Sampling-based Motion Planning for Uncertain Nonlinear Systems via Funnel Control

Christos K. Verginis[1], Dimos V. Dimarogonas[2], and Lydia E. Kavraki[3]

***Abstract*— We integrate sampling-based planning techniques with funnel-based feedback control to develop KDF, a new framework for solving the kinodynamic motion-planning problem via funnel control. The considered systems evolve subject to complex, nonlinear, and uncertain dynamics (aka differential constraints). Firstly, we use a *geometric* planner to obtain a high-level safe path in a user-defined extended free space. Secondly, we develop a low-level funnel control algorithm that guarantees safe tracking of the path by the system. Neither the planner nor the control algorithm use information of the underlying dynamics of the system, which makes the proposed scheme easily distributable to a large variety of different systems and scenarios. Intuitively, the funnel control module is able to implicitly accommodate the dynamics of the system, allowing hence the deployment of purely geometrical motion planners. Extensive computer simulations and experimental results with a 6-DOF robotic arm validate the proposed approach.**

I. INTRODUCTION

Motion planning of autonomous systems is one of the most fundamental problems in robotics, with numerous applications such as exploration, autonomous driving, robotic manipulation, autonomous warehouses, and multi-robot coordination [1], [2]. It has been extensively studied in the related literature; works have been continuously developed for the last three decades, exploring plenty of variations, including feedback control, discrete planning, uncertain environments, and multi-agent systems. One important and active area of research consists of *kinodynamic* motion planning, i.e., when the planning algorithm takes into account the underlying system dynamics, also known as differential constraints [1].

In this paper we develop KDF, an algorithmic framework for the kinodynamic motion-planning problem by integrating sampling-based algorithms with intelligent feedback control. We consider systems that evolve subject to high-dimensional dynamics, which are highly nonlinear and uncertain. Firstly, we develop a funnel-based, feedback-control scheme that achieves confinement of the system state around a given trajectory within *user-defined* bounds, without using any knowledge of the system dynamics. Next, we use these bounds to create a family of *geometric* sampling-based motion planners in an extended free space. A motion-planning query is solved then by obtaining a safe geometric path in this free space, and using the developed feedback-control scheme to track it. The proposed framework guarantees that the system will follow the derived path without colliding with workspace obstacles. Loosely speaking, we augment geometric motion planning algorithms with extended free-space capabilities and intelligent feedback control to provide a new solution to the kinodynamic motion-planning problem. The incorporation of the control scheme relieves the sampling-based motion planner from the system dynamics and their uncertainties.

This work was supported by the H2020 ERC Starting Grant BUCOPHSYS, the European Union's Horizon 2020 Research and Innovation Programme under the GA No. 731869 (Co4Robots), the Swedish Research Council (VR), the Knut och Alice Wallenberg Foundation (KAW), the Swedish Foundation for Strategic Research (SSF), and the National Science Foundation project NSF 2008720 (LEK).

[1]C. K. Verginis is with the Oden Institute for Computational Engineering and Sciences at the University of Texas at Austin, TX, USA christos.verginis@utexas.edu.

[2]D. V. Dimarogonas is with the School of Electrical and Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden dimos@kth.se.

[3]L. E. Kavraki is with the Department of Computer Science at Rice University, Houston, TX, USA kavraki@rice.edu.

II. PROBLEM FORMULATION

Consider a robotic system characterized by the configuration vector $q_1 \in \mathbb{T}\times \subset \mathbb{R}^n$, $n \in \mathbb{N}$. Usual robotic structures (e.g., robotic manipulators) might consist of translational and rotational joints, which we define here as $q^{\mathfrak{t}} = [q_1^{\mathfrak{t}}, \dots, q_{n_{tr}}^{\mathfrak{t}}]^\top \in \mathbb{R}^{n_{tr}}$ and $q^{\mathfrak{r}} = [q_1^{\mathfrak{r}}, \dots, q_{n_r}^{\mathfrak{r}}]^\top \in [0, 2\pi)^{n_r}$, respectively, with $n_{tr} + n_r = n$, and hence $\mathbb{T} \coloneqq \mathcal{W}_{tr} \times [0, 2\pi)^{n_r}$, where $\mathcal{W}_{tr}$ is a closed subset of $\mathbb{R}^{n_{tr}}$. Without loss of generality, we assume that $q_1 = [(q^{\mathfrak{t}})^\top, (q^{\mathfrak{r}})^\top]^\top$.

