Data-driven Adaptive Optimal Control under Model Uncertainty: An Application to Cold Atmospheric Plasmas

Diogo Rodrigues, Kimberly J. Chan, and Ali Mesbah

Abstract— **Cold atmospheric plasmas (CAPs) are increasingly used for applications requiring the processing of heat- and pressure-sensitive (bio)materials. A key challenge in model-based control of CAPs arises from the high computational requirements of theoretical plasma models as well as lack of mechanistic understanding of plasma-surface interactions. Thus, control strategies that rely on simple, physics-based models that can be adapted to mitigate plant-model mismatch will be particularly advantageous for CAP applications. This paper presents an optimal control approach for controlling the nonlinear and cumulative effects of CAPs delivered to a target surface using a simple system model. Through parsimonious input parameterization, the solution to the optimal control problem (OCP) is given by an arc sequence that does not include any singular arcs. A data-driven adaptive algorithm based on modifier adaptation is proposed to deal with the structural plant-model mismatch by estimating the mismatch in the cost and constraints of the OCP. The adaptive approach is shown to converge to a Karush-Kuhn-Tucker (KKT) point of the OCP for the true system. Moreover, a control strategy based on feedback linearization and derivative estimation is proposed for online tracking of path constraints in the presence of disturbances and model uncertainty. The proposed approach is demonstrated by simulations and real-time control experiments on a kHz-excited atmospheric pressure plasma jet in Helium, in which the plasma treatment time is minimized while delivering a desired amount of nonlinear thermal effects to the target surface.**

Index Terms— **optimal control; modifier adaptation; feedback linearization; derivative estimation; cold atmospheric plasmas.**

I. Introduction

Cold atmospheric plasmas (CAPs) are a class of plasmas generated in ambient conditions. These plasmas have been shown to be uniquely promising for the treatment of heat- and pressure-sensitive (bio)materials [1]–[3]. This is due to the non-equilibrium nature of the generated plasma (i.e., the electron temperature is very high, around 1-5 keV, while the temperature of a background gas is close to room temperature). Several reactions with the background gas and the ambient air generate reactive species, as well as heat, radiation, and electric fields. This induces effects that are useful for industrial processing and manufacturing, surface activation and modification [4], [5], as well as biomedical applications such as plasma medicine [6], [7]. Atmospheric pressure plasma jets (APPJs) have emerged as a popular class of CAP discharge devices [4], [8], [9]. In APPJs, an electrode is placed in or around a dielectric tube and an electric field is applied to a flowing gas, which generates a visible plume that extends several centimeters beyond the tip of the dielectric tube [10]. Although the generated plasma and the plasma-surface interactions are difficult to model, APPJs must be operated in a way that is reliable, effective, and time-conscious for treatment, otherwise damage may occur due to the cumulative and nondecreasing nature of the plasma effects on a surface [11]–[13]. This corresponds to the so-called dose delivery problem, where the goal is to deliver a plasma "dose" (i.e., nonlinear and cumulative effects of the plasma) to a target surface in a controllable and reproducible manner. This problem is challenging because the APPJ may need to move across the target surface, the surface properties may change over time and space, and the operation is sensitive to various disturbances such as the tip-to-surface distance and ambient conditions. Moreover, the plasma behavior is intrinsically variable and subject to transition between different modes [14], [15].

Recent work has shown the promise of predictive control and learning-based control approaches for adaptive and automated operation of APPJs [16]–[18]. Reduced-order, physics-based models [19] as well as data-driven models based on subspace identification [20] and linear parameter varying identification [21] have been used to develop model predictive control (MPC) strategies for constrained, multivariable control of the nonlinear thermal effects of plasma on surfaces. Hierarchical predictive control strategies have also been shown to be useful in achieving spatially uniform thermal dose delivery on one-dimensional and two-dimensional surfaces [22], [23]. Furthermore, experimental studies using cell and chemical assays have correlated operational parameters of CAPs such that real-time monitoring of the plasma's chemical effects can be effectively utilized in an adaptive manner for feedback control [24]. These studies were extended to develop data-driven models and MPC-based strategies for targeted cancer cell treatment [25], [26]. In additional studies, reinforcement learning [27] and deep learning methods [28], [29] demonstrated effective control of APPJs for various applications.

Although recent work has shown that MPC facilitates reproducible and effective operation of APPJs, important challenges remain. MPC for controlling the nonlinear plasma effects (i.e., plasma dose delivery) requires solving a nonlinear optimization problem, which can be computationally expensive and may restrict real-time control to cases with relatively large sampling times. This particularly poses a

This work was supported in part by the Swiss National Science Foundation, project number 184521 and by the US National Science Foundation under Grant 1912772.

Diogo Rodrigues, Kimberly J. Chan, and Ali Mesbah are with the Department of Chemical and Biomolecular Engineering, University of California, Berkeley, CA 94720, USA.
{d.rodrigues,kchan45,mesbah}@berkeley.edu

challenge to the control of fast-sampling CAP applications, as well as to point-of-use and portable applications in which embedded control using resource-limited devices is needed. Furthermore, MPC relies on a descriptive model of the system dynamics that is amenable to online computations. However, theoretical plasma models are computationally expensive for real-time control, and models for complex plasma effects on surfaces are not available due to lack of fundamental understanding of plasma-surface interactions. While there has been a growing interest in learning-based MPC approaches that use reduced-order models combined with online learning (see [30] and the citations therein), these approaches still require the online solution of nonlinear optimization problems. Several approaches for adaptive optimal control of nonlinear systems with model uncertainty have been developed recently, mostly using a data-based paradigm. For example, methods based on fuzzy control and combination of feedforward and feedback control [31], [32] or adaptive state observers [33], [34] were developed to implement adaptive, decentralized, and fault-tolerant optimal control for nonlinear systems, including large-scale systems, and a similar method based on neural networks was proposed by [35]. However, these methods typically require a specific structure for either the model or mismatch, which may not encompass the true system. Alternatively, methods based on adaptive dynamic programming and concepts of deep learning and actor-critic neural networks for reinforcement learning were also developed for model-free optimal control [36]–[39]. However, these approaches can be too general, making it challenging to accurately estimate the model parameters. Hence, these methods may not be suitable when the mismatch between the true system and its model is structural, as it is the case in the CAP application at hand due to the unknown model structure for the plasma effects on surfaces. In this case, it may be more effective to adapt the optimization problem itself by considering its optimality conditions.

This paper presents a data-driven, adaptive optimal control approach for controlling the nonlinear thermal effects of a kHz-excited APPJ in Helium on the target surface. The main advantages of the proposed adaptive optimal control approach include: (i) effective handling of significant plant-model mismatch, which is especially important for plasma applications for which high-fidelity models are not available; (ii) elimination of online optimization, which makes the proposed approach useful for control of fast-sampling APPJs; and (iii) convergence to the optimal solution for the true system, which can be used to improve the efficacy of CAP treatments. First, the optimal control problem (OCP), which can handle nonlinear costs and path constraints, is solved offline using a reduced-order system model. Then, the OCP is combined with a data-driven modifier adaptation algorithm [40]–[42] to mitigate the effects of plant-model mismatch. Modifier adaptation enables satisfying the optimality conditions for the true (unknown) system upon convergence by directly using system measurements to update modifiers that are added to the cost and constraints of the OCP [40]. Hence, an important feature of modifier adaptation is the mitigation of infeasibility or suboptimal performance after convergence. Furthermore, we prove that, if the adaptation converges to a single solution, then this point is a KKT point of the OCP for the true system.

Furthermore, to track active path constraints associated with the optimal control sequence, we propose a control strategy that leverages feedback linearization and derivative estimation. Feedback linearization is a control method for nonlinear systems based on exact linearization [43]–[45]. Although feedback linearization typically requires an adequate model of the nonlinear system to avoid losing the property of exact linearization, significant modeling errors and disturbances are present in practice [46]. Nonetheless, since input-output feedback linearization strategies rely on the knowledge of values related to output derivatives, these derivatives can be inferred from measurements via derivative estimation. In particular, efficient algebraic approaches based on finite impulse response (FIR) filters provide a deadbeat property for the derivative estimates [47], [48]. This combination of feedback linearization and derivative estimation has been shown to provide fast setpoint tracking in the presence of disturbances and significant plant-model mismatch [49].

In this paper, these two control strategies (the data-driven, adaptive OCP combined with feedback linearization and derivative estimation for path constraint tracking) are combined and demonstrated in simulations and real-time control experiments using a kHz-excited APPJ in Helium. To the authors' best knowledge, this paper proposes for the first time the combination of a data-driven, adaptive method for solving OCPs based on modifier adaptation and the use of a control strategy based on feedback linearization and derivative estimation for path constraint tracking. This combination is essential to enable the real-time implementation of the proposed approach for adaptive optimal control. In particular, we address the problem of directly minimizing the plasma treatment time while delivering a desired amount of nonlinear thermal effects to the target surface. Minimizing the treatment time is especially important for plasma treatment of biomaterials, for example, in plasma biomedical applications, which has not been addressed thus far.

II. Problem Description

This section presents the kHz-excited APPJ testbed, a reduced-order, physics-based model for describing the nonlinear thermal effects of the APPJ, and the OCP formulation.

