

A Class of SDRE-RRT Based Kinodynamic Motion Planners

Adnan Tahirović¹ and Faris Janjoš²

Abstract—A variety of LQR-RRT kinodynamic motion planners are built on the idea of solving a two point boundary value problem in an LQR manner for affine systems. These planners can also be used for controllable nonlinear systems only if its linearized model at the equilibrium state is also controllable, and the cost function reflects only a time/control trade-off. We propose a class of RRT planners based on the SDRE (State Dependent Riccati Equation) control paradigm. The SDRE control is used both for finding the nearest state in the tree and for the tree expansion. By solving an LQR tracking problem for nonlinear systems within the SDRE framework, instead of a two point boundary value problem, the proposed planners deal with a wider range of controllable nonlinear systems and cost functions. We compare the proposed planners with LQR-RRT-like algorithms by observing the results obtained from the three specific benchmark examples.

I. INTRODUCTION

The rapidly-exploring random trees (RRT) [1] have opened new perspectives in motion planning (MP) community [2]. This type of planner explores the configuration space by randomly distributed samples, avoiding obstacles and robot geometry to be explicitly included in the problem. This is a huge advantage, achieved at a cost of using a collision checking algorithm that secures collision-free connections between any pair of samples. The RRT incrementally expands the tree of collision-free connections starting from an initial configuration towards a goal configuration. The classical RRT algorithm uses the Euclidean distance to find the nearest node in the tree to expand the tree from that node towards a randomly generated point in the configuration space. The tree grows until it finds a final feasible solution by connecting given initial and goal configurations, or it concludes a feasible solution is not possible to find. Although RRT based on the Euclidean metric performs well in configuration space (without differential constraints), it has been shown in [3] and [4] why these algorithms do not efficiently construct trees for exploring state space of dynamical systems.

In order to address this problem, an LQR-based heuristic has been introduced in [4] as a much more efficient pseudo-metric than the Euclidean distance to find the nearest node in the tree. Namely, the authors have found a way to partially include differential constraints used to find the next node from which the tree expands. To do so, they have used a finite

horizon LQR-based control paradigm in the form of an affine quadratic regulator (AQR). As shown in [4], the RRT based on the AQR heuristic was able to construct much sparser trees than the classical RRT algorithm for several different dynamical systems (double integrator, pendulum, cart-pole and acrobot). For clarity of presentation, we shortly explain the AQR-based heuristic algorithm in Section II-B, based on which it is possible to design several LQR-RRT kinodynamic motion planners. However, we encourage the reader to refer to [4] for a more complete understanding of the proposed solution.

Built on the AQR-based metric idea, an RRT* setup using an infinite horizon LQR controller, for both finding the nearest node and expanding the tree, has been presented in [5]. The RRT* algorithm is a modification of the RRT algorithm that secures probabilistically-complete optimal planning, and it was first developed in [6]. However, using an infinite-horizon LQR controller to expand the tree from the nearest node toward a random sample is not optimal for finite-time extension even for linear systems. In that regard, a finite-horizon LQR controller used to expand the tree and to secure a probabilistically-optimal planner for linear systems that involves only a certain class of cost functions (trade off between time and energy) has been proposed in [7]. The extension of this approach to nonlinear systems was not possible either. To widen this class of cost functions including constraints on time, the authors have presented in [8] a probabilistically-optimal LQR-RRT planner for affine dynamics and quadratic cost functions.

We exploit the SDRE-based control (State-Dependent Riccati Equation) to propose a new class of RRT kinodynamic motion planners. We address all pitfalls of the RRTs based on the AQR metric, and explain how the proposed SDRE-RRT class of planners might be used to deal with some of those problems. The complete statement of contributions is given in Section III-E.

Section II gives a short introduction to the problem we address and to some RRT alternatives which are built on the AQR metric paradigm. Section III lists some major pitfalls one such planner might face in different problem setups. This section also contains all contributions of our paper resulting directly from addressing those pitfalls with the proposed SDRE-RRT class of planners. SDRE-based control paradigm is explained in Section IV for both infinite and finite horizon optimization problems. Section V describes the LQR tracking problem and what can be used to build a relevant distance metric. Section VI explains how this metric can be then adapted for nonlinear systems within the SDRE control framework. A comparison between some

*This work was not supported by any organization

¹Adnan Tahirović is with the Faculty of Electrical Engineering, Department for Automatic Control and Electronics, University of Sarajevo, 71000 Sarajevo, Bosnia and Herzegovina, atahirovic@etf.unsa.ba

²Faris Janjoš is with the Chair of Information-Oriented Control, Department of Electrical and Computer Engineering, Technical University of Munich, 80333 Munich, Germany, f.janjos@tum.de

relevant kinodynamic planners is presented in Section VII, and concluding remarks are given in Section VIII.

II. LQR-RRT KINODYNAMIC MOTION PLANNER

A. Problem formulation

We address the kinodynamic MP problem as a control problem of a class of constrained nonlinear systems governed by the state equations

$$\dot{x} = f(x) + B(x)u, \quad (1)$$

which are affine in the m^{th} dimensional control $u \in U \subset \mathbb{R}^m$, where $x \in X \subset \mathbb{R}^{2n}$ is the system state defined as $x = (q, \dot{q})$, for $q \in C$ and C being the n^{th} dimensional configuration space. U represents an admissible subspace that reflects limits imposed on the system actuations in which a control solution is being found to stabilize the system equilibrium state. X is a feasible subspace that takes into account all forbidden state space regions (obstacles) to ensure collision-free trajectories. It is also assumed that the equilibrium point of the system (1) is at the origin, i.e., $f(0) = 0$ for $u = 0$. If the equilibrium point is not at the origin, a coordinate system transform can always be found that shifts this point to the origin.

