IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 10, 2021, accepted August 28, 2021, date of publication August 30, 2021, date of current version September 9, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3109125

Control Based Motion Planning Exploiting Calculus of Variations and Rational Functions: A Formal Approach

BABAK SALAMAT, NUNZIO A. LETIZIA, (Graduate Student Member, IEEE), AND ANDREA M. TONELLO, (Senior Member, IEEE)
Chair of Embedded Communication Systems, University of Klagenfurt, 9020 Klagenfurt, Austria

Corresponding author: Babak Salamat (babaksa@edu.aau.at)

ABSTRACT This paper presents a control-based trajectory generation approach for unmanned aerial vehicles (UAVs) under dynamic constraints. It exploits the concept of optimal control to find closed-form differential equations that satisfy any arbitrary dynamic limitation mapped into kinematic constraints. Pontryagin's Minimum Principle applies to derive a set of differential equations in which the dynamic environment is considered in the constrained Hamiltonian function. In particular, we aim to minimize the L_2-norm of the control input avoiding dynamic obstacles, given initial and final boundary conditions. Lastly, this paper proposes a novel interpolation algorithm based on rational functions, referred to as rational recursive smooth trajectory (RRST) method. The method generates an analytic expression that approximates the control inputs, for which no closed-form solutions are in general attainable.

INDEX TERMS Aerospace, optimal control, calculus of variations, steepest descent, function interpolation, dynamic interpolation, optimization, rational functions, unmanned aerial vehicles (UAV), path planning, trajectory generation.

I. INTRODUCTION

Mission planning has been playing a pivotal role in various application fields, e.g., in air traffic control, aerospace, robotics, and mechanics. In the last two decades, research in the calculus of variation and optimal control has provided several methodologies for trajectory generation under environmental constraints.

In [1] and [2], trajectories avoiding obstacles are obtained with statistical learning and evolutionary algorithms. In the domain of calculus of variations, several methodologies for path planning have been developed [3]–[8]. Second-order and higher order variational problems are studied in [3] and [4], respectively. An optimal control methodology is applied to obtain an optimal path for considered dynamical system models in [5]–[9]. In this class of problems, the objective is to automatically generate smooth trajectories passing through certain waypoints at specific times. In other words, the trajectory planning consists of finding a relation between time and space components. Consequently, the trajectory may be defined as a parametric function of the time. To achieve this goal, several methodologies have been successfully applied by providing only initial and final waypoints or by considering multiple waypoints. In the former case, it is clear that the optimal control problem is equivalent to the solution of the two-waypoint boundary value problem [10]–[12]. In the latter, by specifying the waypoints it is possible to define a trajectory passing through the waypoints either statically (using pure interpolation techniques) or dynamically (through calculus of variations). In the case of multiple waypoints, dynamic interpolation was initially studied in [13] for application to the aircraft motion.

In general, three types of interpolation can be considered:

Static interpolation: the trajectory passes through the waypoints for some values of the time, Fig. 1(a).

Approximation: the trajectory does not necessarily pass through the waypoints, Fig. 1(b).

Dynamic interpolation: the trajectory is specified in terms of discrete, ordered waypoints through which the dynamical system model states must go, Fig. 1(c).

Path primitives such as lines [14], polynomials [15]–[17], and splines [18], have been deployed as interpolation functions, especially for static interpolation. The objective of polynomial interpolation techniques is to obtain a trajectory

The associate editor coordinating the review of this manuscript and approving it for publication was Halil Ersin Soken.

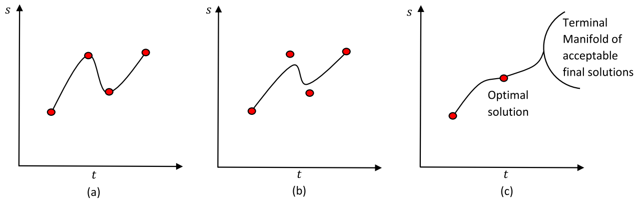


FIGURE 1. Static interpolation (a), approximation (b), and dynamic interpolation (c) of a set of waypoints.

that passes through multiple waypoints. In fact, the differentiability of polynomials makes them a suitable interpolation choice for considering the vehicle dynamics. However, it is necessary to solve an inverse problem to find a unique polynomial trajectory that fulfills the constraints. Therefore, the interpolation problem is often split into splines, i.e., piecewise polynomial trajectories. Splines are easy to be constructed and provide bounded trajectories although the continuity of the derivatives (smoothness) at the waypoints is only guaranteed up to a certain order. A recent methodology that avoids solving an inverse problem has been proposed in [19] and [20]. Therein, the authors developed a recursive methodology to build a unique smooth polynomial trajectory satisfying the kinematic constraints.

Another challenge concerning mission planning is accurate navigation with obstacle avoidance capability in which the unmanned aerial vehicle (UAV) can fulfill and accomplish any given task safely. Artificial Potential Fields (APFs) have been utilized and continued for path planning [21], including application to UAVs with promising results [22]. On the other hand, from the optimal control point of view, Model Predictive Control (MPC) is a promising approach that has been used in local motion planning with kinematic and environmental constraints [23]. However, the previous techniques applied in a local context fail in providing optimal paths and their incremental behavior could lock in local minima and discontinuity in control inputs [21]. To avoid the limitation of proposed methods, research is moving towards a dynamic interpolation based framework.

These limitations motivate us to rethink the problem by formulating a control-based trajectory generation approach for mechanical UAV systems under dynamic constraints. It uses the concept of optimal control to find closed-form differential equations that satisfy any arbitrary dynamic limitations mapped into kinematic constraints. It is fundamentally different from existing approaches [24] in the following aspects:

- The trajectory is defined by a constrained Hamiltonian function that fulfills all the boundary conditions for the dynamical planner system;
- The trajectory is optimal and obtained by solving closed-form differential equations that enable considering the dynamics for the environment such as static and dynamic obstacles;
- To shape the path with environmental constraints there is no need to define extra waypoints;
- There is no discontinuity in the states along the optimal trajectory;
- Lastly, we introduce a novel interpolation method, based on the Recursive Smooth Trajectory (RST) generation algorithm recently introduced in [20], which utilizes as a basis the rational functions to approximate the optimal control inputs.

In detail, the paper is organized as follows. Section II formulates the dynamical system model for the planner block and presents the path motion dynamics. In Section III, the optimal control law and conditions for optimality is rigorously derived. Section IV introduces a novel interpolation approach based on RST and rational functions providing an analytic expression of the optimal control inputs. Simulation results are reported in Section V. Finally, the paper is concluded in Section VI.

Notation: A trajectory is said to have k-th degree parametric continuity in parameter t, if its k-th derivative $\frac{d^k}{dt^k}Q(t)$ is continuous. It is then also called $\mathcal{C}^k$ continuous.

II. PROBLEM FORMULATION

We start with the dynamical system model of the trajectory planner, considering the 1D position and its k derivatives (velocity, acceleration, etc.). The extension to the 3D case will be considered later. In the state-space form, the system model can be written in a linear time-invariant form as:

$$\dot{\mathrm{x}}(t) = A\mathrm{x}(t) + Bu(t) \tag{1}$$

where $\mathrm{x}(t) \in \chi \subset \mathbb{R}^{k+1}$ is the dynamical system state and k is the order of the kinematic constraint, $u(t) \in \mathbb{R}$ is the control input, and the system representation is in the following

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & 1 \\ 0 & \cdots & \cdots & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}. \tag{2}$$

Utilizing the state-space equalization leads to the possibility of analyzing the functionality of planner dynamics in the search space. To proceed with the control based approach we propose the following problem:

Problem 1: Generate a $\mathcal{C}^3$ trajectory $\mathrm{x}(.)$ in 1D, which minimizes the cost functional

$$J_{oc} = \frac{1}{2}\int_0^T u^2(t)dt, \tag{3}$$

amongst all $\mathcal{C}^3$ curves, with the following boundary conditions

$$\begin{aligned} \mathrm{x}(0) &= \mathrm{x}_0, \quad \mathrm{x}(T) = \mathrm{x}_T, \\ \dot{\mathrm{x}}(0) &= v_0, \quad \dot{\mathrm{x}}(T) = v_T, \\ \ddot{\mathrm{x}}(0) &= a_0, \quad \ddot{\mathrm{x}}(T) = a_T, \end{aligned} \tag{4}$$

for the dynamical system

$$\dot{\mathrm{x}}(t) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \mathrm{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t). \tag{5}$$

The functional in (3) is the L_2-norm of the control input and ensures that the trajectory reaches the final state with minimum control input energy. The reason for choosing the L_2-norm consists in the fact that it can easily be deduced from Pontryagin's minimum principle [10]. The objective is to find a $u(t)$, denoted by $u^*(t)$, such that the cost functional in (3) is minimized. Consequently, a trajectory is derived.

III. DERIVATION OF THE OPTIMAL CONTROL LAW AND CONDITIONS FOR OPTIMALITY

In this section, we use Pontryagin's Minimum Principle [10] to derive an optimal control law. The methodology follows by formulating the Hamiltonian system and then applying the first-order necessary conditions. For the sake of completeness, Pontryagin's Minimum Principle is recalled below.