We consider kth-order systems, with $k \geq 2$, of the form

$$\dot{q}_i = f_i(\bar{q}_i, t) + g_i(\bar{q}_i, t)q_{i+1}, \quad \forall i \in \{1, \dots, k-1\} \tag{1a}$$

$$\dot{q}_k = f_k(\bar{q}_k, t) + g_k(\bar{q}_k, t)u, \tag{1b}$$

where $\bar{q}_i \coloneqq [q_1^\top, \dots, q_i^\top]^\top \in \mathbb{T} \times \mathbb{R}^{n(i-1)}$, $\forall i \in \{1, \dots, k\}$, and $u \in \mathbb{R}^n$ is the control input of the system; $q_2, \dots, q_k$ represent high-order variables of the system (e.g., generalized velocities, accelerations, etc). Note that the kth-order model (1) generalizes the simpler 2nd-order Lagrangian dynamics, which is commonly used in the related literature. The vector fields f_i, g_i, which represent various terms in robotic systems (inertia, Coriolis, friction, gravity, centrifugal) are considered to be completely unknown to the designer/planner, $\forall i \in \{1, \dots, k\}$.

We consider that the robot operates in a workspace $\mathcal{W} \subset \mathbb{R}^3$ filled with obstacles occupying a closed set $\mathcal{O} \subset \mathbb{R}^3$. We denote the set of points that consist the volume of the robot at configuration q_1 as $\mathcal{A}(q_1) \subset \mathbb{R}^3$. The collision-free space is defined as the open set $\mathcal{A}_{\text{free}} \coloneqq \{q_1 \in \mathbb{T} : \mathcal{A}(q_1) \cap \mathcal{O} = \emptyset\}$. Our goal is to achieve safe navigation of the robot to a predefined goal region $Q_g \subset \mathcal{A}_{\text{free}}$ from an initial configuration $q_1(0) \in \mathcal{A}_{\text{free}}$ via a path $\boldsymbol{q}_{\text{p}} : [0, \sigma] \to \mathcal{A}_{\text{free}}$ satisfying $\boldsymbol{q}_{\text{p}}(0) = q_1(0)$ and $\boldsymbol{q}_{\text{p}}(\sigma) \in Q_g$, for some positive σ.

The problem we consider is the following:

Problem 1: Given $q(0)) \in \mathcal{A}_{\text{free}}$ and $Q_g \subset \mathcal{A}_{\text{free}}$, respectively, design a control trajectory $u : [0, t_f] \to \mathbb{R}^n$, for some finite $t_f > 0$, such that the solution $q^*(t)$ of (1) satisfies $q^*(t) \in \mathcal{A}_{\text{free}}$, $\forall t \in [0, t_f]$, and $q^*(t_f) \in Q_g$.

III. Main Results

We present here the proposed solution for Problem 1. Our methodology follows a two-layer approach, consisting of a robust trajectory-tracking control design and a higher-level sampling-based motion planner. Firstly, we design an adaptive control protocol that compensates for the uncertain dynamical parameters of the robot and forces the system to evolve in a funnel around a desired trajectory, whose size can be a priori chosen by the user/designer, and is completely independent from the system dynamics (1). Secondly, we develop a geometric sampling-based motion planner that uses this funnel to find a collision free trajectory from the initial to the goal configuration. Intuitively, the robust control design helps the motion planner procedure, which does not have to take into account the complete dynamics (1).

A. Control Design

Let $q_{\text{d}} := [(q_{\text{d}}^{\text{t}})^\top, (q_{\text{d}}^{\text{r}})^\top]^\top := [q_{\text{d}_1}^{\text{t}}, \dots, q_{\text{d}_{n_{tr}}}^{\text{t}}, q_{\text{d}_1}^{\text{r}}, \dots, q_{\text{d}_{n_r}}^{\text{r}}]^\top : [t_0, t_0 + t_f] \to \mathbb{T}$ be a smooth (at least k-times continuously differentiable) reference trajectory, with $q_{\text{d}}^{\text{t}} \in \mathbb{R}^{n_{tr}}$ and $q_{\text{d}}^{\text{r}} \in [0, 2\pi)^{n_r}$ being its translational and rotational parts, respectively. Such a trajectory will be the output of the sampling-based motion-planning algorithm that will be developed in the next section.

