Robust Kinodynamic Motion Planning using Model-Free Game-Theoretic Learning

George P. Kontoudis [1], *Student Member, IEEE,* Kyriakos G. Vamvoudakis [2], *Senior Member, IEEE*

***Abstract*—This paper presents an online, robust, and model-free motion planning framework for kinodynamic systems. In particular, we employ a Q-learning algorithm for a two player zero-sum dynamic game to account for worst-case disturbances and kinodynamic constraints. We use one critic, and two actor approximators to solve online the finite horizon minimax problem with a form of integral reinforcement learning. We then leverage a terminal state evaluation structure to facilitate the online implementation. A static obstacle augmentation, and a local re-planning framework is presented to guarantee safe kinodynamic motion planning. Rigorous Lyapunov-based proofs are provided to guarantee closed-loop stability, while maintaining robustness and optimality. We finally evaluate the efficacy of the proposed framework with simulations and we provide a qualitative comparison of kinodynamic motion planning techniques.**

***Index Terms*—Motion Planning, Q-learning, Game Theory.**

I. Introduction

Autonomous systems have experienced rapid advancements with the use of machine learning in decision making and supervision. Building fully autonomous systems that include smart perception and advanced control systems for motion planning is still in its infancy. Motion planning is a key research topic in autonomy and robotics [1]–[4]. An efficient motion planning algorithm is required to operate in uncertain or even unstructured environments, while ensuring safe autonomy. In realistic systems the kinodynamic constraints compose a challenging problem, especially for a real-time implementation [5], [6]. Indeed, optimality is not always guaranteed, and requires extensive offline computations that are not always feasible. Moreover, the dynamics are often difficult to derive and when obtained they are unreliable and inaccurate, because disturbances and parameter uncertainties may affect the physics of the system [7]. To deal with such problems, a solution is to employ simplified dynamical models, but still compute the optimal solution offline. Our focus in this work is on providing an online, model-free, and robust kinodynamic motion planning algorithm to autonomously perform optimal and safe navigation in environments with obstacles and external disturbances.

[1]G. P. Kontoudis is with the Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA 24060, USA, (email: gpkont@vt.edu).

[2]K. G. Vamvoudakis is with the Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA, (email: kyriakos@gatech.edu.)

This work was supported in part by an NSF CAREER under grant No. CPS-1750789, in part by NATO under grant No. SPS G5176, and in part by NSF under grant No. SATC-1801611.

In high-dimensional systems, motion planning with incremental sampling algorithms has been discussed in probabilistic road-maps (PRM) [1] and rapidly-exploring random trees (RRT) [2]. In [4], the authors proposed RRT*, an asymptotically optimal motion planning algorithm. The aforementioned approaches are not sufficient for dynamical systems with kinodynamic constraints. In [8], the authors introduced the kinodynamic RRT that accounts for the physical constraints of the system, but the control action was randomly selected. The work of [3] presented LQR-trees, a feedback motion planning algorithm. This approach cannot be implemented without complete information of the system dynamics and needs significant computations offline. The authors in [9] proposed the LQR-RRT* which requires complete information of the model, yet this is not always possible in real engineered systems. In [10], the authors proposed a kinodynamic RRT* that deals with linear systems and performs asymptotically optimal motion planning. Kinodynamic RRT* formulates a finite horizon, with a fixed final state and a minimum fuel-time performance. Such an approach requires the system dynamics, and provides a closed form solution of an open-loop controller. A real-time kinodynamic motion planning was proposed in [5]. The authors retrieve the global path offline and compute also offline the optimal solution of the two point boundary value problem (TPBVP), in order to execute the motion planning online. Game theory can potentially offer robustness guarantees for various motion planning problems [11]. Connections between game theory and motion planing were discussed in [12] for planning under sensing and control uncertainties, under environmental uncertainties, and to maintain the visibility of a target. The authors in [13] presented a differential game framework to compute the reachable sets, and perform motion planning under uncertainties.

Adaptive control [14] along with game theory [15] can be efficiently connected by employing the principles of reinforcement learning [16], and approximate dynamic programming, e.g., actor/critic structures [17]–[21]. Specifically, the critic evaluates the cost and the actor performs a policy improvement. In [22], a discrete-time Q-learning formulation was used to solve controlled Markovian systems. However, for continuous time systems the problem is nontrivial. In [23], a relation of Q-learning with nonlinear control was established, based on the observation that the Q-function is related with the Hamiltonian that appears in the minimum principle. In [24], a Q-learning approach for solving the model-free, infinite horizon, continuous time problem was presented.

The contributions of this paper is threefold. First, we formulate a two player zero-sum game for a TPBVP with a continuous Q-learning framework without requiring the solution of the game differential Riccati equation. Next, we provide rigorous Lyapunov-based proofs for global asymptotic stability of the equilibrium point. Finally, we develop a terminal state evaluation framework, a static obstacle augmentation, and a local re-planning technique to alleviate the computational effort and to ensure safe kinodynamic motion planning.

The remainder of this paper is organized as follows. Section II focuses on the problem formulation, Section III discusses the game-theoretic structure, Section IV provides a model-free formulation, Section V presents the motion planning structure, and the algorithmic framework, Section VI shows the efficacy through simulations and discusses a qualitative comparison, while Section VII concludes the paper. The stability proof is provided in the Appendix.

Notation: $\mathbb{R}^+$ is the set of all positive real numbers and $\mathbb{N}$ is the set of natural numbers. The $\underline{\lambda}(A)$ and the $\overline{\lambda}(A)$ are the minimum and the maximum eigenvalues of the matrix A. We denote $\|\cdot\|_p$ as the p-norm of a vector. The vech(A), the vec(A), and the mat(A) are the half-vectorization, the vectorization, and the matrization of a matrix A. The $U \otimes V$ denotes the Kronecker product of two vectors. The $\oplus$ is the Minkowski sum of two sets.

II. Problem Formulation

Consider the following continuous-time kinodynamic autonomous system,

$$\dot{x}(t) = Ax(t) + Bu(t) + Fd(t), \;\; x(0) = x_0, \;\; t \geq 0,$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ is a measurable kinodynamic state vector, $u(t) \in \mathbb{R}^m$ is the control input, $d(t) \in \mathbb{R}^q$ is the disturbance input, $A \in \mathbb{R}^{n\times n}$ is the unknown plant matrix, $B \in \mathbb{R}^{n\times m}$ is the unknown input matrix, and $F \in \mathbb{R}^{n\times q}$ is the unknown disturbance matrix. The game has two players, the control $u(t)$ and the disturbance $d(t)$ [25], [26]. The player 1 (P_1) is the control $u(t)$ that aims to minimize the performance, while the player 2 (P_2) is the disturbance $d(t)$ that desires to maximize the performance.

Since we want to drive the system from an initial state x_0 to a final state $x(T) = x_{\mathrm{r}}$, we define the difference between the state $x(t)$ and the state x_{r}, as the new state $\bar{x}(t) \coloneqq x(t) - x_{\mathrm{r}}$. The final time is denoted by $T \in \mathbb{R}^+$. Similarly, we define the new control as, $\bar{u}(t) \coloneqq u(t) - u_{\mathrm{r}}$, with $u_{\mathrm{r}} = u(T)$, and the new disturbance as, $\bar{d}(t) \coloneqq d(t) - d_{\mathrm{r}}$, with $d_{\mathrm{r}} = d(T)$. The new system has now the form,

$$\dot{\bar{x}}(t) = A\bar{x}(t) + B\bar{u}(t) + F\bar{d}(t), \;\; \bar{x}_0, \; t \geq 0. \tag{1}$$

The finite horizon cost functional is given as,

$$J(\bar{x};\bar{u},\bar{d};t_0,T) = \phi(T) + \frac{1}{2}\int_{t_0}^{T} \bar{x}^\intercal M\bar{x} + \bar{u}^\intercal R\bar{u} - \gamma^2\|\bar{d}\|^2 \mathrm{d}\tau, \tag{2}$$

where $\phi(T) \coloneqq \frac{1}{2}\bar{x}^\intercal(T)P(T)\bar{x}(T)$ is the terminal cost with $P(T) \in \mathbb{R}^{n\times n} \succ 0$ the final Riccati matrix, $M \in \mathbb{R}^{n\times n} \succeq 0$, $R \in \mathbb{R}^{m\times m} \succ 0$ user defined matrices that penalize the state and the control input respectively, and $\gamma^\star \leq \gamma \in \mathbb{R}^+$ is a disturbance rejection constant, where $\gamma^\star$ is the smallest value that stabilizes the system (1), [27].

