Optimal Sample-Based Planning for Non-holonomic Systems: Literature Review

Rebecca Abraham and Evan Lohn

Abstract— **We review a breadth of literature related to non-holonomic motion planning methods, then provide a deep analysis of current solutions to near-optimal sample-based planning with obstacles with the RRT* algorithm. We suggest a method for achieving the desirable properties of RRT* while achieving dynamical feasibility of generated paths in non-holonomic systems.**

I. INTRODUCTION

A. Motivation

Most real-world robotics problems contain motion planning as a subproblem. In other words, it is fundamentally important to be able to be able to determine what sequence of actions will allow a robot to reach some desired state. For example, having a robot to walk, pick up an object, fly through a forest, or change lanes on a highway are all problems that can be solved with some variant of motion planning.

The space of motion planning problems is understandably very large, and potential solutions are typically developed while assuming additional constraints. In this paper, we focus primarily on sample-based planning methods, or in other words methods using samples from the robot's state space or task space to generate a path to the goal state. We also primarily analyze methods that work with non-holonomic systems (systems with at least one constraint such that instantaneous movement in any direction in state space is not always possible). Finally, we also focus on methods that produce near-optimal solutions for reasonable measures of optimality such as minimal distance travelled or input magnitude.

B. Approach

We first review a broad section of literature related to different types of motion planning for non-holonomic systems. We then provide a deep examination of recent improvements to the RRT* algorithm, a popular sampling-based planner which has several of the desirable properties for a solution to our problem. In a slight tangent, we describe ways in which related literature based in systems with only holonomic constraints could be adapted to solve the main non-holonomic motion planning problem. Finally, we provide some preliminary experiments showing the intuitive benefits of the combination we propose.

Our approach to reviewing literature in this paper was not systematic; we started with two main previous literature reviews and followed citations from there to find most of the papers we summarize here [1], [2]. In areas where this approach seemed lacking, we did our own Google searches. Additionally, when looking at early work in the field, we tended to bias our efforts towards papers with more citations.

II. NON-HOLONOMIC MOTION PLANNING

A. Historical Context

Although it is difficult to be sure of the exact dates, our literature review found that the first mention of "non-holonomic" motion planning began in the late 1980s (although prior literature certainly dealt with such robots without using that term). Too restrict our seach, we have primarily focused oon literature that specifically refers to the systems they work with by the term "non-hoonomic". The vast majority of literature on non-holonomic motion planning does experiments on some variant of the Reed-Shepps car [3], so we omit the problem domain for most of the following papers.

B. Late 1980s, Early 1990s

Barraquand et. al [4] are some of the first authors to mention and explicitly define systems with non-holonomic constraints; they present a graph search algorithm over configuration space where successor states are generated with small-time numerical integration. The broader class of kinodynamic motion planning (planning that respects any type of dynamics constraint, such as limits on input force magnitudes) was introduced around the same time [5]. A precursor to the path-biasing method for RRTs is presented in [6]; the authors first construct a dynamically infeasible path, then use a local planner that respects non-holonomic constraints to plan between waypoints on that path. Another path-biasing method involved building and following a skeleton; this approach has also been expanded on with RRTs [7], [8]. In [9], the author specifies open loop and feedback control functions as an average of low and high frequency components, then analytically solves for the magnitudes of those components.

It was also around this time that Professor Sastry published the paper on steering with sinusoids [10], a beautiful mathematical result that unfortunately has practical difficulties with drift and obstacles. Another important early paper [11] established several important facts about the controllability of multi-body robots (i.e. trailers), and provided a motion planning algorithm similar to [4]. Other work uses ideas from Ritz approximation theory to solve a nonlinear optimization problem where the space of inputs u is approximated by a finite basis of functions [12].

C. Late 1990s

Divelbiss & Wen [13] used nonlinear least squares to optimize over the configuration path with penalty functions to ensure dynamic feasibility. Tanner & Kyriakopoulos [14] provide a closed-loop controller for non-holonomic systems and prove its stability using Lyapunov's direct method; however their approach relies on the existence of a navigation function with no local minima that the robot can use to navigate the space. Sekhavat & Laumond [15] show how to update the steering with sinusoids approach for chained systems [10] to deal with obstacles in certain types of systems by breaking up the path and planning between waypoints as necessary (similar to [6]).

