Kinodynamic Planning

Edward Schmerling and Marco Pavone

1 Synonyms

Trajectory planning, planning under differential constraints.

2 Definition

Kinodynamic planning concerns the task of driving a robot from an initial state to a goal region while avoiding obstacles and obeying kinematic and dynamic—in short, kinodynamic—constraints dictating the relationship between a robot's controls and its motion.

3 Overview

As a subfield of robot motion planning, kinodynamic planning is characterized by the explicit consideration of a robot's dynamics throughout the planning process. That is, kinodynamic planning algorithms output guidance trajectories that are not only collision-free with respect to a robot's environment but also feasible with respect to a representative model of the robot's continuous-time dynamics. This model may include kinematic constraints on the types of motion available to a robot

Edward Schmerling
Waymo LLC, 100 Mayfield Avenue, Mountain View, CA 94043, USA.
e-mail: schmrlng@waymo.com

Marco Pavone
Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA.
e-mail: pavone@stanford.edu

(e.g., a robot car cannot translate laterally against the direction of its wheels), as well as dynamics bounds on how those motions may be enacted over time (e.g., the momentum of a free-floating space probe prevents its finite force thrusters from stopping its motion instantaneously). The additional degree of modeling fidelity incorporated by kinodynamic planning algorithms makes them particularly suitable for mobile robots including wheeled vehicles at high speeds, multirotor and fixed wing aircraft, and even multi-link robots that must balance or swing to achieve their goal.

Like other motion planning disciplines, a core task of any kinodynamic planning algorithm is combinatorial search. The earliest works on kinodynamic planning [1, 2] considered robots obeying double integrator dynamics, modeling the motion of a point mass under controlled acceleration. The structure of these dynamics enabled the problem to be formulated as a dynamic programming-based search over a state transition graph defined on a fixed position-velocity grid. In addition to capturing which directions a robot should maneuver around obstacles, this planning representation also enabled decisions such as whether the robot should travel down a long straight corridor where it may build momentum or stick to shorter yet more tortuous paths with lower top speeds. This approach of using systematically applied control inputs to achieve a structured, exhaustive search over the state space, up to the lattice resolution, has been extended to other systems including robotic manipulators and ground vehicles.

Despite their successes, alternatives to lattice constructions have been sought due to their difficulty in generalizing to arbitrary state-space dynamics. Employing a similar approach of constructing global solution trajectories as concatenations of local connections between discrete states, modern sampling-based planners (cf. Sampling-based Roadmap Planners, Sampling-based Tree Planners) use a representative set of probing samples to more efficiently explore trajectories through the free state space. Work in this direction began with the application of the rapidly-exploring random tree (RRT) algorithm to kinodynamic planning [3]; much recent research has centered on sampling-based algorithms that guarantee solution near-optimality in addition to feasibility. Two major research themes that manifest in nearly all sampling-based kinodynamic planning algorithms are (1) how to efficiently evaluate distance between states under kinodynamic constraints, and (2) how to guide exploration towards unvisited sampled states from the graph of visited states. This latter challenge may be approached in an exact or approximate fashion; algorithms that solve exact two-point boundary value problems (*steering* problems) connecting start states to target states for graph expansion are known as *steering-based* planners while algorithms that expand the graph through, e.g., sampling the robot's control space are known as *forward-propagation-based* planners.

3.1 Problem Definition

Let $\mathcal{X} \subseteq \mathbb{R}^n$ and $\mathcal{U} \subseteq \mathbb{R}^m$ be the state space and control space, respectively, of a robotic system, and let us assume the dynamics of the robot are given by

$$\dot{x}(t) = f(x(t), u(t)), \quad x(t) \in \mathcal{X},\ u(t) \in \mathcal{U}. \tag{1}$$