Assumption 1: We assume that the unknown pair (A,B) will be controllable and the unknown pair ($\sqrt{M}$, A) will be detectable. □

We are thus interested in obtaining the saddle-point equilibrium $(\bar{u}^\star, \bar{d}^\star)$ such that, $J(\bar{x};\bar{u}^\star,\bar{d};t_0,T) \leq J(\bar{x};\bar{u}^\star,\bar{d}^\star;t_0,T) \leq J(\bar{x};\bar{u},\bar{d}^\star;t_0,T), \;\; \forall \bar{x},\bar{u},\bar{d}$, which can be described by the min-max problem $J(\bar{x};\bar{u}^\star,\bar{d}^\star;t_0,T) = \min_{\bar{u}}\max_{\bar{d}} J(\bar{x};\bar{u},\bar{d};t_0,T)$ subject to (1). In other words, we want to determine the value function $V^\star$, which is defined by the minimax optimization,

$$V^\star(\bar{x};t_0,T) \coloneqq \tag{3}$$
$$\min_{\bar{u}}\max_{\bar{d}} \Big(\phi(T) + \frac{1}{2}\int_{t_0}^{T} \bar{x}^\intercal M\bar{x} + \bar{u}^\intercal R\bar{u} - \gamma^2\|\bar{d}\|^2 \mathrm{d}\tau\Big),$$

but without any information of the system dynamics, as given in (1).

Consider the $\mathcal{X}_{\text{obs}} \subset \mathcal{X}$ as the obstacle closed space. The free space is defined as, $\mathcal{X}_{\text{free}} \coloneqq (\mathcal{X}_{\text{obs}})^{\complement} = \mathcal{X}\backslash\mathcal{X}_{\text{obs}}$. The output of the RRT* will produce the global path $\pi(x_{0,i}, x_{\mathrm{r},i}) \in \mathbb{R}^{2(k\times n)}$, for $i = 1,\ldots,k$ with $i \in \mathbb{N}$, that has k-sets of i-TPBVPs. The global path $\pi(x_{0,i}, x_{\mathrm{r},i})$ includes the initial states $\mathcal{X}_0 = x_{0,i}$ for all i, with $\mathcal{X}_0 \in \mathbb{R}^{k\times n} \subset \mathcal{X}_{\text{free}}$ and the final states $\mathcal{X}_{\mathrm{G}} = x_{\mathrm{r},i}$ for all i, with $\mathcal{X}_{\mathrm{G}} \in \mathbb{R}^{k\times n} \subset \mathcal{X}_{\text{free}}$. The algorithm also outputs an initial graph $\mathcal{G} = (V, E)$, where V is the initial set of nodes and E the initial set of edges. As a slight abuse of notation, we will refer to nodes $v \in V$ as states $x \in \mathcal{X}$.

Since we are solving a finite horizon optimal control problem with free final state, we can make the following approximation $\lim_{t\to T} x(t) \approx x_{\mathrm{r}}$. This means that the final state $x(T)$ may not obtain the exact desired state x_{r} value. Yet, the system may be fast enough to approximate the desired state, and stay there, until the end of the fixed finite horizon T. To address this problem we define the initial distance as the n-norm of the initial state and the desired state as,

$$D_0(\bar{x}_0) \coloneqq \|\bar{x}_0\|_n, \;\; \forall \bar{x}_0 \in \mathbb{R}^n. \tag{4}$$

Then we measure the relative distance at time $t \geq 0$ with the n-norm of the current state and the desired states as,

$$D(\bar{x}) \coloneqq \|\bar{x}\|_n, \;\; \forall \bar{x} \in \mathbb{R}^n. \tag{5}$$

Lastly, we define the distance error of (4) and (5) as,

$$e_{\mathrm{d}}(\bar{x}_0, \bar{x}) \coloneqq |D_0(\bar{x}_0) - D(\bar{x})|. \tag{6}$$

Our work will formulate, an online implementation framework of robust kinodynamic motion planning given the global path and the initially randomly-sampled graph, with completely unknown dynamics as provided in (1).

III. Two-Player Zero-Sum Game

We employ the continuous-time Hamilton-Jacobi-Isaacs (HJI) [25] equation for the finite horizon optimal control problem with respect to (1) and (3) that yields,

$$\mathcal{H}(\bar{x};\bar{u},\bar{d};\frac{\partial V^\star}{\partial t},\frac{\partial V^\star}{\partial \bar{x}}) = \frac{1}{2}(\bar{x}^\intercal M\bar{x}+\bar{u}^\intercal R\bar{u}-\gamma^2\|\bar{d}^\star\|^2) + \frac{\partial V^{\star\,\intercal}}{\partial \bar{x}}(A\bar{x}+B\bar{u}+F\bar{d})+\frac{\partial V^\star}{\partial t},$$

Considering the linear system (1), let us define a value function that is quadratic in the state $\bar{x}$ as,

$$V^\star(\bar{x};t)=\frac{1}{2}\bar{x}^\intercal P(t)\bar{x},\ \forall \bar{x},t\geq 0, \tag{7}$$

where $P(t)\in\mathbb{R}^{n\times n}\succ 0$ is the time varying Riccati matrix, that can be obtained by, solving,

$$-\dot{P}(t) = P(t)A+A^\intercal P(t)+M+\gamma^{-2}P(t)FF^\intercal P(t) - P(t)BR^{-1}B^\intercal P(t). \tag{8}$$

Theorem 1: Assume that, there exists a positive definite $P(t),\ t\geq 0$ that satisfies the game Riccati equation given in (8) with a final condition $P(T)=P_{\mathrm{T}}$. Then the state feedback policies,

$$\bar{u}^\star(\bar{x};t)=-R^{-1}B^\intercal P(t)\bar{x},\ \forall\bar{x},t, \tag{9}$$

and worst-case disturbance of the form

$$\bar{d}^\star(\bar{x};t)=\gamma^{-2}F^\intercal P(t)\bar{x},\ \forall\bar{x},t, \tag{10}$$

form a saddle-point equilibrium with value $V^\star=\bar{x}_0^\intercal P(0)\bar{x}_0$.

Proof. The proof follows from [26]-(Corollary 17.1). ∎

IV. Model-Free Formulation

Let us define the advantage function as,

$$\mathcal{Q}(\bar{x};\bar{u},\bar{d};t)\coloneqq V^\star(\bar{x};t)+\mathcal{H}(\bar{x};\bar{u},\bar{d};\frac{\partial V^\star}{\partial t},\frac{\partial V^\star}{\partial\bar{x}}), \tag{11}$$

where $\mathcal{Q}(\bar{x};\bar{u},\bar{d};t)$ is an action-dependent value that maps $\mathcal{Q}:\mathbb{R}^{n+m+q}\rightarrow\mathbb{R}^+$.

Lemma 1: The solution of the problem $\mathcal{Q}^\star(\bar{x};\bar{u}^\star,\bar{d}^\star;t)\coloneqq \min_{\bar{u}}\max_{\bar{d}}\mathcal{Q}(\bar{x};\bar{u},\bar{d};t)$ has the same value with the function $V^\star$ in (7) of the min-max problem (3), where $P\succ 0$ is the Riccati matrix found by solving (8), with a final condition given as, $P(T)=P_{\mathrm{T}}$.

