Fast Sample-Based Planning for Dynamic Systems by Zero-Control Linearization-Based Steering

Timothy M. Caldwell and Nikolaus Correll

Abstract We propose linearizing about zero-control trajectories of the dynamics and cost instead of linearizing around just a single point for steering and distance computations in RRT-like motion planning for highly dynamic systems. Formulated as a time-varying Linear Quadratic Regulator optimal control problem, the proposed steering is designed to be efficient and numerically tractable. We describe the computational trade-offs that arise when compared to solving a conventional time-invariant LQR, and provide numerical results for an inverted pendulum with up to three links on a cart for a wide range of look-aheads (from hundredths of a second to a second). We find that planning with longer time horizons requires fewer total vertices, leading to much faster exploration of the search space than when using short look-aheads as are customary when linearizing around a single state. Albeit longer time horizons increase the likelihood of insertion failures due to collisions, we show that overall execution time significantly decreases in environments with moderate obstacle density.

1 Introduction

Planning for dynamic systems is trajectory exploration through nonconvex state spaces. These non-convexities often include configuration space obstacles and regions of inevitable collision [12]. While sample-based planning algorithms differ in many ways, they commonly conduct many short trajectory explorations to connect or nearly connect a large number of stochastically sampled states with the goal of connecting a start state to a final state. Earlier algorithms like RRT [12] guarantee under certain conditions [11] with probability one that if a connecting trajectory ex-

Timothy Caldwell
University of Colorado, Boulder, CO 80309, e-mail: timothy.caldwell@colorado.edu

Nikolaus Correll
University of Colorado, Boulder, CO 80309 e-mail: ncorrell@colorado.edu

ists, that the algorithm will find it. Newer algorithms like RRT*, PRM*, SST*, and FMT* [8–10, 13, 14] guarantee asymptotic convergence to the globally optimal connecting trajectory. For dynamic systems, planners are in need of computationally efficient methods to *compute distances*, e.g. for assessing nearest neighbors, and to *steer*, i.e. to extend to sampled states. Furthermore, each method must be able to handle numerical issues like sensitivities to initial conditions so that the steering trajectories can have a significant time horizon.

The methods proposed in this paper are complementary to [2, 3, 15, 16]. For planning dynamic systems, distance computation and steering are solutions to optimal control problems. They are often approximated through linearizing the nonlinear dynamics and solving linear quadratic regulation (LQR) or tracking (LQT) problems [2, 3, 15]. Each [2, 3, 15] linearize around a single state and so the system is time-invariant. In comparison, we linearize around the zero-control or "free" trajectory so that the linearization remains a good approximation for longer time horizons. Since our approximate system is time-varying we formulate the LQR and LQT so that through precomputation and caching they can be solved efficiently for many distance and steering executions from the same state.

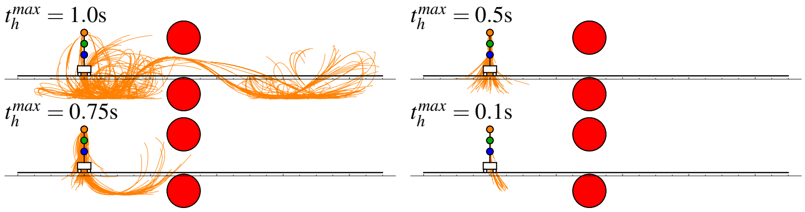

Fig. 1 Trajectories of a triple pendulum on a cart example for executions of RRT with differing max time horizons for a constant number of vertices (1000). Planning with longer time horizons covers significantly more search space for the same number of vertices, at the expense of increased number of insertion failures due to collisions. In this example, computing for $t_h^{max} = 1.0$s takes around three times longer than for $t_h^{max} = 0.1$s.

Both [2, 16] set up the linear quadratic optimal control problem as an optimal state transfer without a running state cost. As [5] indicates, the solution to such a problem is open-loop, which can be numerically intractable as the conditioning of the reachability Gramian becomes poor. Indeed, we show that the reachability Gramian's condition number approaches machine precision for the n-link pendulum on a cart in the examples, and as such, the methods in [2, 16] are infeasible. Instead, we propose implementing a feedback loop and performing the same state transfer procedure on the closed-loop system, which has a reachability Gramian with improved conditioning.

The state and trajectory solutions to the LQR problems are infeasible—i.e. trajectories for the linear system can not be trajectories of a nonlinear system. In order to steer, we implement the trajectory functional projection operator in [4, 6] to project approximate trajectories to feasible trajectories.

The proposed distance computation and steering methods in the paper are contributions to general planners. The methods are designed so that long time horizon exploring is viable. This is reflected in the examples where a small number of vertices of an RRT are needed to plan an n-link pendulum on a cart through a corridor of obstacles (Figure 1). The pendulum on a cart is under-actuated and unstable. A 2-link and 3-link pendulum on a cart are chaotic.

2 Review of Planning for Dynamic Systems

We briefly review planning for dynamic systems. The kinodynamic planning problem is to find a state and control trajectory (x,u) that connects a start state x_{start} to a goal state x_{goal} that satisfies the differential constraint given by the system dynamics

$$\dot{x}(t) = f(x(t),u(t)). \tag{1}$$

If a trajectory satisfies the system dynamics, then we say it is feasible. Sample-based planning algorithms find a path by generating a graph $G=(V,\mathscr{E})$ where the vertices V are explored states $x \in X$ and the edges $\mathscr{E}$ are state and control trajectories that connect two states in V—i.e. $(x,u;t_h) \in \mathscr{E}$ implies for time horizon $t_h > 0$, $x(0) \in V$, and $x(t_h) \in V$. Every edge must be dynamically feasible in that the state and control must satisfy Eq. 1. Additionally, the state and control must remain in the set of *allowable state and control* (X,U) throughout the full trajectory where the allowable state subset $X \subset \mathbb{R}^n$ and control subset $U \subset \mathbb{R}^m$. The set X is free of all obstacles.