In kinodynamic planning contexts there are typically two additional spaces associated with the robot's state space: its configuration space and the workspace. The robot's *configuration* q, derivable from the full dynamic state x, encodes a notion of position and contains all information necessary for checking collision with physical obstacles (possibly including self-collision) at any time instant. This entails computing the robot's occupancy in the workspace, i.e., the 2D or 3D Euclidean space in which the obstacles reside, and checking for intersections. In this way, we may project real-world collision avoidance constraints into an obstacle region $\mathcal{X}_{\text{obs}} \subset \mathcal{X}$ within the state space (which may also include other constraints defined in terms of the state or configuration variables, e.g., velocity limits or state restrictions to regions of stability, enabling robust tracking control of the planned trajectories). Then, the basic kinodynamic planning problem is similar to other robot motion planning formulations, notably with the addition of the dynamics constraint: given an initial state x_{init}, a goal region $\mathcal{X}_{\text{goal}}$ within the obstacle-free space $\mathcal{X}_{\text{free}} = \mathcal{X} \setminus \mathcal{X}_{\text{obs}}$, one seeks state and control trajectories $x : [0, T] \to \mathcal{X},\ u : [0, T] \to \mathcal{U}$ that are dynamically feasible in that they satisfy equation (1), collision free in that $x(t) \in \mathcal{X}_{\text{free}}$ for all $t \in [0, T]$, and satisfy the boundary conditions $x(0) = x_{\text{init}}$, $x(T) \in \mathcal{X}_{\text{goal}}$. As an alternative interpretation, at each point x, one may consider that the state transition function $f(x, u)$ provides a mapping from the control space to the set of all local directions of motion available to the robot. In this sense, incorporating robot dynamics into the planning problem may be thought of as adding additional local constraints to the global state constraints posed by obstacle avoidance considerations.

As in many motion planning contexts, both researchers and practitioners are often concerned not only with solution feasibility for kinodynamic planning problems, but solution quality as well. As early as the coining of the term *kinodynamic planning* [1], algorithms have been designed to find (near-)optimal plans with respect to a trajectory cost functional $c(x(\cdot), u(\cdot), T)$. This cost is typically chosen to reflect system design goals including minimum-time, minimum-energy (i.e., control effort), maximum safety, or some weighted combination thereof. Much literature from the past decade in particular has focused on optimal kinodynamic planning; key research findings will be surveyed in Section 4.

3.2 Context

Kinodynamic planning, in its full generality, subsumes the problem scopes of a number of other planning domains and runs parallel to still others. Here we briefly outline the key distinctions and relationships with these related fields to help clarify how and when kinodynamic planning techniques should be used.

Geometric Planning. (cf. Path/Motion Planning) Though certainly every physically-embodied robot obeys some notion of dynamics, in many cases these considerations may be abstracted away before the motion planning process. Indeed, this decoupling approach—in which the planning problem is decomposed in steps of computing a collision-free geometric path (neglecting the differential constraints), and then smoothing/reparameterizing the trajectory so that the robot can execute it—is often practical for slow-moving or otherwise highly maneuverable robots. Geometric planning is also advisable when the problem's search complexity arises simply from a high dimensional configuration space, as opposed to any dynamics considerations, e.g., when jointly planning the motions of many simple robots. In constrast, kinodynamic planning truly refers to solving the problem in one shot, avoiding any suboptimality or even solution infeasibility that might arise from not directly considering the system dynamics.

Kinematic Planning. Kinematic constraints restrict the local directions of motion available to a robot from a given configuration, i.e., they represent constraints on a robot's velocity in the configuration space. These constraints need not be associated with full continuous-time dynamics; indeed many simple car models including the Dubins and Reeds-Shepp cars and other mobile multi-link wheeled robots are often discussed within the kinodynamic planning literature but in actuality their path geometry may be planned purely in terms of respecting kinematic constraints. Problems containing only nonintegrable kinematic constraints are often referred to as nonholonomic planning problems in the motion planning literature [4].

The study of manifold-constrained planning (cf. Planning under Manifold Constraints) is similar to both kinematic and kinodynamic planning in that robots' local directions of motion are limited to constraint sets. A major distinction, however, is that for kinodynamic planning these motions are explicitly parameterized by the dynamics equation (1), while manifold constraints are often more naturally specified implicitly, e.g., through algebraic or differential algebraic equations.