A. APPJ Testbed

We aim to control a kHz-excited APPJ in Helium (He) with prototypical use in biomedical applications. The APPJ consists of a copper ring electrode wrapped around a quartz tube that serves as both a dielectric barrier and the gas flow channel, as depicted in Fig. 1. He gas flows through the tube, and plasma ignition is achieved by applying a high frequency alternating current (AC) voltage to the copper electrode. The generated plasma is directed out of the tube onto a grounded, glass-covered metal plate at a distance of 4

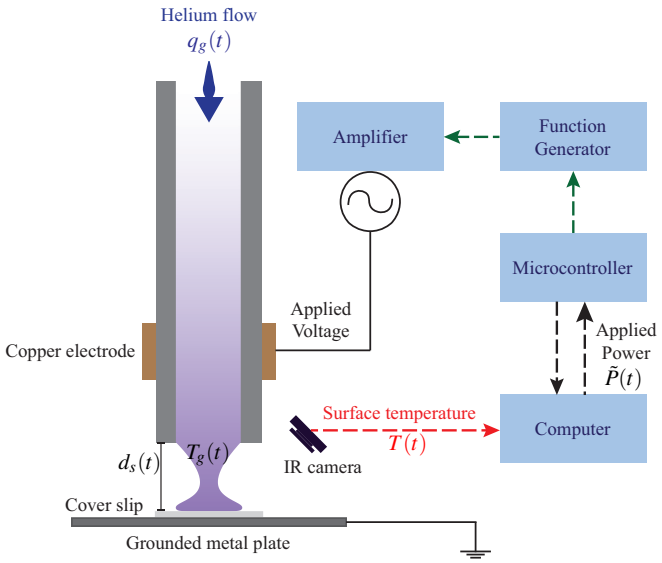

Fig. 1. Schematic of the kHz-excited APPJ in He.

mm below the tip of the tube. In general, the applied power $\tilde{P}$ (W) and He flow rate q_g (slm) are the manipulated inputs to this setup, and the maximum surface temperature T (K) is the measured output. For this study, the He flow rate is set at 3 slm, and the applied power is used to control the maximum surface temperature. Applied power cannot be directly manipulated in the APPJ, but it is implemented using an embedded proportional-integral (PI) controller on a microcontroller (Arduino UNO) that manipulates the applied voltage. Applied voltage is manipulated by generating a sinusoidal waveform at a specified frequency using a function generator (integrated circuit, XR-2602CP). This signal is amplified using an amplifier (TREK 10/40A-HS) before being sent to the copper electrode. Surface temperature is measured through a radiometric infrared thermal camera (Lepton FLIR 3). All advanced control strategies and data acquisition are managed via a standard laptop CPU (2.4 GHz Quad-Core Intel i5 processor) using Python. Offline computation of the OCP is done using custom MATLAB routines, which are compiled to MEX files for efficient computation. Measurements are taken at relatively fast timescales (≈ 0.2 s). Further details on the APPJ testbed can be found in [23].

B. Reduced-Order APPJ Model

We use a reduced-order, physics-based model to describe the thermal effects of plasma on a target surface. The model relies on the fundamental assumption that the mass and energy transport along the plasma occurs at much faster timescales than the surface temperature dynamics [19]. As such, the thermal effects of the APPJ on a surface are described by

$$\frac{\mathrm{d}T}{\mathrm{d}t}(t) = \frac{hq_g(t)^{\gamma}(T_g(t)-T(t))}{\rho c_p d(\pi r^2)^{\gamma}} - \phi\big(T(t)\big), \quad T(t_0)=T_0, \tag{1}$$

and algebraic equations

$$T_{g,0}(t) = \frac{\bar{\eta}}{q_g(t)\rho_g c_{p,g}}\tilde{P}(t) + T_\infty, \tag{2}$$

$$T_g(t) = T_{g,0}(t)\exp\big(-\textstyle\int_0^{d_s(t)} L(z,\frac{q_g(t)}{\pi r^2})^{-1}\mathrm{d}z\big) + T_\infty\Big(1-\exp\big(-\textstyle\int_0^{d_s(t)} L(z,\frac{q_g(t)}{\pi r^2})^{-1}\mathrm{d}z\big)\Big), \tag{3}$$

that imply

$$T_g(t) = \frac{\bar{\eta}}{q_g(t)\rho_g c_{p,g}}\exp\big(-\textstyle\int_0^{d_s(t)} L(z,\frac{q_g(t)}{\pi r^2})^{-1}\mathrm{d}z\big)\tilde{P}(t) + T_\infty, \tag{4}$$

where $T(t)$ is the surface temperature, $\tilde{P}(t)$ is the applied power, $q_g(t)$ is the gas flow rate, $d_s(t)$ is the APPJ tip-to-surface distance, $\phi\big(T(t)\big)$ is an increasing function of $T(t)$, $T_{g,0}(t)$ is the gas temperature at the tip of the APPJ, $T_g(t)$ is the gas temperature at the plasma incident point on the surface, T_0 is the initial surface temperature, $L(z,\frac{q_g(t)}{\pi r^2})$ is a characteristic length that depends on the distance z and $\frac{q_g(t)}{\pi r^2}$, ρ and c_p are the constant density and constant heat capacity of the surface, d is the surface thickness, r is the radius of the plasma jet that results in a section area πr^2, γ is a positive exponent, $hq_g(t)^{\gamma}(\pi r^2)^{-\gamma}$ is a heat transfer coefficient, $\bar{\eta}$ is an efficiency parameter, ρ_g and $c_{p,g}$ are the constant density and constant heat capacity of the gas, and T_∞ is the ambient temperature. A detailed discussion of the model can be found in [19].

Equations (1) and (4) are augmented with the following ordinary differential equation that describes the cumulative and nonlinear thermal effects of the plasma on a surface in the units of cumulative equivalent minutes (CEM)

$$\frac{\mathrm{d}CEM}{\mathrm{d}t}(t) = K^{T_{ref}-T(t)}, \quad CEM(t_0)=0, \tag{5}$$

where K and T_{ref} are constants that relate the surface temperature $T(t)$ to the accumulation of the thermal effects $CEM(t)$ on the surface. According to the definition of CEM in (5), this quantity is used to compare the thermal effects of plasma at different surface temperatures. For example, with the values $K=0.5$ and $T_{ref}=316.15$ K, one can observe that one minute at 316.15 K is equivalent to 15 s at 318.15 K or 4 min at 314.15 K in terms of thermal effects of plasma. We note that CEM was originally proposed to quantify the thermal dose effects in hyperthermia treatment in the 1980's by [50]. Upon fixing a reference value $\bar{d}_s$ for $d_s(t)$ and a constant value $\bar{q}_g$ for $q_g(t)$ and defining

$$C_p := \frac{\rho c_p \pi r^2 d}{\mu}, \tag{6}$$

$$P(t) := \exp\big(-\textstyle\int_{\bar{d}_s}^{d_s(t)} L(z,\frac{\bar{q}_g}{\pi r^2})^{-1}\mathrm{d}z\big)\tilde{P}(t), \tag{7}$$

$$\varphi\big(T(t)\big) := \phi\big(T(t)\big) + \frac{h\bar{q}_g^{\gamma}(T(t)-T_\infty)}{\rho c_p d(\pi r^2)^{\gamma}}, \tag{8}$$

$$\mu := \frac{\bar{\eta}h(\pi r^2)^{1-\gamma}}{\bar{q}_g^{1-\gamma}\rho_g c_{p,g}}\exp\big(-\textstyle\int_0^{\bar{d}_s} L(z,\frac{\bar{q}_g}{\pi r^2})^{-1}\mathrm{d}z\big), \tag{9}$$

the thermal effects of the APPJ can be described by

$$\frac{\mathrm{d}T}{\mathrm{d}t}(t) = \frac{1}{C_p}P(t) - \varphi\big(T(t)\big), \quad T(t_0)=T_0, \tag{10a}$$

$$\frac{\mathrm{d}CEM}{\mathrm{d}t}(t) = K^{T_{ref}-T(t)}, \quad CEM(t_0)=0, \tag{10b}$$

where $T(t)$ and $CEM(t)$ are the states $\mathbf{x}(t)$, $T(t)$ is the measured output $y(t)$, and $P(t)$ is the manipulated input $u(t)$. As such, we arrive at the nonlinear state-space model

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), u(t)\big), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \tag{11a}$$

$$y(t) = \mathbf{c}^{\mathrm{T}}\mathbf{x}(t), \tag{11b}$$

with $\mathbf{x}(t) := \begin{bmatrix} T(t) \\ CEM(t) \end{bmatrix}$, $y(t) := T(t)$, and $u(t) := P(t)$. While the structure of the physics-based model (10) is known, the parameter $C_p > 0$ and the functional relationship $\varphi\big(T(t)\big)$, with $\varphi'\big(T(t)\big) > 0$, are unknown and subject to uncertainty. Thus, although the model (10) is simple in terms of number of states, its structure is fairly general owing to the presence of the term $\varphi\big(T(t)\big)$ with an unknown structure. Additionally, while the power $P(t)$ cannot be directly manipulated in the APPJ, it is possible to control $P(t)$ on fast timescales on the order of milliseconds via the embedded PI controller described in Section II-A. Hence, we can effectively use $P(t)$ as a manipulated input given the slow dynamics of the surface temperature, which is on the order of seconds.

C. Optimal Control Problem

In this work, the control objective is to minimize the plasma treatment time t_f while (i) ensuring that a desired amount CEM_{sp} of thermal plasma effects is delivered to the surface, that is, $CEM_{sp} - CEM(t_f) \leq 0$, and (ii) resolving the surface temperature to its initial value T_0 or lower, that is, $T(t_f) - T_0 \leq 0$. The overarching goal of minimizing treatment time is a significant consideration in various CAP applications (e.g., increasing throughput in biomaterials processing or improving patient experience in plasma medicine). Meanwhile, ensuring the delivery of a specified amount of plasma effects and resolving the surface temperature to its initial value or lower are critical since plasma effects are cumulative and nonlinear (that is, plasma effects cannot be retracted after the effects are delivered). Recall that the dynamics of surface temperature can be relatively slow, which means that a surface temperature above its initial value may continue to affect the substrate despite the end of treatment. Thus, over-administration of plasma effects may result in burning and irreversible damage to the substrate (e.g., tissue), while not delivering enough plasma effects may not produce the desired effects on the substrate (e.g., disinfection) [11]–[14].