Methods based on RRT build the graph through two functions: nearest neighbor and steering. The nearest neighbor calculation relies on the choice of distance function between a state $x_0 \in V$ and the sampled state $x_{samp} \in X$. Ideally, this distance is the cost J to transfer x_0 to x_{samp}. Then, the nearest state is the $x_0 \in V$ with least distance to x_{samp}. Steering computes a trajectory that satisfies Eq. 1 and transfers the system from an initial state $x_0 \in V$ to a neighborhood of a desired state $x_{samp} \in X$. Such a problem can be treated as an optimal control problem which finds a trajectory that minimizes some cost J. The cost for computing distance and steering can be the same. We consider the following cost function:

$$J(x,u;t_h) := \frac{1}{2}\int_0^{t_h} u^T(\tau)R(\tau)u(\tau)\,d\tau + \frac{1}{2}(x(t_h)-x_{samp})^T P_1 (x(t_h)-x_{samp}) \tag{2}$$

where $R = R > 0$ is symmetric positive definite and $P_1 = P_1^T \geq 0$ is symmetric positive semi-definite. The results in this paper can be extended to general costs as long as the cost can be locally approximated by a quadratic.

The problem of minimizing J constrained to the dynamics Eq. 1 is a nonlinear optimization problem for which numerical methods can be slow. For this reason, we approximate by linearizing about one of two candidates. The candidate used in [2,3,15,16] is simply a point x_0. The second candidate, which we propose, is the zero-control trajectory $x_{zero}(t)$ and is the solution to:

$$\dot{x}_{zero}(t) = f(x_{zero}(t), 0), \text{ s.t. } x_{zero}(0) = x_0. \tag{3}$$

Let $x_T(t) = x_0$ or $x_{zero}(t)$ depending on the candidate chosen. The linear terms are $A(t) = \frac{\partial}{\partial x(t)} f(x(t), u(t))|_{(x_T(t),0)}$ and $B(t) = \frac{\partial}{\partial u(t)} f(x(t), u(t))|_{(x_T(t),0)}$ with approximate state and control

$$\begin{gathered} \tilde{x}(t) = x_T(t) + z(t), \text{ and } \tilde{u}(t) = v(t), \\ \text{where } \dot{z} = A(t)z(t) + B(t)v(t), \text{ s.t. } z(0) = 0. \end{gathered} \tag{4}$$

Using the linearization, steering and distance computations are LQT problems. The next section address the LQT problem and formulates an efficient approach for many executions for the same vertex in a planning graph that is numerically tractable.

3 Affine LTV State Exploration

When a system is affine linear it is possible to find a subset of states in X that are reachable from any initial state for a bounded control effort. Later, we will extend the affine linear reachability results to nonlinear system path planning through a local linearization about the zero-control trajectory and a projection.

Suppose the system with state and control $(\tilde{x}, \tilde{u})$ evolves according to

$$\tilde{x}(t) := \tilde{x}_T(t) + z(t) \text{ and } \tilde{u}(t) := v(t)$$

where $\tilde{x}_T$ is some time-varying translation from the origin and z is the solution to the linear differential equation

$$\dot{z}(t) = A(t)z(t) + B(t)v(t), \text{ s.t. } z(0) = 0 \tag{5}$$

where $A(t) \in \mathbb{R}^{n \times n}$ and $B(t) \in \mathbb{R}^{n \times m}$ are piecewise continuous. Through the state-transition matrix (STM) $\Phi(t, \tau)$ of $A(t)$, the solution to the linear differential equation is

$$z(t) = \int_0^t \Phi(t, \tau) B(\tau) v(\tau) d\tau. \tag{6}$$

The STM is the solution to the matrix differential equation

$$\frac{\partial}{\partial \tau} \Phi(t, \tau) = -\Phi(t, \tau) A(\tau), \ \Phi(t, t) = Id_{n \times n} \tag{7}$$

where $Id_{n \times n}$ is the $n \times n$ identity matrix. We wish to find the states $\tilde{x}' \in X$ such that there exists a control v constrained to some set $\mathscr{V} \subset \mathscr{U}$ which transfers $\tilde{x}(0) = \tilde{x}_T(0)$ to $\tilde{x}'$ in t_h time. Define the set reachable at t_h with control constrained to $\mathscr{V}$ as

$$\tilde{X}_{\mathscr{V}} := \{\tilde{x}' \in \mathbb{R}^n | \exists v \in \mathscr{V} \text{ where } \tilde{x}' = \tilde{x}_T(t) + \int_0^{t_h} \Phi(t, \tau) B(\tau) v(\tau) d\tau\}.$$

Consider the control constrained by the control energy

$$\|v\|_R := \frac{1}{2}\int_0^{t_h} v(\tau)^T R(\tau) v(\tau)\, d\tau \tag{8}$$

where $R(\cdot)$ is a piecewise continuous matrix with $R(\tau) = R(\tau)^T > 0$ symmetric positive-definite. The control set with bounded control energy is

$$\mathscr{V}_{\delta v} := \{v \in \mathscr{U} | \frac{1}{2}\int_0^{t_h} v(\tau)^T R(\tau) v(\tau)\, d\tau < \delta v\} \tag{9}$$

where $\delta v > 0$ and $t_h > 0$.

Next, we consider an open-loop and a closed-loop form for the control and analyze their respective bounded reachable sets.

3.1 Open-Loop Exact Linear Shooting

Suppose v has the form

$$v(t) = -R^{-1}(t)B(t)^T \Phi(t_h,t)^T \eta \tag{10}$$

where $\eta \in \mathbb{R}^n$. The significance of this control form is that it is the *minimal-energy control* with control energy Eq. 8 (see [7] Theorem 11.4). This control is open-loop because of its independence on z.