Theorem 1 [10]: Consider the general dynamical system model

$$\dot{\mathrm{x}}(t) = f(\mathrm{x}(t), u(t), t), \tag{6}$$

where $\mathrm{x}(0)$ *is given. The associated cost functional is defined as*

$$J_{P_{(0)}} = \Psi(\mathrm{x}(T),\ T) + \int_0^T \mathcal{D}(\mathrm{x}(t), u(t), t)dt, \tag{7}$$

where the target condition must satisfy

$$\Psi(\mathrm{x}(T),\ T) = 0. \tag{8}$$

If the control signal is unconstrained, then

$$\frac{\partial H_P}{\partial u} = 0, \tag{9}$$

with the Hamiltonian

$$H_P(\mathrm{x}(t), u(t), \lambda(t), t) = \mathcal{D}(\mathrm{x}(t), u(t), t) + \lambda^\top(t) f(\mathrm{x}(t), u(t), t), \tag{10}$$

where $\lambda(t)$ *is the costate. In addition, the more general condition for the constrained optimal control signal can be written*

$$H_P(\mathrm{x}^*(t), u^*(t), \lambda^*(t), t) \leq H_P(\mathrm{x}^*(t), u(t), \lambda^*(t), t), \tag{11}$$

for all admissible $u(t)$. *Indeed, minimizing the Hamiltonian results in optimal values of the state and costate over all admissible* $u(t)$.

Now, let us define $\lambda(t) = [\lambda_1(t)\ \lambda_2(t)\ \lambda_3(t)]^\top \in \mathbb{R}^3$ and a constant $c = [c_1\ c_2\ \ldots\ c_6]^\top$. From (3) and (5), the Hamiltonian can be written as

$$H_{oc} = \frac{1}{2}u^2(t) + \lambda^\top(t)(A\mathrm{x}(t) + Bu(t)). \tag{12}$$

The necessary conditions are derived by differentiating (12) with respect to $\lambda(t)$ and $u(t)$ that are:

$$\dot{\lambda}(t) = -\frac{\partial H_{oc}}{\partial \mathrm{x}}\bigg|_{u=u^*} \tag{13}$$

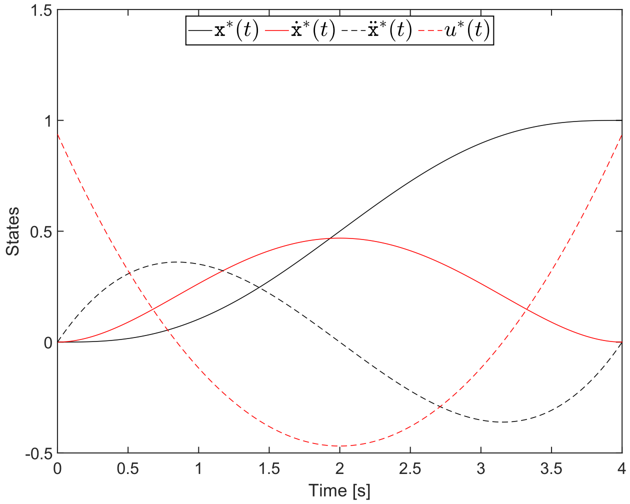


FIGURE 2. 1D optimal states. The corresponding cost is ($J_{oc} = 0.3516$).

$$\frac{\partial H_{oc}}{\partial u} = 0. \tag{14}$$

This results in the following form:

$$\dot{\lambda}_1(t) = -\frac{\partial H_{oc}}{\partial \mathrm{x}_1} = 0 \rightarrow \lambda_1^*(t) = c_1 \tag{15}$$

$$\dot{\lambda}_2(t) = -\frac{\partial H_{oc}}{\partial \mathrm{x}_2} = -\lambda_1(t) \rightarrow \lambda_2^*(t) = c_1 t + c_2 \tag{16}$$

$$\dot{\lambda}_3(t) = -\frac{\partial H_{oc}}{\partial \mathrm{x}_3} = -\lambda_2(t) \rightarrow \lambda_3^*(t) = \frac{1}{2}c_1 t^2 - c_2 t + c_3 \tag{17}$$

$$\frac{\partial H_{oc}}{\partial u} = u(t) + \lambda_3(t) \rightarrow u^*(t) = -\frac{1}{2}c_1 t^2 + c_2 t - c_3. \tag{18}$$

Equations (15)-(17) represent a set of conditions on the adjoint vector $\lambda(t)$. These conditions must be satisfied to get the optimality of the control law. With this control law, the system follows the optimal state trajectory

$$\dot{\mathrm{x}}_3(t) = u^*(t) \rightarrow \mathrm{x}_3^*(t) = -\frac{1}{6}c_1 t^3 + \frac{1}{2}c_2 t^2 - c_3 t + c_4 \tag{19}$$

$$\dot{\mathrm{x}}_2(t) = \dot{\mathrm{x}}_3(t) \rightarrow \mathrm{x}_2^*(t) = -\frac{1}{24}c_1 t^4 + \frac{1}{6}c_2 t^3 - \frac{1}{2}t^2 + c_4 t + c_5 \tag{20}$$

$$\dot{\mathrm{x}}_1(t) = \dot{\mathrm{x}}_2(t) \rightarrow \mathrm{x}_1^*(t) = -\frac{1}{120}c_1 t^5 + \frac{1}{24}c_2 t^4 - \frac{1}{6}c_3 t^3 + \frac{1}{2}c_4 t^2 + c_5 t + c_6. \tag{21}$$

Definition 1: An extremal control law is one that satisfies conditions (13)-(14) of Pontryagin's Minimum Principle. An extremal control law is a globally optimal control law for the dynamical system model in (5).

By applying the boundary conditions (4), the optimal adjoint variables $\lambda^*(t) = [\lambda_1^*(t)\ \lambda_2^*(t)\ \lambda_3^*(t)]^\top$, the optimal trajectory $\mathrm{x}^*(t)$, and the optimal control law $u^*(t)$ can be computed analytically. Fig. 2 shows the extremal curves connecting two waypoints $(\mathrm{x}(0),\ \dot{\mathrm{x}}(0),\ \ddot{\mathrm{x}}(0)) = (0,\ 0,\ 0)$ and $(\mathrm{x}(T),\ \dot{\mathrm{x}}(T),\ \ddot{\mathrm{x}}(T)) = (1,\ 0,\ 0)$ for the value of $T = 4$ s. It can be shown that the boundary conditions are matched.

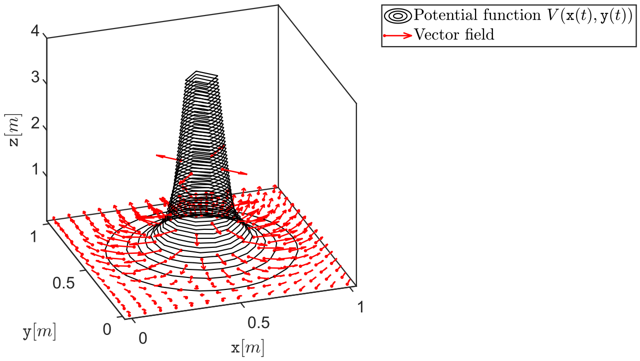


FIGURE 3. Gradient Lines in the case of a potential function $V(\mathrm{x}(t), \mathrm{y}(t), x_{ob}, y_{ob})$ obstacle with $x_{ob} = 0.5$, $y_{ob} = 0.5$, $m = 2$, and $\ell = 0.02$.

Until now, we described a constrained Hamiltonian formalism (12) for the optimal control of the motion planner dynamics in (5). Safety is an essential issue for accomplishing a given mission and this is challenged when we operate in a dynamic unknown environment. For this reason, we are interested in considering the dynamics of obstacles in the constrained Hamiltonian formalism. This methodology allows us to derive a set of closed-form differential equations in which the dynamic environment is considered in the constrained Hamiltonian function. The objective is to find an optimal path from an initial condition $X_0 = [\mathrm{x}(0)\ \ \mathrm{y}(0)]^\top$ to a final destination point $X_T = [\mathrm{x}(T)\ \ \mathrm{y}(T)]^\top$ avoiding collision with obstacles. In this case, the augmented dynamical system model in the state-space form can be written as:

$$\begin{bmatrix}\mathrm{x}(t)\\ \dot{\mathrm{x}}(t)\\ \ddot{\mathrm{x}}(t)\\ \mathrm{y}(t)\\ \dot{\mathrm{y}}(t)\\ \ddot{\mathrm{y}}(t)\end{bmatrix} = \begin{bmatrix}\dot{\mathrm{x}}_1(t)\\ \dot{\mathrm{x}}_2(t)\\ \dot{\mathrm{x}}_3(t)\\ \dot{\mathrm{x}}_4(t)\\ \dot{\mathrm{x}}_5(t)\\ \dot{\mathrm{x}}_6(t)\end{bmatrix} = \begin{bmatrix}0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&0&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\0&0&0&0&0&0\end{bmatrix}\mathrm{x}(t) + \begin{bmatrix}0&0\\0&0\\1&0\\0&0\\0&0\\0&1\end{bmatrix}\begin{bmatrix}u_1\ (t)\\ u_2\ \ (t)\end{bmatrix}. \quad (22)$$

Without losing the generality, we consider an obstacle centered in $(x_{ob},\ y_{ob})$, and we define a simple repulsive potential function [25]. Let $V(\mathrm{x}(t),\ \mathrm{y}(t),\ x_{ob},\ y_{ob})$ be in the form

$$V(\mathrm{x}(t),\ \mathrm{y}(t),\ x_{ob},\ y_{ob}) = \frac{\ell}{(\mathrm{x}(t)-x_{ob})^m + (\mathrm{y}(t)-y_{ob})^m}, \quad (23)$$

where m is a natural positive number and ℓ is a constant parameter (see Fig. 3).