Trajectory Optimization. When appropriately applied, kinodynamic planning and trajectory optimization algorithms (cf. Optimization-based Planners) serve complementary roles in a robot's control stack. While it is true that motion planning may be regarded as a special case of optimal control where the main distinguishing element is a (typically highly non-convex) collision-avoidance constraint on the robot's state trajectory, the global combinatorial search necessitated by this non-convex constraint is truly the hallmark of a motion planning problem. For sufficiently simple planning scenarios, e.g., if the planning horizon of a robot spans only a few upcoming obstacles, the combinatorial search over possible solution trajectories may be elided into a local optimization around an initial guess trajectory. The benefit of such an approach is that it can produce high-quality plans extremely quickly. In general, however, these trajectory optimization algorithms are only able to achieve local optima, and may not even reliably return feasible trajectories if their initialization is too poor. A more robust approach is to combine global motion planning with local trajectory optimization. That is, practitioners may opt to take the output of a kinodynamic planning algorithm as a near-optimal, dynamically feasible solution in the correct homotopy class (i.e., incorporating the correct combinatorial decisions in navigating the obstacle-free space), and further post-process it using local optimization methods before enacting the planned trajectory.

4 Key Research Findings

This section outlines a selection of recent theoretical and practical advances from the kinodynamic planning literature. In light of the context overviewed in Section 3.2, we restrict the present discussion to approaches that frame the problem of planning continuous dynamically feasible trajectories as the construction of and search over a motion graph with discrete states as nodes connected by trajectory segments as edges. In this way the trajectory planning problem may be reduced to a graph traversal problem; solutions connecting the initial state to the goal region are constructed by concatenating the local edges to form collision-free trajectories on a global scale. Though there do exist planning algorithms that account for the continuous-time aspect of trajectory planning directly, e.g., those based on variational methods, differential dynamic programming, or, more generally, full solutions of an appropriately defined Hamilton-Jacobi-Bellman partial differential equation [5], these approaches are more typically discussed in the context of trajectory optimization (cf. Optimization-based Planners).

4.1 Lattice-based Kinodynamic Motion Planning

The earliest, and arguably most natural, formulations of kinodynamic planning as a discrete search problem involve construction of a lattice of states repeated at regular intervals, connected locally by a fixed set of control trajectories [2, 6]. Lattice-based approaches are popular especially for mobile robots where the dynamics are translation-invariant in the spatial state dimensions, and thus regular lattices are practical to devise. The power of these methods comes from their decoupling of dynamics constraints, which depend only on the properties of the robotic system, with collision-avoidance constraints that depend on the particular environmental conditions a robot encounters during its operation. That is, the lattice and associated control trajectories used to traverse it encode a motion graph representation of the robot's dynamics. The graph, which may be computed offline, represents a discrete search space in which differential constraints have been abstracted away so that planning in the presence of obstacles may be reduced to graph search with only edge costs computed online. Modern implementations of lattice-based planning algorithms, e.g., [7], have proven the computational benefits of maintaining an implicit representation of the motion graph and using a heuristic search, e.g., A* or

D* (for planning applications where rapid replanning is desired), to decide which lattice edges to traverse.

4.2 Sampling-based Kinodynamic Motion Planning

Similar to lattice-based approaches, sampling-based motion planning (SBMP) algorithms build a representation of the free state space, and feasible motions within, as a collection of local connections between discrete state samples. These connections are added to a graph of possible motions (where nodes are states and edges are trajectories connecting them) if a collision checking routine verifies that the connection avoids obstacles. Unlike lattices which are typically computed offline and then applied to the planning problem at hand, sampling-based planning algorithms typically include graph construction, including computation of new edges, as part of their online operation. This enables the graph to be tailored to information discovered during the planning process, either explicitly or implicitly as a consequence of randomization in the way that the planning algorithm selects the state samples.[1] The explicit incorporation of graph construction also facilitates analysis of asymptotic completeness and optimality, the focus of much recent research activity in sampling-based kinodynamic planning.