By setting η, the control Eq. 10 transfers $\tilde{x}(0)$ to some $\tilde{x}' \in \mathbb{R}^n$ through Eq. 5. This procedure is shooting. The δv bounded control reachable set $\tilde{X}_{\mathscr{V}_{\delta v}}$ are the points $\tilde{x}' \in \mathbb{R}^n$ for which there exists an η for which both $v \in \mathscr{V}_{\delta v}$ and v transfers $\tilde{x}(0)$ to $\tilde{x}'$. This set $\tilde{X}_{\mathscr{V}_{\delta v}}$ is given in the following Lemma:

Lemma 1. *Supposing $A(\cdot)$ and $B(\cdot)$ are so that Eq. 5 is controllable on $[0,t_h]$ and that v has form Eq. 10, then*

$$\tilde{X}_{\mathscr{V}_{\delta v}} = \{\tilde{x}' \in \mathbb{R}^n | \tfrac{1}{2}(\tilde{x}' - \tilde{x}_T(t_h))^T W_0(t_h)^{-1}(\tilde{x}' - \tilde{x}_T(t_h)) < \delta v\} \tag{11}$$

where the matrix $W_0(t)$, $t \in (0,t_h]$ is symmetric positive-definite and W_0 is the solution to:

$$\dot{W}_0(t) = A(t)W_0(t) + W_0(t)A(t)^T + B(t)R(t)^{-1}B(t)^T \text{ s.t. } W_0(0) = 0.$$

Proof. The symmetric positive-definiteness of $W_0(\cdot)$ follows from $(A(\cdot),B(\cdot))$ controllable and $R(\cdot)$ symmetric positive-definite. Its existence is guaranteed since it is the solution to a linear differential equation. The integral form of $W_0(t)$ is:

$$W_0(t) = \int_0^t \Phi(t,\tau)B(\tau)R(\tau)^{-1}B(\tau)^T \Phi(t,\tau)^T d\tau.$$

An η is allowable if $v \in \mathscr{V}_{\delta v}$ (see Eq. 9). That is, η is allowable if

$$\begin{aligned}\frac{1}{2}\int_0^{t_h} v^T(\tau)R(\tau)v(\tau)d\tau &= \frac{1}{2}\eta^T \int_0^{t_h} \Phi(t_h,\tau)B(\tau)R^{-1}(\tau)B(t)^T \Phi(t_h,\tau)^T d\tau\eta \\ &= \tfrac{1}{2}\eta^T W_0(t_h)\eta < \delta v.\end{aligned} \tag{12}$$

Through the form of v, the solution to z for some $\eta \in \mathbb{R}^n$ is

$$z(t) = -\int_0^t \Phi(t,\tau)B(\tau)R(\tau)^{-1}B(\tau)^T\Phi(t_h,\tau)^T d\tau\eta = -W_0(t)\Phi(t_h,t)^T\eta.$$

Since $W_0(t_h)$ is positive-definite, it is invertible and the η which transfers $z(0) = 0$ to a desired z' is $\eta = -W_0(t_h)^{-1}z'$. Plugging η in Eq. 12 we arrive at the set of reachable z'. The reachable $\tilde{x}'$ are $\tilde{x}' = \tilde{x}'_T(t_h) + z'$ which is the set $\tilde{X}_{\mathscr{V}_{\delta v}}$. □

The set of reachable states $\tilde{X}_{\mathscr{V}_{\delta v}}$ is an ellipsoid centered around $\tilde{x}_T(t_h)$, as seen in Figure 2 for the double integrator system. A planning algorithm can leverage Lemma 1 to choose a reachable state and compute the state and control trajectories $(\tilde{x},\tilde{u})$ that transfer the linear system to that desired state. This exact steering procedure for affine linear systems is shown in Algorithm 1. The procedure relies on a well conditioned $W_0(\cdot)$, especially at t_h. The time-varying matrix W_0 is the reachability Gramian weighted by R^{-1}. It is guaranteed to be invertible when the system $(A(\cdot),B(\cdot))$ is controllable, but numerical inversion requires a well conditioned matrix, which is not the case for the n-link inverted pendulum on a cart analyzed in the examples section. The condition number for time $t_h = 1.0$, $\kappa(W_0(1.0))$, for 1, 2, and 3-link pendulum on the cart at the unstable equilibrium is:

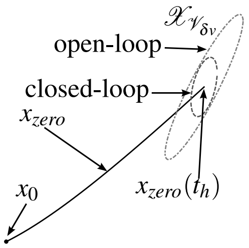

Fig. 2 Reachable set for open-loop and closed-loop double integrator for $x_0 = [0.5,1]^T$, $\delta_v = 0.2$, $t_h = 1.0$.

$$\begin{aligned} 1-link &: \kappa(W_0(1.0)) = 1.32\times 10^6 \\ 2-link &: \kappa(W_0(1.0)) = 1.08\times 10^8 \\ 3-link &: \kappa(W_0(1.0)) = 3.33\times 10^{15}.\end{aligned} \tag{13}$$

Algorithm 1 (Open-Loop Exact Linear Steering)
For $\tilde{x}' \in \tilde{X}_{\mathscr{V}_{\delta v}}$:

```
1. $\eta \leftarrow -W_0(t_h)^{-1}(\tilde{x}' - \tilde{x}_T(t_h))$
2. $\tilde{x}(t) \leftarrow \tilde{x}_T(t) - W_0(t)\Phi(t_h,t)^T\eta$, and
3. $\tilde{u}(t) \leftarrow -R(t)^{-1}B(t)^T\Phi(t_h,t)^T\eta$
```

For the 3-link pendulum, the condition number approaches machine precision on many computing devices. When the conditioning is sufficiently poor, the numerical error between $\tilde{x}'$ and the computed $\tilde{x}(t)$ through Algorithm 1 invalidates the procedure. For unstable systems, the conditioning can be improved through a closed-loop control form with stabilizing feedback. The goal is to design a feedback so that Algorithm 1 becomes a sort of stable shooting algorithm.

3.2 Closed-Loop Exact Linear Steering

In order to make Algorithm 1 numerically tractable, we consider a closed-loop system with a better conditioned reachability Gramian than the open-loop system. The closed-loop linear system is (A_K, B) where $A_K(t) := A(t) - B(t)K(t)$ with time-varying feedback gain $K(t) \in \mathbb{R}^{m\times n}$. For now, we assume K is given. It can be computed as the optimal feedback gain of a LQR problem. With a properly designed K, the closed-loop system (A_K, B) should have better numerical properties—e.g. better conditioning of relevant matrices—than the open-loop system.