D. 2000s

Duleba & Sasiadek [16] introduced a way to use Newton's method to directly optimize trajectories for energy usage. Papadopoulos et. al [17] transformed the state space to make satisfying the nonholonomic constraints simple, and planned with smooth polynomials in that space to get dynamically feasible trajectories. In [18], the authors create a special search space from the original state space that includes only dynamically feasible paths, allowing trajectories to be generated with graph search algorithms. In [19], the authors presented CHOMP, an optimization problem formalization of the motion planning problem solved with gradient descent.

E. Recent methods

More recently, there have been fewer specific references to "non-holonomic" path planning in literature. Many of the recent sample based planners such as RRT(*) [20], [21] (covered in the next section), PRM [22], and FMT [23] employ some justification for the dynamic feasibility of their generated trajectories such as explicit smoothing techniques like splines [24], [25] or implicit ones like the guaranteed existence of a local planner between any sufficiently close states. We have also seen several attempts to solve motion planning with reinforcement learning algorithms such as Q-Learning [26], but these methods are still largely experimental.

III. RAPIDLY-EXPLORING RANDOM TREE (RRT)

A. Generic RRT

All variants of RRT follow the same basic pattern of the following pseudo-code:

Algorithm 1 Generic RRT pseudo-code

```
Require: Planning space S, starting start start, goal state goal
procedure PLAN_RRT
    G ← (V, E)                         ▷ Initialize tree(s)
    while not path_found(G) do
        s ← sample(S)                  ▷ sample from planning space
        maybe_update_tree(G, s)
    end while
    return construct_path(G)
end procedure
```

The diversity in RRT literature largely comes from the variety of approaches to sampling either the configuration or task space, as well as the usage of those samples in constructing a path. The original RRT paper [20] initializes a tree with a vertex at the starting configuration. For sampling, it samples uniformly from the task space. It then finds the tree node n closest to that sample, picks an input u that would take the robot from n to a state s' close to the sampled s while avoiding collisions, then adds to the tree a vertex s' and an edge (n, s') that stores the calculated input u.

The original RRT algorithm is probabilistically complete, meaning that with enough samples the algorithm will generate a path to the goal. However, RRT's do not by default guarantee that generated paths are dynamically feasible in systems with non-holonomic constraints. Additionally, basic RRT's can be inefficient in scenarios where planning through narrow gaps between obstacles is required, because the uniform sampling process may take longer than necessary to generate samples that allow the RRT to extend through those gaps. Finally, the basic RRT algorithm does not guarantee optimality for any generated paths.

*B. RRT**

The problem of optimality for RRT's was addressed through the introduction of the RRT* algorithm [21]. In RRT*, each node n in the tree maintains a cost metric c (i.e. distance) of the path taken from the tree's root to n. Instead of simply selecting the nearest node in the planning space with respect to that cost (as is done in basic RRT), new nodes n are connected to a parent node p such that the cost of node n $c(n) = d(p, n) + c(p)$ is minimized, where $d(p, n)$ is the cost of connecting p to n. Then, each node in the tree changes its parent to the new node n if doing so would decrease its total cost. These two additions to the algorithm make RRT* asymptotically optimal, i.e. as the number of samples taken increases, the cost of the path obtained by RRT* approaches the cost of the optimal path. However, further improvements are necessary to solve the problems of sample efficiency and dynamical feasibility.

C. RRT* with improved sample efficiency

An early adaptation to vanilla RRT biases the sampling procedure towards the portion of the planning space closer to obstacles [27]; this allows the RRT to more efficiently enter and explore complicated spaces cluttered with obstacles. Karaman et. al [28] used the asymptotic optimality of RRT* to re-plan more optimal trajectories as the robot executed an initial "committed" trajectory segment, and use branch-and-bound to keep the RRT's growth focused on improvements to the main trajectory. Nasir et. al [29] introduced RRT*-SMART, which biases the sampling process towards states near the best path found. RRT*-SMART also uses "Intelligent Sampling", a process by which more samples are drawn near several key points selected near obstacles on the current path. Intuitively, this allows the algorithm to more quickly generate tighter turns around obstacles if such a maneuver is optimal. In work done by Ma et. al, a problem-specific rule-based biasing on the sampling process led to an efficient RRT variant called Fast-RRT that yielded impressive results when run in real time on autonomous vehicles [30]. Another variant known as Guided RRT [31] uses the path-biasing approach by having geometric trees explore that planning space first and then biasing the growth of the RRT along the explored paths.