We wish to design the control input u of (1) such that $q(t)$ converges to $q_{\text{d}}(t)$, despite the unknown terms f_i, g_i. We start by defining the appropriate error metric between q_1 and q_{d}, which represents their distance. Regarding the translational part, we define the standard Euclidean error $e^{\text{t}} := q^{\text{t}} - q_{\text{d}}^{\text{t}}$. For the rotation part, however, the the same error $e^{\text{r}} := q^{\text{r}} - q_{\text{d}}^{\text{r}}$ does not represent the minimum distance metric, since q^{r} evolves on the n_r-dimensional sphere. Hence, we use the chordal metric $d_C(x) := 1 - \cos(x) \in [0, 2], \forall x \in [0, 2\pi)$, extended for vector arguments $x = [x_1, \dots, x_{\text{n}}] \in [0, 2\pi)^{\text{n}}$ to $\bar{d}_C(x) := \sum_{\ell \in \{1,\dots,\text{n}\}} d_C(x_j)$, and introduce

$$\eta_\ell^{\text{r}} := d_C(e_\ell^{\text{r}}) = 1 - \cos(e_\ell^{\text{r}}),$$

where $e_\ell^{\text{r}} := q_\ell^{\text{r}} - q_{\text{d}_\ell}^{\text{r}}$ is the ℓth element of e^{r}, $\ell \in \{1, \dots, n_r\}$.

The funnel is defined by the functions $\rho_j^{\text{t}} : [t_0, t_0 + t_f] \to [\underline{\rho}_j^{\text{t}}, \bar{\rho}_j^{\text{t}}]$, $\rho_\ell^{\text{r}} : [t_0, t_0 + T] \to [\underline{\rho}_\ell^{\text{r}}, \bar{\rho}_\ell^{\text{r}}]$ with

$$\rho_j^{\text{t}}(t_0) = \bar{\rho}_j^{\text{t}}, \rho_\ell^{\text{r}}(t_0) = \bar{\rho}_\ell^{\text{r}} \tag{2a}$$
$$\rho_j^{\text{t}}(t_0 + t_f) = \underline{\rho}_j^{\text{t}}, \rho_\ell^{\text{r}}(t_0 + t_f) = \underline{\rho}_\ell^{\text{r}} \tag{2b}$$
$$0 < \underline{\rho}_j^{\text{t}} \leq \bar{\rho}_j^{\text{t}}, 0 < \underline{\rho}_\ell^{\text{r}} \leq \bar{\rho}_\ell^{\text{r}} < 2 \tag{2c}$$
$$|e_j^{\text{t}}(t_0)| < \bar{\rho}_j^{\text{t}}, \eta_\ell^{\text{r}}(t_0) < \bar{\rho}_\ell^{\text{r}} \tag{2d}$$

$\forall j \in \{1, \dots, n_{tr}\}$, $\ell \in \{1, \dots, n_r\}$. Our aim is to design a control protocol such that

$$|e_j^{\text{t}}(t)| < \rho_j^{\text{t}}(t), \quad \forall j \in \{1 \dots, n_{tr}\}, \tag{3a}$$
$$\eta_\ell^{\text{r}}(t) < \rho_\ell^{\text{r}}(t), \quad \forall \ell \in \{1 \dots, n_r\}, \tag{3b}$$

$\forall t \in [t_0, t_0 + t_f]$. The funnel functions can be defined a priori by a user, specifying the performance of the system in terms of overshoot and steady-state value of the errors e_j^{t}, e_ℓ^{r}.