Let $\Phi_K(\cdot,\cdot)$ be the state-transition matrix corresponding to A_K. The closed-loop control form is

$$v(t) = -K(t)z(t) - R^{-1}(t)B(t)^T\Phi_K(t_h,t)^T\eta. \tag{14}$$

The closed-loop linear system dynamics are

$$\dot{z}(t) = A_K(t)z(t) - B(t)R(t)^{-1}B(t)^T\Phi_K(t_h,t)^T\eta,\ \text{s.t. } z(0) = 0$$

with solution

$$\begin{aligned} z(t) &= -\int_0^t \Phi_K(t,\tau)B(\tau)R(\tau)^{-1}B(\tau)^T\Phi_K(t_h,\tau)^T d\tau\eta \\ &= -W_K(t)\Phi_K(t_h,t)^T\eta \end{aligned}$$

where W_K is the reachability Gramian corresponding to A_K, weighted by R^{-1}, and has differential form:

$$\dot{W}_K(t) = A_K(t)W_K(t) + W_K(t)A_K(t)^T + B(t)R(t)^{-1}B(t)^T \text{ s.t. } W_K(0) = 0. \tag{15}$$

As with Lemma 1, we wish to find the reachable subset of $\mathbb{R}^n$ in t_h time with bounded control energy, except for the closed-loop form:

Lemma 2. *Supposing $A(\cdot)$ and $B(\cdot)$ are so that Eq. 5 is controllable on $[0,t_h]$ and that v has form Eq. 14, then*

$$\tilde{X}_{\mathcal{V}_{\delta v}} = \{\tilde{x}' \in \mathbb{R}^n | (\tilde{x}' - \tilde{x}_T(t_h))^T S_K(t_h)^{-1}(\tilde{x}' - \tilde{x}_T(t_h)) < \delta v\} \tag{16}$$

where the matrix $S_K(t)$, $t \in (0,t_h]$ is symmetric positive-definite and is the solution to: (omitting $(\cdot)$)

$$\dot{S}_K = AS_K + S_KA^T + (KW_K - R^{-1}B^T)^T R(KW_K - R^{-1}B^T) \text{ s.t. } S_K(0) = 0. \tag{17}$$

The proof of Lemma 2 is similar to that of Lemma 1 and for the sake of brevity is omitted.

Similar to Lemma 1, Lemma 2 finds that the reachable states with the closed-loop control form an ellipsoid centered at $\tilde{x}_T(t_h)$, except where the ellipsoid's axes are parameterized by $S_K(t_h)^{-1}$ as opposed to $W_0(t_h)^{-1}$.

The analog to Algorithm 1 is

```
Algorithm 2 (Closed-Loop Exact Linear Steering)
For $\tilde{x}' \in \tilde{X}_{\mathcal{V}_{\delta v}}$ and feedback gain $K$:

1. $\eta \leftarrow -W_K(t_h)^{-1}(\tilde{x}' - \tilde{x}_T(tf))$
2. $\tilde{x}(t) \leftarrow \tilde{x}_T(t) - W_K(t)\Phi_K(t_h,t)^T\eta$, and
3. $\tilde{u}(t) \leftarrow -K(t)z(t) - R(t)^{-1}B(t)^T\Phi(t_h,t)^T\eta$
```

The conditioning of W_K and S_K for the n-link cart pendulum is a significant improvement over W_0 (see Eq. 13):

$$\begin{array}{lll} 1-link: & \kappa(W_K(1.0)) = 5.84 & \kappa(S_K(1.0)) = 3.80 \times 10^4 \\ 2-link: & \kappa(W_K(1.0)) = 5.49 \times 10^2 & \kappa(S_K(1.0)) = 2.52 \times 10^6 \\ 3-link: & \kappa(W_K(1.0)) = 1.03 \times 10^5 & \kappa(S_K(1.0)) = 2.57 \times 10^6. \end{array}$$

The closed loop form should be used when the open loop reachability Gramian is poorly conditioned.

3.3 Inexact Linear Steering

The reachability results provide the minimal energy control to a reachable set. When the desired state is not within the reachable set a new objective can be considered which weights the importance of tracking the desired state. Consider the cost function

$$J(\tilde{x},\tilde{u};t_h) := \frac{1}{2}\int_0^{t_h} \tilde{u}^T(\tau)R(\tau)\tilde{u}(\tau)\,d\tau + \frac{1}{2}(\tilde{x}(t_h) - \tilde{x}_{des})^T P_1(\tilde{x}(t_h) - \tilde{x}_{des}) \tag{18}$$

where $P_1 = P_1 \geq 0$ is symmetric positive semi-definite. Parameterized by (z,v), the cost is

$$\begin{aligned} J(z,v;t_h) = &\frac{1}{2}\int_0^{t_h} v^T(\tau)R(\tau)v(\tau)\,d\tau \\ &+\tfrac{1}{2}(\tilde{x}_T(t_h) + z(t_h) - \tilde{x}_{des})^T P_1(\tilde{x}_T(t_h) + z(t_h) - \tilde{x}_{des})\Big]. \end{aligned} \tag{19}$$

The problem is to find a state and control trajectory that satisfies the linear dynamics Eq. 5. That is, we wish to solve the problem

Problem 1. Solve

$$\min_{z,v;t_h} J(z,v;t_h)$$
$$\text{s.t. } \dot{z}(t) = A(t)z(t) + B(t)v(t),\, z(0) = 0.$$

For a fixed t_h, this problem is a linear quadratic tracking LQT problem [1], the solution of which is well known. The optimal control is

$$v^\star(t) = -K^\star(t)z^\star(t) - R(t)^{-1}B(t)^T \Phi_K(t,t_h)P_1(\tilde{x}_T(t_h) - \tilde{x}_{des}) \tag{20}$$

with optimal feedback gain $K^\star(t)$

$$K^\star(t) = R(t)^{-1}B(t)^T P(t) \tag{21}$$

where $P(t)$ is the solution to the Riccati equation

$$-\dot{P}(t) = A(t)^T P(t) + P(t)A(t) - P(t)B(t)R(t)^{-1}B(t)^T P(t) \text{ s.t. } P(t_h) = P_1. \tag{22}$$

Notice that $v^\star$ has the closed-loop form Eq. 14 except for a specific feedback gain $K(t) = K^\star(t)$ and where $\eta = P_1(\tilde{x}_T(t_h) - \tilde{x}_{des})$. The gain K in Eq. 14 can be chosen as the optimal feedback gain from this LQT problem. The $K(t)$ would depend on the choice of $\tilde{x}_{des}$ as well as P_1. If $P_1 = 0$—i.e. the cost function reduces to just the control energy—then $K^\star \equiv 0$—i.e. there is no feedback—and the optimal control is the open-loop control Eq. 10. As such, the procedure Algorithm 1 does indeed compute the minimal control energy trajectory that transfers the system to a reachable state.