Proof. Substitute (9) and (10), in (11) to obtain $\mathcal{Q}^\star(\bar{x};\bar{u}^\star,\bar{d}^\star;t)=V^\star(\bar{x},t)$. ∎

Next, we define $U\coloneqq[\bar{x}^\intercal\ \ \bar{u}^\intercal\ \ \bar{d}^\intercal]^\intercal$ to obtain the Q-function (11) in a compact quadratic form,

$$\mathcal{Q}(\bar{x};\bar{u},\bar{d};t)=\frac{1}{2}U^\intercal\begin{bmatrix}Q_{\mathrm{xx}}(t) & Q_{\mathrm{xu}}(t) & Q_{\mathrm{xd}}(t)\\ Q_{\mathrm{ux}}(t) & Q_{\mathrm{uu}} & Q_{\mathrm{ud}}\\ Q_{\mathrm{dx}}(t) & Q_{\mathrm{du}} & Q_{\mathrm{dd}}\end{bmatrix}U \coloneqq \frac{1}{2}U^\intercal\bar{\mathcal{Q}}(t)U=\frac{1}{2}\mathrm{vech}(\bar{\mathcal{Q}}(t))^\intercal(U\otimes U), \tag{12}$$

where $Q_{\mathrm{xx}}(t)=\dot{P}(t)+P(t)+M+P(t)A+A^\intercal P(t)+P(t)B$, $Q_{\mathrm{xu}}=P(t)B$, $Q_{\mathrm{xd}}(t)=P(t)F$, $Q_{\mathrm{ux}}(t)=B^\intercal P(t)$, $Q_{\mathrm{uu}}=R$, $Q_{\mathrm{ud}}=Q_{\mathrm{du}}=0$, $Q_{\mathrm{dx}}(t)=F^\intercal P(t)$, and $Q_{\mathrm{dd}}-\gamma^2$. We can parametrize the control $u^\star$ given in (9) and the disturbance $d^\star$ given in (10) with respect to the Q-function, by employing the stationarity conditions $\frac{\partial\mathcal{Q}(\bar{x};\bar{u},\bar{d};t)}{\partial\bar{u}}=0$, and $\frac{\partial\mathcal{Q}(\bar{x};\bar{u},\bar{d};t)}{\partial\bar{d}}=0$, that yield $\bar{u}^\star(\bar{x};t)=\arg\min_{\bar{u}}\mathcal{Q}(\bar{x};\bar{u},\bar{d};t)=-Q_{\mathrm{uu}}^{-1}Q_{\mathrm{ux}}(t)\bar{x}$ and, $\bar{d}^\star(\bar{x};t)=\arg\max_{\bar{d}}\mathcal{Q}(\bar{x};\bar{u},\bar{d};t)=-Q_{\mathrm{dd}}^{-1}Q_{\mathrm{dx}}(t)\bar{x}$, respectively.

A. Actor/Critic Structure

Let us define, $\nu(t)^\intercal W_{\mathrm{c}}\coloneqq\frac{1}{2}\mathrm{vech}(\bar{\mathcal{Q}}(t))$, where $\nu(t)$ is a bounded radial basis function of proper dimensions that depends explicitly on time $t\geq 0$. Since the ideal weight parameters are unknown, we will be motivated by adaptive control techniques [14] to find tuning laws for the current weight values. Therefore, the estimated Q-function yields,

$$\hat{\mathcal{Q}}(\bar{x};\bar{u},\bar{d};t)=\hat{W}_{\mathrm{c}}^\intercal\nu(t)(U\otimes U). \tag{13}$$

Similarly, we define the actor that approximates the control policy with $W_{\mathrm{a}}^\intercal\mu(t)\coloneqq-Q_{\mathrm{uu}}^{-1}Q_{\mathrm{ux}}(t)$, where $\mu(t)$ is a bounded radial basis function of appropriate dimensions. Thus, the control policy actor yields,

$$\hat{\bar{u}}(\bar{x};t)=\hat{W}_{\mathrm{a}}^\intercal\mu(t)\bar{x},\ \forall\bar{x}\in\mathbb{R}^n. \tag{14}$$

The actor to approximate the disturbance can be written as,

$$\hat{\bar{d}}(\bar{x};t)=\hat{W}_{\mathrm{d}}^\intercal\xi(t)\bar{x},\ \forall\bar{x}\in\mathbb{R}^n, \tag{15}$$

where $\hat{W}_{\mathrm{d}}^\intercal\xi(t)=-Q_{\mathrm{dd}}^{-1}Q_{\mathrm{dx}}(t)$, where $\xi(t)$ is a bounded radial basis function of proper dimensions that depends explicitly on time $t\geq 0$.

Remark 1: The critic and the actor approximators described in (13), , and (14) respectively, do not include any approximations errors. Therefore, we use the whole space and not just a compact set. With this structure, the approximations will converge to the optimal policies, thus the superscript $\star$ that denotes the ideal values of the adaptive weight estimation renders similarly with the optimal solutions. □

We leverage the integral Bellman equation from [20], and the result of Lemma 1 to obtain,

$$\mathcal{Q}^\star(\bar{x}(t);\hat{\bar{u}}^\star(t),\hat{\bar{d}}^\star(t);t) = \mathcal{Q}^\star(\bar{x}(t-\Delta t);\hat{\bar{u}}^\star(t-\Delta t),\hat{\bar{d}}^\star(t-\Delta t);t-\Delta t) - \frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^\intercal M\bar{x}+\hat{\bar{u}}^\intercal R\hat{\bar{u}}-\gamma^2\|\hat{\bar{d}}\|^2)\,\mathrm{d}\tau, \tag{16}$$

$$\mathcal{Q}^\star(\bar{x}(T),T)=\frac{1}{2}\bar{x}^\intercal(T)P(T)\bar{x}(T), \tag{17}$$

where $\Delta t\in\mathbb{R}^+$ is a small time interval.

We define the critic estimation errors $e_{\mathrm{c}_1},e_{\mathrm{c}_2}\in\mathbb{R}$, that we want to eventually drive to zero by tuning the parameters of

the critic in (13). Define the first critic error e_{c_1} as,

$$e_{c_1} \coloneqq \hat{W}_c^\intercal \nu(t)\Big((\hat{U}(t)\otimes\hat{U}(t)) - (\hat{U}(t-\Delta t)\otimes\hat{U}(t-\Delta t)\Big) + \frac{1}{2}\int_{t-\Delta t}^{t}(\bar{x}^\intercal M\bar{x} + \hat{u}^\intercal R\hat{u} - \gamma^2\|\hat{d}\|^2)\,\mathrm{d}\tau,$$

and the second critic error as, $e_{c_2} \coloneqq \frac{1}{2}\bar{x}^\intercal(T)P(T)\bar{x}(T) - \hat{W}_c(T)^\intercal\nu(T)(U(T)\otimes U(T))$. Next, we write the actor approximator error as, $e_a \coloneqq \hat{W}_a^\intercal\mu(t)\bar{x} + \hat{Q}_{uu}^{-1}\hat{Q}_{ux}(t)\bar{x}$, where $e_a \in \mathbb{R}^m$, and the values of $\hat{Q}_{uu}, \hat{Q}_{ux}$ will be obtained from the critic weights $\hat{W}_c$. Similarly, we write the disturbance errors as, $e_d \coloneqq \hat{W}_d^\intercal\xi(t)\bar{x} + \hat{Q}_{dd}^{-1}\hat{Q}_{dx}(t)\bar{x}$, where $e_d \in \mathbb{R}^q$, and the values of $\hat{Q}_{dd}, \hat{Q}_{dx}$ will be obtained from the critic weights $\hat{W}_c$. Then, we define the squared-norm of the errors,

$$K_1(\hat{W}_c, \hat{W}_c(T)) = \frac{1}{2}\|e_{c_1}\|^2 + \frac{1}{2}\|e_{c_2}\|^2, \tag{18}$$

$$K_2(\hat{W}_a) = \frac{1}{2}\|e_a\|^2, \tag{19}$$

$$K_3(\hat{W}_d) = \frac{1}{2}\|e_d\|^2. \tag{20}$$

B. Learning Framework

The learning framework consists of three "plug-n-play" tuning laws. We apply a normalized gradient descent technique [14] in (18) for the critic estimation weights,

$$\dot{\hat{W}}_c = -\alpha_c\Big(\frac{1}{(1+\sigma^\intercal\sigma)^2}\sigma e_{c_1} + \frac{1}{(1+\sigma_f^\intercal\sigma_f)^2}\sigma_f e_{c_2}\Big), \tag{21}$$

where, $\sigma \coloneqq \nu(t)(\hat{U}(t)\otimes\hat{U}(t) - \hat{U}(t-\Delta t)\otimes\hat{U}(t-\Delta t))$, $\sigma_f = \nu(T)(U(T)\otimes U(T))$, and $\alpha_c \in \mathbb{R}^+$ is a constant gain that determines the convergence rate. The critic tuning law (21) guarantees that as $e_{c_1} \to 0$ and $e_{c_2} \to 0$, then $\hat{W}_c \to W_c$. We derive a tuning law for the weights of the actor $\hat{W}_a$, by applying a gradient descent algorithm in (19) that yields,

$$\dot{\hat{W}}_a = -\alpha_a\frac{\partial K_2}{\partial\hat{W}_a} = -\alpha_a\bar{x}e_a^\intercal, \tag{22}$$

where $\alpha_a \in \mathbb{R}^+$ is a constant gain that specifies the convergence rate. The actor tuning law (22) guarantees that as $e_a \to 0$ then $\hat{W}_a \to W_a$. The weights of the actor $\hat{W}_d$ for the worst-case disturbance, by applying a gradient descent algorithm in (20) yields,

$$\dot{\hat{W}}_d = -\alpha_d\frac{\partial K_3}{\partial\hat{W}_d} = -\alpha_d\bar{x}e_d^\intercal, \tag{23}$$

where $\alpha_d \in \mathbb{R}^+$ is a constant gain that specifies the convergence rate. The actor tuning law (23) guarantees that as $e_d \to 0$ then $\hat{W}_d \to W_d$.