To improve sampling efficiency by searching the planning space from both the start and goal, bidirectional versions of RRT and RRT* (RRT-Connect and B-RRT* respectively) were proposed [32], [33]. Building off this previous work, Qureshi & Ayaz combined both the aforementioned intelligent sampling with B-RRT* and created IB-RRT* [34]. In the space of the same year, Qureshi & Ayaz also published work on P-RRT* [35], an algorithm which uses Artificial Potential Fields to bias the sampling of RRT* for increased efficiency. Somewhat predictably, in the following year Qureshi & Ayaz were both authors on the paper that introduced PIB-RRT*, which combined the three improvements of artificial potential field sampling bias, intelligent sampling, and bidirectionality for improving RRT* sample efficiency [36].

D. RRT* with dynamical feasibility

Some early work modified the routine for choosing an min-cost parent p for a new node n. The new algorithm considered only p such that it was provably possible to steer from p to n within a certain amount of time while satisfying the non-holonomic constraints of the system [37]. However, this methodology assumes the existence of a local planner for steering between nodes.

Other approaches have approximated system dynamics as linear in order to create an approximate local planner. The authors of these paper justify the dynamical feasibility of the returned paths based on the validity of their planners in the neighborhood of the linearization point. Perez et. al [38] introduce LQR-RRT* with a local planner and extension heuristic based on Linear-Quadratic Regulators (LQR). Webb & van den Burg [39] introduced Kinodynamic RRT*, which used optimal control for a system's linearized dynamics as a local planner.

More recently, there have been several RRT variants introduced that construct an initial path that may violate dynamic constraints, then use an RRT that is biased to grow along that path to construct a dynamically feasible path. Palmieri et. al [40] introduced Theta*-RRT, which uses any-angle path search for initial paths, and RRT with smooth geodesics for local steering. Dong et. al [8] used the wavefront algorithm [41] to build a skeleton of the search space that an RRT then follows. Both algorithms rely on the assumption that the robot is small-time controllable; in other words that it can reach a neighborhood of its initial state in any finite amount of time.

There have also been several approaches that build paths with either splines [24], [25] or motion primitives to create dynamically feasible paths with RRTs. In particular, an approach called RRT-MP [42] generated motion primitives via Central Pattern Generators (CPGs) and used those motion primitives to expand the RRT.

For the experiments we conducted based on this literature review, we combined several common ideas from the wide array of prior literature on path planning. Namely, we hypothesized that the sample-efficient properties of certain types of RRTs would serve as a good initial path for path biasing a secondary RRT that guaranteed dynamical feasibility through motion primitives.

IV. Motion Primitives

One approach to constructing dynamically feasible paths is to chain together dynamically feasible motion primitives. This is the approach we used in our experimentation, and we briefly discuss approaches to building sets of motion primitives in this section. Given a dictionary of interesting motion primitives, one approach to planning is to sequence these primitives. A drawback of this approach is that generated paths have no smoothness guarantees. Sometimes, sequencing primitives requires the use of transitional motions between primitives. In [43], trim trajectories (steady motions) and maneuvers (unsteady motions) are used, so that generated paths alternate between trims and maneuvers.

Any dictionary of motion primitives must be expressive enough to span the state space, up to some desired resolution. Systematic approaches to achieving this include twogrid based approaches. In [44], a suggested approach is to discretize the state space and pre-compute optimal motion primitives to all the discrete points within a neighborhood of a start point. This approach requires an local planner. In [45], Vukosavljev et al hierarchically construct increasingly expressive motion primitives from a finite set of basic motion primitives. This work focuses on a cluttered, multi-agent setting, where computing a centralized control policy built up of simple primitives becomes exceedingly computationally expensive with added agents. To manage this complexity, building up motion primitives allows more abstract motions to be constructed.

V. EXPERIMENTATION

In this section, we discuss the methodology behind our experimentation. Since the focus of this work is on reviewing relevant literature, we did not run comprehensive experiments. Rather, we share example results to demonstrate the intuitive benefits of carefully selected motion primitives and sampling schemes.