The ultimate task is to find an admissible control $u \in U$ to feasibly stabilize the system at the origin given an initial system state $x(0)$. For a solvable problem setup, we also assume the system (1) is stabilizable. However, in most cases one can impose a bit stricter assumption to let the system be controllable. In MP terms, one should find admissible system inputs u to drive the system towards the goal state space $x = 0$ by avoiding state space obstacles, for MP setups where the initial and the goal states are connectible.

In this paper, we address the kinodynamic MP problem by an RRT-like algorithm. The most widely used heuristic to find the nearest node in the tree is based on the solution of an affine quadratic regulator problem (AQR) [4], [7], [5]. The AQR-based distance heuristic used within these planners is shortly described in the sequel.

B. AQR-based pseudometric

The AQR-based pseudometric is used to compute the distances between all nodes in the tree and a random sample x_s , in terms of the nonlinear dynamical system (1). The best possible distance heuristic would be computed by solving a boundary value problem from each node of the tree to x_s . However, this would be too costly, so the problem was simplified to solve an LQR-like problem for the nonlinear system linearized at x_s (2)

$$\dot{\bar{x}} = A_s \bar{x} + B_s \bar{u} + c, \quad \bar{x} = x - x_s, \quad \bar{u} = u - u_s, \quad (2)$$

where

$$A_s = \frac{\partial f}{\partial x}(x_s) + \frac{\partial B}{\partial x}(x_s)u_s, \quad B_s = B(x_s), \quad c = f(x_s) + B_s u_s. \quad (3)$$

System (2) is an affine system in terms of constant c , which reflects the fact that the system is not stabilizable in

x_s . Such a problem can be solved in an AQR manner with respect to the cost function (4)

$$J(\bar{x}, t_f) = \int_0^{t_f} [1 + \frac{1}{2} \bar{u}^T(t) R \bar{u}(t)] dt, \quad R = R^T \succ 0, \quad (4)$$

which includes a trade-off between the minimum time needed by the system to reach x_s , and the control effort used to guide the system. The optimal cost-to-go can be then computed as

$$J^*(\bar{x}, t_f) = t_f + \frac{1}{2} d^T(\bar{x}, t_f) P^{-1}(t_f) d(\bar{x}, t_f), \quad (5)$$

where

$$d(\bar{x}_0, t_f) = e^{A t_f} \bar{x}_0 + \int_0^{t_f} e^{A(t_f - \tau)} c d\tau, \quad (6)$$

and

$$\dot{P}(t) = AP(t) + P(t)A^T + BR^{-1}B^T, \quad P(0) = 0. \quad (7)$$

Moreover, if the optimization is set to use a free final time t_f , then the minimal distance from the node x_0 to the random sample x_s can be obtained as

$$J^* = J^*(\bar{x}_0, T^*), \quad (8)$$

where

$$T^* = \arg \min_{t_f} J^*(\bar{x}_0, t_f), \quad t_f \in [0, t_f^{\max}]. \quad (9)$$

In accordance to the AQR-based distance metric, the nearest node in the tree is then the node with the smallest value of (8).

C. LQR-RRT variants

The first kinodynamic motion planner built on the AQR-based pseudometric for finding the nearest nodes from random samples during the tree expansion has been proposed in [7]. The authors have also used a finite-horizon optimal controller to compute the three expansions from the nearest node towards the random sample. The RRT* variant of the algorithm is proven to be optimal for the systems with linear differential constraints. However, this approach requires the computation of the optimal time horizon for each tree expansion, and can be used only for class of functions represented by (4). For the purpose of this presentation, we use this approach without implementing the RRT*-based *rewiring stage*, and denote by *AQR-RRT* such a planner.

The second LQR-RRT variant that includes a modified AQR-based pseudometric idea both for discovering the nearest nodes in the tree and for the tree expansion has been presented in [8]. The authors have also used a finite-horizon LQR to measure cost and to extend the tree within RRT* framework. However, they have modified the LQR optimization to cover the class of more general cost functions, including a classical quadratic cost function in terms of states and control, and time as an additional dimension of the state space.

The third kinodynamic planner simplifies the problem by omitting the constant c in the linearized system (2), and

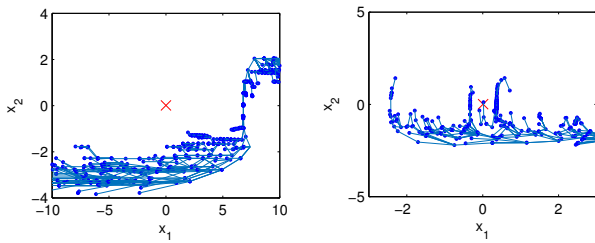


Fig. 1. One realization of the tree grown by the AQR-RRT (left) and a variant of the tracking based SDRE-RRT (right), while solving the control problem for system (10).

conducts an infinite LQR optimization both for minimum distance between two nodes to find the nearest node, and for the tree expansion [5]. Although the computation is simplified, this method does not produce locally optimal trajectories for finite-time extensions, which is a condition for an optimal MP RRT* framework in terms of Frazzoli and Karaman’s conditions [9]. For the purpose of this presentation, we use this approach without implementing the RRT*-based *rewiring stage*, and denote by *LQR-RRT* such a planner.