Interestingly, we can bring the potential function in (23) to the constrained Hamiltonian formalism. To do so, we propose the following problem:

Problem 2: Generate a $\mathcal{C}^3$ trajectory $X(.)$ in $\mathbb{R}^6$, which minimizes the cost functional

$$J_{ob} = \frac{1}{2}\int_0^T \Big(\mathrm{x}_1^2(t) + \mathrm{x}_4^2(t) + u_1^2(t) + u_2^2(t) + \frac{\ell}{(\mathrm{x}(t)-x_{ob})^m + (\mathrm{y}(t)-y_{ob})^m}\Big)dt, \quad (24)$$

amongst all $\mathcal{C}^3$ curves, with the following boundary conditions

$$\begin{aligned} X(0) &= X_0, \quad X(T) = X_T,\\ \dot{X}(0) &= V_0, \quad \dot{X}(T) = V_T,\\ \ddot{X}(0) &= A_0, \quad \ddot{X}(T) = A_T, \end{aligned} \quad (25)$$

for the dynamical system in (22).

To proceed, let us define

$$p(t) = [p_1(t)\ \ p_2(t)\ \ p_3(t)\ \ p_4(t)\ \ p_5(t)\ \ p_6(t)]^\top \in \mathbb{R}^6. \quad (26)$$

From (24) and (22), the Hamiltonian can be written as

$$\begin{aligned}\mathcal{H} = {}& \frac{1}{2}\mathrm{x}_1^2(t) + \frac{1}{2}\mathrm{x}_4^2(t) + \frac{1}{2}u_1^2(t) + \frac{1}{2}u_2^2(t)\\ &+ \frac{1}{2}\frac{\ell}{(\mathrm{x}_1(t)-x_{ob})^m + (\mathrm{x}_4(t)-y_{ob})^m}\\ &+ p_1(t)\mathrm{x}_2(t) + p_2(t)\mathrm{x}_3(t)\\ &+ p_3(t)u_1(t) + p_4(t)\mathrm{x}_5(t) + p_5(t)\mathrm{x}_6(t) + p_6(t)u_2(t).\end{aligned} \quad (27)$$

Following the necessary conditions and after straightforward computation resulting in closed-form differential equations

$$\frac{\partial\mathcal{H}}{\partial u_1} = 0 \rightarrow u_1(t) + p_3(t) = 0 \quad (28)$$

$$\frac{\partial\mathcal{H}}{\partial u_2} = 0 \rightarrow u_2(t) + p_6(t) = 0 \quad (29)$$

$$\dot{p}_1(t) = -\mathrm{x}_1(t) + \frac{m\ell(\mathrm{x}_1(t)-x_{ob})^{m-1}}{\Big((\mathrm{x}_1(t)-x_{ob})^m + (\mathrm{x}_4(t)-y_{ob})^m\Big)^2} \quad (30)$$

$$\dot{p}_2(t) = -p_1(t) \quad (31)$$

$$\dot{p}_3(t) = -p_2(t) \quad (32)$$

$$\dot{p}_4(t) = -\mathrm{x}_4(t) + \frac{m\ell(\mathrm{x}_4(t)-y_{ob})^{m-1}}{\Big((\mathrm{x}_1(t)-x_{ob})^m + (\mathrm{x}_4(t)-y_{ob})^m\Big)^2} \quad (33)$$

$$\dot{p}_5(t) = -p_4(t) \quad (34)$$

$$\dot{p}_6(t) = -p_5(t) \quad (35)$$

It is not possible to solve analytically the resulting closed-form differential equations for Problem 2. In the next section, we propose two iterative learning-based algorithms for solving the mentioned optimal control problem that involves both terminal constraints on the state variables and inequality constraints on the states (navigation function) along the entire trajectory.

A. STEEPEST DESCENT APPROACH

Let

$$\dot{p}(t) = -\frac{\partial \mathcal{H}}{\partial \mathrm{x}} \tag{36}$$

$$\frac{\partial \mathcal{H}}{\partial u} = \nabla f(u(t)), \tag{37}$$

and $u^{(i)}(t) = [u_1^{(i)}(t) \;\; u_2^{(i)}(t)]^\top$ be the i-th iteration of the optimal control $u^{(0)}(t) = [u_1^{(0)}(t) \;\; u_2^{(0)}(t)]^\top$. Using the steepest descent methodology [26], our algorithm reads as follows:

Step 1: Define an arbitrary control input $u^{(0)}(t) = [u_1^{(0)}(t) \;\; u_2^{(0)}(t)]^\top$.

Step 2: By having $u^{(0)}(t) = [u_1^{(0)}(t) \;\; u_2^{(0)}(t)]^\top$, it is possible to update the states in (22).

Step 3: Update the co-state equation $\dot{p}(t) = -\frac{\partial \mathcal{H}}{\partial \mathrm{x}}$.

Step 4: By having the information of control inputs $u^{(i)}(t)$, states $\mathrm{x}^{(i)}(t)$, and co-states $p^{(i)}(t)$ from the previous iteration we can update the control inputs in the following form

$$u^{(i+1)}(t) = u^{(i)}(t) - \gamma \nabla f(u^{(i)}(t)), \tag{38}$$

where $\gamma = [\gamma_1 \;\; \gamma_2]^\top > 0$ is the learning rate vector.

Step 5: Check if $\|\mathcal{H}^{(i)}\| \leq \epsilon$ or $\|J_{ob}^{(i-1)} - J_{ob}^{(i)}\| \leq \epsilon$ where ϵ is a small positive constant. If this is true, then $u^*(t) = u^{(i+1)}(t)$ else $i = i+1$, $u^{(i)}(t) = u^{(i+1)}(t)$ and return to Step 3.

B. CONJUGATE GRADIENT APPROACH

Inspired by the conjugate gradient methodology [27], our algorithm reads as

Step 1: Define an arbitrary control input $u^{(0)}(t) = [u_1^{(0)}(t) \;\; u_2^{(0)}(t)]^\top$.

Step 2: By having $u^{(0)}(t) = [u_1^{(0)}(t) \;\; u_2^{(0)}(t)]^\top$, it is possible to update the states in (22).

Step 3: Update the co-state equation $\dot{p}(t) = -\frac{\partial \mathcal{H}}{\partial \mathrm{x}}$.

Step 4: By having the information of control inputs $u^{(i)}(t)$, states $\mathrm{x}^{(i)}(t)$, and co-states $p^{(i)}(t)$ from the previous iteration we can update the control inputs in the following form

$$u^{(i+1)}(t) = u^{(i)}(t) + \gamma r^{(i)}, \tag{39}$$

where

$$r^{(i)} = -\nabla f(u^{(i)}(t)) + \zeta^{(i-1)} r^{(i-1)} \tag{40}$$

$$\zeta^{(i-1)} = \frac{\|\nabla f(u^{(i)}(t))\|^2}{\|\nabla f(u^{(i-1)}(t))\|^2}, \tag{41}$$

with the initial condition $r^{(0)} = -\nabla f(u^{(0)}(t))$.

Step 5: Check if $\|\mathcal{H}^{(i)}\| \leq \epsilon$ or $\|J_{ob}^{(i-1)} - J_{ob}^{(i)}\| \leq \epsilon$ where ϵ is a small positive constant. If this is true, then $u^*(t) = u^{(i+1)}(t)$ else $i = i+1$, $u^{(i)}(t) = u^{(i+1)}(t)$ and return to Step 3.

C. CONVERGENCE

In this subsection, we prove the convergence of the proposed algorithm.

Proposition 1: Let $\|\gamma\|$ be sufficiently small, and $\nabla f(u^{(i)}(t)) \neq 0$ then $J_{ob}(u^{(i+1)}(t)) < J_{ob}(u^{(i)}(t))$.

Proof: Using the Taylor's expansion we can write

$$\begin{aligned} &f(u^{(i)}(t) - \gamma \nabla f(u^{(i)}(t))) \\ &\quad = f(u^{(i)}(t)) + \nabla f^\top(u^{(i)}(t))\big(-\gamma \nabla f(u^{(i)}(t))\big). \end{aligned} \tag{42}$$

Observing (42), we can say that the multiplication of two gradients in the RHS of (42) is always positive. Also, $\gamma_i > 0$ for $i = 1, 2$ is positive and, therefore, we can write

$$f(u^{(i)}(t)) + \nabla f^\top(u^{(i)}(t))\big(-\gamma \nabla f(u^{(i)}(t))\big) < f(u^{(i)}(t)), \tag{43}$$

which means that the second term in the LHS of (43) is always negative and the result follows. With both algorithms we guarantee that f is decreasing, therefore also J_{ob} is decreasing since the integrand is always semi-definite positive. ■

Furthermore, the terms in the cost functional in (24) are convex. Therefore, it is possible to check whether $u^*(t)$ is an isolated local or global minimum. This statement is supported by the following result.