To demonstrate the potential advantages of different approaches to dynamically-feasible path planning around obstacles for non-holonomic systems, we implemented some of the algorithms we found during review and compared the results qualitatively. Our simulations were created by extending the RRT implementations within the Python Robotics package [46]. To ensure dynamic feasibility, we built paths by chaining together feasible unicycle-model motion primitives. We ran each planning algorithm for a set number of iterations, and chose the best path by selecting the shortest path (if any) that ended within a threshold distance of the goal. We defined the distance between states for the unicycle system as a weighted sum of Euclidian distance between the states' x, y coordinates and the minimum distance between the states' orientation, $\theta \pmod{2\pi}$:

$$\text{MinAngle}(\theta_1, \theta_2) = \begin{cases} \text{abs}(\theta_1 - \theta_2) & \text{abs}(\theta_1 - \theta_2) \leq \pi \\ 2\pi - abs(\theta_1 - \theta_2) & 2\pi < \text{abs}(\theta_1 - \theta_2) \end{cases}$$

So that the total distance between states (x_1, y_1, θ_1) and (x_2, y_2, θ_2) is

$$\left\| \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} - \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \right\| + 0.1 * \text{MinAngle}(\theta_1, \theta_2)$$

This distance metric was used for selecting motion primitives by determining which primitive gets closest to a newly sampled state. For the final goal check, only Euclidian distance between the x, y coordinates was calculated, so that final orientation wasn't taken into account.

We investigated planning for two layouts:

1) Placing the goal within a U-shaped obstacle so that the agent needs to find the opening of the U
2) Positioning obstacles that require the agent to navigate through narrow openings to reach the goal

As a baseline, we ran the basic RRT algorithm with uniform samples and motion primitives that were pre-generated by setting system inputs at discrete intervals (Fig. 1). RRT was run for 200 iterations in each map (Fig. 2).

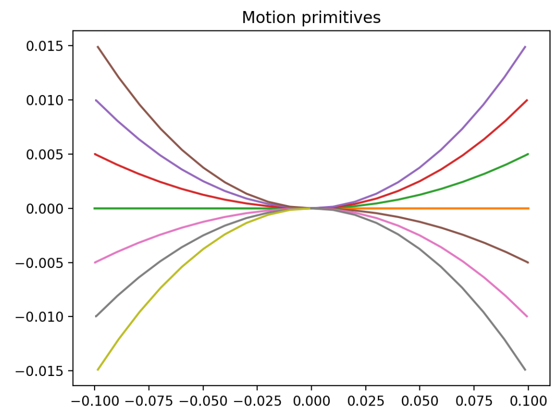

Fig. 1. Motion primitives for baselines tests

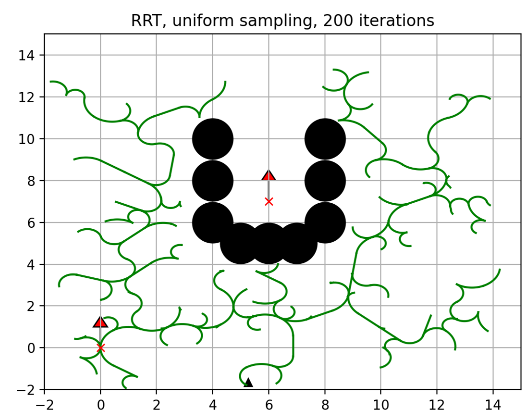

Fig. 2. Paths generated after 200 iterations of RRT, no full path found

To improve upon the baseline, we turned to path-biased sampling, which requires a two-step process. First, a preliminary path is generated with any path-planning algorithm. Then, RRT is run with motion primitives, with sampling biased towards points along the preliminary path. For our first pass, we ran RRT* to generate a preliminary dynamically infeasible path. At this step, any of the RRT* variants discussed above could be used for faster convergence. Then, we ran RRT with the same motion primitives as before, this time sampling uniformly from along the length of the pre-generated path, setting θ in the direction of the path. We also added some uniform noise to each state variable of each sample. An example of RRT* path is in Fig. 3 and a path generated by sampling from this path is in Fig. 4.