The following algorithm computes the optimal cost $\tilde{J}^\star(t_h) := J(\tilde{x}^\star, \tilde{u}^\star, t_h)$, and trajectory $(\tilde{x}^\star, \tilde{u}^\star)$ for fixed t_h:

Algorithm 3 (Fixed t_h Inexact Linear Steering)
For $\tilde{x}_{des} \in X$ and $t_h > 0$:

```
1. $P(t) \leftarrow$ solve Riccati equation Eq. 22
2. $K^\star(t) \leftarrow R(t)^{-1}B(t)^T P(t)$
3. $z^\star(t) \leftarrow$ solve $\dot{z}^\star(t) = A_{K^\star}(t) - B(t)R(t)^{-1}B(t)^T \Phi_K(t,t_h)P_1(\tilde{x}_T(t_h) - \tilde{x}_{des})$
4. $\tilde{x}^\star(t) \leftarrow \tilde{x}_T(t) + z^\star(t)$, and
5. $\tilde{u}^\star(t) \leftarrow -K^\star(t)z^\star(t) - R(t)^{-1}B(t)^T \Phi_K(t,t_h)P_1(\tilde{x}_T(t_h) - \tilde{x}_{des})$
6. $\tilde{J}^\star(t_h) \leftarrow$ solve cost function Eq. 19
7. Return $\tilde{J}^\star(t_h)$ and $(\tilde{x}^\star, \tilde{u}^\star)$
```

In comparison to Algorithms 1 and 2, this algorithm computes a trajectory that tracks any state in X as opposed to just the states in a reachable set. However, the trajectory does not transfer the state to $\tilde{x}_{des}$ and as such is an inexact steering.

Every execution of Algorithm 3 solves the Riccati equation, the linear state equation, and integrates the running cost, which is computationally expensive. Next, we propose an efficient inexact linear steering algorithm that relies on precomputation and caching so that the algorithm does not rely on solving any differential equations or integrations. The precomputations are also invariant on $\tilde{x}_{des}$.

3.4 Efficient Inexact Linear Steering

For an affine linear system—i.e. for a set x_T, A and B—Algorithm 3 must be executed anew for distinct t_h and $\tilde{x}_{des}$. As we show here, inexact linear steering gains efficiency through precomputation and caching to remove redundancies from multiple algorithm execution.

As previously noted, the closed-loop control Eq. 14 has the same form as the solution to the LQT problem with optimal control Eq. 20. The efficiency is gained through precomputing and fixing a feedback gain $K(t)$ and solving for η as was done in Section 3.2. In other words, the problem is minimize the cost Eq. 19 constrained to the closed-loop control Eq. 14:

Problem 2. With fixed $K(t)$, solve

$$\begin{aligned} &\min\nolimits_{\eta;t_h} J(z,v,t_h) \\ \text{s.t.}\;\; &\dot{z}(t) = A_K(t)z(t) - B(t)v(t)R^{-1}(t)B(t)^T\Phi_K(t_h,t)^T\eta,\, z(0)=0 \\ &v(t) = -K(t)z(t) - R^{-1}(t)B(t)^T\Phi_K(t_h,t)^T\eta. \end{aligned}$$

The solution to Problem 2 is not the same as the solution to Problem 1 unless the feedback gain $K(t)$ happens to be the optimal $K^\star(t)$.

The solution to Problem 2 is given in the following Lemma:

Lemma 3. *For fixed $t_h > 0$ and K and supposing $A(\cdot)$ and $B(\cdot)$ are so that Eq. 5 is controllable on $[0,t_h]$, the solution to Problem 2 is*

$$\eta^\star = P_{t_h}(\tilde{x}_T(t_h) - \tilde{x}_{des}) \tag{23}$$

where

$$P_{t_h} = (W_K(t_h)P_1W_K(t_h) + S_K(t_h))^{-1}W_K(t_h)P_1.$$

Additionally,

$$\begin{aligned} z^\star(t) &= -W_K(t)\Phi_K(t_h,t)^T\eta^\star, \\ v^\star(t) &= [K(t)W_K(t) - R(t)^{-1}B(t)^T]\Phi_K(t_h,t)^T\eta^\star \end{aligned} \tag{24}$$

and

$$\begin{aligned} J(z^\star,v^\star;t_h) = &\tfrac{1}{2}(\eta^\star)^TS_K(t_h)\eta^\star \\ &+\tfrac{1}{2}(\tilde{x}_T(t_h) - W_K(t_h)\eta^\star - \tilde{x}_{des})^TP_1(\tilde{x}_T(t_h) - W_K(t_h)\eta^\star - \tilde{x}_{des}). \end{aligned} \tag{25}$$

Proof. The integral form of z is Eq. 6:

$$\begin{aligned} z(t) &= \textstyle\int_0^t \Phi_K(t,\tau)B(\tau)v(\tau)d\tau = -\int_0^t \Phi_K(t,\tau)B(\tau)R(\tau)^{-1}B(\tau)^T\Phi(t_h,\tau)^Td\tau\eta \\ &= -W_K(t)\Phi_K(t_h,t)^T\eta. \end{aligned}$$