A generic formulation of the OCP is given as

$$\min_{u(\cdot), t_f} \quad \mathscr{J}\big(u(\cdot), t_f\big) = \phi\big(\mathbf{x}(t_f), t_f\big), \tag{12a}$$

$$\text{s.t.} \quad \mathscr{T}\big(u(\cdot), t_f\big) = \boldsymbol{\psi}\big(\mathbf{x}(t_f), t_f\big) \leq \mathbf{0}_{n_\psi}, \tag{12b}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), u(t)\big), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \tag{12c}$$

$$\mathbf{g}\big(u(t)\big) = \begin{bmatrix} \underline{u} - u(t) \\ u(t) - \overline{u} \end{bmatrix} \leq \mathbf{0}_{n_g}, \tag{12d}$$

$$\mathbf{h}\big(\mathbf{x}(t)\big) \leq \mathbf{0}_{n_h}, \tag{12e}$$

where t_0 is the initial time; $t_f \in [t_0, t_{max}]$ is the finite final time with upper bound t_{max}; $u(t)$ is the piecewise-continuous input for all $t \in \big[t_0, t_f\big)$ with lower and upper bounds $\underline{u}$ and $\overline{u}$; $\mathbf{x}(t)$ is the n_x-dimensional vector of piecewise-continuously differentiable states for all $t \in \big[t_0, t_f\big)$; $\mathbf{f}(\mathbf{x}, u)$ is an n_x-dimensional vector function, smooth for all $(\mathbf{x}, u) \in \mathbb{R}^{n_x} \times \mathbb{R}$; $\mathbf{g}(u)$ is an n_g-dimensional vector function, smooth for all $u \in \mathbb{R}$; $\mathbf{h}(\mathbf{x})$ is an n_h-dimensional vector function, smooth for all $\mathbf{x} \in \mathbb{R}^{n_x}$; and $\phi(\mathbf{x}, t)$, $\boldsymbol{\psi}(\mathbf{x}, t)$ are a scalar function and an n_ψ-dimensional vector function, respectively, smooth for all $(\mathbf{x}, t) \in \mathbb{R}^{n_x} \times [t_0, t_{max}]$. We assume that $\mathbf{h}^{(1)}(\mathbf{x}, u) := \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x})\mathbf{f}(\mathbf{x}, u)$ depends explicitly on u.

In view of the aforementioned control objective, the OCP (12) is cast as the minimization of the final time t_f needed to reach a desired thermal effect CEM_{sp} in the units of CEM and to return to a surface temperature below the initial temperature T_0 that ensures no accumulation of thermal effects after the final time. Thus, the corresponding cost function is simply the final treatment time

$$\mathscr{J}\big(u(\cdot), t_f\big) = \phi\big(\mathbf{x}(t_f), t_f\big) = t_f, \tag{13}$$

whose minimization is subject to the terminal constraints

$$\mathscr{T}\big(u(\cdot), t_f\big) = \boldsymbol{\psi}\big(\mathbf{x}(t_f), t_f\big) = \begin{bmatrix} T(t_f) - T_0 \\ CEM_{sp} - CEM(t_f) \end{bmatrix} \leq \mathbf{0}_2. \tag{14}$$

In addition, the power $P(t)$ is subject to a lower bound $\underline{P}$ and an upper bound $\overline{P}$, that is, the input constraints

$$\mathbf{g}\big(u(t)\big) = \begin{bmatrix} \underline{P} - P(t) \\ P(t) - \overline{P} \end{bmatrix} \leq \mathbf{0}_2, \tag{15}$$

while the surface temperature is subject to an upper bound $\overline{T}$ to ensure that the surface is not subject to excessive temperatures, which corresponds to the state constraint

$$h\big(\mathbf{x}(t)\big) = T(t) - \overline{T} \leq 0. \tag{16}$$

The OCP is also subject to the dynamics $\mathbf{f}\big(\mathbf{x}(t), u(t)\big)$ given by (10). The APPJ model (10) is a significantly simplified model of the complex plasma-surface interactions, and therefore, is subject to significant plant-model mismatch. To this end, we resort to a data-driven adaptive algorithm to update the OCP via modifier adaptation, as illustrated in Fig. 2. Modifier adaptation relies on the estimation of the mismatch of the terminal cost function and terminal constraint functions between the true plasma treatment and the model (10) after a treatment. Updates to the initial OCP are determined offline using the estimated mismatches.

Additionally, as shown later in this paper, a portion of the optimal control sequence relies on a state constraint-seeking arc, specifically for tracking the surface temperature $T(t)$. Within this arc, we must keep $T(t)$ at some reference point without explicit knowledge of the dynamics of $T(t)$. For effective reference tracking, fast control computations on par with system dynamics on the order of seconds are essential since a control algorithm with computation times on the order of seconds would be inappropriate. Here, we propose a feedback linearization method with derivative estimation for reference tracking. This tracking strategy ensures elimination of steady-state error and rejection of input disturbances despite incomplete knowledge of the model [49]. The overall control strategy does not require solving any optimization problem online, as the online computations are limited to

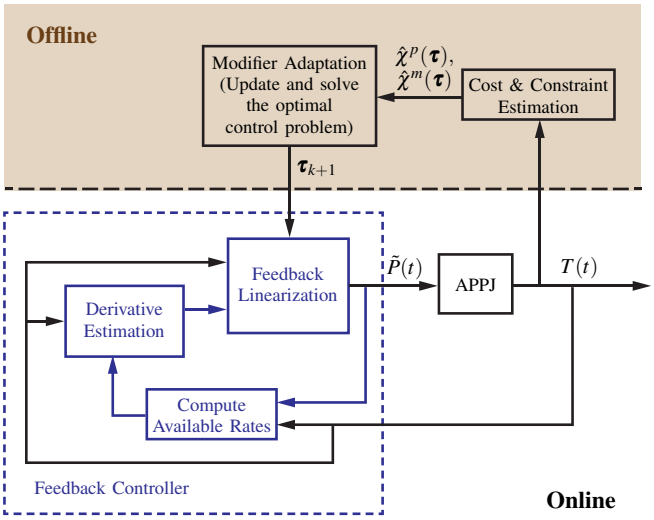

Fig. 2. Schematic of the adaptive optimal control approach combined with feedback linearization and derivative estimation for path constraint tracking. The shaded section includes modifier adaptation, which updates and solves the optimal control problem after receiving system measurements from a treatment. The enclosed blue section is the feedback control scheme for reference tracking that involves feedback linearization and derivative estimation.

simple, small-scale vector operations. This can be particularly advantageous towards control of fast-sampling APPJ applications using resource-limited embedded systems in the pursuit of portable and automated biomaterials processing and biomedical devices.

III. Control Strategy

This section presents the data-driven, adaptive optimal control approach as well as the feedback linearization strategy with derivative estimation for path constraint tracking.

A. OCP Reformulation as a Numerical Optimization Problem

According to Pontryagin's maximum principle, a solution to (12) is optimal only if there exists an n_ψ-dimensional vector $\boldsymbol{\nu} \geq \mathbf{0}_{n_\psi}$, an n_x-dimensional vector $\boldsymbol{\lambda}(t)$, an n_g-dimensional vector $\boldsymbol{\mu}(t) \geq \mathbf{0}_{n_g}$, and an n_h-dimensional vector $\boldsymbol{\eta}(t) \geq \mathbf{0}_{n_h}$ such that, for

$$\chi\big(\mathbf{x}(t_f),t_f\big) = \phi\big(\mathbf{x}(t_f),t_f\big) + \boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{\psi}\big(\mathbf{x}(t_f),t_f\big), \tag{17}$$

the following conditions are satisfied [51].

(i) For almost every t, $\big(\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t),\boldsymbol{\eta}(t)\big) \neq \mathbf{0}_{n_x+n_g+n_h}$.

(ii) The adjoint variables $\boldsymbol{\lambda}(t)$ are described, for almost every t, by the differential equations

$$\begin{aligned}\dot{\boldsymbol{\lambda}}(t) &= -\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\boldsymbol{\lambda}(t) - \frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{x}}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\boldsymbol{\eta}(t),\\ \boldsymbol{\lambda}(t_f) &= \frac{\partial \chi}{\partial \mathbf{x}}\big(\mathbf{x}(t_f),t_f\big)^{\mathrm{T}},\end{aligned} \tag{18}$$

and, for each entry point η such that $h_k\big(\mathbf{x}(\eta^-)\big) < 0$ and $h_k\big(\mathbf{x}(\eta)\big) = 0$ for some $k = 1,\ldots,n_h$, it holds that

$$\boldsymbol{\lambda}(\eta^-) = \boldsymbol{\lambda}(\eta) - \frac{\partial h_k}{\partial \mathbf{x}}\big(\mathbf{x}(\eta^-)\big)^{\mathrm{T}} \frac{\big(\mathbf{f}\big(\mathbf{x}(\eta^-)\big) - \mathbf{f}\big(\mathbf{x}(\eta)\big)\big)^{\mathrm{T}}\boldsymbol{\lambda}(\eta)}{h_k^{(1)}\big(\mathbf{x}(\eta^-)\big)}. \tag{19}$$

(iii) The Lagrangian function, defined as

$$\begin{aligned}\mathscr{L}\big(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t),\boldsymbol{\eta}(t)\big) &= \boldsymbol{\lambda}(t)^{\mathrm{T}}\mathbf{f}\big(\mathbf{x}(t),u(t)\big)\\ &+ \boldsymbol{\mu}(t)^{\mathrm{T}}\mathbf{g}\big(u(t)\big) + \boldsymbol{\eta}(t)^{\mathrm{T}}\mathbf{h}^{(1)}\big(\mathbf{x}(t),u(t)\big),\end{aligned} \tag{20}$$

satisfies, for almost every t,

$$\begin{aligned}0 = \frac{\partial \mathscr{L}}{\partial u}\big(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t),\boldsymbol{\eta}(t)\big) &= \frac{\partial \mathbf{f}}{\partial u}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\boldsymbol{\lambda}(t)\\ &+ \frac{\partial \mathbf{g}}{\partial u}\big(u(t)\big)^{\mathrm{T}}\boldsymbol{\mu}(t) + \frac{\partial \mathbf{h}^{(1)}}{\partial u}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\boldsymbol{\eta}(t).\end{aligned} \tag{21}$$

(iv) The Hamiltonian function, defined as

$$\mathscr{H}\big(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t)\big) = \boldsymbol{\lambda}(t)^{\mathrm{T}}\mathbf{f}\big(\mathbf{x}(t),u(t)\big), \tag{22}$$

satisfies, for almost every t,

$$\mathscr{H}\big(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t)\big) + \frac{\partial \chi}{\partial t}\big(\mathbf{x}(t_f),t_f\big) = 0. \tag{23}$$