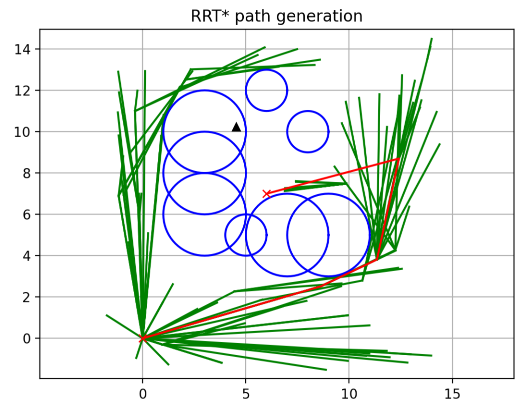

Fig. 3. Paths generated after 200 iterations of RRT*

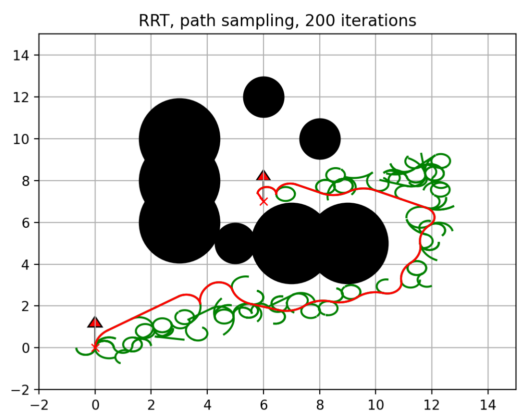

Fig. 4. Paths generated after 200 iterations of RRT, sampling a pre-generated path

For further exploration, we used hierarchical motion primitives, as done in [45]. For our Level 0 primitives, we used a subset of the primitives used previously. Our Level 1 primitives were generated from every combination of two primitives from the Level 0 set, including each primitive with itself (Fig. 5). At each iteration, the planner chose from all Level 0 and Level 1 primitives. We used the same scheme for path-biasing, and ran RRT for 200 iterations. Sample results are depicted in Fig. 6 and Fig. 7.

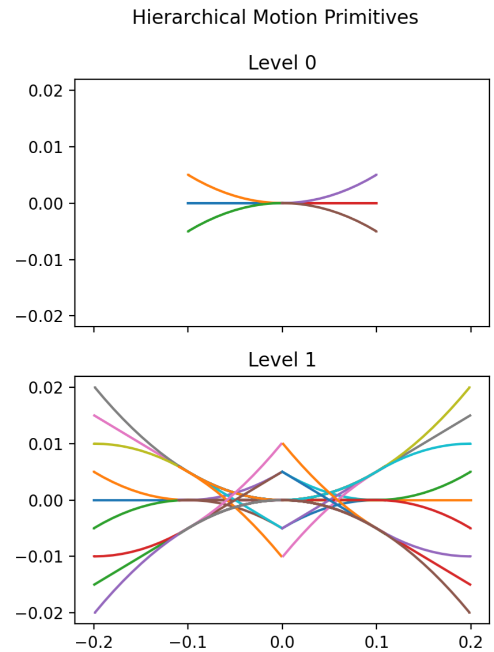

Fig. 5. Hierarchically generated motion primitives

VI. CONCLUSIONS AND FUTURE WORK

A. Conclusion

After a broad survey of non-holonomic motion planning techniques, we noted several broad trends. The path biasing technique (which we made use of in our own experiments) has been used in various forms since the earliest mention of non-holonomic motion. Additionally, local approximations and linearizations have been popular, likely due to the difficulty of solving general motion planning. Finally, optimization techniques and analytical methods have tended to produce excellent results, but at the cost of lacking the efficiency necessary to be run online. For this reason, a large portion of recent literature has focused on improving the efficiency and dynamical feasibility of sampling based planners such as RRTs.

B. Future Work

While we attempted to provide a general survey of the historical progress of non-holonomic motion planning, we

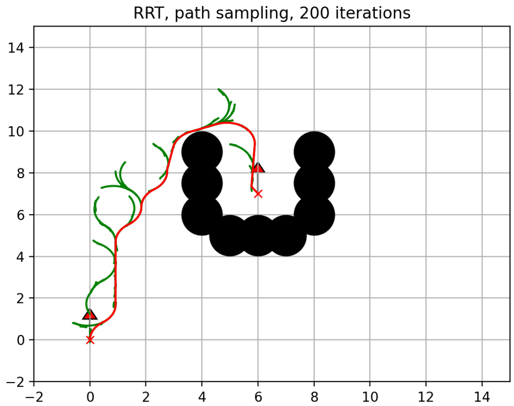

Fig. 6. RRT, path sampling and hierarchical motion primitives, 200 iterations on first map