In this paper, we aim to examine some novel distance heuristics for kinodynamic MP within the SDRE-RRT framework. To do so, we do not put the focus on optimality but rather on examination of a potential these planners have in exploring the state space. For this reason, we will only use the basic variants of AQR-RRT and LQR-RRT planners as benchmarks for comparison.

III. LQR-RRT PITFALLS AND STATEMENTS OF CONTRIBUTIONS

A. Controllability of linearized system

The first pitfall of the AQR-based RRT regards the assumption under which we address the systems with stabilizable (controllable) dynamics. This is a very important assumption since it would not be possible to grow the tree towards the goal state from any of the selected tree nodes. Namely, since the control action in the AQR-based RRT is found by linearization of the system around the equilibrium point, each time the algorithm tends to connect the selected nearest node to the goal state, it is very important to check whether the obtained linearized system is controllable (or stabilizable). If not, the AQR-based RRT can struggle to find its way to the final goal position. For example, eq. (10) represents an unconstrained and locally controllable nonlinear system whose linearization around the equilibrium state is uncontrollable [10].

$$\dot{x}_1 = x_2^3, \quad \dot{x}_2 = u \quad (10)$$

Fig.1-left shows an AQR-RRT generated tree of 1000 nodes, which was not able to grow to connect the initial state with the equilibrium state (0,0). It should be noted that the final success depends on the chance to drive the nonlinear system from the nodes on the tree towards the

equilibrium state by using the actions found for its linearized counterpart. The problem would be even more difficult with underactuated and state constrained dynamics. Later in this work, this system is analyzed from the perspective of the proposed class of SDRE-RRT planners (Fig.1-right).

B. Fixed sample point for finding the nearest node

The second pitfall pertains to the assumption imposed on a newly generated sample state. Namely, in AQR-based heuristic and all other state-of-the-art works based on this approach, the sampled state is considered as a fixed point in the state space. Such an assumption completely ignores information on how the dynamical system would behave once being in that state (or how the tree would grow starting from that state). However, by using the AQR-based distance, this information is partially recovered in comparison to the pure Euclidean distance. In our paper, we use the LQR-tracking problem instead of finite horizon optimization in a boundary value problem, letting the sampling state move within the state space in accordance to the dynamics of the system.

C. Linearization for the tree expansion

Although the AQR-RRT algorithm includes some information on the system dynamics while expanding the tree and as such it performs much better than the classical RRT algorithm based on the pure Euclidean distance, the AQR-RRT uses a linearized dynamics around the sample point when it expands the tree from the nearest node. This way the algorithm certainly gives up from the complete nonlinear dynamics in favor of computational simplicity, so we consider this as the third pitfall.

D. Limited class of cost functions

The fourth pitfall regards the cost function (4) upon which the AQR-RRT algorithm is built on, to find the nearest node and to expand the tree towards the generated sample point. As explained in Section II-B, the authors have been forced to use this function since the random point with a nonzero velocity is not stabilizable. Although the performance of the final control solution is not addressed with the AQR-RRT approach as in many other classical RRT algorithms (except RRT* variants), the use of the criteria (4) seems somewhat artificial especially because it would be more natural to use the form of the objective function which is initially imposed on the control problem. In most control problems, the objective function is given in a quadratic form in terms of system states and control effort. In the objective function (4), the integrand function includes a constant to penalize long lasting trajectories towards the random point. This cost function is indeed a natural choice for some systems such as spacecraft in deep space, helicopters, and even ground vehicles [11]. However, to hope for the optimality in RRT* manner, it would be arguably more convenient to select the same form of the objective function for the tree expansion as the one given for the initial control problem.

Moreover, the classical LQR framework does not address the cost functions which are nonlinear and nonquadratic. In case we deal with such a cost, it would not be possible to approach an optimal sample-based planner even for a linear system.

E. Statements of contributions

In this paper we propose a class of kinodynamic motion planners built on the idea of using SDRE-based control for unconstrained nonlinear systems. By using an appropriate factorization of the system within the LQR framework instead of a linearization, the proposed SDRE-RRT planners might even stabilize controllable nonlinear systems that have uncontrollable linearizations at the equilibrium state, which would not be possible by AQR-based RRT planners.

Unlike the AQR-based heuristic, the heuristic proposed in this paper lets the sampled state move in the state space in accordance with the system dynamics. With this in mind, we propose a heuristic based on the tracking problem instead of solving boundary value problem as in case of AQR. Such an assumption helps to avoid pitfalls of the Euclidian distance metric on one side, and provides a simplified heuristic based on a classical LQR manner on the other.

By using factorization at the nearest node instead of linearization (case with LQR-like RRTs) during the tree expansion step, the proposed SDRE-RRT class of planners preserve full dynamics of a nonlinear system at this particular node. Following the same arguments as in debate of LQR vs. SDRE [12], we argue the SDRE might improve suboptimality of the expansion trajectories, which can be important once one deals with the overall optimality.