(v) The Lagrange multipliers $\boldsymbol{\nu}$, $\boldsymbol{\mu}(t)$, and $\boldsymbol{\eta}(t)$ satisfy

$$\mathrm{diag}\Big(\boldsymbol{\psi}\big(\mathbf{x}(t_f),t_f\big)\Big)\boldsymbol{\nu} = \mathbf{0}_{n_\psi}, \tag{24}$$

and, for almost every t,

$$\mathrm{diag}\Big(\mathbf{g}\big(u(t)\big)\Big)\boldsymbol{\mu}(t) = \mathbf{0}_{n_g}, \tag{25}$$

$$\mathrm{diag}\Big(\mathbf{h}\big(\mathbf{x}(t)\big)\Big)\boldsymbol{\eta}(t) = \mathbf{0}_{n_h}, \tag{26}$$

$$\dot{\boldsymbol{\eta}}(t) \leq \mathbf{0}_{n_h}. \tag{27}$$

These conditions imply that, for almost every t such that the constraints $\mathbf{g}\big(u(t)\big) \leq \mathbf{0}_{n_g}$ and $\mathbf{h}\big(\mathbf{x}(t)\big) \leq \mathbf{0}_{n_h}$ are not active, $\boldsymbol{\mu}(t) = \mathbf{0}_{n_g}$, $\boldsymbol{\eta}(t) = \mathbf{0}_{n_h}$, and

$$\begin{aligned}\frac{\partial \mathscr{H}}{\partial u}\big(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t)\big) &= \frac{\partial \mathscr{L}}{\partial u}\big(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t),\boldsymbol{\mu}(t),\boldsymbol{\eta}(t)\big)\\ &= 0.\end{aligned} \tag{28}$$

If this condition (28) is sustained over an interval, then the time derivatives $\frac{\mathrm{d}^q\left(\frac{\partial \mathscr{H}}{\partial u}(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t))\right)}{\mathrm{d}t^q}$ for $q = 1,2,\ldots$ must also vanish over that interval, which corresponds to a singular arc.

Below, we show by contradiction that a singular arc cannot occur for the APPJ modeled by (10). Firstly, we compute

$$\begin{aligned}\frac{\partial \mathscr{H}}{\partial u}\big(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t)\big) &= \frac{\partial \mathbf{f}}{\partial u}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\boldsymbol{\lambda}(t)\\ &= \begin{bmatrix}\frac{1}{C_p} & 0\end{bmatrix}\boldsymbol{\lambda}(t)\end{aligned} \tag{29}$$

and

$$\begin{aligned}\frac{\mathrm{d}\left(\frac{\partial \mathscr{H}}{\partial u}(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t))\right)}{\mathrm{d}t} &= \mathbf{f}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\frac{\partial^2 \mathbf{f}}{\partial u \partial \mathbf{x}}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\boldsymbol{\lambda}(t)\\ &- \frac{\partial \mathbf{f}}{\partial u}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\boldsymbol{\lambda}(t)\\ &= -\frac{1}{C_p}\frac{\partial \mathbf{f}}{\partial T}\big(\mathbf{x}(t),u(t)\big)^{\mathrm{T}}\boldsymbol{\lambda}(t)\\ &= \begin{bmatrix}\frac{\varphi'(T(t))}{C_p} & \frac{\log(K)K^{Tref-T(t)}}{C_p}\end{bmatrix}\boldsymbol{\lambda}(t).\end{aligned} \tag{30}$$

This implies that, in a singular arc,

$$\begin{bmatrix}0\\0\end{bmatrix} = \begin{bmatrix}\frac{\partial \mathscr{H}}{\partial u}\big(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t)\big)\\ \frac{\mathrm{d}\left(\frac{\partial \mathscr{H}}{\partial u}(\mathbf{x}(t),u(t),\boldsymbol{\lambda}(t))\right)}{\mathrm{d}t}\end{bmatrix} = \mathscr{M}\boldsymbol{\lambda}(t), \tag{31}$$

with

$$\mathscr{M} = \begin{bmatrix} \frac{1}{C_p} & 0 \\ \frac{\varphi'(T(t))}{C_p} & \frac{\log(K)K^{T_{ref}-T(t)}}{C_p} \end{bmatrix}. \tag{32}$$

Since $\boldsymbol{\mu}(t) = \mathbf{0}_{n_g}$, $\boldsymbol{\eta}(t) = \mathbf{0}_{n_h}$, and $\big(\boldsymbol{\lambda}(t), \boldsymbol{\mu}(t), \boldsymbol{\eta}(t)\big) \neq \mathbf{0}_{n_x+n_g+n_h}$, the condition $\boldsymbol{\lambda}(t) \neq \mathbf{0}_{n_x}$ holds. This is possible only if $\mathscr{M}$ is singular, that is, $\det(\mathscr{M}) = 0$. However, since

$$\det(\mathscr{M}) = \frac{\log(K)K^{T_{ref}-T(t)}}{C_p^2} > 0, \tag{33}$$

$\mathscr{M}$ cannot be singular. Hence, singular arcs are not possible for the model in (10). Note that this is the case independently of (i) the cost function and constraints and (ii) the functional relationship $\varphi\big(T(t)\big)$ and the parameter C_p, that is, independently of the model uncertainty.

Hence, at least one of the constraints $\mathbf{g}\big(u(t)\big) \leq \mathbf{0}_{n_g}$ and $\mathbf{h}\big(\mathbf{x}(t)\big) \leq \mathbf{0}_{n_h}$ is active for all times t. If a constraint $g_k\big(u(t)\big) \leq 0$ is active, then the input $u(t)$ is such that $g_k\big(u(t)\big) = 0$, for $k = 1, \ldots, n_g$. If a constraint $h_k\big(\mathbf{x}(t)\big) \leq 0$ is active, then the input $u(t)$ is such that $h_k^{(1)}\big(\mathbf{x}(t), u(t)\big) = 0$, for $k = 1, \ldots, n_h$. Thus, the optimal input $u(t)$ is composed of a sequence of arcs where each arc can be of type 1) input constraint-seeking, such that it is determined by an equality $u(t) = \underline{u}$ or $u(t) = \overline{u}$ that specifies an input bound, or 2) state constraint-seeking, such that it is determined by an equality $h_k^{(1)}(\mathbf{x}(t), u(t)) = 0$ for some $k = 1, \ldots, n_h$, that is, such that a state constraint remains active [52]. Consequently, there is a finite number of arc types from which arc sequences can be formed. If we consider as plausible arc sequences only sequences with a number of arcs no larger than some upper bound $\bar{n}_a$ and without consecutive arcs of the same type, it follows that the number of plausible sequences is also finite.

Parsimonious input parameterization [53] is an effective approach for describing the optimal input using only a few decision variables, in contrast to infinite-dimensional variables in the OCP (12). For a given plausible arc sequence composed of $n_s + 1$ input constraint-seeking arcs, the input is defined by the following decision variables: the switching times $\bar{t}_1, \ldots, \bar{t}_{n_s}$ to these arcs and the final time $\bar{t}_{n_s+1} = t_f$ [54]. The entry points in state constraint-seeking arcs are given by the n_η-dimensional vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{n_\eta})$, but switching to these arcs cannot occur at arbitrary times since it depends on the states $\mathbf{x}(t)$. Consequently, the OCP reduces to the problem of determining the optimal arc sequence and the optimal switching times and final time.

For a given arc sequence, we describe the input in the i-th time interval $[\bar{t}_{i-1}, \bar{t}_i)$, for $i = 1, \ldots, n_s + 1$, by using the control law $u(t) = \tilde{c}\big(\mathbf{x}(t)\big)$. For input constraint-seeking arcs, $\tilde{c}\big(\mathbf{x}(t)\big) = \underline{u}$ or $\tilde{c}\big(\mathbf{x}(t)\big) = \overline{u}$. For state constraint-seeking arcs, $\tilde{c}\big(\mathbf{x}(t)\big)$ is such that $h_k^{(1)}\big(\mathbf{x}(t), \tilde{c}\big(\mathbf{x}(t)\big)\big) = 0$ for some $k = 1, \ldots, n_h$. Then, upon eliminating input dependencies and rewriting the OCP (12) in terms of the states $\mathbf{x}$, the system dynamics take the form

$$\tilde{\mathbf{f}}\big(\mathbf{x}(t)\big) := \mathbf{f}\big(\mathbf{x}(t), \tilde{c}\big(\mathbf{x}(t)\big)\big). \tag{34}$$

Since the input parameters for the given arc sequence are $\boldsymbol{\tau} := \big(\bar{t}_1, \ldots, \bar{t}_{n_s}, t_f\big)$, the OCP (12) can be reformulated in terms of the new decision variables as

$$\min_{\boldsymbol{\tau}} \quad \hat{\phi}(\boldsymbol{\tau}) := \phi\big(\mathbf{x}(t_f), t_f\big), \tag{35a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\psi}}(\boldsymbol{\tau}) := \boldsymbol{\psi}\big(\mathbf{x}(t_f), t_f\big) \leq \mathbf{0}_{n_\psi}, \tag{35b}$$

$$\bar{t}_{i-1} \leq \bar{t}_i, \quad i = 1, \ldots, n_s + 1, \tag{35c}$$

$$\dot{\mathbf{x}}(t) = \tilde{\mathbf{f}}\big(\mathbf{x}(t)\big), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \tag{35d}$$

which is convenient for numerical optimization since there are only $N := n_s + 1$ decision variables. The decision variables $\boldsymbol{\tau}$ are considered as continuous variables. Also, they do not need to be recast to a sampling time in real time since the switching between arcs can take place at any time, including between sampling times.

However, the explicit form of $h_k^{(1)}\big(\mathbf{x}(t), u(t)\big)$ is not exactly known due to uncertainty of $\frac{1}{C_p}$ and lack of knowledge about $\varphi\big(T(t)\big)$ in (10). Thus, to implement a state constraint-seeking arc such that $h_k^{(1)}\big(\mathbf{x}(t), u(t)\big) = 0$, for $k = 1, \ldots, n_h$, the constraint $h_k\big(\mathbf{x}(t)\big) = 0$ must be tracked via feedback control (see Section III-C). Recall that we consider a single state constraint $h\big(\mathbf{x}(t)\big) = T(t) - \overline{T} \leq 0$ that consists in an upper bound for the temperature. Note that a similar constraint for $CEM(t)$ is not needed since $CEM(t)$ is nondecreasing, which means that a state path constraint $CEM(t) - \overline{CEM} \leq 0$ can be recast as a terminal constraint $CEM(t_f) - \overline{CEM} \leq 0$.