Next, we provide the main Theorem for the proposed framework.

Theorem 2: Consider the system (1), with the critic, the control actor, and the disturbance actor approximators given by (13), (14), and (1) respectively. The weights of the critic, the control actor, and the disturbance actor estimators are tuned by (21), (22), and (23) respectively. Then, the origin is a globally uniformly asymptotically stable equilibrium point of the closed-loop system with state $\psi = [\bar{x}^\intercal\ \tilde{W}_c^\intercal\ \tilde{W}_a^\intercal\ \tilde{W}_d^\intercal]^\intercal$ and for all initial conditions $\psi(0)$, given that the critic constant gain α_c is sufficiently larger than the actor gains $\alpha_a,\ \alpha_d$ and the following inequalities hold with δ_1 and δ_2 constants of unity order,

$$0 < \alpha_a < \frac{\underline{\lambda}(M + Q_{xu}R^{-1}Q_{xu}^\intercal - \gamma^{-2}Q_{xd}Q_{xd}^\intercal) - \overline{\lambda}(Q_{xu}Q_{xu}^\intercal)}{\delta_1\overline{\lambda}\Big(\frac{\mu(t)R^{-1}}{\|1+\mu(t)^\intercal\mu(t)\|^2}\Big)},$$

$$0 < \alpha_d < \frac{\underline{\lambda}(Q_{xd}Q_{xd}^\intercal)}{\delta_2\overline{\lambda}\Big(\frac{\xi(t)\gamma^{-2}}{\|1+\xi(t)^\intercal\xi(t)\|^2}\Big)};\ \ \alpha_c \gg \alpha_a.$$

Proof. See the Appendix. □

V. Motion Planning Framework

In this section, we discuss the structure of the proposed model-free, online kinodynamic motion planning with Q-learning, game-theory, and optimal sampling-based path planners. The motion planning structure is shown in Fig. 1. The structure consists of an offline RRT* computation; an online actor/critic structure; an online terminal state evaluation; an online static obstacle augmentation; and an online local re-planning.

First, we compute offline the global optimal path $\pi(x_{0,i}, x_{r,i})$, using the RRT* algorithm. Then, we continue with the online model-free learning of the optimal policy for the worst-case disturbance. More specifically, we evaluate the policy with a critic and we improve the policy with an actor by considering the worst-case disturbance obtained by another actor. The critic's objective is to estimate the Q-function, which is obtained from the Equations (16), and (17). The critic approximates the $\hat{\mathcal{Q}}$ using (13), where $\hat{W}_c$ are the critic parameters that can be computed online by (21). The control actor computes the action $\hat{u}$ according to (14), where $\hat{W}_a$ are the actor parameters that can be estimated online by (22). The disturbance actor computes the action $\hat{d}$ according to (1), where $\hat{W}_d$ are the actor parameters that can be estimated online by (23). The critic parameters include intrinsic dynamics, which can be obtained by taking the time derivative that yields,

$$\dot{p} = \bar{x}^\intercal(t)M\bar{x}(t) - \bar{x}^\intercal(t-\Delta t)M\bar{x}(t-\Delta t) + \bar{u}^\intercal(t)R\bar{u}(t) - \bar{u}^\intercal(t-\Delta t)R\bar{u}(t-\Delta t). \tag{24}$$

A distance metric will be used to evaluate the final state x_r. The initial distance D_0 is computed by (4). Next, the relative distance D is obtained online at every iteration Δt by (5). In the case that the distance error (6) decreases below an admissible value of the initial distance $e_d \leq \beta D_0$, $\beta \in \mathcal{B} \coloneqq \{\beta \in \mathbb{R} \mid 0 \leq \beta \leq 1\}$, we continue to the next i-TPBVP, by assigning the current state value as the new initial state $x_{0,i+1} = x(t)$. It is to be noted that the i-TPBVP is specified by the i-set of the initial and the final states x_0, x_r, which were initially provided by the global planning with RRT*.

The RRT* algorithm is proved to compute the optimal path, which most of the times passes very close to the obstacles.

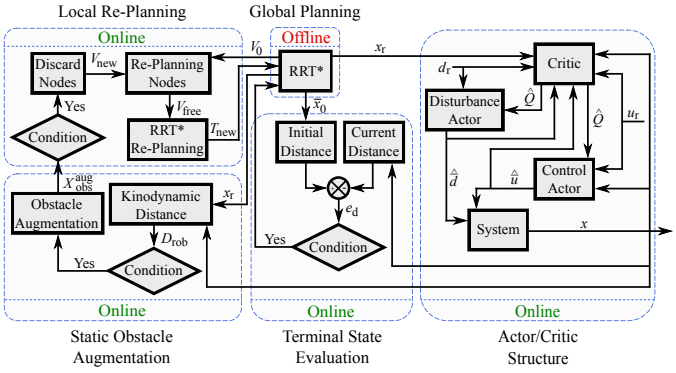

Fig. 1. The structure of the motion planning algorithm. The sequence operation is clockwise, starting from the global planning. The structure blends five stages: 1) the offline global RRT* computation, 2) the online actor/critic structure, 3) the online terminal state evaluation, 4) the static obstacle augmentation, and 5) the online local RRT* re-planning.

Inherently, in kinodynamic motion planning straight lines cannot be tracked, due to the constraints imposed by the physics of the system. Therefore, when the robot navigates closely to the obstacle and deviates from the given global path, then a collision may occur with the obstacle. To address this problem we propose a static augmentation of the obstacle space and a local re-planning strategy using the precomputed randomly sampled nodes that were employed for the global planning.

For the static obstacle augmentation, we compute the maximum deviation of the robot motion from the straight line at every TPBVP, that we term as kinodynamic distance,

$$D_{\text{rob}}(\bar{x}_0, \bar{x}) = \frac{|\bar{x}_0 \times \bar{x}|}{D_0(\bar{x}_0)}. \tag{25}$$

Next, if the kinodynamic distance is greater than the previously measured deviations in motion $D_{\text{rob},i} > \max\{D_{\text{rob},1}, \ldots, D_{\text{rob},i-1}\}$, we compute the augmented closed obstacle space, $\mathcal{X}_{\text{obs}}^{\text{aug}} := \mathcal{X}_{\text{obs}} \oplus \mathcal{X}_{\text{rob}}$, where $\mathcal{X}_{\text{rob}} \in \mathbb{R}^2$ is the kinodynamic distance space that is constructed as a rectangle with sides $\delta = 2D_{\text{rob}}$. That is a conservative approach, because we limit the navigation considering the maximum kinodynamic distance. Although, since we tackle the model-free problem, the system is unknown for offline computations. Therefore, the agent may deviate from the optimal path, yet we guarantee collision-free navigation.

We continue on the local re-planning stage that will provide a safe path in the open diminished free space $\mathcal{X}_{\text{free}}^{\text{dim}} := (\mathcal{X}_{\text{obs}}^{\text{aug}})^{\complement} = \mathcal{X} \backslash \mathcal{X}_{\text{obs}}^{\text{aug}}$. We start by evaluating whether the global path collides with the augmented obstacle space. Then, if a collision occurs, the graph $\mathcal{G}(V, E)$ is pruned by discarding the nodes in the augmented obstacle space from the initial list of nodes, $V_{\text{new}} = V \backslash V_{\text{aug}}$, $V_{\text{aug}} = V \in \mathcal{X}_{\text{obs}}^{\text{aug}}$. Now since it is required to perform online the algorithm, we cannot computationally afford to perform the RRT* even in the diminished free state space $\mathcal{X}_{\text{free}}^{\text{dim}}$. Therefore, a significantly reduced free state space needs to be specified.