Many sampling-based planning algorithms originally designed for geometric motion planning (cf. Sampling-based Roadmap Planners, Sampling-based Tree Planners) extend naturally to kinodynamic planning applications as well. The algorithm pseudocode may be similar, or even identical, but differences arise in the implementation of each planner subroutine, including methods for state sampling, near-neighbor computation, and edge construction (i.e., graph extension). Moreover, in the theoretical analysis of these algorithm adaptations, quantification of the reachable set of states from a graph node is critical—unlike geometric planning where local robot motions are unconstrained, kinodynamic planning algorithms must appropriately compensate for the restrictions brought on by dynamic constraints and widen their exploration strategies lest they insufficiently cover the search space.

4.2.1 Forward-propagation-based SBMP Algorithms

For the most general formulations of robot dynamics, solving an exact steering problem driving a robot from a start state to a target state may be essentially as difficult as solving the full motion planning problem itself, additional collision-avoidance constraints notwithstanding. Since the introduction of the rapidly-exploring random tree (RRT) [3] and expansive space tree (EST) [8] algorithms for kinodynamic planning in the early 2000s, researchers have grown a family of planning algorithms that employ only forward dynamics propagation to explore the free state space. In [9] the

[1] We note that lattice-based planning algorithms may also be categorized as deterministic sampling-based planners.

authors rigorously establish the probabilistic completeness of a variant of RRT under differential constraints, which expands the motion graph by propagating random control steps for random durations, subject to a minimal regularity assumption that the robot dynamics are Lipschitz continuous in both arguments. It was discovered by [10] that judicious pruning of the RRT throughout construction, resulting in the Stable-Sparse-RRT (SST) and SST* algorithms, achieves efficient asymptotic near-optimality and optimality respectively for Lipschitz-continuous cost functions under the same system assumptions. An alternative avenue to asymptotic optimality has been recently proposed as the AO-x meta-algorithm [11], which converts an optimal planning problem into a feasibility problem (to which any probabilistically complete kinodynamic planners, including RRT and EST, may be applied) by augmenting the state space with an addition dimension tracking accumulated cost on which upper bound constraints are considered.

4.2.2 Steering-based SBMP Algorithms

For robotic systems where efficient online steering subroutines exist, kinodynamic planning algorithms may take advantage of this domain knowledge to more explicitly guide graph construction. Many adaptations of algorithms originally proposed for geometric planning, where the steering subroutine is trivial, to planning under differential constraints fall into this category, including kinodynamic extensions of the asymptotically optimal RRT*, probabilistic roadmap (PRM*), and fast marching tree (FMT*) algorithms [12, 13, 14, 15]. A key parameter for these algorithms is the size of the local neighborhood in which steering connections between states should be considered—too small and the algorithms will not be considering enough connections to approach optimality, too large and far more steering connections will be computed than are necessary negatively impacting planner efficiency. Generalizing results for their geometric planning counterparts, those works derive the appropriate scaling for the connection radius r_n in terms of sample count n in order to guarantee asymptotic optimality, assuming uniform random sampling:

$$r_n \propto \left(\mathrm{Volume}(\mathcal{X}_{\mathrm{free}}) \left(\frac{\log n}{n} \right) \right)^{1/D}$$

where D is a constant, depending only on the robot dynamics and trajectory cost function, such that the r_n-bounded-cost forward/backward reachable sets from a state contain on average $\log(n)$ other states (though it is noted by, e.g., [14, 15] that this is not a sufficient condition in itself). In order to make this radius selection practical, however, there is a need for methods to identify near states, according to the steering metric, without computing all pairwise steering connections. Recent work [16] has investigated the application of k-d trees, spatial partitioning data structures designed for efficient near-neighbor lookups, for this purpose.

5 Examples of Application

Lattice-based planning methods have seen particular success in the field applied to trajectory planning problems for wheeled robots. In addition to driving prototypes for extraterrestrial research rovers [7], lattice-based planners were employed by many competitors in the DARPA Grand Challenge and DARPA Urban Challenge for autonomous vehicles in the mid-2000s, including the winning CMU [17] and runner-up Stanford [18] Urban Challenge teams.