Since the model in the OCP (35) is not an exact representation of the APPJ, the solution to (35) is not the optimal solution for the true system. Next, we propose an adaptive algorithm based on modifier adaptation to compute the optimal solution to (35) for the true system despite the presence of plant-model mismatch. In summary, the input constraints (15) are satisfied by the choice of arc sequences, the state constraints (16) are satisfied by the control strategy for tracking of state constraints, and the terminal cost (13) is optimized and the terminal constraints (14) are satisfied by the adaptive algorithm based on modifier adaptation.

Remark 1: The reader may note that the developments in this section were a consequence of the structure of the dynamical system only and are valid independently of the cost and constraints of the problem.

B. Adaptive Optimal Control via Modifier Adaptation

We propose an adaptive approach to deal with the mismatch between the APPJ and its model (10) so as to converge to the optimum of the true system. Note that this mismatch can be structural, which implies that the model of $\varphi\big(T(t)\big)$ can be structurally incorrect. To this end, we first estimate the mismatch between the cost and constraint functions for the true system and the model; and then use this estimation of the mismatch to implement an adaptive algorithm to solve the OCP (35) for the true system.

Each terminal cost and constraint function of the OCP is expressed for the true system and the model, respectively, as

$$\hat{\chi}^p(\boldsymbol{\tau}) := \tilde{\chi}\big(\mathbf{x}^p(t_f), t_f\big), \tag{36}$$

$$\hat{\chi}^m(\boldsymbol{\tau}) := \tilde{\chi}\big(\mathbf{x}^m(t_f), t_f\big), \tag{37}$$

where $\mathbf{x}^p(t)$ and $\mathbf{x}^m(t)$ are the states $\mathbf{x}(t)$ that correspond to $\boldsymbol{\tau}$ in the true system and the model, respectively. Suppose that, at each iteration k of the algorithm, data for $\hat{\chi}^p(\boldsymbol{\tau})$ are available in the neighborhood of a set of points $\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_k$ evaluated by the algorithm. More precisely, the data are available not only for the nominal points $\boldsymbol{\tau}_i^0=\boldsymbol{\tau}_i$, for $i=1,\ldots,k$, but also for the auxiliary points $\boldsymbol{\tau}_i^j$, for $i=1,\ldots,k$ and $j=1,\ldots,N$, where $\boldsymbol{\tau}_1$ is the model optimum. Hence, the goal is to describe the mismatch between the functions $\hat{\chi}^p(\boldsymbol{\tau})$ and $\hat{\chi}^m(\boldsymbol{\tau})$ from measurements of $\hat{\chi}^p(\boldsymbol{\tau}_i^j)$ and knowledge of $\hat{\chi}^m(\boldsymbol{\tau}_i^j)$, for $i=1,\ldots,k$ and $j=0,\ldots,N$. To this end, we use a method known as modifier adaptation, which uses so-called modifiers to provide a description of the mismatch in the cost and constraint functions [40].

Suppose that the unknown function

$$g^{\chi*}(\boldsymbol{\tau})=\hat{\chi}^p(\boldsymbol{\tau})-\hat{\chi}^m(\boldsymbol{\tau})\simeq\begin{bmatrix}1 & (\boldsymbol{\tau}-\boldsymbol{\tau}_1)^{\mathrm{T}}\end{bmatrix}\begin{bmatrix}\varepsilon^{\chi*}\\ \boldsymbol{\lambda}^{\chi*}\end{bmatrix}, \quad (38)$$

with $\varepsilon^{\chi*}:=\hat{\chi}^p(\boldsymbol{\tau}_1)-\hat{\chi}^m(\boldsymbol{\tau}_1)$, $\boldsymbol{\lambda}^{\chi*}=\frac{\partial\hat{\chi}^p}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_1)^{\mathrm{T}}-\frac{\partial\hat{\chi}^m}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_1)^{\mathrm{T}}$, is sampled at $\boldsymbol{\tau}=\boldsymbol{\tau}_i^j$, for $i=1,\ldots,k$ and $j=0,\ldots,N$, as

$$y_j^{\chi}(i)=g^{\chi*}(\boldsymbol{\tau}_i^j)+e_j^{\chi}(i), \quad i=1,\ldots,k,\ j=0,\ldots,N, \quad (39)$$

where $y_j^{\chi}(i)$ is the measurement of $g^{\chi*}(\boldsymbol{\tau}_i^j)$ with noise $e_j^{\chi}(i)$. Note that, if $\boldsymbol{\tau}-\boldsymbol{\tau}_k$ had been used in (38) instead of $\boldsymbol{\tau}-\boldsymbol{\tau}_1$, the unknown function $g^{\chi*}(\boldsymbol{\tau})$ would still be approximated by an affine function of $\boldsymbol{\tau}$. In that case, the only difference would be that $\varepsilon^{\chi*}$ would vary for each value of k. However, the formulation in (38) allows using the same regressor $\boldsymbol{\tau}-\boldsymbol{\tau}_1$ and obtaining a nearly constant value of $\varepsilon^{\chi*}$ for each value of k. Based on the data $y_j^{\chi}(i)$ and the mismatch model

$$y_j^{\chi}(i)=\mathbf{h}_j(i)^{\mathrm{T}}\begin{bmatrix}\varepsilon_k^{\chi}\\ \boldsymbol{\lambda}_k^{\chi}\end{bmatrix}+e_j^{\chi}(i), \quad i=1,\ldots,k,\ j=0,\ldots,N, \quad (40)$$

with $\mathbf{h}_j(i):=\begin{bmatrix}1 & (\boldsymbol{\tau}_i^j-\boldsymbol{\tau}_1)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, we aim to estimate an affine approximation $g_k^{\chi}(\boldsymbol{\tau})$ of $g^{\chi*}(\boldsymbol{\tau})$. Defining $\boldsymbol{\theta}_k^{\chi}:=\begin{bmatrix}\varepsilon_k^{\chi}\\ \boldsymbol{\lambda}_k^{\chi}\end{bmatrix}$, $\mathbf{y}^{\chi}(i):=\begin{bmatrix}y_0^{\chi}(i) & \cdots & y_N^{\chi}(i)\end{bmatrix}^{\mathrm{T}}$, $\mathbf{e}^{\chi}(i):=\begin{bmatrix}e_0^{\chi}(i) & \cdots & e_N^{\chi}(i)\end{bmatrix}^{\mathrm{T}}$, $\mathbf{H}(i):=\begin{bmatrix}\mathbf{h}_0(i) & \cdots & \mathbf{h}_N(i)\end{bmatrix}^{\mathrm{T}}$, we then have

$$\mathbf{y}^{\chi}(i)=\mathbf{H}(i)\boldsymbol{\theta}_k^{\chi}+\mathbf{e}^{\chi}(i), \quad i=1,\ldots,k. \quad (41)$$

Additionally, upon defining $\mathbf{y}_k^{\chi}:=\begin{bmatrix}\mathbf{y}^{\chi}(1)^{\mathrm{T}} & \cdots & \mathbf{y}^{\chi}(k)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, $\mathbf{e}_k^{\chi}:=\begin{bmatrix}\mathbf{e}^{\chi}(1)^{\mathrm{T}} & \cdots & \mathbf{e}^{\chi}(k)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, $\mathbf{H}_k:=\begin{bmatrix}\mathbf{H}(1)^{\mathrm{T}} & \cdots & \mathbf{H}(k)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$, we have

$$\mathbf{y}_k^{\chi}=\mathbf{H}_k\boldsymbol{\theta}_k^{\chi}+\mathbf{e}_k^{\chi}. \quad (42)$$

Assuming that the noise realizations $e_0^{\chi}(1),\ldots,e_N^{\chi}(1)$, $\ldots,e_0^{\chi}(k),\ldots,e_N^{\chi}(k)$ for $\boldsymbol{\tau}_1^0,\ldots,\boldsymbol{\tau}_1^N,\ldots,\boldsymbol{\tau}_k^0,\ldots,\boldsymbol{\tau}_k^N$ are independent and identically distributed (i.i.d.) and drawn from a normal distribution with zero mean and variance σ_{χ}^2, the maximum-likelihood estimate of $\boldsymbol{\theta}_k^{\chi}$ is

$$\bar{\boldsymbol{\theta}}_k^{\chi}=\left(\mathbf{H}_k^{\mathrm{T}}\mathbf{H}_k\right)^{-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{y}_k^{\chi}. \quad (43)$$

Also, the maximum-likelihood estimate of $g_k^{\chi}(\boldsymbol{\tau})$ is

$$\bar{g}_k^{\chi}(\boldsymbol{\tau})=\begin{bmatrix}1 & (\boldsymbol{\tau}-\boldsymbol{\tau}_1)^{\mathrm{T}}\end{bmatrix}\bar{\boldsymbol{\theta}}_k^{\chi}. \quad (44)$$

Since $\bar{g}_k^{\chi}(\boldsymbol{\tau})$ is an estimate of the mismatch $\hat{\chi}^p(\boldsymbol{\tau})-\hat{\chi}^m(\boldsymbol{\tau})$, $\hat{\chi}_k^m(\boldsymbol{\tau}):=\hat{\chi}^m(\boldsymbol{\tau})+\bar{g}_k^{\chi}(\boldsymbol{\tau})$ represents an estimate for $\hat{\chi}^p(\boldsymbol{\tau})$. This can be used to obtain estimates of the cost and constraint functions $\hat{\phi}^p(\boldsymbol{\tau})$ and $\hat{\psi}_j^p(\boldsymbol{\tau})$ of the OCP (35) for the true system. Hence, based on the data available in the neighborhood of the set of points $\boldsymbol{\tau}_1,\ldots,\boldsymbol{\tau}_k$ at each iteration k of the adaptive algorithm, the following OCP is solved:

$$\begin{aligned}
\boldsymbol{\tau}_{k+1}:=\arg\min_{\boldsymbol{\tau}} \quad & \hat{\phi}^m(\boldsymbol{\tau})+\bar{g}_k^{\phi}(\boldsymbol{\tau}), && (45a)\\
\text{s.t.} \quad & \hat{\psi}_j^m(\boldsymbol{\tau})+\bar{g}_k^{\psi_j}(\boldsymbol{\tau})\le 0, \quad j=1,\ldots,n_{\psi}, && (45b)\\
& \bar{t}_{i-1}\le\bar{t}_i, \quad i=1,\ldots,n_s+1, && (45c)\\
& \dot{\mathbf{x}}^m(t)=\tilde{\mathbf{f}}\big(\mathbf{x}^m(t)\big), \quad \mathbf{x}^m(t_0)=\mathbf{x}_0. && (45d)
\end{aligned}$$

Assume that the quality of the initial model specified by the cost and constraint functions $\hat{\chi}^m(\boldsymbol{\tau})$ and the modified models specified by the functions $\hat{\chi}_k^m(\boldsymbol{\tau})$ is such that the model optimum $\boldsymbol{\tau}_1$ and the subsequent solutions $\boldsymbol{\tau}_{k+1}$ are in the basin of attraction of a global optimum of the OCP (35) for the true system. Then, if the adaptive algorithm converges to a single point, the true system converges to its global optimum, as shown in the following theorem.