For the sake of exposition simplicity, we consider the 1D cost functional with $m = 2$, then we have,

$$J_{ob} = \frac{1}{2}\int_0^T \left(\mathrm{x}_1^2(t) + u_1^2(t) + \frac{\ell}{(\mathrm{x}(t) - x_{ob})^2}\right) dt, \tag{44}$$

and the Jacobian and Hessian matrix of (44) are

$$\tilde{J} = \begin{bmatrix} \mathrm{x}_1(t) - \dfrac{\ell}{(\mathrm{x}_1(t) - x_{ob})^3} \\ u_1(t) \end{bmatrix}, \tag{45}$$

$$\tilde{H} = \begin{bmatrix} \dfrac{3\ell}{(\mathrm{x}_1(t) - x_{ob})^4} + 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{46}$$

that shows the sufficiency condition for $u_1^*(t)$ to be an isolated global minimum. Indeed, the determinant of the first principal minor is positive and the determinant of the Hessian is as well always positive, therefore J_{ob} is strictly convex.

D. COMPUTATIONAL COMPLEXITY

We consider the following assumptions on Problem 2.

A 1: $f(u(t))$ is differentiable and continuous, and $f(u(t))$ is bounded below, that there exists a constant δ_{lb} such that

$$f(u(t)) \geq \delta_{lb}.$$

A 2: $\nabla f(u(t))$ is Lipschitz continuous, therefore, there exists a constant $\gamma_n \geq 0$, for all $u(t)$, $\bar{u}(t)$

$$\|\nabla f(u(t)) - \nabla f(\bar{u}(t))\| \leq \gamma_n \|u(t) - \bar{u}(t)\|.$$

Theorem 2: Let us assume that A 1-A 2 hold. Then, it exists a constant δ_{up} depending on $u^0(t)$, for all $\epsilon \in (0, 1)$, at most $\frac{\delta_{up}}{\epsilon^2} \approx \mathcal{O}(\epsilon^{-2})$ iterations are needed to obtain in the iteration $u^i(t)$ such that at most $\|\mathcal{H}^{(i)}\| \leq \epsilon$.

Proof: Using Taylor's expansion gives that, for each $i \geq 0$,

$$\begin{aligned} &f(u^{(i)}(t)) - f(u^{(i)}(t) - \gamma \nabla f(u^{(i)}(t))) \\ &\quad \geq f(u^{(i)}(t)) - f(u^{(i)}(t)) + \gamma \nabla f^\top(u^{(i)}(t))\big(\nabla f(u^{(i)}(t))\big) \end{aligned}$$

$$-\frac{1}{2}\gamma^2\gamma_n\nabla f^\top(u^{(i)}(t))\big(\nabla f(u^{(i)}(t))\big), \tag{47}$$

for any $\gamma \geq 0$. Maximizing the right-hand side of the previous inequality with respect to γ yields

$$f(u^{(i)}(t)) - f(u^{(i)}(t)) - \frac{1}{\gamma_n}\nabla f(u^{(i)}(t)))$$
$$\geq \frac{1}{2\gamma_n}\nabla f^\top(u^{(i)}(t))\big(\nabla f(u^{(i)}(t))\big) \geq \frac{\epsilon^2}{2\gamma_n}, \tag{48}$$

for each iteration i as long as $\nabla f^\top(u^{(i)}(t))\big(\nabla f(u^{(i)}(t))\big) \geq \epsilon$. Therefore, the maximum number of iterations can be defined as $\frac{2\gamma_n(f(u^{(0)}(t) - \delta_{lb})}{\epsilon^2} \triangleq \frac{\delta_{up}}{\epsilon^2}$. ■

E. LAGRANGIAN FORMALISM AND VARIATION OF THE REPULSIVE FUNCTION DUE TO DYNAMICS OF THE OBSTACLE

The goal of this section is twofold. We first propose an alternative solution to Problem 2. Then we address the motion path in a dynamic environment, where an obstacle can change its position over time.

We investigated how to solve Problem 2 with Hamiltonian formalism. The link between the Hamiltonian and Lagrangian is the *Legendre transformation* [28]

$$\mathcal{H} = \dot{X}\frac{\partial \mathcal{L}}{\partial \dot{X}} - \mathcal{L}, \tag{49}$$

where $\mathcal{L}$ is the Lagrangian. So we have:

$$\begin{aligned}\mathcal{L}(X,\dot{X}) = {} & \frac{1}{2}\mathrm{x}_1^2(t) + \frac{1}{2}\mathrm{x}_4^2(t) + \frac{1}{2}u_1^2(t) + \frac{1}{2}u_2^2(t)\\ & + V(\mathrm{x}(t),\ \mathrm{y}(t),\ x_{ob},\ y_{ob}) + \theta_1(t)(\dot{\mathrm{x}}_1(t) - \mathrm{x}_2(t))\\ & + \theta_2(t)(\dot{\mathrm{x}}_2(t) - \mathrm{x}_3(t)) + \theta_3(t)(\dot{\mathrm{x}}_3(t) - u_1(t))\\ & + \theta_4(t)(\dot{\mathrm{x}}_4(t) - \mathrm{x}_5(t)) + \theta_5(t)(\dot{\mathrm{x}}_5(t) - \mathrm{x}_6(t))\\ & + \theta_6(t)(\dot{\mathrm{x}}_6(t) - u_2(t))\end{aligned} \tag{50}$$

where $\theta(t) = [\theta_1(t)\ \theta_2(t)\ \theta_3(t)\ \theta_4(t)\ \theta_5(t)\ \theta_6(t)]^\top$ is the Lagrange multiplier vector. From calculus of variations theory, solving problem (24) is equal to solving the Euler-Lagrange equation

$$\frac{\partial \mathcal{L}}{\partial \mathrm{x}} - \frac{d}{dt}\left(\frac{\mathcal{L}}{\partial \dot{\mathrm{x}}}\right) + \frac{d^2}{dt^2}\left(\frac{\mathcal{L}}{\partial \ddot{\mathrm{x}}}\right) + \frac{\partial \mathcal{L}}{\partial \theta} - \frac{d}{dt}\left(\frac{\mathcal{L}}{\partial \dot{\theta}}\right) = 0 \tag{51}$$

and by substituting the Lagrangian defined in (50) into (51) it follows that

$$u_1(t) = \theta_3(t) \tag{52}$$
$$u_2(t) = \theta_6(t) \tag{53}$$
$$\dot{\theta}_1(t) = \mathrm{x}_1(t) - \frac{m\ell(\mathrm{x}_1(t) - x_{ob})^{m-1}}{\Big((\mathrm{x}_1(t) - x_{ob})^m + (\mathrm{x}_4(t) - y_{ob})^m\Big)^2} \tag{54}$$
$$\dot{\theta}_2(t) = \theta_1(t) \tag{55}$$
$$\dot{\theta}_3(t) = \theta_2(t) \tag{56}$$
$$\dot{\theta}_4(t) = \mathrm{x}_4(t) - \frac{m\ell(\mathrm{x}_4(t) - y_{ob})^{m-1}}{\Big((\mathrm{x}_1(t) - x_{ob})^m + (\mathrm{x}_4(t) - y_{ob})^m\Big)^2} \tag{57}$$
$$\dot{\theta}_5(t) = \theta_4(t) \tag{58}$$
$$\dot{\theta}_6(t) = \theta_5(t). \tag{59}$$

Interestingly equations (52)-(59) are equivalent to those obtained in the Hamiltonian formalism. We now proceed to consider the motion path in a dynamic environment. To do so, let $V(\mathrm{x}(t),\ \mathrm{y}(t),\ x_{ob},\ y_{ob})$ be in the form

$$V(\mathrm{x}(t),\ \mathrm{y}(t),\ x_{ob}(t),\ y_{ob}(t)) = \frac{\ell}{(\mathrm{x}(t) - x_{ob}(t))^m + (\mathrm{y}(t) - y_{ob}(t))^m}. \tag{60}$$

The time derivative of (60) is

$$\frac{\partial V(t)}{\partial t} = \frac{\partial V(t)}{\partial \mathrm{x}}\dot{\mathrm{x}}(t) + \frac{\partial V(t)}{\partial \mathrm{x}_{ob}}\dot{x}_{ob}(t) + \frac{\partial V(t)}{\partial y_{ob}}\dot{y}_{ob}(t). \tag{61}$$

The first term of (61) represents the variations on $V(t)$ due to changes in the motion path. The last two terms indicate the dynamics of the obstacle in the environment and which given by

$$\begin{aligned}&\frac{\partial V(t)}{\partial x_{ob}}\dot{x}_{ob}(t) + \frac{\partial V(t)}{\partial y_{ob}}\dot{y}_{ob}(t)\\ &= \frac{m\ell(\mathrm{x}_1(t) - x_{ob}(t))^{m-1}}{\Big((\mathrm{x}_1(t) - x_{ob}(t))^m(\mathrm{x}_4(t) - y_{ob}(t))^m\Big)^2}\dot{x}_{ob}(t)\\ &\quad + \frac{m\ell(\mathrm{x}_4(t) - y_{ob}(t))^{m-1}}{\Big((\mathrm{x}_1(t) - x_{ob}(t))^m + (\mathrm{x}_4(t) - y_{ob}(t))^m\Big)^2}\dot{y}_{ob}(t).\end{aligned} \tag{62}$$

To calculate (62), it is necessary to have the velocities of the obstacle in the x and y-direction. By having the variation of the impulsive potential function in (61) and adding to the functional in (24), we can handle the dynamic obstacle avoidance.