The underlying idea for the local path planning problem

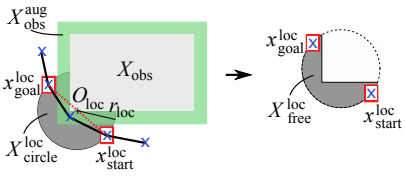

Fig. 2. The construction of the local circular space $\mathcal{X}_{\text{circle}}^{\text{loc}}$ and the local free space $\mathcal{X}_{\text{free}}^{\text{loc}}$. The local free space $\mathcal{X}_{\text{free}}^{\text{loc}}$ contains the local start state $x_{\text{start}}^{\text{loc}}$ and the local goal state $x_{\text{goal}}^{\text{loc}}$.

exploits the precomputed nodes and the global path, toward defining a new local free state space $\mathcal{X}_{\text{free}}^{\text{loc}}$. First, we search for the two closest states of the initial global path outside the area of collision with the augmented obstacle space. These two states will serve as the local start state $x_{\text{start}}^{\text{loc}}$ and the local goal state $x_{\text{goal}}^{\text{loc}}$, while the rest path will not be affected. If any states of the global path are located in the augmented obstacle space $\mathcal{X}_{\text{obs}}^{\text{aug}}$ we discard them from the updated list of nodes V_{new}. Next, we establish a circle with center point at $O_{\text{loc}} = (x_{\text{start}}^{\text{loc}} + x_{\text{goal}}^{\text{loc}})/2$ and radius $r_{\text{loc}} = \|x_{\text{start}}^{\text{loc}} - x_{\text{goal}}^{\text{loc}}\|$, that forms the local closed circular space $\mathcal{X}_{\text{circle}}^{\text{loc}} := \{x \in \mathcal{X} \mid \|x - O_{\text{loc}}\|^2 \leq r_{\text{loc}}^2\}$ as shown in Fig. 2. Then, the local closed free path planning space is defined as the relative complement of the augmented obstacle space in the local circular space, $\mathcal{X}_{\text{free}}^{\text{loc}} := \mathcal{X}_{\text{circle}}^{\text{loc}} \backslash \mathcal{X}_{\text{obs}}^{\text{aug}}$. To assess the $\mathcal{X}_{\text{free}}^{\text{loc}}$ we introduce the following definitions.

Definition 1: If A is a subset of a metric space X, and if ∂A denotes the set of all its limit points, then $\bar{A}$ said to be closure of A if $\overline{A} = A \cup \partial A$. □

Definition 2: Two subsets A, B of a metric space X are said to be separated if both $A \cap \overline{B} = \emptyset$, $\overline{A} \cap B = \emptyset$ hold. □

Definition 3: A set A is connected if it is not the union of two separated sets. □

Lemma 2: For a given set of states in the open diminished free space $\mathcal{X}_{\text{free}}^{\text{dim}}$, the local start state $x_{\text{start}}^{\text{loc}}$, and the local goal state $x_{\text{goal}}^{\text{loc}}$, if there exists a sufficient, connected, and closed local free space $\mathcal{X}_{\text{free}}^{\text{loc}}$ that forms a ring, based on the fixed incremental distance ϵ of the RRT*, then we can obtain a collision-free path with the local re-planning framework.

Proof. The proof follows from [28]. ■

Remark 2: The determination of a significantly small local free space $\mathcal{X}_{\text{free}}^{\text{loc}}$ that is connected and guarantees the existence of a local path π^{loc} from the local initial state $x_{\text{start}}^{\text{loc}}$ to the local goal state $x_{\text{goal}}^{\text{loc}}$ is a challenging problem. This difficulty lies in the unknown kinodynamic distances D_{rob} due to the model-free approach that augments the obstacle space, the unknown number of states that will be discarded from the initial global path π, and the requirements for reduced computational effort that will allow the online implementation of the algorithm. In this paper, we assess the local candidate path planning space $\mathcal{X}_{\text{cand}}^{\text{loc}}$ and we discuss the case of a connected space. □

Since we obtained a small local free space $\mathcal{X}_{\text{free}}^{\text{loc}}$ that is guaranteed to contain the local start state $x_{\text{start}}^{\text{loc}}$, the local goal state $x_{\text{goal}}^{\text{loc}}$, and sufficient space for the implementation of the path planning with incremental distance ϵ, we move on the

local re-planning step with RRT*. We equip the algorithm with the local free nodes, that is the global nodes which are located inside the local free space, $V_{\text{free}} = V_{\text{new}} \in \mathcal{X}_{\text{free}}^{\text{loc}}$. The output is a local path π^{loc} that connects the local start state $x_{\text{start}}^{\text{loc}}$ with the local goal state $x_{\text{goal}}^{\text{loc}}$, which along with the previously computed global path π produces the new tree $\mathcal{T}_{\text{new}}$.

A. Motion Planning Algorithm

The algorithmic framework consists of five phases, the offline computation of the global path planning; the online path tracking with game-theoretic learning; a terminal state evaluation framework; a static obstacle augmentation; and the local re-planning procedure.

The main routine is presented in Algorithm 1. Its subroutines can be found in [28]. The global graph $\mathcal{G}(V, E)$ is obtained offline by the `RRT*`, that provides all the TPBVPs with initial and final states. Next, we continue with the online implementation. The function `NoCollision` monitors if there exist a collision in the entire augmented obstacle space with the global path through the whole procedure and returns a binary value. The function `InitialDistance` calculates the distance of the initial and final state according to (4). Then, follows the online approximation of the optimal policy with full state feedback (lines 7-14). The function `Critic` estimates the critic parameters from (21). The function `EstimateQ` approximates the parameters of the $\hat{\mathcal{Q}}$ from (13). `DisturbanceActor` calculates the disturbance actor parameters from (23), that lead the `Disturbance` to justify the disturbance $\hat{\bar{d}}$ from (1). The `ControlActor` calculates the control actor parameters from (22), that lead the function `Control` to produce the control action $\hat{\bar{u}}$ from (14). Nest, we perform the terminal state evaluation (lines 15-20). The function `KinodynamicDistance` returns the deviation of the agent from the straight line that connects the initial and final states, by employing (25). The distance error is calculated by the function `DistanceError`, which allow the terminal state evaluation to proceed to the next problem. The primitive `Augment` inflates the obstacle space by comparing the maximum distance of the previously obtained deviations.

When a collision of the global path occurs with the augmented obstacle space, then the algorithm continues to the next phase of the online local re-planning. A critical aspect for the feasibility of the online implementation, is to perform the re-planning procedure sufficiently fast. Thus, we obtain the local free space with `LocalNodes`, that also provides the local free nodes for feasible local re-planning. Then, the `RRT*` computes the local path, yet in a reduced space. Lastly, the primitive `Connect` employs the global path and the locally established path, to find a safe tree $\mathcal{T}_{\text{new}}$ with respect to the kinodynamic constraints.