Let us define first the normalized and transformed signals

$$\xi_j^{\text{t}} := \frac{e_j^{\text{t}}}{\rho_j^{\text{t}}}, \quad \xi_\ell^{\text{r}} := \frac{\eta_\ell^{\text{r}}}{\rho_\ell^{\text{r}}}, \tag{4a}$$
$$\varepsilon_j^{\text{t}} := \ln\left(\frac{1 + \xi_j^{\text{t}}}{1 - \xi_j^{\text{t}}}\right), \quad \varepsilon_\ell^{\text{r}} := \ln\left(\frac{1}{1 - \xi_\ell^{\text{r}}}\right), \tag{4b}$$
$$r_j^{\text{t}} := \frac{\partial \varepsilon_j^{\text{t}}}{\partial \xi_j^{\text{t}}} \quad r_\ell^{\text{r}} := \frac{\partial \varepsilon_\ell^{\text{r}}}{\partial \xi_\ell^{\text{r}}}, \tag{4c}$$

for $j \in \{1 \dots, n_{tr}\}$, $\ell \in \{1 \dots, n_r\}$, and set $\alpha_1 := [(\alpha^{\text{t}})^\top, (\alpha^{\text{r}})^\top]^\top$, with

$$\alpha^{\text{t}} := -K^{\text{t}} \widetilde{r}^{\text{t}} (\widetilde{\rho}^{\text{t}})^{-1} \varepsilon^{\text{t}} \tag{5a}$$
$$\alpha^{\text{r}} := -K^{\text{r}} \widetilde{s}^{\text{r}} (\widetilde{\rho}^{\text{r}})^{-1} r^{\text{r}}, \tag{5b}$$

where $K^{\text{t}} := \text{diag}\{[k_j^{\text{t}}]_{j \in \{1,\dots,n_{tr}\}}\} \in \mathbb{R}^{n_{tr} \times n_{tr}}$, $K_{\text{r}} := \text{diag}\{[k_\ell^{\text{r}}]_{\ell \in \{1,\dots,n_r\}}\} \in \mathbb{R}^{n_r \times n_r}$ are diagonal positive definite gain matrices, $\widetilde{r}^{\text{t}} := \text{diag}\{[r_j^{\text{t}}]_{j \in \{1,\dots,n_{tr}\}}\}$, $r^{\text{r}} := [r_1^{\text{r}}, \dots, r_{n_r}^{\text{r}}]$, $\widetilde{\rho}^{\text{t}} := \text{diag}\{[\rho_j^{\text{t}}]_{j \in \{1,\dots,n_{tr}\}}\}$, $\widetilde{\rho}^{\text{r}} := \text{diag}\{[\rho_\ell^{\text{r}}]_{\ell \in \{1,\dots,n_r\}}\}$, $\varepsilon^{\text{t}} := [\varepsilon_1^{\text{t}}, \dots, \varepsilon_{n_{tr}}^{\text{t}}]^\top$, and $\widetilde{s}^{\text{r}} := \text{diag}\{[\sin(e_\ell^{\text{r}})]_{\ell \in \{1,\dots,n_r\}}\}$.

The rest of the algorithm proceeds recursively: for $i \in \{2, \dots, k\}$, we define the error $e_i = [e_{i_1}, \dots, e_{i_n}]^\top$, with

$$e_i := q_i - \alpha_{i-1} \in \mathbb{R}^n, \tag{6}$$

where α_{i-1} will be given subsequently in (8). We design funnel functions $\rho_{i_m} : [t_0, t_0 + t_f] \to [\underline{\rho}_{i_m}, \bar{\rho}_{i_m}]$, $\underline{\rho}_{i_m} \leq \bar{\rho}_{i_m}$, such that $\rho_{i_m}(t_0) = \bar{\rho}_{i_m} > |e_{i_m}(t_0)|$[1], $\forall m \in \{1, \dots, n\}$, and define

$$\xi_i := \rho_i^{-1} e_i, \tag{7a}$$
$$\varepsilon_i := \left[\ln\left(\tfrac{1+\xi_{i_1}}{1-\xi_{i_1}}\right), \dots, \ln\left(\tfrac{1+\xi_{i_n}}{1-\xi_{i_n}}\right)\right]^\top \tag{7b}$$
$$r_i := \text{diag}\left\{\left[\frac{\partial \varepsilon_{i_m}}{\partial \xi_{i_m}}\right]_{m \in \{1,\dots,n\}}\right\}, \tag{7c}$$

