the average time taken over ten trials for each planner in two different environments. In the istance that a planner did not solve, we recorded the time taken to generate 15 thousand nodes. Our time is shown as total time taken between user and planner interaction as:

- constant feedback was being given by the planner to the user
- constant input was given by the user to the planner.

The environments we used are shown in Figure 3 and Figure 4. The construction queries are maked by a start configuration (red) and a goal configuration (blue).

- 2D Environment (Figure 3):
  - Small squared robot.
  - Restricted to movement about the XY plane.
  - 6 dimensional state-space with 4 controls.
  - Controls only allow movement; forward, backward, rotation left and right.
  - Robot had to steer itself in an environment that allowed to build a great amount of velocity making inevitable collision a big factor.
- 3D Environment (Figure 4):
  - Small fully controlable rectangular robot.
  - 12 dimensional state-space with 12 controls.
  - Robot had to traverse through several narrow passages, as well as avoid some areas of light clutter.

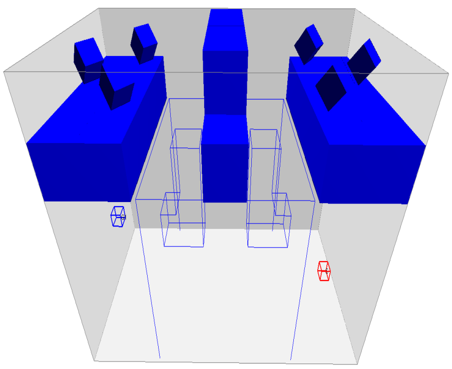

Fig. 4. 3D Environment

B. Results

The data for our 2D environment trials showed that our planner's solving time was three times faster than KRRT's, and six times faster in our 3D environment. In the 2D environment, Kinodynamic RRT solved 60 percent of the time, and had an average time of 155.5s. Our planner solved 100 percent of the time, and had an average time of 56s. In our 3D environment, KRRT solved only 10 percent of the time with an average time of 631s. Our planner solved 100 percent of the time with an average time of 92.3s.

V. Discussion

Our planner showed that user guidence increases a planners precision and speed making problems that would otherwise take minutes to solve, only took seconds with a users input. In our trials, the user used multiple regions and dynamic regions in environment throughout the planning until the goal was reached. Since this is just the basis of a user guided non-holonomic planner, real life implemintation will be left to future work (i.e. semi-autonomous vehicles, drones, robotics, etc.).

VI. Conclusion

In this paper, we introduced the very first planner that allows Region Steering be applied to non-holonomic constraints. Our combination of KRRT and Region Steering, had a dramatic decrease in computational time compared to KRRT. We demonstrated that it is possible to guide a high-dimensional, non-holonomic problem with low dimensional input.

Acknowledgements

Supported in part by an NSF Graduate Research Fellowship,

This research was supported by the Department of Computer Science and Engineering of Texas A&M University College Station.

References

[1] O. B. Bayazit, G. Song, and N. M. Amato. Enhancing randomized motion planners: Exploring with haptic hints. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 529–536, 2000.

[2] A. Buss, Harshvardhan, I. Papadopoulos, O. Pearce, T. Smith, G. Tanase, N. Thomas, X. Xu, M. Bianco, N. M. Amato, and L. Rauchwerger. STAPL: Standard template adaptive parallel library. In *Proc. Annual Haifa Experimental Systems Conference (SYSTOR)*, pages 1–10, New York, NY, USA, 2010. ACM.

[3] J. Denny, R. Sandstrom, N. Julian, and N. Amato. A region-based strategy for collaborative roadmap construction. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Istanbul, Turkey, August 2014.

[4] C. Guo, T. Tarn, N. Xi, and A. Bejczy. Fusion of human and machine intelligence for telerobotic systems. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3110–3115, 1995.

[5] Y. Hwang, K. Cho, S. Lee, S. Park, and S. Kang. Human computer cooperation in interactive motion planning. In *Proc. IEEE Int. Conf. Adv. Robot. (ICAR)*, pages 571–576, 1997.

[6] I. Ivanisevic and V. J. Lumelsky. Configuration space as a means for augmenting human performance in teleoperation tasks. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 30(3):471–484, Jun 2000.

[7] L. E. Kavraki, P. Švestka, J. C. Latombe, and M. H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, August 1996.

[8] S. M. LaValle and J. J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 473–479, 1999.