Algorithm 1 RRT_Q*

1: $V_{\text{free}} \leftarrow \emptyset$; $\mathcal{X}_{\text{obs}}^{\text{aug}} \leftarrow \mathcal{X}_{\text{obs}}$;
2: $\mathcal{G}, \pi \leftarrow$ `RRT*`$(\mathcal{G}, N, V_{\text{free}})$;
3: $D_{\text{rob}}^{\text{kin}} \leftarrow \emptyset$;
4: **while** `NoCollision`(π) **do**
5: **for** $i = 1$ to k **do**
6: $D_0 \leftarrow$`InitialDistance`(x_0);
7: **for** $t \in T$ **do**
8: $\hat{W}_{\text{c}} \leftarrow$ `Critic`$(x_{\text{r}}, M, R, \Delta t, P(T), \alpha_{\text{c}})$;
9: $\hat{\mathcal{Q}} \leftarrow$ `EstimateQ`$(\hat{W}_{\text{c}}, x_{\text{r}})$;
10: $\hat{W}_{\text{d}} \leftarrow$ `DisturbanceActor`$(x_{\text{r}}, \hat{\mathcal{Q}}, \alpha_{\text{d}})$;
11: $\hat{\bar{d}} \leftarrow$ `Disturbance`$(\hat{W}_{\text{d}}, x_{\text{r}})$;
12: $\hat{W}_{\text{a}} \leftarrow$ `ControlActor`$(x_{\text{r}}, \hat{\mathcal{Q}}, \alpha_{\text{a}})$;
13: $\hat{\bar{u}} \leftarrow$ `Control`$(\hat{W}_{\text{a}}, x_{\text{r}})$;
14: Return $\hat{\bar{u}}$;
15: $D_{\text{rob}} \leftarrow$ `KinodynamicDistance`(x_0, x_{r}, D_0);
16: $e_d \leftarrow$ `DistanceError`(D_0, x_{r});
17: **if** $e_d \leq \beta D_0$ **then**
18: $x_{0,i+1} \leftarrow x(t)$;
19: break;
20: **end if**
21: **end for**
22: **if** $D_{\text{rob}} > D_{\text{rob}}^{\text{kin}}$ **then**
23: $\mathcal{X}_{\text{obs}}^{\text{aug}} \leftarrow$ `Augment`$(\mathcal{X}_{\text{obs}})$; $D_{\text{rob}}^{\text{kin}} \leftarrow D_{\text{rob}}$;
24: **end if**
25: **end for**
26: **end while**
27: $V_{\text{free}} \leftarrow$ `LocalNodes`$(\pi, \mathcal{X}_{\text{obs}}^{\text{aug}}, \epsilon)$;
28: $\pi^{\text{loc}} \leftarrow$ `RRT*`(V_{free});
29: $\mathcal{T}_{\text{new}} \leftarrow$ `Connect`(π, π^{loc});
30: Return $\mathcal{T}_{\text{new}}$;

VI. Results and Discussion

In this section, we demonstrate the efficiency of the proposed online and robust kinodynamic motion planning technique. We also provide a qualitative comparison of our framework with other kinodynamic motion planning techniques.

A. Simulations

Consider now the Maxwell-slip model [29], where the robot slips on a frictioned flat surface. While the mass m is translating along the x-axis and the y-axis direction, a spring-damper system models the friction with coefficients k_{x}, c_{x}, and k_{y}, c_{y} respectively. The system is described by,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \\ \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = A \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & 0 \\ 0 & \frac{1}{m} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} d, \qquad (26)$$

with plant matrix,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{k_{\text{x}}}{m} & 0 & -\frac{c_{\text{x}}}{m} & 0 \\ 0 & -\frac{k_{\text{y}}}{m} & 0 & -\frac{c_{\text{y}}}{m} \end{bmatrix},$$

where x_1, y_1 are the translations (kinematic constraints), $\dot{x}_1 = x_2$, $\dot{y}_1 = y_2$ are the velocities (dynamic constraints), and $\ddot{x}_1 = \dot{x}_2$, $\ddot{y}_1 = \dot{y}_2$ are the accelerations (dynamic constraints). The vector $\begin{bmatrix} f_1 & f_2 \end{bmatrix}^{\intercal}$ is the input force.

We set the finite horizon $T = 10$ s for every run and the

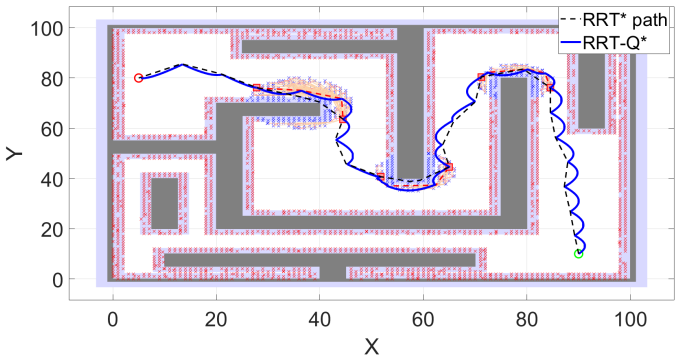

Fig. 3. The online kinodynamic motion planning framework with completely unknown dynamics and worst-case disturbance in a complex environment. The motion planning framework performs local re-planning at three areas according to the static obstacle augmentation and avoids obstacle collision.

admissible window $\beta = 5\%$. The user-defined matrices are $M = I$, and $R = 0.1I$ with proper dimensions for the identity matrix. The final Riccati matrix is $P(T) = 0.5I$ and the final control action is $u(T) = 0.001$. We set $\alpha_c = 50$, $\alpha_a = 4$, and $\alpha_a = 2.5$ by following Theorem 2. The small fixed value of the internal dynamics is $\Delta t = 0.05\,\text{s}$. The initial values of $\hat{W}_c$, $\hat{W}_d$, and $\hat{W}_a$ are randomly selected. The initial and the final set of states $\mathcal{X}_0$, $\mathcal{X}_G$ are given by the offline computation of the RRT*. The stiffness, and the damping coefficients are $k_x = k_y = 20\,\text{N/m}$, and $c_x = c_y = 45\,\text{kg/s}$ respectively. The mass is $m = 40\,\text{kg}$. The state space is described by the Cartesian space $\mathcal{X} \in [0, 100] \times [0, 100]$. We consider exact knowledge of the obstacle space $\mathcal{X}_{\text{obs}}$ and we require full state feedback. We compute offline the global path using RRT* with fixed incremental distance $\epsilon = 2$ and neighborhood radius $r = 10$.

The proposed framework efficiently performs robust, kinodynamic motion planning in challenging obstacle environments and with external disturbances, as depicted in Figure 3. The motion of the robot is illustrated with a blue solid line, the start state x_{start} with a green circle, the goal state x_{goal} with a red circle, and the global path π with a dashed black line. The red crosses represent the discarded nodes V_{aug} that are located in the augmented obstacle space $\mathcal{X}_{\text{obs}}^{\text{aug}}$. The inflated space is drawn with light purple. The local start state $x_{\text{start}}^{\text{loc}}$ and the local goal state $x_{\text{goal}}^{\text{loc}}$ are presented with red rectangles. The feasible local path π^{loc} is illustrated with a red dashed line. This performance reveals that the governing dynamics and the disturbances do not affect the performance of the proposed motion planning technique even in challenging obstacle environments.

B. Qualitative Comparison

In Table I we provide a qualitative comparison of the proposed technique with other kinodynamic motion planning works. We consider four specifications, the optimality; the online implementation; the robustness; and the system's model. We select optimality as a basis of this comparison, yet some approaches evaluate different performance. More specifically, the LQR-Trees, LQR-RRT*, and our approach solve the minimum-energy problem as given in (2), while Kinodynamic RRT* evaluates a minimum time-fuel performance. We also provide Theorem 2, that guarantees closed-loop stability of the equilibrium point. Minimum energy problems penalize the control and the states simultaneously, while minimum time-fuel problems penalize only the control. Thus, minimizing the energy corresponds to better performance [15]. Online implementation can be only achieved with the proposed framework, as it requires the computation of three simple gradient descent laws given by (21), (22), (23), and the local re-planning at a relatively small free space without any re-sampling. The other works need to solve the Riccati equation that inherits extensive offline computation and comprises the model of the system, which for the finite time horizon is a nonlinear PDE and must be integrated backwards in time. Therefore, the Ricatti equation yields extensive offline computation. The infinite horizon case is not ideal for motion planning, as the time is always a requirement for real systems. We consider robustness as a mean to reject disturbance. Our solution implements closed-loop feedback, considers disturbance with a two-player zero-sum game, and approximates the worst-case disturbance with an actor. Note that LQR-Trees and LQR-RRT* exploit closed-loop controllers, that provide some level of robustness, yet they do not explicitly model disturbances in the system. Kinodynamic RRT* does not reject disturbances and they make use of open-loop control. The proposed framework is model-free, as we approximate the optimal policy in (13) without any information of the system dynamics. To this end, our technique is suitable for any unmanned vehicle, while the other works require the system's model for their calculations.