IV. ANALYTIC EXPRESSION OF THE OPTIMAL CONTROL INPUTS

Majority of motion planning problems cannot be solved analytically. The differential equations related to the optimal control problem are generally non-linear and therefore the solutions (if they exist) can be only represented numerically. However, analytic closed-form expressions approximating the numerical solutions offer several advantages: 1. Oversampling, thus, they provide a quick methodology to evaluate intermediate points. 2. In the context of optimal control, they can be used to express the control inputs, allowing derivation and integration for state estimation. 3. They are often smooth functions (e.g. polynomials [17]) and smoothness in control inputs is a desired property [19]. Furthermore, they can be easily implemented in microcontrollers.

To approximate the numerical solutions obtained in Section III, we firstly use polynomials as basis functions and we exploit the Recursive Smooth Trajectory (RST) generation method recently introduced in [19] and [20]. Lastly, we introduce a novel interpolation method, based on the RST algorithm, which utilizes as basis the rational functions.

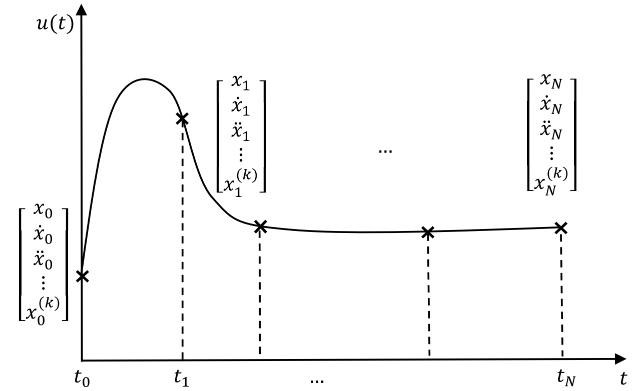

$u(t)$
$\begin{bmatrix} x_0 \\ \dot{x}_0 \\ \ddot{x}_0 \\ \vdots \\ x_0^{(k)} \end{bmatrix}$
$\begin{bmatrix} x_1 \\ \dot{x}_1 \\ \ddot{x}_1 \\ \vdots \\ x_1^{(k)} \end{bmatrix}$
...
$\begin{bmatrix} x_N \\ \dot{x}_N \\ \ddot{x}_N \\ \vdots \\ x_N^{(k)} \end{bmatrix}$
t_0
t_1
...
t_N
t

FIGURE 4. Sampling of the control input.

A. RECURSIVE SMOOTH TRAJECTORY

The RST algorithm was proposed in [19] and [20] as an iterative interpolation method. The idea is to recursively build an unique polynomial function $f_k(t)$ knowing the associated kinematic constraints $\frac{d^i}{dt^i} f_k(t)\Big|_{t=t_j} = \sigma_{i,j}$ for each point in time t_j with $j = 0, 1, \dots, N$ and $i = 0, 1, \dots k$, where $N + 1$ and k are the number of waypoints and the number of derivative constraints, respectively. Indeed, it has been shown in [19] and [20] that, defined the polynomial function as

$$f_k(t) = \sum_{i=0}^{k} p_i(t), \tag{63}$$

if

$$p_i(t) = \frac{1}{i!}\left(\prod_{n=0}^{N}(t - t_n)\right)^i \cdot s_i(t), \tag{64}$$

then the i-partial function (See Def.1 in [20]) $f_i(t)$ depends recursively on $f_{i-1}(t)$. In particular,

$$s_i(t_j) = \frac{\frac{d^i}{dt^i} f_k(t)\Big|_{t=t_j} - \frac{d^i}{dt^i} f_{i-1}(t)\Big|_{t=t_j}}{\left(\prod_{\substack{n=0 \\ n\neq j}}^{N}(t_j - t_n)\right)^i}. \tag{65}$$

Therefore, it is possible to component-wise approximate the optimal control input $u(t)$ with a polynomial $f_k(t)$. The idea is to sample the desired input into $N + 1$ waypoints $(x_0, x_1, \dots, x_N = x_T)$ and for each of them provide the information on its derivatives up to the k-th derivative, denoted as kinematic constraints (see Fig. 4).

Unfortunately, when the number of waypoints $N + 1$ is large and when t_j are equally spaced, the Runge's phenomenon may occur [29]. The Runge's phenomenon represents an unwanted oscillation near the endpoints of the polynomial interpolation function. To avoid it, one possibility relies on spline interpolation methods [18] that connect low order polynomial functions. However, if the estimation of the states is required, it is necessary to solve a set of linear equations to find the constants of integration. With a unique function, the number of linear equations to solve reduces significantly. Another way to tackle the oscillation problem consists of changing the distribution of the nodes t_j more densely towards the edges of the interval $[t_0;\ t_N]$ [30]. Since the sampling process is designed by the user, a standard choice considers the set of points in time as the set of Chebyshev nodes. In particular, for $N + 1$ points in the interval $[t_0;\ t_N]$, nodes are transformed into

$$\hat{t}_j = \frac{1}{2}(t_0 + t_N) + \frac{1}{2}(t_N - t_0)\cos\left[\frac{2j+1}{2(N+1)}\pi\right], j = 0, \dots, N. \tag{66}$$

Runge's phenomenon can be avoided also with optimization on the polynomial interpolant. Indeed, it has been shown that the set of feasible functions satisfying the kinematic constraints $\sigma_{i,j}$ can be expressed as an induced set by the polynomial $q(t)$ as follows

$$f_{\text{ext}}(t) = f_k(t) + \frac{\left(\prod_{n=0}^{N}(t_j - t_n)\right)^{k+1}}{(k+1)!} \cdot q(t). \tag{67}$$

To minimize the oscillation effect, we can look for the minimal energy error function as the solution to

$$\min_{q(t)} \int_{t_0}^{t_N} \left\| \left(u(t) - f_k(t) - \frac{\left(\prod_{n=0}^{N}(t_j - t_n)\right)^{k+1}}{(k+1)!} \cdot q(t) \right) \right\|^2 dt. \tag{68}$$

However, such optimization process is computationally expensive, therefore we propose to use RST, exploiting its efficiency advantage.

B. RATIONAL RECURSIVE SMOOTH TRAJECTORY

Polynomial interpolation is in general a simple and fast process to implement. Nevertheless, when the degree of the interpolant function is high, oscillation at the edges may occur as mentioned before. For this reason, we consider a different basis function which may take advantage of the simplicity of polynomials but also provide more flexibility and degrees of freedom to tackle Runge's phenomenon. We identify and propose such basis as the rational basis function

$$R_{n,d}(t) = \frac{N(t)}{D(t)}, \tag{69}$$

where $N(t)$ is the numerator, a polynomial of degree n, and $D(t)$ is the denominator, a polynomial of degree d. Such choice allows us to exploit some of the polynomial properties for both numerator and denominator but most importantly, enables the development of a new algorithm, referred to as rational recursive smooth trajectory (RRST). To find the coefficients of both numerator and denominator, the idea is to pick the denominator $D(t)$ and use RST to find the coefficients of the numerator $N(t)$. Intuitively, the new kinematic constraints for building $N(t)$ are a weighted sum of the kinematic constraints $\frac{d^i}{dt^i}f_k(t)\Big|_{t=t_j}$ (given) and the kinematic constraints $\frac{d^i}{dt^i}D(t)\Big|_{t=t_j}$ (designed as input). The following Lemma provides the mathematical formulation for the RRST.

Lemma 1: Let t_j be a point in time, for $j = 0, 1, \ldots, N$, such that $\frac{d^i}{dt^i}f_k(t)\big|_{t=t_j}$ is the associated given kinematic constraint, for $i = 0, 1, \ldots, k$. Let $N(t)$ be a polynomial and $D(t)$ be a given polynomial of degree d. If $f_k(t)$ is a rational function defined as

$$f_k(t) = R_{n,d}(t) = \frac{N(t)}{D(t)}, \tag{70}$$

with $n = (k+1)(N+1) - 1$, then the coefficients of $N(t)$ can be obtained with RST, in particular its associated kinematic constraint has expression

$$\frac{d^i}{dt^i}N(t)\bigg|_{t=t_j} = \sum_{l=0}^{i}\binom{i}{l}\left(\frac{d^l}{dt^l}f_k(t)\big|_{t=t_j}\right)\cdot\left(\frac{d^{i-l}}{dt^{i-l}}D(t)\big|_{t=t_j}\right). \tag{71}$$

Proof: For simplicity of notation, the rational function $R_{n,d}(t)$ will be denoted with $R(t)$. We proceed by induction on the kinematic constraint. Consider the case when $i = 0$, then

$$N(t_j) = R(t_j) \cdot D(t_j) \tag{72}$$

represents the value that $N(t)$ needs to assume at the time t_j. For the case $i = 1$

$$\frac{d}{dt}N(t)\bigg|_{t=t_j} = \frac{d}{dt}\Big(R(t)\cdot D(t)\Big)\bigg|_{t=t_j} \tag{73}$$

which is equal to

$$\frac{d}{dt}N(t)\bigg|_{t=t_j} = \binom{1}{0}R(t_j)\cdot\left(\frac{d}{dt}D(t)\big|_{t=t_j}\right) + \binom{1}{1}\left(\frac{d}{dt}R(t)\big|_{t=t_j}\right)\cdot D(t_j). \tag{74}$$

Suppose that the statement of the lemma is true for the case i, which means that

$$\frac{d^i}{dt^i}N(t)\bigg|_{t=t_j} = \sum_{l=0}^{i}\binom{i}{l}\left(\frac{d^l}{dt^l}R(t)\big|_{t=t_j}\right)\cdot\left(\frac{d^{i-l}}{dt^{i-l}}D(t)\big|_{t=t_j}\right). \tag{75}$$