Due to their relative computational expense, sampling-based algorithms have hitherto largely been limited to simulation-based planning applications, but there are some notable practical deployments. The MIT team in the DARPA Urban Challenge employed a variant of the kinodynamic RRT algorithm, propagating the closed-loop dynamics of the car with a stabilizing controller and employing situation-specific biased sampling for improved computational efficiency [19]. Real-time sampling-based motion planning has also been deployed to guide quadrotor flight in cluttered environments [20].

6 Future Directions for Research

Despite sustained research interest in their theoretical properties and a limited degree of practical deployment in mobile robot planning applications, the dominant research challenge in developing kinodynamic planning algorithms remains how to achieve truly real-time performance for general robotic systems in complex obstacle environments. Ongoing work in fusing motion planning and trajectory optimization (cf. Optimization-based Planners) targets that goal. Researchers have also begun to incorporate machine learning into kinodynamic planning algorithms in order to benefit from prior experience (beyond trajectory replanning in the immediate sense). Data-driven heuristics hold promise in accelerating sampling-based planning algorithms by biasing sampling distributions towards critical areas of the state space and providing faster learned approximations of algorithm subroutines such as near-neighbor computation and collision-checking that may be used to optimistically speed up exploration.

References

1. J. Canny, B. Donald, J. Reif, and P. Xavier, “On the complexity of kinodynamic planning,” in *IEEE Symp. on Foundations of Computer Science*, 1988.
2. B. Donald, P. Xavier, J. Canny, and J. Reif, “Kinodynamic motion planning,” *Journal of the Association for Computing Machinery*, vol. 40, no. 5, pp. 1048–1066, 1993.
3. S. M. LaValle and J. J. Kuffner, “Randomized kinodynamic planning,” *Int. Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
4. S. M. LaValle, *Planning Algorithms*. Cambridge Univ. Press, 2006.

5. R. Takei and R. Tsai, "Optimal trajectories of curvature constrained motion in the Hamilton-Jacobi formulation," *SIAM Journal on Scientific Computing*, vol. 54, no. 2-3, pp. 622–644, 2013.
6. B. Donald and P. Xavier, "Provably good approximation algorithms for optimal kinodynamic planning: Robots with decoupled dynamics bounds," *Algorithmica*, vol. 14, no. 6, pp. 443–479, 1995.
7. M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2009.
8. D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *Int. Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.
9. M. Kleinbort, K. Solovey, Z. Littlefield, K. E. Bekris, and D. Halperin, "Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, 2019.
10. Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *Int. Journal of Robotics Research*, vol. 35, no. 5, p. 528âĂŞ564, 2016.
11. K. Hauser and Y. Zhou, "Asymptotically optimal planning by feasible kinodynamic planning in a state-cost space," *IEEE Transactions on Robotics*, vol. 32, no. 6, 2016.
12. S. Karaman and E. Frazzoli, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *Proc. IEEE Conf. on Robotics and Automation*, 2013.
13. D. J. Webb and J. van den Berg, "Kinodynamic RRT*: Optimal motion planning for systems with linear differential constraints," in *Proc. IEEE Conf. on Robotics and Automation*, 2013.
14. E. Schmerling, L. Janson, and M. Pavone, "Optimal sampling-based motion planning under differential constraints: the driftless case," in *Proc. IEEE Conf. on Robotics and Automation*, 2015, Extended version available at http://arxiv.org/abs/1403.2483/.
15. ——, "Optimal sampling-based motion planning under differential constraints: the drift case with linear affine dynamics," in *Proc. IEEE Conf. on Decision and Control*, 2015.
16. V. Varricchio, B. Paden, D. Yershov, and E. Frazzoli, "Efficient nearest-neighbor search for dynamical systems with nonholonomic constraints," in *Workshop on Algorithmic Foundations of Robotics*, 2016.
17. C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y. Woo Seo, S. Singh, J. Snider, A. Stentz, W. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, "Autonomous driving in urban environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 8, 2008.
18. D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.
19. J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams, "A perception-driven autonomous urban vehicle," *Journal of Field Robotics*, vol. 25, no. 10, 2008.
20. R. Allen and M. Pavone, "A real-time framework for kinodynamic planning in dynamic environments with application to quadrotor obstacle avoidance," *Robotics and Autonomous Systems*, vol. 115, pp. 174–193, 2019.