TABLE I
KINODYNAMIC MOTION PLANNING COMPARISON

	LQR-Trees [3]	LQR-RRT* [9]	Kinodynamic RRT* [10]	Proposed Solution
Optimality	✓	✓	✓	✓
Online	✗	✗	✗	✓
Robustness	CL	CL	✗	✓
Model-Free	✗	✗	✗	✓

VII. CONCLUSION

This paper proposed an online kinodynamic motion planning framework. We employed a game-based Q-learning approach to approximate the optimal policy for the worst-case disturbance of a continuous linear system with a finite horizon performance. We discussed the mathematical formulation that guarantees asymptotic stability and optimality of kinodynamic motion planning for systems with completely unknown dynamics. We also presented the algorithmic framework, we proposed a terminal state evaluation that reduces significantly the computational effort, and a static obstacle augmentation along with and a local re-planning framework that to facilitates the online and collision-free implementation. Simulation examples validated the efficiency of the proposed framework and a qualitative comparison outlined the benefits of our approach.

Future research efforts will focus on the extension of the

online kinodynamic motion planning algorithm in 3D underwater environments with moving obstacles.

REFERENCES

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

[2] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation*, vol. 2, 2000, pp. 995–1001.

[3] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "LQR-Trees: Feedback motion planning via sums-of-squares verification," *The Int. Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[4] 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.

[5] R. Allen and M. Pavone, "A real-time framework for kinodynamic planning with application to quadrotor obstacle avoidance," in *AIAA Guidance, Navigation, and Control Conference*, 2016, p. 1374.

[6] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 528–564, 2016.

[7] F. Berkenkamp and A. P. Schoellig, "Safe and robust learning control with Gaussian processes," in *Europ. Con. Conf.*, 2015, pp. 2496–2501.

[8] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The Int. J. of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[9] 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 *IEEE International Conference on Robotics and Automation*, 2012, pp. 2537–2542.

[10] D. J. Webb and J. van den Berg, "Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics," in *IEEE Int. Conference on Robotics and Automation*, 2013, pp. 5054–5061.

[11] S. M. LaValle, "Robot motion planning: A game-theoretic foundation," *Algorithmica*, vol. 26, no. 3-4, pp. 430–465, 2000.

[12] S. M. Lavalle, "A game-theoretic framework for robot motion planning," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1995.

[13] J. Ding, E. Li, H. Huang, and C. J. Tomlin, "Reachability-based synthesis of feedback policies for motion planning under bounded disturbances," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 2160–2165.

[14] P. A. Ioannou and J. Sun, *Robust Adaptive Control*. Courier Corporation, 2012.

[15] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*, 3rd ed. John Wiley & Sons,, 2012.

[16] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[17] W. B. Powell, *Approximate Dynamic Programming: Solving the curses of dimensionality*. John Wiley & Sons, 2007, vol. 703.

[18] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2010, vol. 39.

[19] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Systems*, vol. 32, no. 6, pp. 76–105, 2012.

[20] D. Vrabie, K. G. Vamvoudakis, and F. L. Lewis, *Optimal adaptive control and differential games by reinforcement learning principles*. IET, 2013, vol. 2.

[21] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 6, pp. 2042–2062, 2018.

[22] C. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[23] P. Mehta and S. Meyn, "Q-learning and pontryagin's minimum principle," in *IEEE Conf. on Decision and Control*, 2009, pp. 3598–3605.

[24] K. G. Vamvoudakis, "Q-learning for continuous-time linear systems: A model-free infinite horizon optimal control approach," *Systems & Control Letters*, vol. 100, pp. 14–20, 2017.

[25] T. Başar and P. Bernhard, *H-infinity optimal control and related minimax design problems: A dynamic game approach*. Springer Science & Business Media, 2008.

[26] J. P. Hespanha, *Noncooperative Game Theory: An introduction for Engineers and Computer Scientists*. Princeton Press, 2017.

[27] A. J. Van Der Schaft, "L_2-gain analysis of nonlinear systems and nonlinear state-feedback H_∞ control," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 770–784, 1992.

[28] G. P. Kontoudis and K. G. Vamvoudakis, "Kinodynamic motion planning with continuous-time Q-learning: An online, model-free, and safe navigation framework," *IEEE Transactions on Neural Networks and Learning Systems*, 2019.

[29] F. Al-Bender, V. Lampaert, and J. Swevers, "The generalized Maxwell-slip model: A novel model for friction simulation and compensation," *IEEE Tr. on Automatic Control*, vol. 50, no. 11, pp. 1883–1887, 2005.

[30] H. Khalil, *Nonlinear Systems*. Prentice Hall, 2002.

APPENDIX

For the theoretical analysis we define the weight estimation error for the critic $\tilde{W}_\mathrm{c} \coloneqq W_\mathrm{c} - \hat{W}_\mathrm{c}$, for the control actor $\tilde{W}_\mathrm{a} \coloneqq W_\mathrm{a} - \hat{W}_\mathrm{a}$, and for the disturbance actor $\tilde{W}_\mathrm{d} \coloneqq W_\mathrm{d} - \hat{W}_\mathrm{d}$. The estimation error dynamics of the critic yields,

$$\dot{\tilde{W}}_\mathrm{c} = -\alpha_\mathrm{c} \frac{1}{(1+\sigma^\intercal\sigma)^2}\sigma\sigma^\intercal\tilde{W}_\mathrm{c}.$$

The estimation error dynamics of the control actor becomes,

$$\dot{\tilde{W}}_\mathrm{a} = -\alpha_\mathrm{a}\bar{x}\bar{x}^\intercal\mu(t)^\intercal\tilde{W}_\mathrm{a} - \alpha_\mathrm{a}\bar{x}\bar{x}^\intercal\frac{\mu(t)\tilde{Q}_\mathrm{xu}R^{-1}}{\|1+\mu(t)^\intercal\mu(t)\|^2}.$$

Next, the estimation error dynamics of the disturbance yields,

$$\dot{\tilde{W}}_\mathrm{d} = -\alpha_\mathrm{d}\bar{x}\bar{x}^\intercal\xi(t)^\intercal\tilde{W}_\mathrm{d} - \alpha_\mathrm{d}\bar{x}\bar{x}^\intercal\frac{\xi(t)\tilde{Q}_\mathrm{xd}(-\gamma)^{-2}}{\|1+\xi(t)^\intercal\xi(t)\|^2},$$

where $Q_\mathrm{dd} = -\gamma^{-2}$ and $Q_\mathrm{uu} = R$.

Lemma 3: Considering any control input $u(t) \in \mathcal{U}$, then the estimation error dynamics of the critic $\tilde{W}_\mathrm{c}$ has an exponentially stable equilibrium point at the origin that is bounded by, $\|\tilde{W}_\mathrm{c}\| \leq \|\tilde{W}_\mathrm{c}(t_0)\|\kappa_1 \mathrm{e}^{-\kappa_2(t-t_0)}$, where $\kappa_1, \kappa_2 \in \mathbb{R}^+$. The signal $\Delta(t) \coloneqq \frac{\sigma(t)}{1+\sigma(t)^\intercal\sigma(t)}$ needs to be persistently exciting (PE) at $[t, t+T_\mathrm{PE}]$, where $T_\mathrm{PE} \in \mathbb{R}^+$ the excitation period and if there exists a $\beta \in \mathbb{R}^+$ such that $\beta\mathrm{I} \leq \int_t^{t+T_\mathrm{PE}} \Delta(\tau)\Delta(\tau)^\intercal \mathrm{d}\tau$.