Then, it is true also for the case $i + 1$. Indeed

$$\begin{aligned}
\frac{d^{i+1}}{dt^{i+1}}N(t)\bigg|_{t=t_j}
&= \frac{d}{dt}\sum_{l=0}^{i}\binom{i}{l}\left(\frac{d^l}{dt^l}R(t)\big|_{t=t_j}\right)\cdot\left(\frac{d^{i-l}}{dt^{i-1}}D(t)\big|_{t=t_j}\right)\\
&= \sum_{l=0}^{i}\binom{i}{l}\frac{d}{dt}\left[\left(\frac{d^l}{dt^l}R(t)\big|_{t=t_j}\right)\cdot\left(\frac{d^{i-l}}{dt^{i-1}}D(t)\big|_{t=t_j}\right)\right]\\
&= \sum_{l=0}^{i}\binom{i}{l}\left(\frac{d^{l+1}}{dt^{l+1}}R(t)\big|_{t=t_j}\right)\cdot\left(\frac{d^{i-l}}{dt^{i-1}}D(t)\big|_{t=t_j}\right)\\
&\quad + \sum_{l=0}^{i}\binom{i}{l}\left(\frac{d^l}{dt^l}R(t)\big|_{t=t_j}\right)\cdot\left(\frac{d^{i+1-l}}{dt^{i+1-l}}D(t)\big|_{t=t_j}\right)
\end{aligned} \tag{76}$$

where we used the linearity of the differential operator and the product rule. With a change of variable in the first term of the RHS, $h = l + 1$, it follows that

$$\begin{aligned}
\frac{d^{i+1}}{dt^{i+1}}N(t)\bigg|_{t=t_j}
&= \sum_{h=1}^{i+1}\binom{i}{h-1}\left(\frac{d^h}{dt^h}R(t)\big|_{t=t_j}\right)\cdot\left(\frac{d^{i+1-h}}{dt^{i+1-h}}D(t)\big|_{t=t_j}\right)\\
&\quad + \sum_{l=0}^{i}\binom{i}{l}\left(\frac{d^l}{dt^l}R(t)\big|_{t=t_j}\right)\cdot\left(\frac{d^{i+1-l}}{dt^{i+1-l}}D(t)\big|_{t=t_j}\right)\\
&= \sum_{l=0}^{i+1}\binom{i+1}{l}\left(\frac{d^l}{dt^l}R(t)\big|_{t=t_j}\right)\cdot\left(\frac{d^{i+1-l}}{dt^{i+1-l}}D(t)\big|_{t=t_j}\right)
\end{aligned}$$

where we used the Pascal's identity

$$\binom{i+1}{l} = \binom{i}{l-1} + \binom{i}{l}. \tag{77}$$

Hence the result is true for $i+1$ and by induction is true for all positive integers. From Corollary 1.1 of [20], the minimum degree n of $N(t)$ is $(k + 1)(N + 1) - 1$. ■

Lemma 1 provides the general expression of the kinematic constraints associated to $N(t)$, however it assumes that the denominator $D(t)$ is given. The choice of the denominator remains an open question in this paper although some considerations can be made. The denominator represents a whole set of degrees of freedom and therefore the choice of the coefficients should in principle consider some strategies. For example, a fundamental aspect is the position of the roots inside the interval $[t_0, t_N]$. Indeed, if one real pole (denominator root) falls inside the desired interval, it may cause discontinuities in the rational interpolant. To avoid this, a possible strategy relies on the selection of multiple complex conjugate roots. Further studies have to be made in the roots locus analysis for such rational function but they go out of the scope of this paper therefore we postpone these questions to future work. Finally, it is interesting to notice that if the denominator $D(t)$ is constant, we lead back to the classical polynomial interpolation via RST, therefore we can tract RRST as a rational basis extension of the RST algorithm.

To facilitate the implementation of the RRST algorithm, we report the pseudo code in Tab. 1.

To show how the RRST tackles the oscillation problem, we report in Fig. 5 an example of function approximation with polynomials (RST) and rational functions (RRST). In particular, we select as function to interpolate $f_k(t) = \arctan(\pi t)$, with $t \in [-1, 1]$. Fig. 5 illustrates the resulting interpolants when the number of waypoints is set to 10 and no kinematic constraints (from velocity on) are imposed. The denominator

Algorithm 1 Rational Recursive Smooth Trajectory (RRST)

```
Inputs:
    $N+1$ points in time $t_0 < t_1 < \cdots < t_N$;
    Number of derivatives $k$ to fulfill;
    Kin. constr.
    $\frac{d^i}{dt^i} f_k(t)\big|_{t=t_0}, \ldots, \frac{d^i}{dt^i} f_k(t)\big|_{t=t_N}$;
    Denominator $D(t)$ of degree $d$.
Initialize:
    Kin. constr.
    $\frac{d^i}{dt^i} D(t)\big|_{t=t_0}, \ldots, \frac{d^i}{dt^i} D(t)\big|_{t=t_N}$;
for $i = 0$ to $k$ do
    for $j = 0$ to $N$ do
        $\frac{d^i}{dt^i} N(t)\Big|_{t=t_j} =$
        $\sum_{l=0}^{i} \binom{i}{l} \left( \frac{d^l}{dt^l} f_k(t)\big|_{t=t_j} \right) \cdot \left( \frac{d^{i-l}}{dt^{i-l}} D(t)\big|_{t=t_j} \right)$;
    end for
end for
Get $N(t)$ with RST given the kinematic constraints $\frac{d^i}{dt^i} N(t)\Big|_{t=t_j}$ as input;
$f_k(t) = \frac{N(t)}{D(t)}$.
```

of the rational function is set to $D(t) = t^2 + 0.1$ and for such choice, RRST shows to perform better than the polynomial interpolant at the edges.

V. RESULTS

This section presents several simulation results to show how our methodology can solve the problem of generating optimal and smooth trajectories avoiding an obstacle without the need to define extra waypoints in the search space. First, we start with an obstacle avoidance scenario that can be used for real-time applications. We perform a computer simulation on the path planner dynamics using the optimal control methodology which was designed in Section III. Then, we compare the performance of the proposed optimal method concerning other strategies.

A. OBSTACLE AVOIDANCE

We solve the proposed motion planning problem with different values of ℓ. We start by considering $\ell = 0.01$, and then we increase ℓ until the influence of the potential function (navigation function) in (23) is robust enough to avoid the obstacle. Therefore, we choose $\ell = 0.02, 0.04, 0.06, 0.08$ and the total time is set to $T = 4$s. The obstacle is centered at the point $(x_{ob}, y_{ob}) = (0.5, 0.5)$ m in the search space. The objective is to generate a feasible trajectory from an initial to a final condition. To do so, we impose the initial and final states to be $\mathrm{x}_1(0) = 0, \mathrm{x}_2(0) = 0, \mathrm{x}_3(0) = 0, \mathrm{x}_4(0) = 0, \mathrm{x}_5(0) = 0, \mathrm{x}_6(0) = 0$ and $\mathrm{x}_1(T) = 1, \mathrm{x}_2(T) = 0, \mathrm{x}_3(T) = 0, \mathrm{x}_4(T) = 1, \mathrm{x}_5(T) = 0, \mathrm{x}_6(T) = 0$.

The optimal trajectory is shown in Fig. 6. It should be noted that increasing the parameter ℓ results in a higher related cost functional $J_{ob} = \frac{1}{2} \int_0^T \Big(\mathrm{x}_1^2(t) + \mathrm{x}_4^2(t) + u_1^2(t) + u_2^2(t) + V(\mathrm{x}(t), \mathrm{y}(t), x_{ob}, y_{ob}) \Big) dt$ and safety margin from the obstacle. The states (position, velocity and acceleration) and the control inputs for different values of ℓ are shown in Fig. 7, 8. The results illustrate how our methodology is capable of generating an optimal path avoiding an obstacle, without extra waypoints.

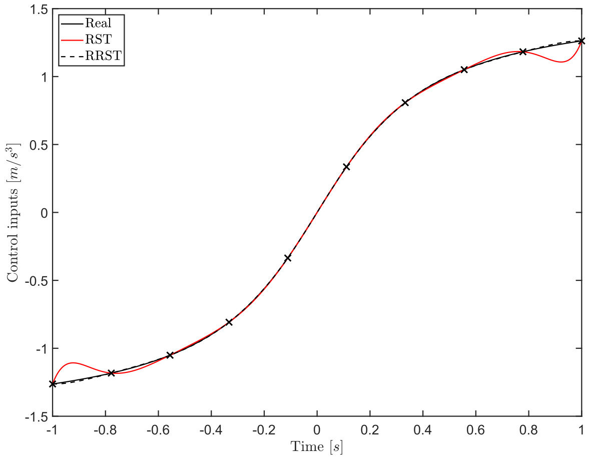


FIGURE 5. Comparison between polynomial (RST) and rational (RRST) interpolation of 10 waypoints, obtained as samples of the analytic control input arctan(πt).