Theorem 1: Suppose that the OCP (45) is solved at each iteration k of the adaptive algorithm. If the algorithm converges to a single point $\boldsymbol{\tau}_{\infty}$, then $\boldsymbol{\tau}_{\infty}$ is a Karush-Kuhn-Tucker (KKT) point of the OCP (35) for the true system. Moreover, assuming that the points $\boldsymbol{\tau}_k$ are in the basin of attraction of a global optimum of the OCP (35) for the true system, $\boldsymbol{\tau}_{\infty}$ is a global optimum of the OCP (35) for the true system.

Proof: Assume that the adaptive algorithm converges to a single point $\boldsymbol{\tau}_{\infty}$. This implies that the algorithm evaluates the cost and constraint functions $\hat{\chi}^p$ an arbitrarily large number of times not only for $\boldsymbol{\tau}_{\infty}$ but also for the auxiliary points $\boldsymbol{\tau}_{\infty}^j$, for $j=1,\ldots,N$. From the properties of maximum-likelihood estimation and (43)-(44), this implies that $\bar{g}_{\infty}^{\chi}(\boldsymbol{\tau}_{\infty})=g^{\chi*}(\boldsymbol{\tau}_{\infty})$, $\frac{\partial\bar{g}_{\infty}^{\chi}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_{\infty})=\frac{\partial g^{\chi*}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_{\infty})$, which in turn implies that $\hat{\chi}_{\infty}^m(\boldsymbol{\tau}_{\infty})=\hat{\chi}^p(\boldsymbol{\tau}_{\infty})$, $\frac{\partial\hat{\chi}_{\infty}^m}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_{\infty})=\frac{\partial\hat{\chi}^p}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_{\infty})$, and $\boldsymbol{\tau}_{\infty}$ is a KKT point of the OCP (45) for $k\to\infty$ if and only if $\boldsymbol{\tau}_{\infty}$ is a KKT point of the OCP (35) for the true system. Since $\boldsymbol{\tau}_{\infty}$ is the solution to the OCP (45) for $k\to\infty$, $\boldsymbol{\tau}_{\infty}$ is also a KKT point of the OCP (35) for the true system. Then, since the only KKT point of the OCP (35) for the true system that is in the basin of attraction of a global optimum of that OCP is the global optimum itself, $\boldsymbol{\tau}_{\infty}$ is a global optimum of the OCP (35) for the true system. ■

Note that the convergence of the algorithm to a single point is an easily verifiable condition, which then guarantees convergence to a KKT point of the OCP (35) for the true system. Upon assuming a relatively mild technical condition related to the basin of attraction of a global optimum of that OCP, this KKT point is also guaranteed to be a global optimum of the OCP (35) for the true system. Hence, the proposed approach provides a way to account for the mismatch between the true system and the model.

In summary, the interface between the blocks of the control architecture in Fig. 2 is as follows: the "APPJ" block sends the signal $T(t)$ to the "Cost & Constraint Estimation"

block; the "Cost & Constraint Estimation" block sends the information about the plant and model cost and constraints $\hat{\chi}^p(\boldsymbol{\tau})$, $\hat{\chi}^m(\boldsymbol{\tau})$ that correspond to the points $\boldsymbol{\tau}_k^0,\ldots,\boldsymbol{\tau}_k^N$; the "Modifier Adaptation" block uses the information received for the previously evaluated points $\boldsymbol{\tau}_i^j$ to update and solve the OCP and sends the information $\boldsymbol{\tau}_{k+1}$ to the "Feedback Controller" block; the "Feedback Controller" block activates feedback control when the state constraint is active according to the switching times in $\boldsymbol{\tau}_{k+1}$ and computes the signal $\tilde{P}(t)$ for the "APPJ" block based on $T(t)$, as specified next.

C. Feedback Control for Path Constraint Tracking

As discussed in Sections II-C and III-A, one of the objectives is to control the output $y_c(t) := T(t)$ to the setpoint $r(t) := \overline{T} - \Delta T(t)$, where $\Delta T(t) \geq 0$ is a piecewise-constant back-off parameter, to satisfy the state constraint $h\big(\mathbf{x}(t)\big) = T(t) - \overline{T} \leq 0$, by manipulating the input $u(t) := P(t)$. The constraint tightening enabled by a nonzero back-off parameter has been shown to be an effective way for ensuring path constraint satisfaction in the presence of plant-model mismatch [55]. Note that $\mathbf{f}\big(\mathbf{x}(t),u(t)\big)$ is a smooth function and the relative degree of $y_c(t)$ with respect to $u(t)$ is one, which implies that $y_c(t)$ is an absolutely continuous function of t that depends only on the absolutely continuous states $\mathbf{x}(t)$. Furthermore, one can write

$$\dot{y}_c(t) = \dot{y}_u(t) + \dot{y}_a(t), \tag{46}$$

with

$$\dot{y}_u(t) = s_u\big(\mathbf{x}(t)\big) := -\varphi\big(T(t)\big), \tag{47}$$

where $s_u\big(\mathbf{x}(t)\big)$ is a fully unknown function of the absolutely continuous states $\mathbf{x}(t)$ only, and

$$\dot{y}_a(t) = s_a\big(y(t),u(t)\big) := \tfrac{1}{C_p} P(t), \tag{48}$$

where $s_a\big(y(t),u(t)\big)$ is a function of the current outputs $y(t)$ and inputs $u(t)$ that is partially known with some modeling error. In addition, $s_a\big(y(t),u(t)\big)$ is an affine function of the inputs, that is,

$$s_a\big(y(t),u(t)\big) = \beta_a\big(y(t)\big) + B_a\big(y(t)\big)u(t), \tag{49}$$

for $\beta_a\big(y(t)\big) := 0$ and the nonzero scalar $B_a\big(y(t)\big) := \frac{1}{C_p}$. We note that $s_a\big(y(t),u(t)\big)$ is partially known with some modeling error as

$$\tilde{s}_a\big(y(t),u(t)\big) = \tilde{\beta}_a\big(y(t)\big) + \tilde{B}_a\big(y(t)\big)u(t), \tag{50}$$

where $\tilde{\beta}_a\big(y(t)\big) := 0$ and $\tilde{B}_a\big(y(t)\big) := \frac{1}{\tilde{C}_p}$ are the modeled counterparts of $\beta_a\big(y(t)\big)$ and $B_a\big(y(t)\big)$.

The goal is to control the output $y_c(t)$ to the setpoint $r(t)$ while the function $s_u\big(\mathbf{x}(t)\big)$ that corresponds to the derivative $\dot{y}_u(t)$ is fully unknown, the functions $\beta_a\big(y(t)\big)$ and $B_a\big(y(t)\big)$ are partially known with some modeling error, and the input $u(t)$ and the output $y(t)$ are subject to disturbances. The feedback control strategy for tracking must also ensure elimination of steady-state error and rejection of constant input disturbances without any integral term, as well as fast convergence of the controlled output to the setpoint.

Note that $r(t)$ is piecewise-constant, but may be discontinuous, which implies that $\dot{r}(t) = 0$ between time instants with discontinuity of $r(t)$. If the true system (11) were perfectly known, one could use the input-output feedback linearization law [45]

$$u(t) = B_a\big(y(t)\big)^{-1}\Big(v(t) - \dot{y}_u(t) - \beta_a\big(y(t)\big)\Big) \tag{51}$$

to set the rates of variation $v(t)$ for $\dot{y}_c(t)$ as

$$\dot{y}_c(t) = v(t). \tag{52}$$

For the variation rate $v(t)$, one could choose the control law [56]

$$v(t) = \tau_c^{-1}\big(r(t) - y_c(t)\big), \tag{53}$$

which would ensure exponential convergence of $y_c(t)$ to $r(t)$ with time constant τ_c in the ideal case of perfectly known functions $s_u\big(\mathbf{x}(t)\big)$, $\beta_a\big(y(t)\big)$, and $B_a\big(y(t)\big)$, and no input or output disturbances. However, the function $s_u\big(\mathbf{x}(t)\big)$ that corresponds to the derivative $\dot{y}_u(t)$ is fully unknown, while the functions $\beta_a\big(y(t)\big)$ and $B_a\big(y(t)\big)$ are partially known with some modeling error. Furthermore, there is a difference between the system input $u(t)$ and the actuator input $\tilde{u}(t) := \tilde{P}(t)$ due to the input disturbance $d(t) := u(t) - \tilde{u}(t)$; and a difference between the sensor output $\tilde{y}(t) := \tilde{T}(t)$ and the system output $y(t) := T(t)$ due to the output disturbance $w(t) := \tilde{y}(t) - y(t)$. We define the sensor output $\tilde{y}_c(t) := \tilde{y}(t)$ and the disturbance $w_c(t) := \tilde{y}_c(t) - y_c(t)$.