where $\rho_i := \text{diag}\{[\rho_{i_m}]_{i \in \{1,\dots,n\}}\} \in \mathbb{R}^{n \times n}$. Finally, we design the intermediate reference signals as

$$\alpha_i := -K_i \rho_i^{-1} r_i \varepsilon_i, \forall i \in \{2, \dots, k-1\}, \tag{8}$$

and the control law

$$u = -K_k \rho_k^{-1} r_k \varepsilon_k, \tag{9}$$

where $K_i \in \mathbb{R}^{n \times n}$, $i \in 2, \dots, k$, are positive definite gain matrices. We can guarantee the correctness of the proposed protocol under mild assumptions on the positive definiteness of g_i, $i \in \{1, \dots, k\}$ [3].

[1]Note that $e_{i_m}(t_0)$ can be measured at the time instant t_0 and the functions ρ_{i_m} can be designed accordingly.

B. Motion Planner

We introduce now the framework of *KinoDynamic motion planning via Funnel control*, or *KDF* motion-planning framework; The framework uses the control design of Section III-A to augment geometric sampling-based motion-planning algorithms and solve the kinodynamic motion-planning problem.

Before presenting the framework, we define the extended-free space, which will be used to integrate the results from the feedback control of the previous subsection. In order to do that, we define first the open polyhedron as

$$\begin{aligned}\mathcal{P}(\mathsf{z},\bar{\rho}) \coloneqq \{\mathsf{y}\in\mathbb{T} :& |\mathsf{y}_j^{\mathfrak{t}}-\mathsf{z}_j^{\mathfrak{t}}|<\bar{\rho}_j^{\mathfrak{t}},\\ & 1-\cos(\mathsf{y}_\ell^{\mathfrak{r}}-\mathsf{z}_\ell^{\mathfrak{r}})<\bar{\rho}_\ell^{r},\\ & \forall j\in\{1,\dots,n_{tr}\},\ell\in\{1,\dots,n_r\}\} \end{aligned} \tag{10}$$

where $\mathsf{y},\mathsf{z}\in\mathbb{T}$ consist of translational and rotational terms (similarly to q_1), and $\bar{\rho}\coloneqq[\bar{\rho}_1^{\mathfrak{t}},\dots,\bar{\rho}_{n_{tr}}^{\mathfrak{t}},\bar{\rho}_1^{\mathfrak{r}},\dots,\bar{\rho}_{n_r}^{\mathfrak{r}}]^\top\in\mathbb{R}^{n_{tr}+n_r}$ is the vector of maximum funnel values. We define now the $\bar{\rho}$-extended free space

$$\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})\coloneqq\{\mathsf{z}\in\mathbb{T}:\bar{\mathcal{A}}(\mathsf{z},\bar{\rho})\cap\mathcal{O}=\emptyset\}, \tag{11}$$

where $\bar{\mathcal{A}}(\mathsf{z},\bar{\rho})\coloneqq\bigcup_{\mathsf{y}\in\mathcal{P}(\mathsf{z},\bar{\rho})}\mathcal{A}(\mathsf{y})$. In addition, define the distance metric for vectors $x=[(x^{\mathfrak{t}})^\top,(x^{\mathfrak{r}})^\top]^\top$, $y=[(y^{\mathfrak{t}})^\top,(y^{\mathfrak{r}})^\top]^\top\in\mathbb{T}$ as $d_{\mathbb{T}}:\mathbb{T}^2\to\mathbb{R}_{\geq 0}$, with

$$d_{\mathbb{T}}(x,y)=\|x^{\mathfrak{t}}-y^{\mathfrak{r}}\|^2+\bar{d}_C(x^{\mathfrak{r}}-y^{\mathfrak{r}}), \tag{12}$$

which captures accurately the proximity of $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})\subset\mathbb{T}$.