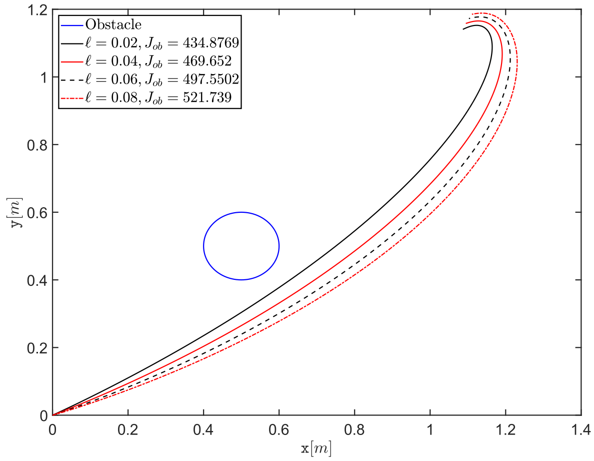


FIGURE 6. The optimal path obtained solving the closed-from differential equations in (28-35).

So far, the control inputs $u(t) = [u_1(t) \; u_2(t)]^\top$ are obtained in an offline stage and lack of an analytic expression. To ease the implementation of control laws in microcontrollers, we propose to consider a set of waypoints sampled from the control signals and interpolate them with unique polynomial and rational functions, exploiting the RST and RRST algorithms, respectively. As representative scenarios, we analyze the impact of the number of waypoints $N + 1$ and the depth of the kinematic constraints k on the goodness of approximation for both the basic functions. We choose as denominator $D(t) = t^2 + 10$. Fig. 9 illustrates an example of interpolation of the control input $u_1(t)$ (with no closed-form

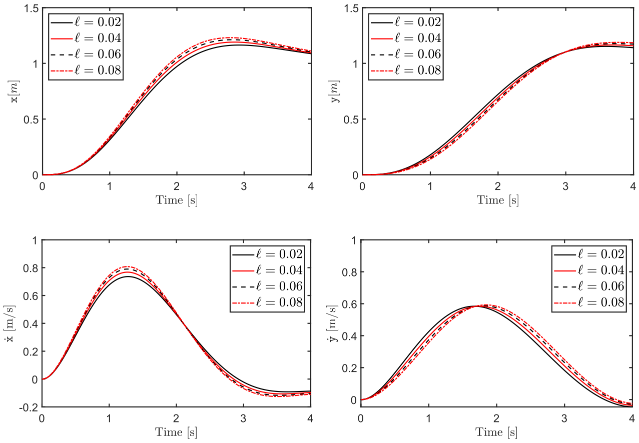


FIGURE 7. Evolution of the states obtained solving the closed-from differential equations in (28-35).

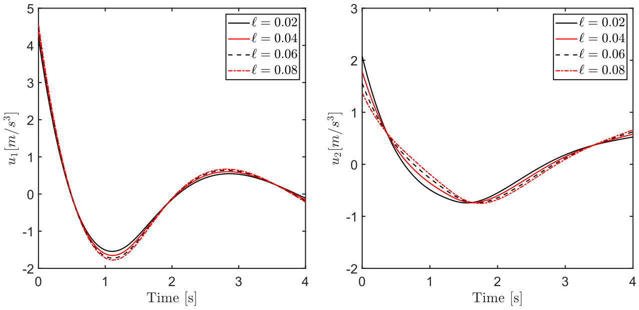


FIGURE 8. Evolution of the control inputs $u(t) = [u_1(t)\;\; u_2(t)]^\top$ obtained solving the closed-from differential equations in (28-35).

expression available). In particular, four cases have been studied: Fig. 9 (a) shows the two interpolants passing through 12 waypoints with no information on the kinematic constraints (only position is given); Fig. 9 (b) compares the two functions passing through 6 waypoints where only the information on the velocity was given on each of them ($k = 1$); Fig. 9 (c) shows how both functions try to approximate the control signal when only 4 waypoints are selected but with information on velocity and acceleration; Lastly, Fig. 9 (d) displays the two interpolants passing through only 3 waypoints with information up to jerk on each of them ($k = 3$). It is easy to notice that under the same polynomial degree ($(k + 1)(N + 1) - 1$ is constant in the 4 examples), the best approximation is obtained when we consider more waypoints, as expected. However, when N is large, to avoid oscillation, rational functions show better performance. Moreover, in all the other cases (b)-(d), the coefficients of the denominator in the rational function have significant influence on the interpolant behaviour and therefore this suggests to conduct future studies.

B. CLUTTERED ENVIRONMENT

An effective comparison of performance is considered between the proposed method and other strategies: the MPC [23] and the APF [21], [22] approaches in a cluttered

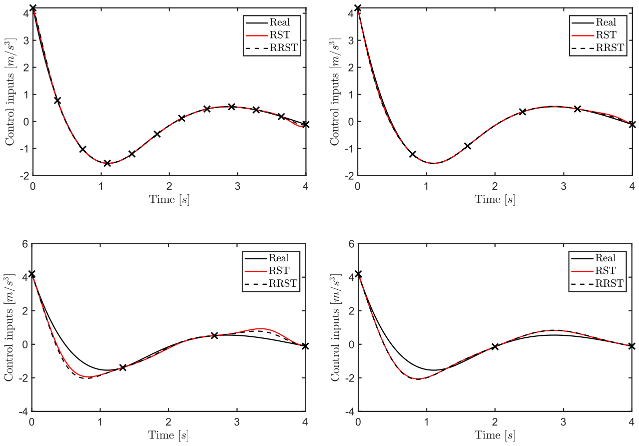


FIGURE 9. Comparison of control input approximation via RST and RRST. (a) Top-left $N = 11$, $k = 0$. (b) Top-right $N = 5$, $k = 1$. (c) Bottom-left $N = 3$, $k = 2$. (d) Bottom-right $N = 2$, $k = 3$.

environment. The planner starts at (0, 0) m encountering six obstacles at $(x_{ob_1}, y_{ob_1}) = (0.5, 0.5)$ m, $(x_{ob_2}, y_{ob_2}) = (0.8, 0.3)$ m, $(x_{ob_3}, y_{ob_3}) = (0.6, 0.2)$ m, $(x_{ob_4}, y_{ob_4}) = (0.8, 0.8)$ m, $(x_{ob_5}, y_{ob_5}) = (0.8, 0.1)$ m, and $(x_{ob_6}, y_{ob_6}) = (0.2, 0.8)$ m and reaches the target at (1, 1) m. The Fig. 10 shows the trajectories generated by the MPC [23], the APF [21], [22], along with the proposed method. It is clear that the APF generates a trajectory that suffers from discontinuity in the control inputs. However, as shown in Fig. 10, the summation of a repulsive potential field may result in local minima. The trajectory generated by APF is not optimal and passes between two obstacles from the starting position to the target position. Two obstacles are close enough to have zero gradients in the summation of attractive and repulsive potentials. The proposed methodology avoids the well-known local minima phenomena due to the superposition of potential fields. Indeed, in Problem 2, we added repulsive potential functions and other constraints into the dynamic interpolation framework and following the constrained Hamiltonian formalism that allows obtaining the optimal trajectory.

In the MPC methodology, the trajectory is generated from the initial condition (0, 0) m and fed to the interior-point solver to solve the constrained nonlinear optimization during the navigation. The discontinuity also can be observed by the MPC technique. The significant difference between the performance of the proposed framework over the MPC and APF strategies is the lack of oscillations when approaching the obstacles, with our approach resulting in smooth controllers. In [21], authors suggested switching potential functions to avoid local minima. However, the discontinuity in the control inputs due to the switching functions cannot be removed.

Moreover, Fig. 10 (top) shows that the settling time is also sluggish and significant overshoot exists for the APF method in regions of the relatively flat gradient. The MPC method fails in avoiding obstacles while the proposed technique generates smooth and feasible trajectory. The MPC, on the other

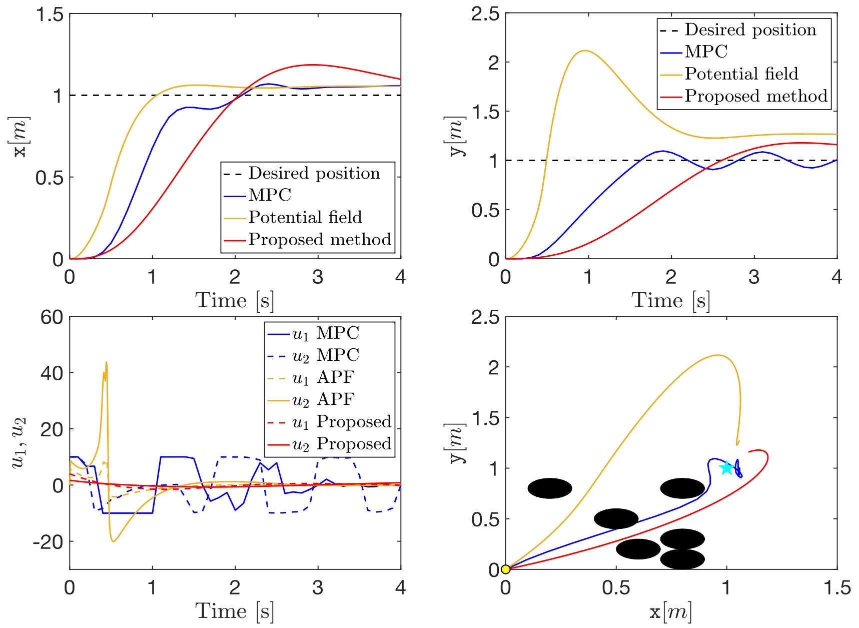


FIGURE 10. Comparison of trajectories generated by different methodologies: the MPC [23] and the APF [21], [22] based methods in a cluttered environment over the proposed one.