The proposed class of SDRE-RRT planners all use a classical LQR framework which rather reduces computational cost, unlike most of the AQR-based RRT planners implemented for affine systems. Moreover, the SDRE-based control setup also allows for nonlinear and nonquadratic cost functions to be used in the optimization. To do so, one needs to factorize the cost and to get a pointwise quadratic function (see eq. (12) in Section IV), which can be then solved in LQR manner.

IV. SDRE-BASED CONTROL FOR UNCONSTRAINED NONLINEAR SYSTEMS

A. Unconstrained SDRE control setup

The SDRE control strategy has been successfully used in a wide range of fields, where a nonlinear control was inevitable as a control choice. Autopilot design [13], integrated guidance and control design [14], robotics [15], to name a few, wherein the results of the SDRE implementations have been successfully demonstrated. Much more real world examples have been reported in [12].

The SDRE approach requires parametrization of the full-state observable nonlinear system (1) into the state vector and the state-dependent state matrix as follows

$$\dot{x} = A(x)x + B(x)u. \quad (11)$$

The objective function used within a classical SDRE control is an infinite horizon objective function. This function also requires an appropriate factorization as shown in (12)

$$J_{SDRE} = \frac{1}{2} \int_{t_0}^{\infty} (x^T Q(x)x + u^T R(x)u) dt, \quad (12)$$

where $Q(x)$ and $R(x)$ must be selected such that $Q(x) = D^T(x)D(x) \geq 0$ and $R(x) > 0$, $\forall x$.

The main idea behind the factorizations of the nonlinear system (11) and the objective function (12) is to represent the nonlinear system as a pointwise linear system by assuming $A(x)$ and $B(x)$ are constant matrices for each state x ; and to consider the objective function as pointwise quadratic in cost and control by assuming $Q(x)$ and $R(x)$ are constant matrices for each x as well.

The unconstrained SDRE problem can be now formulated as a minimization problem of the given state-dependent quadratic cost (12) subject to differential constraints (11) given the initial system states $x(t_0)$. In order to solve the problem, one has to solve a pointwise (or state-dependent) Riccati equation

$$A^T(x)P + PA(x) - PB(x)R^{-1}(x)B^T(x)P + Q(x) = 0. \quad (13)$$

Such an optimization represents an infinite-horizon LQR problem and it is solvable if and only if the pair (A, B) is controllable and $(A, Q^{\frac{1}{2}})$ is detectable, where $Q^{\frac{1}{2}}$ denotes the Cholesky decomposition of the matrix Q . The control sequence can be easily obtained in the form (14)

$$u_{SDRE}(x) = -R^{-1}(x)B^T(x)P(x)x, \quad (14)$$

where $P(x)$ is a symmetric positive definite solution of (13) for each x .

The LQR-based value function, which is the minimum of the cost (12) in case of frozen Riccati equation, has a quadratic form

$$V_{SDRE}(x) = \frac{1}{2} x^T P(x)x. \quad (15)$$

This Lyapunov-based function has been used in [16] to prove local asymptotic stability of the given nonlinear system, and recently combined with the model predictive control approach to obtain a globally stabilizing model predictive control for unconstrained nonlinear systems [17].

B. Finite-time unconstrained SDRE control setup

Unlike the unconstrained SDRE control setup, the finite horizon SDRE control framework [18] optimizes the objective function over a finite horizon. The unconstrained finite SDRE problem can be formulated as a minimization problem of the given state-dependent quadratic cost over a finite horizon (16) subject to differential constraints (11) given the initial system states $x(t_0)$. To solve this problem, one has now to solve a state-dependent differential Riccati equation defined by (17-18).

$$J_{SDRE}^{t_f} = \frac{1}{2} \int_{t_0}^{t_f} (x^T Q(x)x + u^T R(x)u) dt + \frac{1}{2} x^T(t_f) S x(t_f) \quad (16)$$

$$A^T(x)P + PA(x) - PB(x)R^{-1}(x)B^T(x)P + Q(x) = -\dot{P} \quad (17)$$

$$P(x, t_f) = S \quad (18)$$

The matrices $A(x)$, $B(x)$, $Q(x)$ and $R(x)$ have the same properties as for the unconstrained SDRE control setup from Section II-B. $P(x, t)$ is now a unique symmetric positive-definite solution of (17-18), where S is a positive definite or positive semi-definite final state penalty matrix.

The pointwise optimal control sequence can be then computed by the form

$$u_{SDRE}^{t_f}(x) = -R^{-1}(x)B^T(x)P(x)x \quad (19)$$

to solve for the finite-time unconstrained SDRE-based control setup.

V. LQR TRACKING BASED HEURISTICS FOR LINEAR SYSTEMS

A. Problem formulation

The main idea upon which we build a novel distance heuristic in the state space for nonlinear systems, will be first explained for linear systems, and then extended in Section VI to kinodynamic MP of nonlinear systems by using the system factorization and the SDRE control approach.

Let (20) be a linear system for which we propose a novel distance heuristic between two points in the state space, the tree node x_n and the sample point x_s

$$\dot{x} = Ax + Bu, \quad (20)$$

having an initial condition in the tree node $x(0) = x_n$. We now try to answer the question on how difficult for the system would be to go from the node x_n to the sample point x_s , by means of an appropriate cost-to-go value.