Plugging $z(t)$ into $v(t)$,

$$v(t) = (K(t)W_K(t) - R(t)^{-1}B(t)^T)\Phi_K(t_h,t)^T\eta.$$

As such, $z(t)$ and $v(t)$ depend linearly on η. The control energy is

$$\begin{aligned}&\tfrac{1}{2}\int_0^{t_h} v^T(\tau)R(\tau)v(\tau)\,d\tau\\ &\quad= \tfrac{1}{2}\eta^T \int_0^{t_h} \Phi_K(t_h,\tau)(K(\tau)W_K(\tau)-R(\tau)^{-1}B(\tau)^T)^T\\ &\qquad\cdot R(\tau)(K(\tau)W_K(\tau)-R(\tau)^{-1}B(\tau)^T)\Phi_K(t_h,\tau)d\tau\eta\\ &\quad= \tfrac{1}{2}\eta^T S_K(t_h)\eta.\end{aligned}$$

The cost is thus

$$J = \tfrac{1}{2}(\eta)^T S_K(t_h)\eta + \tfrac{1}{2}(\tilde{x}_T(t_h) - W_K(t_h)\eta - \tilde{x}_{des})^T P_1 (\tilde{x}_T(t_h) - W_K(t_h)\eta - \tilde{x}_{des})$$

with optimal $\eta^\star$ where $\frac{\partial}{\partial\eta}J|_{\eta\to\eta^\star} = 0$:

$$\tfrac{\partial}{\partial\eta}J|_{\eta\to\eta^\star} = [(S_K(t_h) + W_K(t_h)P_1W_K(t_h))\eta - W_K(t_h)P_1(\tilde{x}_T(t_h) - \tilde{x}_{des})]_{\eta\to\eta^\star} = 0.$$

Since $S_K(t_h)$ and $W_K(t_h)$ are positive definite through the controllability assumption, $(S_K(t_h) + W_K(t_h)P_1W_K(t_h))$ is invertible. Solving for $\eta^\star$ results in Eq. 23. □

The lemma instructs how to do efficient fixed t_h inexact steering:

```
Algorithm 4 (Efficient Fixed $t_h$ Inexact Linear Steering)
For $\tilde{x}_{des} \in X$ and $t_h > 0$:

1. $P_{t_h} \leftarrow (W_K(t_h)P_1W_K(t_h) + S_K(t_h))^{-1}W_K(t_h)P_1$
2. $\eta^\star \leftarrow P_{t_h}(\tilde{x}_T(t_h) - \tilde{x}_{des})$
3. $\tilde{x}^\star(t) \leftarrow \tilde{x}_T(t) - W_K(t)\Phi_K(t_h,t)^T\eta^\star$
4. $\tilde{u}^\star(t) \leftarrow [K(t)W_K(t) - R(t)^{-1}B(t)^T]\Phi_K(t_h,t)^T\eta^\star$
5. $\tilde{J}^\star(t_h) \leftarrow$ solve Eq. 25
6. Return $\tilde{J}^\star(t_h)$ and $(\tilde{x}^\star, \tilde{u}^\star)$
```

The algorithm is efficient compared to Algorithm 3 because it does not rely on solving any differential equations assuming certain functions have been precomputed. The functions to be precomputed and saved in memory are $K(t)$, $W_K(t)$, $S_K(t)$ and $\Phi_K(t_h^{max},t)$ for a specified long time horizon $t \in [0,t_h^{max}]$, $t_h^{max} > 0$. As such, Algorithm 4 relies solely on matrix manipulations to return a fixed t_h inexact linear steering trajectory assuming $t_h < t_h^{max}$. Note that $\Phi_K(t_h,t) = \Phi_K(t_h^{max},t_h)^{-1}\Phi_K(t_h^{max},t)$. Also, $\Phi(t_h,t_h) = Id_{n\times n}$ and so $\tilde{x}^\star(t_h)$, $\tilde{u}^\star(t_h)$ and $\tilde{J}^\star(t_h)$ do not rely on precomputing Φ.

As of yet, the feedback gain is assumed to have been chosen in some way. A reasonable choice is the optimal feedback gain to the LQT problem, Problem 1, for the max time horizon $t_h = t_h^{max}$. It is the case that the solution to Problem 2 is equivalent to the solution to Problem 1 when $P_1 = P_{t_h}$. In other words, if $P_1 = P_{t_h}$, then $K = K^\star$, which will be the case at least at t_h^{max} for this choice of K.

4 Extending to Nonlinear Systems

We extend the affine linear steering results in Section 3 to nonlinear dynamics $\dot{x}(t) = f(x(t), u(t))$ by projecting inexact and exact linear steering trajectories to feasible trajectories. For an initial state x_0 that may be a vertex of a graph generated by a planner, the dynamics are linearized about the zero-control trajectory for $t_h > 0$ time. The zero-control trajectory x_{zero} is given by Eq. 2 and the linearization is in Eq. 4.

A planner can pick which linear steering algorithm, Algorithms 1-4, to approximately transfer the system to a desired state x_{des}. The resulting approximate trajectories $(\tilde{x}, \tilde{u})$ are not feasible—i.e. $(\tilde{x}, \tilde{u})$ do not satisfy the system dynamics—unless the dynamics are linear.

When the approximate trajectories are within a region where the linearization is reasonable, there are nearby feasible trajectories. The trajectory functional projection operator $\mathscr{P}$ proposed in [6] maps $(\tilde{x}, \tilde{u})$ to a feasible trajectory.

$$\begin{bmatrix} x \\ u \end{bmatrix} = \mathscr{P}\left(\begin{bmatrix} \tilde{x} \\ \tilde{u} \end{bmatrix}\right) := \begin{cases} \dot{x} = f(x,u) \\ u = \tilde{u} - K(x - \tilde{x}). \end{cases} \tag{26}$$

The projection is a feedback loop with gain K, reference signal $\tilde{x}$, and feedforward term $\tilde{u}$. The feedback gain may be chosen as the optimal feedback of an LQR or LQT problem.

Inexact steering transfers a vertex $x_0 \in V$ to a state near a desired state x_{des}. Suppose $K(t)$, $W_K(t)$, $S_K(t)$ and $\Phi_K(t_h^{max}, t)$ have been computed for the vertex x_0. Then, the efficient inexact steering trajectory is computed by projecting the approximate trajectories given by efficient inexact linear steering, Algorithm 4.