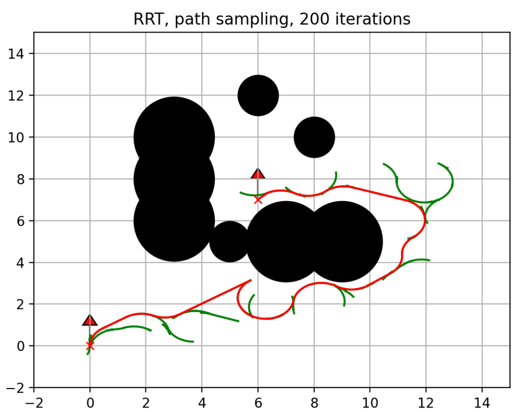

Fig. 7. RRT, path sampling and hierarchical motion primitives, 200 iterations on second map

certainly did not manage to cover all the available literature. Future researchers wanting an even more thorough treatment of non-holonomic motion planning would likely want to look into literature that aims to solve specific non-holonomic motion planning problems, or possibly look into more ways to adapt motion planning techniques that produce dynamically infeasible trajectories to solve problems with non-holonomic constraints. In terms of our own experiments, future work might provide more empirical validation of our ideas by using PIB-RRT* or a similar RRT* variant with high sample efficiency for the planning stage, and compare its performance against optimization strategies, graph search, and analytical methods.

VII. Website

For more detailed results and code, see our project website:

https://bcylincoln.github.io/eecs106b_final_project/

VIII. Acknowledgments

We would like to thank Valmik Prabhu, Amay Saxena, Jun Zeng, and Tiffany Cappellari for their knowledge and support as TAs this semester. We are also extremely grateful to Professors Ruzena Bajcsy and Shankar Sastry for taking time out of their busy schedules to teach this class. We hope this literature review is some convex combination of interesting and helpful!

References

[1] L. G. D. O. Véras, F. L. L. Medeiros, and L. N. F. Guimaráes, "Systematic literature review of sampling process in rapidly-exploring random trees," *IEEE Access*, vol. 7, pp. 50 933–50 953, 2019.

[2] M. Elbanhawi and M. Simic, "Sampling-based robot motion planning: A review," *IEEE Access*, vol. 2, pp. 56–77, 2014.

[3] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards." *Pacific Journal of Mathematics*, vol. 145, pp. 367–393, 1990.

[4] J. Barraquand and J. . Latombe, "On nonholonomic mobile robots and optimal maneuvering," in *Proceedings. IEEE International Symposium on Intelligent Control 1989*, 1989, pp. 340–347.

[5] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic motion planning," *J. ACM*, vol. 40, no. 5, p. 1048–1066, Nov. 1993. [Online]. Available: https://doi.org/10.1145/174147.174150

[6] J. . Laumond, P. E. Jacobs, M. Taix, and R. M. Murray, "A motion planner for nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 577–593, 1994.

[7] B. Mirtich and J. Canny, "Using skeletons for nonholonomic path planning among obstacles," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2533–2540 vol.3.

[8] Y. Dong, E. Camci, and E. Kayacan, "Faster rrt-based nonholonomic path planning in 2d building environments using skeleton-constrained path biasing," *Journal of Intelligent & Robotic Systems*, vol. 89, pp. 387–401, 2018.

[9] L. Gurvits, "Averaging approach to nonholonomic motion planning," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 2541–2546 vol.3.

[10] R. M. Murray and S. S. Sastry, "Steering nonholonomic systems using sinusoids," in *29th IEEE Conference on Decision and Control*, 1990, pp. 2097–2101 vol.4.

[11] J. Barraquand and J. . Latombe, "Nonholonomic multibody mobile robots: controllability and motion planning in the presence of obstacles," in *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, 1991, pp. 2328–2335 vol.3.

[12] C. Fernandes, L. Gurvits, and Zexiang Li, "Near-optimal nonholonomic motion planning for a system of coupled rigid bodies," *IEEE Transactions on Automatic Control*, vol. 39, no. 3, pp. 450–463, 1994.

[13] A. W. Divelbiss and J. T. Wen, "A path space approach to nonholonomic motion planning in the presence of obstacles," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 443–451, 1997.

[14] H. G. Tanner and K. J. Kyriakopoulos, "Nonholonomic motion planning for mobile manipulators," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 1233–1238 vol.2.

[15] S. Sekhavat and J. . Laumond, "Topological property for collision-free nonholonomic motion planning: the case of sinusoidal inputs for chained form systems," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 671–680, 1998.

[16] I. Duleba and J. Z. Sasiadek, "Nonholonomic motion planning based on newton algorithm with energy optimization," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 3, pp. 355–363, 2003.