In the presence of these disturbances, it is not possible to use directly the control laws (51) and (53). However, one can approximate these control laws by replacing $u(t)$ and $y(t)$ by $\tilde{u}(t)$ and $\tilde{y}(t)$, $\beta_a\big(y(t)\big)$ and $B_a\big(y(t)\big)$ by $\tilde{\beta}_a\big(\tilde{y}(t)\big)$ and $\tilde{B}_a\big(\tilde{y}(t)\big)$, as well as the unknown derivative $\dot{y}_u(t)$ by its estimate $\hat{\dot{y}}_u(t)$. This yields

$$\tilde{u}(t) = \tilde{B}_a\big(\tilde{y}(t)\big)^{-1}\Big(\tilde{v}(t) - \hat{\dot{y}}_u(t) - \tilde{\beta}_a\big(\tilde{y}(t)\big)\Big), \tag{54}$$

$$\tilde{v}(t) = \tau_c^{-1}\big(r(t) - \tilde{y}_c(t)\big). \tag{55}$$

Now, (46)–(48) can be used to estimate the value $\hat{\dot{y}}_u(t)$ of the unknown derivative by applying an finite impulse response (FIR) filter to the sensor outputs $\tilde{y}_c(t)$ and the available rates

$$\tilde{\dot{y}}_a(t) = \tilde{s}_a\big(\tilde{y}(t),\tilde{u}(t)\big) \tag{56}$$

in the interval $[t-\Delta t, t]$, where Δt is the size of the filter window. We use the definition

$$\begin{aligned}\tilde{\dot{y}}_u(t) &= \dot{\tilde{y}}_c(t) - \tilde{\dot{y}}_a(t)\\ &= \dot{w}_c(t) + \dot{y}_u(t) + \dot{y}_a(t) - \tilde{\dot{y}}_a(t).\end{aligned} \tag{57}$$

Appendix A provides more details about the continuous-time derivative estimation of $\hat{\dot{y}}_u(t)$. If $\tilde{y}_c$ is absolutely continuous and $\mathscr{D}(\tilde{y}_c,t)$ is defined as in Appendix A, then one can show that

$$\begin{aligned}\mathscr{D}(\tilde{y}_c,t) &= \tilde{\dot{y}}_u(t) + \mathscr{R}(\tilde{\dot{y}}_u,t) + \mathscr{W}(\tilde{\dot{y}}_a,t)\\ &= \mathscr{W}(\tilde{\dot{y}}_u,t) + \mathscr{W}(\tilde{\dot{y}}_a,t),\end{aligned} \tag{58}$$

with

$$\mathscr{R}(\tilde{y}_u,t) := \int_0^{\Delta t} 6\left(\frac{\tau}{\Delta t} - \left(\frac{\tau}{\Delta t}\right)^2\right)\frac{\tilde{y}_u(t-\Delta t+\tau)-\tilde{y}_u(t)}{\Delta t}\mathrm{d}\tau, \quad (59)$$

$$\mathscr{W}(\tilde{y}_a,t) := \int_0^{\Delta t} 6\left(\frac{\tau}{\Delta t} - \left(\frac{\tau}{\Delta t}\right)^2\right)\frac{\tilde{y}_a(t-\Delta t+\tau)}{\Delta t}\mathrm{d}\tau. \quad (60)$$

We now note that

$$\hat{y}_u(t) = \mathscr{D}(\tilde{y}_c,t) - \mathscr{W}(\tilde{y}_a,t), \quad (61)$$

which implies that, from (58),

$$\hat{y}_u(t) = \tilde{y}_u(t) + \mathscr{R}(\tilde{y}_u,t) = \mathscr{W}(\tilde{y}_u,t). \quad (62)$$

From (62) and the definition of $\mathscr{R}(\tilde{y}_u,t)$, one can observe that, if $\tilde{y}_u(t-\Delta t+\tau)$ is constant for $\tau \in [0,\Delta t]$, the estimate $\hat{y}_u(t)$ yields the desired value $\tilde{y}_u(t)$. This results in a deadbeat property. In addition, one can show that this estimate minimizes both the mean squared error and the effect of measurement noise $w_c(t)$ for a given Δt [56].

In summary, the proposed controller via feedback linearization and continuous-time derivative estimation is given by (54), (55) with (50), (56), (61) for the system (46)–(49). It can be shown that this feedback control strategy eliminates steady-state error and rejects constant input disturbances without any integral term and forces the controlled output to converge exponentially to its setpoint [56].

IV. Application to the Optimal Control of the Thermal Effects of an APPJ

This section demonstrates the performance of the proposed data-driven, adaptive optimal control approach in simulations and real-time control experiments for controlling the nonlinear thermal effects of the kHz-excited APPJ in He depicted in Fig. 1. The hardware and software platform used to deploy the online part of the proposed control architecture has been described in detail in Section II-A.

A. Control Objective

As discussed in Section II-C, in CAP applications for biomaterials processing, it is important to treat a surface in a minimal time to deliver a desired amount of cumulative, nonlinear plasma effects. In this work, the control objective consists in minimizing the treatment time t_f such that a desired amount of thermal effects is delivered and that the surface temperature is resolved back to its initial temperature. The cost and constraints of the OCP for the control of the APPJ are given in (13)–(16). The APPJ is subject to state (surface temperature $T \le \overline{T}$) and input bounds (power $P \in [\underline{P},\overline{P}]$).

It follows from Section III-A that the optimal input $u(t)$ is composed of a sequence of constraint-seeking arcs that include:

- $g_1\big(u(t)\big) = \underline{P} - P(t) \le 0$ is active and $P(t) = \underline{P}$,
- $g_2\big(u(t)\big) = P(t) - \overline{P} \le 0$ is active and $P(t) = \overline{P}$, or
- $h\big(\mathbf{x}(t),u(t)\big) = T(t) - \overline{T} \le 0$ is active and $\dot{T}(t) = 0$.

From the structure of the APPJ model (10) and the OCP, the optimal arc sequence is composed of three arcs: in the first arc, $P(t) = \overline{P}$ so that $T(t)$ increases as quickly as possible until the path constraint $T(t) - \overline{T} \le 0$ becomes active; in the second arc, $\dot{T}(t) = 0$ so that the path constraint $T(t) - \overline{T} \le 0$ remains active; and, in the third arc, $P(t) = \underline{P}$ so that $T(t)$ decreases as quickly as possible until the terminal constraints are satisfied. This arc sequence is optimal independently of the functional relationship $\varphi\big(T(t)\big)$ and the parameter C_p. Switching from the first arc to the second arc cannot occur at an arbitrary time since it depends on the temperature $T(t)$. Moreover, since the explicit form of $\dot{T}(t)$ is not exactly known, control via feedback linearization and derivative estimation is used to track the constraint $T(t) - \overline{T} = 0$ in the second arc (see the enclosed blue area in Fig. 2). As such, since the optimal arc sequence is known, the OCP reduces to the problem of determining the optimal switching time $\bar{t}_1$ from the second arc to the third arc and the optimal final time t_f. Hence, the decision variables are $\boldsymbol{\tau} := (\bar{t}_1, t_f)$. Finally, modifier adaptation is used to converge to the optimal values of these decision variables for the true system despite the presence of plant-model mismatch.

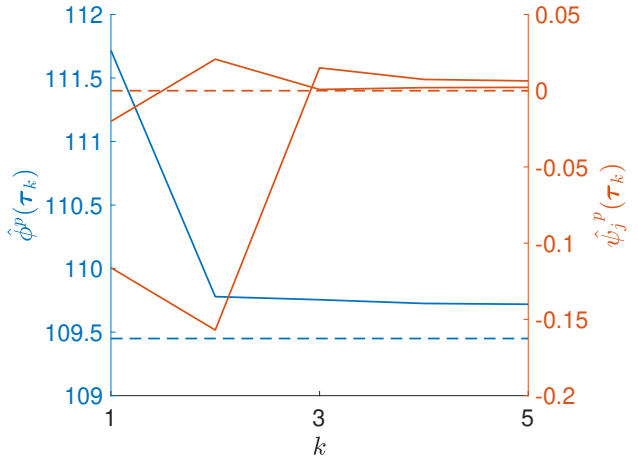

Fig. 3. Evolution of the cost (in blue) and constraints (in red) as a function of the iteration number k of the adaptive optimal control algorithm. The dashed lines represent the optimal values.

B. Closed-loop Simulations

We simulate the true system with the following equation for the unknown function $\varphi\big(T(t)\big)$,

$$\varphi\big(T(t)\big) = \frac{UA}{C_p}\left(\overline{T} - \frac{T_b+T_\infty}{2}\right)\frac{\log(\overline{T}-T_\infty)-\log(\overline{T}-T_b)}{\log(T(t)-T_\infty)-\log(T(t)-T_b)}, \quad (63)$$

with $C_p = \frac{\rho c_p \pi r^2 d}{\mu}$ and $UA = \frac{2\pi r d k \beta}{\mu}$. Parameters for this simulated true system are given in Table I of Appendix B, while system constraints and objective values are given in Table II. Uncertainty in the APPJ model (10) is considered by describing the unknown function $\varphi\big(T(t)\big)$ with structural mismatch as $\tilde{\varphi}\big(T(t)\big) = \frac{\tilde{U}A}{C_p}\left(T(t) - \frac{T_b+T_\infty}{2}\right)$ and the values of the model parameters as $\tilde{C}_p = 0.8C_p$ and $\tilde{U} = 0.8U$. Additionally, the true system input $u(t) = P(t)$ is not known due to input disturbances $d(t)$. These input disturbances can represent the effect of a varying tip-to-surface distance and are given by zero-mean Gaussian noise with a standard deviation $\sigma_d = 0.1$ W. Thus, the decision variable in the OCP is the known actuator input $\tilde{u}(t) = \tilde{P}(t)$, for which the input bounds $\underline{P} - \tilde{P}(t) \le 0$ and $\tilde{P}(t) - \overline{P} \le 0$ are applicable.