Considering a newly generated sample point x_s as fixed in the state space is quite counterintuitive since the state may include a nonzero velocity preventing the system to stay in it. To deal with that assumption in AQR-based heuristic, the linearized system is rendered to a non-standard structure with an affine term (2). In our structure, we will be able to preserve the standard linear system description, by letting the sample point moves through the state space and be generated by the autonomous system (21)

$$\dot{x}_s = Ax_s. \quad (21)$$

Such a system starts at the given sample point $x_s(0)$ and has the same state matrix A as our initial system.

The proposed distance heuristic now pertains to the question on how hard for the system would be to follow x_s , which in case of using a quadratic cost (22), both in states and control, reduces to a classical finite horizon LQR tracking problem along horizon t_f .

$$V_{LQR} = \frac{1}{2} \int_0^{t_f} \{(x - x_s)^T Q(x - x_s) + u^T R u\} dt \quad (22)$$

We can now define the LQR tracking problem in the context of the proposed distance heuristics for linear systems as follows. *Given the linear system (20), find an optimal control*

u to minimize the cost function (22), where the trajectory x_s is generated by the system (21). The cost obtained by using the optimal tracking control for different values of t_f will be used to form a class of distance pseudometrics between x_n and $x_s(0)$.

B. Solution to the LQR tracking problem for linear systems

The dynamics (21) can be subtracted from (20) to get

$$\dot{e} = Ae + Bu, \quad (23)$$

where $e = x - x_s$ is the tracking error and the cost function is given with (22).

We can now augmented the system (20) with (21) to obtain

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{B}u, \quad (24)$$

where

$$\hat{x} = \begin{bmatrix} x \\ x_s \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} A & 0 \\ 0 & A \end{bmatrix}, \quad \hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad (25)$$

and

$$\hat{Q} = \begin{bmatrix} Q & -Q \\ -Q & Q \end{bmatrix}. \quad (26)$$

Note that the cost (22) is equal to the cost (27) formed for the system (24).

$$\hat{V} = \frac{1}{2} \int_0^{t_f} \{\hat{x}^T \hat{Q} \hat{x} + u^T R u\} dt \quad (27)$$

By solving (24) with respect to (27) in LQR manner, one obtains

$$u^* = -R^{-1} \hat{B}^T \hat{P} \hat{x}(t), \quad (28)$$

where now \hat{P} is the solution of

$$-\dot{\hat{P}} = \hat{A}^T \hat{P} + \hat{P} \hat{A} - \hat{P} \hat{B} R^{-1} \hat{B}^T \hat{P} + \hat{Q}, \quad \hat{P}(t_f) = \mathbf{0} \quad (29)$$

If \hat{P} is decomposed into

$$\hat{P} = \begin{bmatrix} P & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}, \quad (30)$$

then it would be easier to see from (29) that we have the following three matrix differential equations to solve

$$-\dot{P} = A^T P + PA - PBR^{-1}B^T P + Q, \quad P(t_f) = \mathbf{0} \quad (31)$$

$$-\dot{P}_{12} = A^T P_{12} + P_{12} A - PBR^{-1}B^T P_{12} - Q, \quad P_{12}(t_f) = \mathbf{0} \quad (32)$$

$$-\dot{P}_{22} = A^T P_{22} + P_{22} A - P_{12}^T B R^{-1} B^T P_{12} + Q, \quad P_{22}(t_f) = \mathbf{0} \quad (33)$$

The LQR optimal control law (28) can be derived to the form

$$u^* = -R^{-1} B^T P x - R^{-1} B^T P_{12} x_s, \quad (34)$$

which in turn gives the optimal cost for the defined augmented system

$$\hat{V}^* = \frac{1}{2} x_n^T P x_n + x_n^T P_{12} x_s(0) + \frac{1}{2} x_s(0)^T P_{22} x_s(0). \quad (35)$$

The cost (35) will be used to find the nearest node in the tree, while control (28) will be used to expand the tree from that node towards the sample state.

VI. SDRE-RRT FOR KINODYNAMIC MOTION PLANNING

A. SDRE-based distance heuristics

In this paper, we propose several different SDRE-based distance heuristics. The first one is based on the same idea as the heuristic used for the infinite-horizon LQR-RRT [5]. Instead of linearization of the system around the sample x_s , we now factorize the system in x_s to find an infinite horizon cost-to-go from each node in the tree, and to find the nearest node x_{near} for which the cost-to-go has a minimal value. To expand the tree, we now factorize the system in x_{near} , to solve for an infinite-horizon SDRE-based control problem, and to drive the system for a fixed expansion step. For the purpose of this presentation, we denote by SDRE-RRT the planner built on this heuristic.

The remaining heuristics are derived from the LQR tracking control problem, shown in Section V for a linear system, derived by an appropriate factorization of the nonlinear system. To find the nearest node, the nonlinear system is factorized in x_s , and an optimal cost-to-go is computed for each node in the tree by (35) for the *finite-horizon LQR*. The expansion is performed by solving an SDRE-tracking control problem based on the factorization of the nonlinear system in x_{near} , by using control (28) for the *infinite-horizon LQR*. We refer to the SDRE-RRT kinodynamics motion planners based on this heuristics as *tracking-based tSDRE-RRT-T*, where T is the selected horizon used for the tracking control optimization.