[17] E. Papadopoulos, I. Poulakakis, and I. Papadimitriou, "On path planning and obstacle avoidance for nonholonomic platforms with manipulators: A polynomial approach," *International Journal of Robotic Research - IJRR*, vol. 21, pp. 367–383, 04 2002.

[18] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Optimal , smooth , nonholonomic mobile robot motion planning in state lattices," 2007.

[19] N. Ratliff, M. Zucker, J. A. D. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, May 2009.

[20] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Tech. Rep., 1998.

[21] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," 2011.

[22] L. E. Kavraki, P. Svestka, J. . Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[23] L. Janson and M. Pavone, "Fast marching trees: a fast marching sampling-based method for optimal motion planning in many dimensions - extended version," *CoRR*, vol. abs/1306.3532, 2013. [Online]. Available: http://arxiv.org/abs/1306.3532

[24] K. Yang, S. Moon, S. Yoo, J. Kang, N. Doh, H. Kim, and S. Joo, "Spline-based rrt path planner for non-holonomic robots," *Journal of Intelligent and Robotic Systems*, vol. 73, pp. 763–782, 01 2014.

[25] A. Piazzi, C. Guarino Lo Bianco, and M. Romano, "η^3-splines for the smooth path generation of wheeled mobile robots," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1089–1095, 2007.

[26] E. S. Low, P. Ong, and K. C. Cheah, "Solving the optimal path planning of a mobile robot using improved q-learning," *Robotics and Autonomous Systems*, vol. 115, pp. 143 – 161, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889018308285

[27] S. Khanmohammadi and A. Mahdizadeh, "Density avoided sampling: An intelligent sampling technique for rapidly-exploring random trees," in *2008 Eighth International Conference on Hybrid Intelligent Systems*, 2008, pp. 672–677.

[28] S. Karaman, M. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the rrt*," 06 2011, pp. 1478–1483.

[29] J. Nasir, F. Islam, U. E. Malik, Y. Ayaz, O. Hasan, M. A. Khan, and M. S. Muhammad, "Rrt*-smart: A rapid convergence implementation of rrt*," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.

[30] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1961–1976, 2015.

[31] J. Zhang, M. Wisse, and M. Bharatheesha, "Guided rrt: A greedy search strategy for kinodynamic motion planning," *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 480–485, 2014.

[32] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, 2000, pp. 995–1001 vol.2.

[33] M. Jordan and A. Perez, "Optimal bidirectional rapidly-exploring random trees," 2013.

[34] A. H. Qureshi and Y. Ayaz, "Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments," *Robotics and Autonomous Systems*, vol. 68, pp. 1 – 11, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889015000317

[35] ——, "Potential functions based sampling heuristic for optimal path planning," *CoRR*, vol. abs/1704.00264, 2017. [Online]. Available: http://arxiv.org/abs/1704.00264

[36] Z. Tahir, A. H. Qureshi, Y. Ayaz, and R. Nawaz, "Potentially guided bidirectionalized rrt* for fast optimal path planning in cluttered environments," *Robotics and Autonomous Systems*, vol. 108, pp. 13 – 27, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889017309387

[37] S. Karaman and E. Frazzoli, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 5041–5047.

[38] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "Lqr-rrt*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 2537–2542.

[39] D. J. Webb and J. van den Berg, "Kinodynamic rrt*: Optimal motion planning for systems with linear differential constraints," 2012.

[40] L. Palmieri, S. Koenig, and K. O. Arras, "Rrt-based nonholonomic motion planning using any-angle path biasing," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2775–2781.

[41] T. Biedl, M. Held, S. Huber, D. Kaaser, and P. Palfrader, "A simple algorithm for computing positively weighted straight skeletons of monotone polygons," *Information Processing Letters*, vol. 115, no. 2, pp. 243 – 247, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020019014001987

[42] V. Vonásek, M. Saska, K. Košnar, and L. Přeučil, "Global motion planning for modular robots with local motion primitives," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 2465–2470.

[43] E. Frazzoli, M. A. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 1, 2001, pp. 43–49 vol.1.

[44] B. Sakcak, L. Bascetta, G. Ferretti, and M. Prandini, "Sampling-based optimal kinodynamic planning with motion primitives," 2018.

[45] M. Vukosavljev, A. P. Schoellig, and M. E. Broucke, "Hierarchically consistent motion primitives for quadrotor coordination," 2019.

[46] A. Sakai, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques, "Pythonrobotics: a python code collection of robotics algorithms," 2018.