The intuition behind the KDF framework is as follows. The control scheme of the previous subsection guarantees that the robot can track a trajectory within the prescribed funnel bounds. In other words, given a desired trajectory signal $q_{\text{d}}:[t_0,t_0+t_f]\to\mathbb{T}$, the control algorithm (4)-(9) guarantees that $q_1(t)\in\mathcal{P}(q_{\text{d}}(t),\bar{\rho})$, for all $t\in[t_0,t_0+t_f]$. Therefore, by the construction of $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$, if $q_{\text{d}}(t)$ belongs to $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$, $q_1(t)$ belongs to $\mathcal{A}_{\text{free}}$. The proposed sampling-based framework aims at finding a path in $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$; this path will be then endowed with time constraints in order to form the trajectory $q_{\text{d}}:[t_0,t_0+t_f]\to\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$, which will then safely tracked by the system using the designed controller.

Common geometric sampling-based motion-planning algorithms follow a standard iterative procedure that builds a discrete network $\mathcal{G}=(\mathcal{V},\mathcal{E})$, (tree, roadmap) of points in the free space connecting the initial configuration to the goal; $\mathcal{V}$ and $\mathcal{E}$ denote the nodes (points) and edges, respectively, of the network. The most common functions are $\mathsf{Sample}()$, and $\mathsf{ObstacleFree}(\mathsf{y},\mathsf{z})$, sampling a random point from a distribution in $\mathcal{A}_{\text{free}}$, and checking whether the path from y to z belongs to the free space $\mathcal{A}_{\text{free}}$, respectively.

As stated before, we aim to find a path in the extended free space $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$. To this end, we need to sample points and perform collision checking in $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$. Therefore, we define the functions $\mathsf{SampleExt}(\bar{\rho})$ and $\mathsf{ObstacleFreeExt}(\mathsf{y},\mathsf{z},\bar{\rho})$; $\mathsf{SampleExt}(\bar{\rho})$ samples a point from a uniform distribution in the extended free space $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$; $\mathsf{ObstacleFreeExt}(\mathsf{y},\mathsf{z},\bar{\rho})$ checks whether the path $X_{\text{Line}}:[0,\sigma]\to\mathbb{T}$, for some positive σ, from y to z is collision free with respect to the extended free space, i.e., check whether $\mathsf{y}'\in\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$, $\forall\mathsf{y}'\in X_{\text{Line}}$.

The new functions $\mathsf{SampleExt}(\cdot)$ and $\mathsf{ObstacleFreeExt}(\cdot)$ can be used in any geometric sampling-based motion planning algorithm, giving thus rise to a new family of algorithms, which produce a safe path in an extended free space $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$. This path is then tracked by the system using the control algorithm of Section III-A. Note, however, that the control algorithm guarantees tracking of a *time-varying* smooth (at least k-times continuously differentiable) trajectory $q_{\text{d}}(t)$, whereas the output of the respective motion planning algorithm is a path, i.e., a sequence of points in $\mathbb{T}$. Therefore, we smoothen this path and endow it with a time behavior, producing hence a time trajectory (see [4]). The combination of the aforementioned steps, namely the family of geometric sampling-based motion planning algorithms in $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$, the time endowment, and the funnel-control algorithm of Section III-A, constitute the framework of *KinoDynamic motion planning via Funnel control*, or *KDF* motion-planning framework. This framework solves the kinodynamic motion-planning problem, without resorting to sampling of control inputs or employment of the system dynamics (either in the motion-planning or the control module).

An example of a KDF motion-planning algorithm is KDF-RRT, presented in Algorithm 1.

Algorithm 1 KDF-RRT

Input: $\bar{\rho}$, $\mathcal{A}_{\text{free}}$, Q_g, $q_1(0)$
Output: Tree $\mathcal{G}$ in $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$