Algorithm 5 (Efficient Inexact Steering)
For $x_0 \in X$, $x_{des} \in X$, and $t_h \in (0, t_h^{max}]$:

```
1. $(\tilde{x}^\star, \tilde{u}^\star, \tilde{J}^\star(t_h)) \leftarrow$ Algorithm 4 for $t_h$
2. $(x, u) \leftarrow \mathscr{P}(\tilde{x}^\star, \tilde{u}^\star)$
3. return $(x, u, \tilde{x}, \tilde{u}, \tilde{J}^\star(t_h), t_h)$
```

The nearest neighbor computation chooses the vertex $x_0 \in V$ of the graph $G = (V, \mathscr{E})$ nearest a sampled state $x_{samp} \in X$ where nearest is specified by a distance function $d : X \times X \to \mathbb{R}$. The distance between two states x_0 and x_{samp}, $d(x_0, x_{samp})$, is the optimal cost to transfer the system from x_{samp} to or nearby x_{samp}. Choose the cost J as the cost for inexact linear steering, Eq. 18, where the desired state is x_{samp}:

$$J(x, u; t_h) := \frac{1}{2}\int_0^{t_h} u^T(\tau) R(\tau) u(\tau)\, d\tau + \frac{1}{2}(x(t_h) - x_{samp})^T P_1 (x(t_h) - x_{samp}). \tag{27}$$

Ideally, the distance is $d(x_0, x_{samp}) = J(x^\star, u, t_h)$ for the optimal feasible trajectory.Since such a nonlinear optimal control problem is slow to compute, we instead approximate it as $d(x_0, x_{samp}) = J(\tilde{x}^\star, \tilde{u}^\star, t_h)$, which is the distance given by inexact linear steering (see Section 3.4 and Algorithm 4)

5 Examples and Algorithm

Using the steering and nearest neighbor methods proposed in the paper we construct an RRT to plan a trajectory that transfers an n-link pendulum on a cart through a corridor of obstacles to a goal state. The state is composed of the pendulum angles θ_i, their angular velocities $\dot{\theta}_i$, the cart position p and the cart velocity $\dot{p}$. The control is the force applied to the cart, accelerating it forward or backward. The pendulums are in inverted equilibrium for both the start and goal states, with cart position at $p_{start} = 0$m as seen in Figure 1. The system must find a path to connect the start and goal at $p_{goal} = 6$ without the pendulum heads colliding with the obstacles. The obstacles are circles of radius 0.6 at $(3, -0.85)$ and $(3, 0.85)$. We consider 1, 2, and 3-link pendulums, with n=4, 6, and n=8 states, respectively. Each pendulum head has mass 0.1 kg and the pendulum lengths are assumed massless. All pendulums have total length 1 m, that is the 2-link pendulum consists of two 0.5 m links and so on. The cart has mass 1 kg.

We execute an RRT-like algorithm with inexact steering. The matrices R and P_1 in the cost J, Eqs. 18 and 27, are $R = [0.025]$ and $P_1 = Id_n$ the $n \times n$ identity matrix.

An RRT formulation with efficient inexact steering through precomputing follows. At each iteration, a sample state $x \in X$ and a sample time $t_{h,samp} \in (0, t_h^{max}]$ over a uniform distribution is taken. The vertex that is the nearest neighbor is computed and the system is steered using the efficient inexact steering algorithm. The resulting approximate trajectory is projected to the set of feasible trajectories. An insertion failure occurs if the projected trajectory collides with a boundary of X such as an obstacle. Otherwise, the precomputations for the new vertex are made and the new vertex is appended to the graph. As we focus exclusively on distance calculation and steering, the methods in this paper can also be implemented for more complex planners like RRT*.

Algorithm 6 (RRT with Efficient Inexact Steering) *for $x_{start} \in X$:*

```
Precompute $x_{zero}, K, W_K, S_K, \Phi_K$ for $x_{start}$ over $t \in [0, t_h^{max}]$. (Eqs. 2, 21, 15, 17, 7)
$N_{init} = \{x_{start}, x_{zero}, K, W_K, S_K, \Phi_K\}$
$V \leftarrow \{N_{init}\}$; $\mathscr{E} \leftarrow \emptyset$
while $x_{goal}$ not found:
  $x_{samp} \leftarrow sample(x)$, $t_{h,samp} \leftarrow sample((0, t_h^{max}])$
  $N_{near} \leftarrow nearestneighbor(V, x_{samp})$.
  $(x_{new}, u_{new}, \tilde{x}_{new}, \tilde{u}_{new}, \tilde{J}^\star, t_{h,samp}) \leftarrow$ Alg. 5 from $N_{near}$
  $(x, u) \leftarrow \mathscr{P}(\tilde{x}, \tilde{u})$ (Eq.26)
  if $(x_{new}(t), u_{new}(t)) \in (X, U)$ for all $t \in [0, t_{h,samp}]$:
      Precompute $x_{zero}, K, W_K, S_K, \Phi_K$ for $x_{new}(t_{h,samp})$
          over $t \in [0, t_h^{max}]$.
      $N_{new} \leftarrow \{x_{new}(t_{h,samp}), x_{zero}, K, W_K, S_K, \Phi_K\}$
      $V \leftarrow V \cup \{N_{new}\}$; $\mathscr{E} \leftarrow \mathscr{E} \cup \{(x_{new}, u_{new}; t_{h,samp})\}$
return $G = (V, \mathscr{E})$
```

Qualitative results of applying Algorithm 6 are shown in Figure 1, which shows the explored trajectories for the top pendulum head for a triple pendulum on a cart. We conducted similar simulations for the one and two-link pendulums, and performed systematic experiments for the one-link case. For max time horizons $\bar{t}_h = 0.05, 0.1, 0.25, 0.5, 1.0$s, we execute the RRT 100 times. Instead of exploring as much of the search space as possible as shown in Figure 1, we now stop execution once a state $x' \in X$ is discovered nearby a goal state—i.e. when $\|x' - x_{goal}\| < \delta = 2$. Here, x_{goal} is an unstable equilibrium with cart position on the right side of the obstacles. The results for the average execution time, average number of vertices, and average number of insertion failures is shown in Figure 3. It is evident that the number of vertices required reduces significantly for the greater time horizons, which is reflected in the execution time. Comparing the results for linearizing about x_0 and $x_{zero}(t)$, we find that the number of required vertices is nearly the same, but the number of insertion failures is significantly lesser for $x_{zero}(t)$, which is also reflected in the execution time. This increase in the number of insertion failures is due to the quality of the approximation.