Proof. The proof follows from [24]. ∎

Proof of Theorem 2. Consider the Lyapunov function,

$$\mathcal{L}(\psi;t) = V^\star(\bar{x};t) + \frac{1}{2}\|\tilde{W}_\mathrm{c}\|^2 + \frac{1}{2}\mathrm{tr}\{\tilde{W}_\mathrm{a}^\intercal\tilde{W}_\mathrm{a}\} + \frac{1}{2}\mathrm{tr}\{\tilde{W}_\mathrm{d}^\intercal\tilde{W}_\mathrm{d}\} > 0,$$

for all $t \geq 0$, where $\psi = [\bar{x}^\intercal \ \tilde{W}_\mathrm{c}^\intercal \ \tilde{W}_\mathrm{a}^\intercal \ \tilde{W}_\mathrm{d}^\intercal]^\intercal$ is the augmented state, and $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{\frac{(n+m+q)(n+m++q1)}{2}} \times \mathbb{R}^{n\times m} \to \mathbb{R}$. The orbital derivative for the closed-loop dynamics by using $\hat{u}$ yields, $\dot{\mathcal{L}} = T_1 + T_2 + T_3 + T_4$ where,

$$T_1 = \bar{x}^\intercal\dot{P}(t)\bar{x} + \bar{x}^\intercal P(t)(A\bar{x} + B\hat{\bar{u}}), \tag{27}$$

$$T_2 = -\alpha_\mathrm{c}\tilde{W}_\mathrm{c}^\intercal\frac{1}{(1+\sigma^\intercal\sigma)^2}\sigma\sigma^\intercal\tilde{W}_\mathrm{c},$$

$$T_3 = -\alpha_\mathrm{a}\mathrm{tr}\Big\{\tilde{W}_\mathrm{a}^\intercal\bar{x}\bar{x}^\intercal\mu(t)^\intercal\tilde{W}_\mathrm{a} - \tilde{W}_\mathrm{a}^\intercal\bar{x}\bar{x}^\intercal\frac{\mu(t)\tilde{Q}_\mathrm{xu}R^{-1}}{\|1+\mu(t)^\intercal\mu(t)\|^2}\Big\}.$$

$$T_4 = -\alpha_\mathrm{d}\mathrm{tr}\Big\{\tilde{W}_\mathrm{d}^\intercal\bar{x}\bar{x}^\intercal\xi(t)^\intercal\tilde{W}_\mathrm{d} + \tilde{W}_\mathrm{d}^\intercal\bar{x}\bar{x}^\intercal\frac{\xi(t)\tilde{Q}_\mathrm{xd}(-\gamma)^{-2}}{\|1+\xi(t)^\intercal\xi(t)\|^2}\Big\}.$$

It is easy to see that,

$$T_2 \leq -\frac{\alpha_{\mathrm{c}}}{4}\|\tilde{W}_{\mathrm{c}}\|^2, \tag{28}$$

$$T_3 \leq -\frac{\alpha_{\mathrm{a}}}{2}\|\bar{x}^\intercal\sqrt{\mu(t)}^\intercal\tilde{W}_{\mathrm{a}}\|^2+\frac{\alpha_{\mathrm{a}}\delta_1}{2}\overline{\lambda}\Big(\frac{\mu(t)R^{-1}}{\|1+\mu(t)^\intercal\mu(t)\|^2}\Big)\|\bar{x}\|^2, \tag{29}$$

$$T_4 \leq -\frac{\alpha_{\mathrm{d}}}{2}\|\bar{x}^\intercal\sqrt{\xi(t)}^\intercal\tilde{W}_{\mathrm{d}}\|^2+\frac{\alpha_{\mathrm{d}}\delta_2}{2}\overline{\lambda}\Big(\frac{\xi(t)(-\gamma)^{-2}}{\|1+\xi(t)^\intercal\xi(t)\|^2}\Big)\|\bar{x}\|^2. \tag{30}$$

The estimated control action $\hat{u}$ and disturbance $\hat{\bar{d}}$ result,

$$\begin{aligned}\hat{u} =&\hat{W}_{\mathrm{a}}^\intercal\mu(t)\bar{x} &(31)\\ =&-(Q_{\mathrm{xu}}Q_{\mathrm{uu}}^{-1}+\mu(t)^\intercal\tilde{W}_{\mathrm{a}})^\intercal\bar{x}\\ =&-Q_{\mathrm{uu}}^{-1}Q_{\mathrm{ux}}\bar{x}-\tilde{W}_{\mathrm{a}}^\intercal\mu(t)\bar{x}\\ =&\bar{u}^\star-\tilde{W}_{\mathrm{a}}^\intercal\mu(t)\bar{x}, &(32)\\ & &(33)\end{aligned}$$

and similarly the estimated disturbance becomes,

$$\hat{\bar{d}} =\bar{d}^\star-\tilde{W}_{\mathrm{d}}^\intercal\xi(t)\bar{x}. \tag{34}$$

Using the Riccati equation (8), the estimated control and disturbance (31), (33), and Young's inequality, the (27) yields,

$$\begin{aligned}T_1 \leq -\Big(&\frac{1}{2}\underline{\lambda}(M+Q_{\mathrm{xu}}R^{-1}Q_{\mathrm{xu}}^\intercal-\gamma^2Q_{\mathrm{xd}}Q_{\mathrm{xd}}^\intercal)-\frac{1}{2}\overline{\lambda}(Q_{\mathrm{xu}}Q_{\mathrm{xu}}^\intercal)\\ &-\frac{1}{2}\overline{\lambda}(Q_{\mathrm{xd}}^\intercal Q_{\mathrm{xd}})\Big)\|\bar{x}\|^2. &(35)\end{aligned}$$

Next, from (28), (29), (30), and (35) we obtain,

$$\begin{aligned}\dot{\mathcal{L}}(\psi,t) \leq -\Big[&\frac{1}{2}\underline{\lambda}(M+Q_{\mathrm{xu}}R^{-1}Q_{\mathrm{xu}}^\intercal-\gamma^2Q_{\mathrm{xd}}Q_{\mathrm{xd}}^\intercal)-\frac{1}{2}\overline{\lambda}(Q_{\mathrm{xd}}Q_{\mathrm{xd}}^\intercal)\\ &-\frac{\alpha_{\mathrm{a}}\delta_1}{2}\overline{\lambda}\Big(\frac{\mu(t)R^{-1}}{\|1+\mu(t)^\intercal\mu(t)\|^2}\Big)-\frac{\alpha_{\mathrm{d}}\delta_2}{2}\overline{\lambda}\Big(\frac{\xi(t)(-\gamma)^{-2}}{\|1+\xi(t)^\intercal\xi(t)\|^2}\Big)\\ &-\frac{1}{2}\overline{\lambda}(Q_{\mathrm{xu}}^\intercal Q_{\mathrm{xu}})\Big]\|\bar{x}\|^2-\frac{\alpha_{\mathrm{c}}}{4}\|\tilde{W}_{\mathrm{c}}\|^2-\frac{\alpha_{\mathrm{a}}}{2}\|\bar{x}^\intercal\sqrt{\mu(t)}^\intercal\tilde{W}_{\mathrm{a}}\|^2\\ &-\frac{\alpha_{\mathrm{d}}}{2}\|\bar{x}^\intercal\sqrt{\xi(t)}^\intercal\tilde{W}_{\mathrm{d}}\|^2= W_3(\psi;t). &(36)\end{aligned}$$

Considering the inequality in (2) then $\dot{\mathcal{L}}(\psi;t)$ is non-positive for all ψ and $t \geq t_0$. From $W_1(\psi;t) = W_2(\psi;t) = V^\star(\bar{x},t)+\frac{1}{2}\|\tilde{W}_{\mathrm{c}}\|^2+\frac{1}{2}\mathrm{tr}\{\tilde{W}_{\mathrm{a}}^\intercal\tilde{W}_{\mathrm{a}}\}+\frac{1}{2}\mathrm{tr}\{\tilde{W}_{\mathrm{d}}^\intercal\tilde{W}_{\mathrm{d}}\} > 0$, we get $W_1(\psi;t) \leq \mathcal{L}(\psi;t) \leq W_2(\psi;t)$. In this way, we can conclude that the origin $\psi_{\mathrm{e}} = 0$ is uniformly stable according to the Lyapunov stability theorem. Since $\mathcal{L}(\psi;t)$ is lower-bounded and non-increasing, inequality (36) is also bounded, which implies that $\mathcal{L}(\psi;t)$ is uniformly continuous. According to Barbalat's lemma, $\mathcal{L}(\psi;t) \to 0$ as $t \to \infty$. Since $W_3(\psi;t)$ is positive definite, so asymptotic stability holds from the Lyapunov stability theorem. Next, $W_1(\psi;t)$ is radially unbounded with respect to $\|\bar{x}\|$, $\|\tilde{W}_{\mathrm{c}}\|$, $\|\tilde{W}_{\mathrm{a}}\|$ $\|\tilde{W}_{\mathrm{d}}\|$ and globally properties also hold. Therefore, the equilibrium at the origin $\psi_{\mathrm{e}} = 0$ of the closed-loop system is globally uniformly asymptotically stable [30]. ■