procedure TREE
  $\mathcal{V}\leftarrow\{q_1(0)\}$; $\mathcal{E}\leftarrow\emptyset$; $\mathsf{ReachGoal}\leftarrow\mathsf{False}$;
  while not $\mathsf{ReachGoal}$ **do**
    $\mathcal{G}\leftarrow(\mathcal{V},\mathcal{E})$;
    $q_{\text{rand}}\leftarrow\mathsf{SampleExt}(\bar{\rho})$;
    $q_{\text{nearest}}\leftarrow\mathsf{Nearest}(\mathcal{G},q_{\text{rand}})$;
    $q_{\text{new}}\leftarrow\mathsf{Steer}(q_{\text{nearest}},q_{\text{rand}})$;
    if $\mathsf{ObstacleFreeExt}(q_{\text{nearest}},q_{\text{new}},\bar{\rho})$ **then**
      $V\leftarrow V\cup\{q_{\text{new}}\}$;$\mathcal{E}\leftarrow\mathcal{E}\cup\{(q_{\text{nearest}},q_{\text{new}})\}$;
    for $q'\in\mathcal{V}$ **do**
      if $\mathsf{ObstacleFreeExt}(q',Q_g,\bar{\rho})$ **then**
        $V\leftarrow V\cup\{Q_g\}$; $\mathcal{E}\leftarrow\mathcal{E}\cup\{(q',Q_g)\}$;
        $\mathsf{ReachGoal}\leftarrow\mathsf{True}$;

IV. Experiments

This section is devoted to the experimental validation of the proposed framework using a 6DOF manipulator from HEBI-Robotics subject to 2nd-order dynamics, which consists of 6 rotational joints (see Fig. 1) operating in $[-\pi,\pi]$, resulting in $q_1=[q_1^{\mathfrak{t}},\dots,q_6^{\mathfrak{t}}]^\top$.

We consider that the robot has to perform a pick-and-place task, where it has to pick an object from region T_1 and deliver it in region T_2 (see Fig. 1). We use the KDF-RRT algorithm, with $\bar{\rho}=[0.15,0.1,0.1,0.2,0.2,0.2]$ rad, to generate two paths: from the initial configuration to a point close to T_1 (to avoid collision with the object), and from T_1 to T_2.

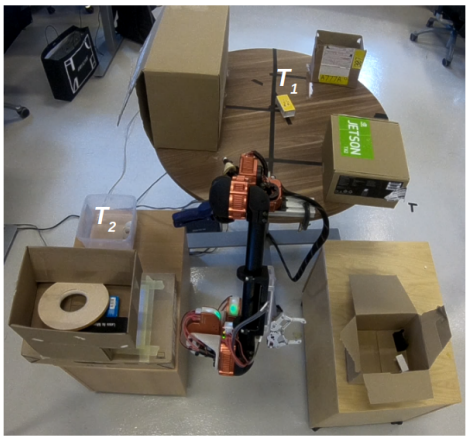

Fig. 1: The initial configuration of the HEBI robot in an obstacle-cluttered environment.

Regarding the collision checking in $\bar{\mathcal{A}}_{\text{free}}(\bar{\rho})$, we check 10 samples around each point of the resulting path for collision. We next fit smooth trajectories for the two paths $q_{\text{d}}^{\text{r},1}(t)$, $q_{\text{d}}^{\text{r},2}(t)$, with duration of $t_{f_1} = 7$ and $t_{f_2} = 11$ seconds, respectively. For grasping the object, we use a simple linear interpolation to create an additional time-varying trajectory segment to T_1 with duration of 3 seconds.

For the execution of the control algorithm, we choose constant funnel functions $\rho^{\text{t},i} = [\rho_1^{\text{t},i}, \dots, \rho_6^{\text{t},i}]^\top = \bar{\rho} = [0.15, 0.1, 0.1, 0.2, 0.2, 0.2]$ rad, for the two paths $i \in \{1, 2\}$. Moreover, we choose $\rho_{2_j} = 15$ for all $j \in \{1, \dots, 6\}$, and the control gains as $K^{\text{t}} = \text{diag}(1.25, 1.5, 1, 2, 1, 1)$, $K_2 = \text{diag}(250, 200, 150, 50, 20, 10)$.

The results of the experiment are depicted in Fig. 2, which depicts the normalized signals $\xi^{\text{t}}(t) = [\xi_1^{\text{t}}, \dots, \xi_6^{\text{t}}]^\top$ and $\xi_2(t) = [\xi_{2_1}, \dots, \xi_{2_6}]^\top$ (top and bottom, respectively) for $t \in [0, 21]$ seconds. It can be observed that for the entire motion, it holds that $\xi_j^{\text{t}} \in (-1, 1)$, $\xi_{2_j}(t) \in (-1, 1)$, for all $j \in \{1, \dots, 6\}$, which implies that $-\rho_j^{\text{t}} < e_j^{\text{t}}(t) = q_1(t) - q_{\text{d}}(t) < \rho_j^{\text{t}}$, $-\rho_{2_j} < e_2(t) = q_{2_j}(t) - \alpha_{1_j}(t) < \rho_j^{\text{t}}$, for all $j \in \{1, \dots, 6\}$ and $t \in [0, 21]$ seconds, with α_1 as in (5). Therefore, we conclude that the robot tracks the path output by the KDF-RRT algorithm within the prescribed funnel, avoiding thus collisions.