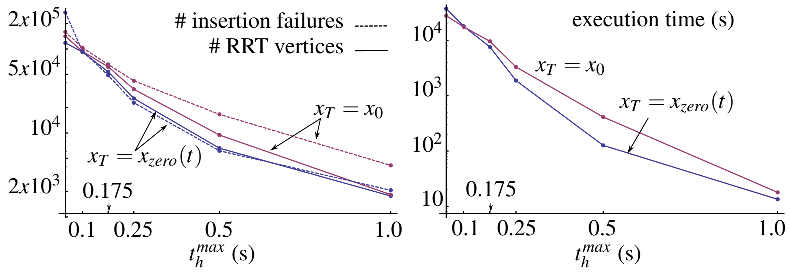

Fig. 3 Comparison of average number of RRT vertices, average number of insertion failures and average execution time for 100 runs of RRT at differing max time horizons. We stop the RRT once a state $x' \in X$ was $\|x' - x_{goal}\| < \delta$.

6 Discussion

The results compare the capability of an RRT-based planner to search the allowable state space of a dynamic system for differing max time horizons and differing linearizations. Qualitatively, Figure 1 provides evidence that planning with longer time horizons will search a space with far fewer number of vertices. When benchmarked in a feasible-path context, Figure 3 shows that the execution time to search near a goal state is over 10^3 times faster for maximum time horizon $t_h^{max} = 1.0$s compared to $t_h^{max} = 0.05$s, which is largely due to needing nearly 10^2 times more vertices. However, this gained efficiency depends greatly on the density of the obstacles since the likelihood of colliding with an obstacle increases for longer trajectories. A future direction is benchmarking this approach in environments with varying densities of obstacles, and to develop extensions that compute the best compromising maximum time horizon.

Figure 3 also compares planning with the linearization about a single state x_0 and the zero-control trajectory $x_{zero}(t)$. For longer maximum time horizons, the number of failed insertions is significantly fewer (roughly half). Depending on the severity of the nonlinearities, this difference can be more pronounced. Though, as the maximum time horizon decreases, the number of failed insertions evens out between the two linearizations types since for short enough time horizons, the two linearizations are effectively the same. Therefore, planning with the single state linearization is more desirable as it avoids the additional time-varying complexity of dealing with the zero-control trajectory linearization.

7 Conclusion

Complementing [2, 3, 15, 16], we approximate the optimal control problems that arise in sample-based planning through a linearization. However, we linearize about the zero-control trajectory, making the approximation valid for longer time horizons. Moreover, we steer a closed-loop system instead of the numerically less tractable open-loop system. Each of these decisions were made so that steering is viable over long time horizons. With longer time horizons, a planner's exploration can be coarser, thereby requiring fewer nodes in a graph, thereby decreasing a sample-based planner's computation time.

Acknowledgements This work has been supported by NASA Early Career Faculty fellowship NNX12AQ47GS02 and ARO grant W911NF1410203.

References

1. B. D. O. Anderson and J. B. Moore. *Optimal Control: Linear Quadratic Methods*. Dover Publications, INC, 1990.
2. E. Glassman and R. Tedrake. A quadratic regulator-based heuristic for rapidly exploring state space. *IEEE International Conference on Robotics and Automation*, pages 5021–5028, 2010.
3. G. Goretkin, A. Perez, R. Platt Jr., and G. Konidaris. Optimal sampling-based planning for linear-quadratic kinodynamic systems. *IEEE International Conference on Robotics and Automation*, pages 2429–2436, 2013.
4. J. Hauser. A projection operator approach to the optimization of trajectory functionals. *IFAC Wolrd Congress*, 2002.
5. J. Hauser. On the computation of optimal state transfers with application to the control of quantum spin systems. *American Control Conference*, pages 2169 – 2174, 2003.
6. J. Hauser and D. G. Meyer. The trajectory manifold of a nonlinear control system. *IEEE Conference on Decision and Control*, pages 1034–1039, 1998.
7. J. P. Hespanha. *Linear Systems Theory*. Princeton university press, 2009.

8. L. Janson and M. Pavone. Fast marching trees: a fast marching sampling-based method for optimal motion planning in many dimensions. *International Symposium on Robotics Research*, 2013.
9. J. H. Jeon, H. Karaman, and E. Frazzoli. Anytime computation of time-optimal off-road vehicle maneuvers using the rrt*. *IEEE Conference on Decision and Control*, pages 3276 – 3282, 2011.
10. S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *The International Journal of Robotics Research*, pages 846–894, 2011.
11. T. Kunz and M. Stilman. Kinodynamic rrts with fixed time step and bestinput extension are not probabilistically complete. *International Workshop on the Algorithmic Foundations of Robotics*, 2014.
12. S. M. Lavalle and J. J. Kuffner. Randomized kinodynamic planning. *The International Journal of Robotics Research*, pages 378–400, 2001.
13. Y. Li, Z. Littlefield, and K. E. Bekris. Sparse methods for efficient asymptotically optimal kinodynamic planning. *Workshop on the Algorithmic Foundations of Robotics*, 2014.
14. Z. Littlefield, Y. Li, and K. E. Bekris. Efficient sampling-based motion planning with asymptotic near-optimality guarantees for systems with dynamics. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1779 – 1785, 2013.
15. A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez. LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics. *IEEE International Conference on Robotics and Automation*, pages 2537 – 2542, 2012.
16. D. J. Webb and J. van den Berg. Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics. *IEEE International Conference on Robotics and Automation*, pages 5054–5061, 2013.