TABLE 1. Comparison of execution time for different methodologies.

Strategies	Execution time
MPC [23]	11.375627 s
APF [21], [22]	0.013104 s
Proposed method	0.424412 s

hand, requires a very high computational execution time (see table 1) compared to our proposed method.

Remark: The limitation of the proposed methodology is the tuning parameter ℓ in the repulsive potential function in (23). As the number of obstacles increases, there is a need to find the optimal values for each obstacle function.

VI. CONCLUSION

We investigated the problem of path planning for mechanical UAV systems. We showed that by using Pontryagin's Minimum Principle, it is possible to derive a set of closed-form differential equations in which the dynamic environment is considered in the constrained Hamiltonian function. The same closed-from differential equations are obtained via Lagrangian formalism. Finally, we proposed a novel interpolation algorithm based on rational functions, referred to as a rational recursive smooth trajectory (RRST) method. The algorithm can effectively generate an analytic expression that approximates control inputs for which no closed-form solution is in general attainable.

REFERENCES

[1] B. Salamat and A. Tonello, "Stochastic trajectory generation using particle swarm optimization for quadrotor unmanned aerial vehicles (UAVs)," *Aerospace*, vol. 4, no. 2, p. 27, May 2017.

[2] B. Salamat and A. M. Tonello, "A generalized multi-objective framework for UAV mission planning," in *Proc. IEEE Aerosp. Conf.*, Mar. 2019, pp. 1–6.

[3] C. Altafini, "Reduction by group symmetry of second order variational problems on a semidirect product of lie groups with positive definite Riemannian metric," *ESAIM, Control, Optim. Calculus Variat.*, vol. 10, no. 4, pp. 526–548, Oct. 2004.

[4] L. Colombo and D. M. de Diego, "Higher-order variational problems on lie groups and optimal control applications," *J. Geometric Mech.*, vol. 6, no. 4, p. 451, 2014.

[5] P. Crouch and F. S. Leite, "Geometry and the dynamic interpolation problem," in *Proc. Amer. Control Conf.*, Jun. 1991, pp. 1131–1136.

[6] I. I. Hussein and A. M. Bloch, "Dynamic coverage optimal control for multiple spacecraft interferometric imaging," *J. Dyn. Control Syst.*, vol. 13, no. 1, pp. 69–93, Feb. 2007.

[7] L. Noakes, G. Heinzinger, and B. Paden, "Cubic splines on curved spaces," *IMA J. Math. Control Inf.*, vol. 6, no. 4, pp. 465–473, Dec. 1989.

[8] M. Zefran, V. Kumar, and C. B. Croke, "On the generation of smooth three-dimensional rigid body motions," *IEEE Trans. Robot. Autom.*, vol. 14, no. 4, pp. 576–589, Aug. 1998.

[9] K. Biswas and I. Kar, "An optimal solution to fixed time horizon moving target tracking with obstacle avoidance," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2016, pp. 1610–1615.

[10] D. Kirk, D. Kirk, and D. Kreider, *Optimal Control Theory: An Introduction*. Upper Saddle River, NJ, USA: Prentice-Hall, 1970.

[11] S. Liu, N. Atanasov, K. Mohta, and V. Kumar, "Search-based motion planning for quadrotors using linear quadratic minimum time control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 2872–2879.

[12] S. Liu, K. Mohta, N. Atanasov, and V. Kumar, "Search-based motion planning for aggressive flight in SE(3)," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 2439–2446, Jul. 2018.

[13] P. E. Crouch and J. W. Jackson, *A Non-Holonomic Dynamic Interpolation Problem*. Boston, MA, USA: Birkhäuser, 1991, pp. 156–166.

[14] G. Hoffmann, S. Waslander, and C. Tomlin, "Quadrotor helicopter trajectory tracking control," in *Proc. AIAA Guid., Navigat. Control Conf. Exhibit*, Aug. 2008, p. 7410.

[15] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "A prototype of an autonomous controller for a quadrotor UAV," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2007, pp. 4001–4008.

[16] B. Salamat and A. M. Tonello, "Novel trajectory generation and adaptive evolutionary feedback controller for quadrotors," in *Proc. IEEE Aerosp. Conf.*, Mar. 2018, pp. 1–8.

[17] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2520–2525.

[18] Y. Bouktir, M. Haddad, and T. Chettibi, "Trajectory planning for a quadrotor helicopter," in *Proc. 16th Medit. Conf. Control Autom.*, Jun. 2008, pp. 1258–1263.

[19] N. A. Letizia, B. Salamat, and A. M. Tonello, "A new recursive framework for trajectory generation of UAVs," in *Proc. IEEE Aerosp. Conf.*, Mar. 2020, pp. 1–8.

[20] N. A. Letizia, B. Salamat, and A. M. Tonello, "A novel recursive smooth trajectory generation method for unmanned vehicles," *IEEE Trans. Robot.*, early access, Feb. 22, 2021, doi: 10.1109/TRO.2021.3053649.

[21] G. Fedele, L. D'Alfonso, F. Chiaravalloti, and G. D'Aquila, "Obstacles avoidance based on switching potential functions," *J. Intell. Robot. Syst.*, vol. 90, nos. 3–4, pp. 387–405, Jun. 2018.

[22] A. C. Woods and H. M. La, "A novel potential field controller for use on aerial robots," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 4, pp. 665–676, Apr. 2019.

[23] S. Lai, M. Lan, and B. M. Chen, "Model predictive local motion planning with boundary state constrained primitives," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3577–3584, Oct. 2019.

[24] N. Boizot and J. Gauthier, "Motion planning for kinematic systems," *IEEE Trans. Autom. Control*, vol. 58, no. 6, pp. 1430–1442, Jun. 2013.

[25] K. Lynch and F. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[26] H. B. Curry, "The method of steepest descent for non-linear minimization problems," *Quart. Appl. Math.*, vol. 2, no. 3, pp. 258–261, 1944.

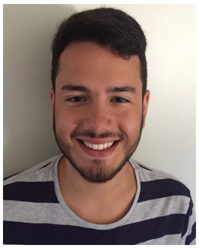
[27] D. Bertsekas, "Partial conjugate gradient methods for a class of optimal control problems," *IEEE Trans. Autom. Control*, vol. AC-19, no. 3, pp. 209–217, Jun. 1974.

[28] K. Vogtmann, A. Weinstein, and V. Arnol'd, *Mathematical Methods of Classical Mechanics* (Graduate Texts in Mathematics). New York, NY, USA: Springer, 1997.

[29] C. Runge, "Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten," *Zeitschrift Math. Phys.*, vol. 46, pp. 224–243, 1901.

[30] J.-P. Berrut and L. N. Trefethen, "Barycentric Lagrange interpolation," *SIAM Rev.*, vol. 46, no. 3, pp. 501–517, 2004.

BABAK SALAMAT received the B.S. degree in mechanical engineering and the M.S. degree in aerospace engineering from the Air-force University of Shahid Sattari, Tehran, Iran, in 2012 and 2014, respectively, and the Ph.D. degree in information technology with application to aerospace vehicles from the University of Klagenfurt, Austria, in 2021. His current research interests include control theory, path planning, robust control for autonomous aerial vehicles, and evolutionary algorithms. He was a co-recipient with A. Tonello of the 2018 Best Paper Award in the *Aerospace* journal.



NUNZIO A. LETIZIA (Graduate Student Member, IEEE) received the Laurea Magistrale degree *(summa cum laude)* in electrical engineering from the University of Udine, Italy, in 2018. He is currently pursuing the Ph.D. degree in information technology with the University of Klagenfurt, Austria. During his studies, he was awarded with a full scholarship at Scuola Superiore dell'Universitá degli studi di Udine. His research interests include spread from mathematics, physics to communication engineering, signal processing, and statistical learning.

ANDREA M. TONELLO (Senior Member, IEEE) received the D.Eng. degree (Hons.) in electronics and the D.Res. degree in electronics and telecommunications from the University of Padova, Italy, in 1996 and 2002, respectively. From 1997 to 2002, he was with Bell Labs-Lucent Technologies, Whippany, NJ, USA, as a member of the Technical Staff. Then, he was promoted to the Technical Manager and appointed to the Managing Director of the Bell Labs Italy Division. In 2003, he joined the University of Udine, Udine, Italy, where he became an Aggregate Professor, in 2005, and an Associate Professor, in 2014. He is currently the Chair of the Embedded Communication Systems Group, University of Klagenfurt, Klagenfurt, Austria. He is also the Founder of a spinoff company, WiTiKee. He received several awards, including the Distinguished Visiting Fellowship from the Royal Academy of Engineering, U.K., in 2010, the IEEE Vehicular Technology Society (VTS) and the ComSoc Distinguished Lecturer Awards (2011, 2015, and 2018), the Chair of Excellence from UC3M (2019–2020), and nine best paper awards. He served as the Chair for the IEEE ComSoc TC-PLC. He currently serves as the Chair for the IEEE ComSoc TC-SGC, and he has been appointed as the Director of the Industry Outreach of ComSoc. He served/s as an Associate Editor for the IEEE Transactions on Vehicular Technology, IEEE Transactions on Communications, IEEE Access, and *IET Smart Grid*.