In order to further evaluate the proposed control algorithm, we compared our results with a standard well-tuned PID controller as well as the parametric adaptive control scheme (PAC) of our previous work [4]. The signals ξ^{t} for these two control schemes are depicted in Fig. 3. Note that the controllers fail to retain the normalized errors $\xi_j^{\text{t}}(t)$ in the interval $(-1, 1)$. Although in the particular instance this did not lead to collisions, it jeopardizes the system motion, since it does not comply with the bounds set in the KDF-RRT algorithm.

V. Discussion

By using funnel control, we open the way of using geometric sampling-based motion planners for solving the kinodynamic motion-planning problem for high-dimensional

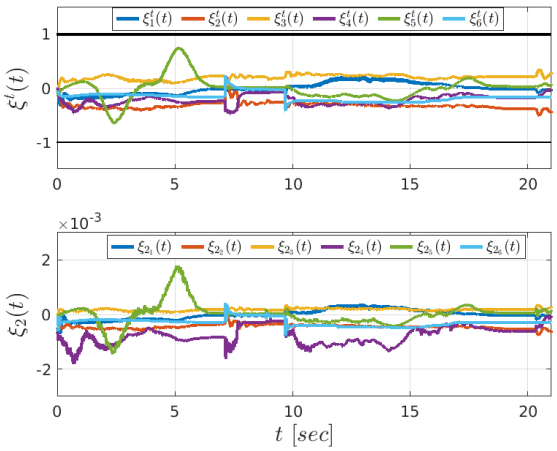

Fig. 2: The evolution of the normalized errors ξ_j^{t} (top) and $\xi_{2_j}(t)$, for $j \in \{1, \dots, 6\}$, of the hardware experiment.

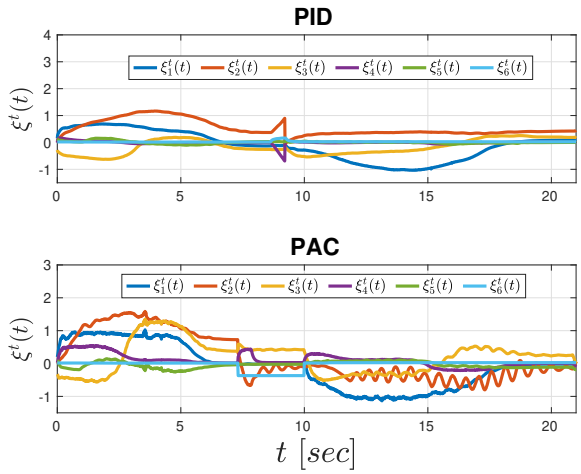

Fig. 3: The evolution of the normalized errors ξ_j^{t} (top) and $\xi_{2_j}(t)$, for $j \in \{1, \dots, 6\}$, of the hardware experiment, when using a PID controller (top), and the adaptive control algorithm from [4].

robots with uncertain dynamics. Limitations that need to be addressed in future work include taking into account explicit input constraints and considering underactuated or nonholonomic dynamics.

References

[1] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[2] H. M. Choset, S. Hutchinson, K. M. Lynch, G. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of robot motion: theory, algorithms, and implementation*. MIT press, 2005.

[3] C. K. Verginis, D. V. Dimarogonas, and L. E. Kavraki, "Kdf: Kinodynamic motion planning via geometric sampling-based algorithms and funnel control," 2021.

[4] C. Verginis, D. V. Dimarogonas, and L. Kavraki, "Sampling-based motion planning for uncertain high-dimensional systems via adaptive control," *The 14th International Workshop on the Algorithmic Foundations of Robotics*, 2021.