B. tSDRE-RRT algorithm

The SDRE algorithm has the same flow as all other classical RRT algorithms including the AQR-RRT. The *RandomNodeInStateSpace()* function is used in Line 4 (Algorithm 1) to generate a random sample from the state space at the beginning of each iteration. We sample the state space uniformly although some other distributions may be used. The *ClosestNode()* function used in Line 5 is shown in Algorithm 2. First, we factorize the system in the sample point x_s^0 (Line 1 of Algorithm 2), then solve Riccati equation (29) (Line 2) for the *finite-horizon*, which is represented by the three counterparts (31-33). Then, we find the nearest node x_{near} in the tree from which it is the least expensive to track x_s in terms of the optimal cost-to-go function (35).

After the nearest node x_n is found, the expansion function (Line 6 in Algorithm 1, shown in Algorithm 3) is conducted by factorizing the nonlinear system in x_n (Line 1 of Algorithm 3), solving again Riccati equation (29) (Line 2) but now for the *infinite-horizon*, and then using the optimal control u_{exp} computed by (28) (Line 3) to drive the system for a fixed step T_s (Line 4).

C. Discussion

In case the system linearized at the equilibrium point (the node to which we want to stabilize the system) loses the controllability as for the system (10), then the LQR-RRT struggles to reach this node, as shown in Fig. 1-left. However, the appropriate factorization of the system in the equilibrium node may inherit controllability feature from

Algorithm 1 SDRETree(x_0, x_{goal})

```

1:  $\mathcal{T}.init(x_0)$ 
2:  $x_{new} \leftarrow x_0$ 
3: while  $\|x_{new} - x_{goal}\| < \epsilon$  do
4:    $x_{sample}^0 \leftarrow \text{RANDOMNODEINSTATESPACE}()$ 
5:    $x_{near} \leftarrow \text{CLOSESTNODE}(x_{sample}^0, \mathcal{T})$ 
6:    $x_{new} \leftarrow \text{EXTENDTREE}(x_{near}, x_{sample}^0)$ 
7:    $\mathcal{T}.add(x_{new})$ 
8: end while
9: return  $\mathcal{T}$ 

```

Algorithm 2 ClosestNode($x_{sample}^0, \mathcal{T}$)

```

1:  $A_s \leftarrow A(x_{sample}^0); B_s \leftarrow B(x_{sample}^0); Q_s \leftarrow Q(x_{sample}^0);$   

    $R_s \leftarrow R(x_{sample}^0)$ 
2:  $\hat{P}_s \leftarrow \text{SOLVERICCATTI}(A_s, B_s, Q_s, R_s)$ 
3:  $x_{near} \leftarrow \text{ArgMinCostToGo}(x_{sample}^0, \hat{P}_s, \mathcal{T})$ 

```

the controllable nonlinear system ensuring all SDRE-RRT planners are successful in reaching the goal state (Fig.1-right).

We let the random node x_s to move in the state space by using the tracking problem instead of using two point boundary value problem for a nonstabilizable state (nonzero velocity). By doing so, we take into account the dynamics the system might have at this randomly sampled state. Arguably, by using a more realistic scenario leads to a more reliable distance heuristic used to find the nearest node in the tree.

To expand the tree from the nearest node, we factorize the system in this node instead of using a linearized model. Since the factorization does not approximate the system dynamics at the state in which we factorize the system, we feel that the expansion stage will exploit more of the system dynamics than in case of system linearization. This especially becomes true when the linearization is conducted at x_s which is too far from the nearest state.

By using (22) instead of (4), we perform classical LQR optimization for both finding the nearest node and the tree expansion. However, to take into account time within the optimization process, one might augment the system as in [8] and perform a classical LQR optimization. Moreover, one can also use a cost function which is nonlinear and nonquadratic in any of the SDRE-RRT planners. Namely, by an appropriate factorization (12 or 16), the optimization is again solved pointwise in LQR manner.

Algorithm 3 ExtendTree(x_{sample}^0, x_{near})

```

1:  $A_n \leftarrow A(x_{near}); B_n \leftarrow B(x_{near}); Q_n \leftarrow Q(x_{near}); R_n \leftarrow$   

    $R(x_{near})$ 
2:  $\hat{P}_n \leftarrow \text{SOLVERICCATTI}(A_n, B_n, Q_n, R_n)$ 
3:  $u_{exp} = -R_n^{-1} B_n^T \hat{P}_n x_{near} - R_n^{-1} B_n^T P_n^{12} x_{sample}$ 
4:  $x_{new} \leftarrow \mathbf{f}(x, u_{exp}, \Delta t = T_s)$ 

```

VII. SIMULATIONS

In all presented simulations, we have used 4-core system with i5-6300HQ with 8GB RAM.

A. Non-controllable linearized system

We first show a nonlinear system (10), factorized as in (36), which would not be possible to stabilize by using any of LQR-RRT variants based on the AQR metric. This is the case when the nonlinear system is controllable but its linearized counterpart is not, which can be easily shown.

$$A(x) = \begin{bmatrix} 0 & x_2^2 \\ x_2 & -x_1 \end{bmatrix} B(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} R = \mathbf{I}. \quad (36)$$

Table I includes a comparison between AQR-RRT, LQR-RRT, SDRE, and two tSDRE-RRT planners constructed by horizons, $t_f = 0.2$ and $t_f = 0.5$. All planners have been run 20 times for three different initial conditions, x_1 , x_2 and x_3 . Then, we observe how many times a planner was able to reach the goal position w.r to the total number of runs under a limited number of nodes used to expand the trees ($N = 1000$). Table I depicts the success percentage for each planner.

As expected, the AQR-RRT and LQR-RRT planners were not able to solve the problem. However, there is a small percentage when the LQR-RRT was able to reach the goal position by chance. On the other side, the tracking-based planners were able to stabilize the system with a high percentage of success. One realization of the tSDRE-RRT-0.2 planner used to stabilize (10) can be seen in Fig.(1)-right.

TABLE I
SUCCESS STATISTICS DERIVED FOR THE SYSTEM (10)

	AQR	LQR	SDRE	tSDRE-0.2	tSDRE-0.5
$x_1 = [4.88; 1.58]$	0%	5%	5%	60%	80%
$x_2 = [-3.17; -0.49]$	0%	0%	50%	100%	30%
$x_3 = [3.47; 0.29]$	0%	5%	10%	85%	85%
$x_4 = [1.26; -0.97]$	0%	5%	10%	40%	90%

B. Pendulum

This is an underactuated nonlinear system frequently addressed within control and robotics community. A factorized version of this system is given by (37), where $x = [\theta, \dot{\theta}]^T$, θ being the pendulum displacement angle, and g the gravitational constant.

$$A(x) = \begin{bmatrix} 0 & 1 \\ -g \frac{\sin(\theta)}{\theta} & -b \end{bmatrix} B(x) = \begin{bmatrix} 0 \\ 1 \end{bmatrix} Q = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix} R = \mathbf{I}. \quad (37)$$

Table II depicts a comparison between the AQR-RRT, the LQR-RRT, the infinite horizon SDRE-RRT and one tracking-based tSDRE-RRT planner for $T = 0.5$. We first compare the potential of the planners to cover the state space going from the equilibrium state outwards for 50 iterations, and use the same coverage metric as defined in [4]. The given statistics

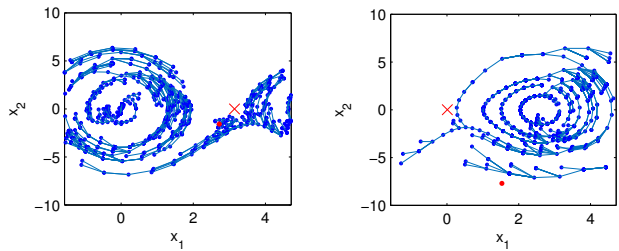


Fig. 2. One coverage realization of the tree for AQR-RRT (left) and tSDRE-RRT-0.2 (right) planners, obtained while expanding the tree from the desired state outwards for pendulum.

suggest a similar exploration potential all planners have, which can be also seen from Fig.(2) where the obtained trees by the AQR-RRT and the tSDRE-RRT-0.2 are presented.

TABLE II
COVERAGE AND SUCCESS STATISTICS DERIVED FOR PENDULUM

		AQR	LQR	SDRE	tSDRE-0.2
Coverage [%] <i>N=1000</i>	Mean	42	55	68	61
	Median	42	56	69	61
	Min	36	48	62	47
	Max	47	64	71	66
Number of nodes	Mean	743	198	615	482
	Median	749	326	532	470
	Min	510	198	239	271
	Max	983	517	987	964
Execution time [s]	Mean	242.9	1.7	5.6	9.8
	Median	239.4	1.8	4.9	9.55
	Min	110.5	0.88	1.9	4.9
	Max	407.1	2.97	5.6	22.02

Furthermore, Table II suggests the LQR-RRT is more efficient for this example in guiding the system towards the goal state (upward position at $(\pi/2, 0)$) in terms of total number of nodes and time required to solve the problem. However, the other two SDRE-based algorithms can be also used in terms of their total execution time. Fig.(3) illustrates the trees computed by these four algorithms while solving the control problem. All trees have been started at the same initial condition. Note that for all tSDRE-RRT realizations $(\pi/2, 0)$ needs to be shifted into $(0, 0)$ in accordance to the tracking problem settings.

C. 3rd order nonlinear system

The last example is taken from [19] and represents a highly nonlinear system. The nonlinear system is

$$\dot{x}_1 = x_1 - x_1^3 + x_2 + u_1, \quad \dot{x}_2 = x_1 + x_1^2 x_2 - x_2 + u_2 \quad (38)$$

where $Q = \mathbf{I}$, and $R = 2\mathbf{I}$ are the constant matrices. The extended linear parametrization of the system is given by the state dependent matrix

$$A(x) = \begin{bmatrix} 1 - x_1^2 & 1 \\ 1 + x_1 x_2 & -1 \end{bmatrix}. \quad (39)$$

The relevant statistics are shown in Table III for the same four RRT-based planners growing from $x^T = [2 \ 1]$. One

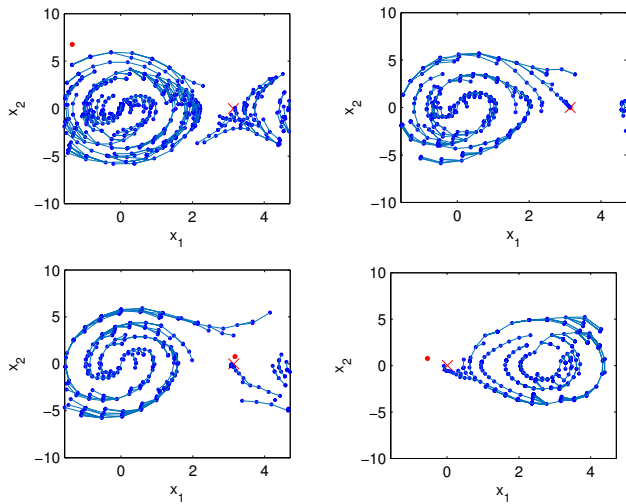


Fig. 3. One realization of the tree for AQR-RRT (up-left), LQR-RRT (up-right), SDRE-RRT (down-left) and tSDRE-RRT-0.5 (down-right), obtained while guiding the pendulum towards the upward position.

can again see the same level of exploration capacity the planners have in terms of the coverage metric. However, in this example, the tracking-based tSDRE-RRT planner for $T = 0.5$ is much more efficient than other planners in terms of both total number of nodes and time needed to stabilize the system.

TABLE III

COVERAGE AND SUCCES STATISTICS DERIVED FOR THE SYSTEM (38)

		AQR	LQR	SDRE	tSDRE-0.2
Coverage [%] <i>N=1000</i>	Mean	13	20	23	20
	Median	13	19	23	20
	Min	9	16	21	17
	Max	17	28	24	22
Number of nodes	Mean	902	99	105	28
	Median	964	91	108	26
	Min	684	14	12	15
	Max	996	218	213	70
Execution time [s]	Mean	944.0	1.7	1.8	0.5
	Median	1010.0	1.6	1.9	0.48
	Min	521.6	0.4	0.3	0.33
	Max	1179.0	1.7	3.5	1.11

VIII. CONCLUSION

In this paper we exploit the SDRE-based control paradigm to form a class of kinodynamic motion planners. For finding the nearest state in the tree and for the tree expansion, we solve the LQR tracking problem for nonlinear systems within the SDRE framework. By using the SDRE framework to solve the LQR problem in a pointwise manner for a factorized system, instead of using the AQR framework to solve a two point boundary value problem for a linearized affine system, the proposed SDRE-RRT class of planners deal with a wider range of controllable nonlinear systems and cost functions.

We present the statistics derived by applying several RRT planners on three specific nonlinear systems, a controllable nonlinear system with uncontrollable linearization, pendulum

and on a 3rd order nonlinear system. The results indicates a relevance the tSDRE-RRT kinodynamic motion planners might have in solving highly difficult control problems.

REFERENCES

- [1] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.
- [2] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, May 2006.
- [3] —, "From dynamic programming to rrt: Algorithmic design of feasible trajectories," in *Control Problems in Robotics*. Springer, 2003, pp. 19–37.
- [4] E. Glassman and R. Tedrake, "A quadratic regulator-based heuristic for rapidly exploring state space," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5021–5028.
- [5] 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 *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2537–2542.
- [6] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.
- [7] D. J. Webb and J. van den Berg, "Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5054–5061.
- [8] G. Goretkin, A. Perez, R. Platt, and G. Konidaris, "Optimal sampling-based planning for linear-quadratic kinodynamic systems," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2429–2436.
- [9] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 7681–7687.
- [10] A. Megretski. Dynamics of nonlinear systems. Massachusetts Institute of Technology. [Online]. Available: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-243j-dynamics-of-nonlinear-systems-fall-2003/lecture-notes/lec12_6243_2003.pdf
- [11] E. Schmerling, L. Janson, and M. Pavone, "Optimal sampling-based motion planning under differential constraints: the drift case with linear affine dynamics," in *Decision and Control (CDC), 2015 IEEE 54th Annual Conference on*. IEEE, 2015, pp. 2574–2581.
- [12] T. Cimen, "State-dependent riccati equation (sdre) control: a survey," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 3761–3775, 2008.
- [13] C. P. Mracek and J. R. Cloutier, "Full envelope missile longitudinal autopilot design using the state-dependent riccati equation method," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 1997, pp. 1697–1705.
- [14] N. F. Palumbo and T. D. Jackson, "Integrated missile guidance and control: A state dependent riccati differential equation approach," in *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, vol. 1. IEEE, 1999, pp. 243–248.
- [15] E. B. Erdem and A. G. Alleyne, "Experimental real-time sdre control of an underactuated robot," in *Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, vol. 3. IEEE, 2001, pp. 2986–2991.
- [16] J. R. Cloutier, C. N. D'ÁzSouza, and C. P. Mracek, "Nonlinear regulation and nonlinear h_∞ control via the state-dependent riccati equation technique: Part 1, theory," in *Proceedings of the First International Conference on Nonlinear Problems in Aviation and Aerospace*. Embry-Riddle Aeronautical Univ. Press Daytona Beach, FL, 1996, pp. 117–130.
- [17] A. Tahirovic and S. Dzuzdanovic, "A globally stabilizing nonlinear model predictive control framework," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 4033–4039.
- [18] A. Heydari and S. N. Balakrishnan, "Approximate closed-form solutions to finite-horizon optimal control of nonlinear systems," in *American Control Conference (ACC), 2012*. IEEE, 2012, pp. 2657–2662.
- [19] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.