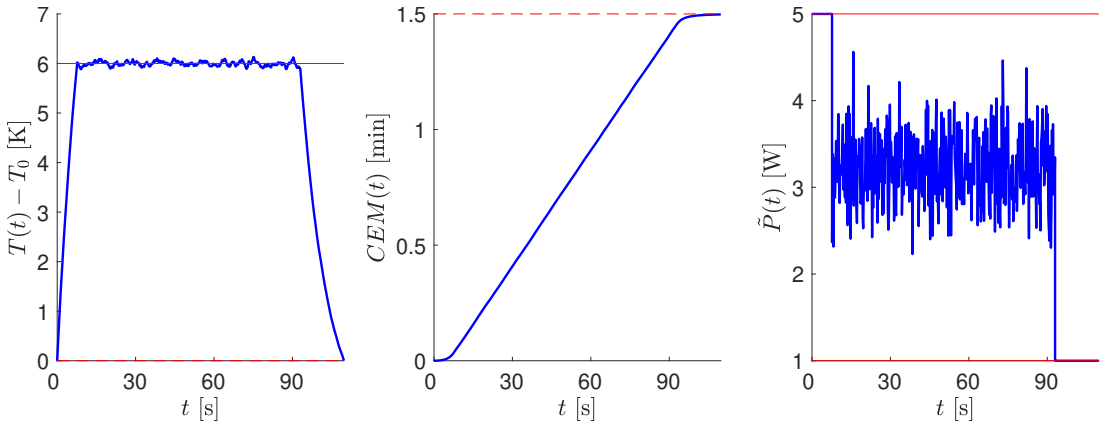

Fig. 4. Closed-loop simulation results. Evolution of $T(t)$, $CEM(t)$, and $\tilde{P}(t)$ over time for the nominal point of iteration $k = 5$, with the setpoint in the second arc and upper bound $\overline{T}$ for $T(t)$ and the lower and upper bounds $\underline{P}$ and $\overline{P}$ for $\tilde{P}(t)$ represented by solid red lines and the terminal constraints for $T(t)$ and $CEM(t)$ represented by dashed red lines.

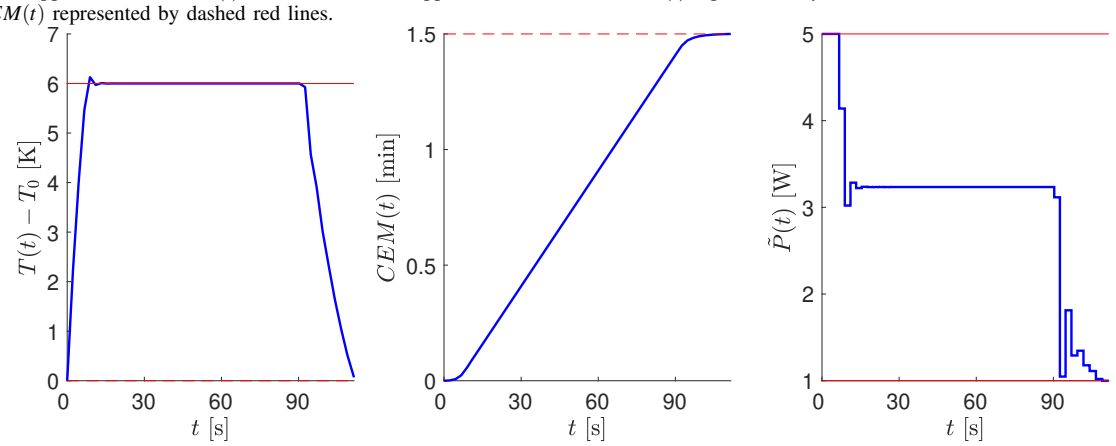

Fig. 5. Closed-loop simulation results for nominal MPC without measurement noise. Evolution of $T(t)$, $CEM(t)$, and $\tilde{P}(t)$ over time, with the upper bound $\overline{T}$ for $T(t)$ and the lower and upper bounds $\underline{P}$ and $\overline{P}$ for $\tilde{P}(t)$ represented by solid red lines and the terminal constraints for $T(t)$ and $CEM(t)$ represented by dashed red lines.

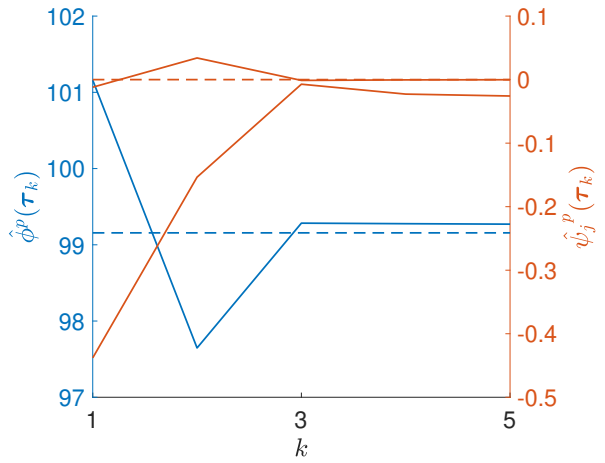

Fig. 6. Evolution of the cost (in blue) and constraints (in red) as a function of the iteration number k of the adaptive optimal control algorithm for different initial conditions ($T_0 = 312.15$ K). The dashed lines represent the optimal values.

Finally, the measurements of $T(t)$ are corrupted by zero-mean Gaussian noise with a standard deviation $\sigma_T = 0.1$ K.

The OCP (35) is first solved using the uncertain model (10). This procedure yields the optimal values of the decision variables $\bar{t}_1^* = 93.99$ s and $t_f^* = 111.72$ s. This corresponds to a worse cost than the optimal cost of 109.45 s in the absence of the plant-model mismatch, while the constraints are inactive and feasible with $T^*(t_f^*) - T_0 = -0.120$ K, $CEM_{sp} - CEM^*(t_f^*) = -0.017$ min. These values of the decision variables become the first point $\boldsymbol{\tau}_1$ evaluated by the adaptive algorithm, from which the algorithm should converge to the true solution. To this end, the algorithm uses measurements of the terminal constraints. At each iteration k of the adaptive algorithm, besides the nominal point $\boldsymbol{\tau}_k$, $N = 2$ auxiliary points $\boldsymbol{\tau}_k^j$ are evaluated. The step away from each nominal point to obtain an auxiliary point corresponds to $+1.0$ s for $\bar{t}_1$ and $+1.0$ s for t_f. Since the proposed OCP for the considered setup is solved offline, that is, between consecutive runs of the APPJ, the time needed to solve it does not need to be extremely low. Nevertheless, the OCP is solved in less than one second.

Five iterations of the adaptive algorithm are performed to examine the convergence to the true system. Fig. 3 shows the evolution of the cost and constraints for the true system. We observe that the cost for the true system already approaches its optimal value at the iteration steps $k = 2$ and $k = 3$, while the constraints become active with a slight violation. From

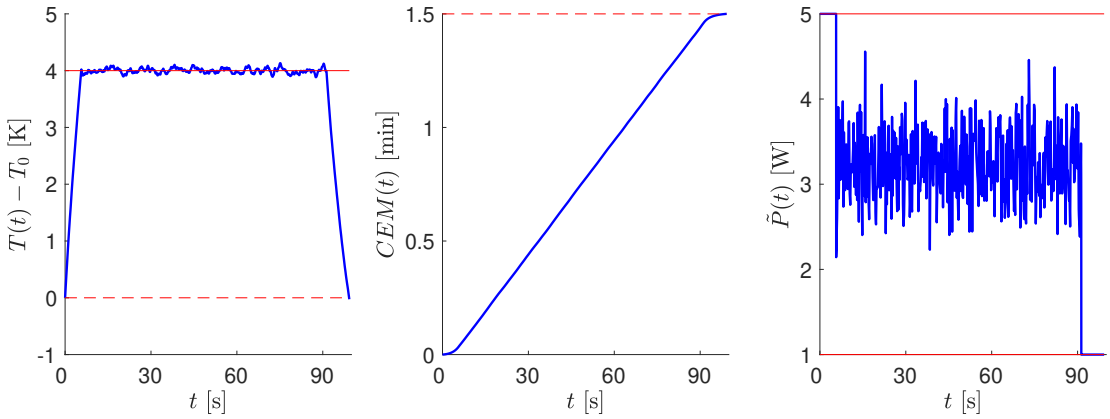

Fig. 7. Closed-loop simulation results for different initial conditions ($T_0 = 312.15$ K). Evolution of $T(t)$, $CEM(t)$, and $\tilde{P}(t)$ over time for the nominal point of iteration $k = 5$, with the setpoint in the second arc and upper bound $\overline{T}$ for $T(t)$ and the lower and upper bounds $\underline{P}$ and $\overline{P}$ for $\tilde{P}(t)$ represented by solid red lines and the terminal constraints for $T(t)$ and $CEM(t)$ represented by dashed red lines.

iteration $k = 4$, both the cost and the constraints for the true system become almost equal to their optimal values, which indicates convergence to the optimal solution to the nonlinear thermal effects delivery problem for the true system. After five iterations, the cost for the true system converges to $t_f^* = 109.72$ s, which is nearly equal to the optimal cost of 109.45 s for the true system. The violation of the constraints $T^*(t_f^*) - T_0 \leq 0$ and $CEM_{sp} - CEM^*(t_f^*) \leq 0$ remains under 0.02 K and 0.025 min for all the iterations, respectively.

For each point $\boldsymbol{\tau}_k^j$ evaluated by the adaptive algorithm, the proposed control strategy via feedback linearization and derivative estimation is used to track the constraint $T(t) - \overline{T} = 0$ in the second arc. This control strategy ensures elimination of steady-state error and rejection of input disturbances despite the fact that the controller does not include any integral term and uses an incomplete model, that is, without knowledge of $\varphi\big(T(t)\big)$. The nominal values $\tau_c = 1$ s and $\Delta t = 3$ s of the design parameters are used. This choice is based on the assumption that the unknown rate $\varphi\big(T(t)\big)$ remains approximately constant in a window of $\Delta t = 3$ s. Fig. 4 shows the evolution of $T(t)$, $CEM(t)$, and $\tilde{P}(t)$ over time in the true system for the nominal point of iteration $k = 5$. The control strategy successfully tracks the constraint $T(t) - \overline{T} = 0$ in the second arc despite the existence of measurement noise and the lack of knowledge about the true system. In this case, it is adequate to set the back-off parameter to $\Delta T(t) = 0$ since the violation of the path constraint $T(t) - \overline{T} \leq 0$ is negligible.

To demonstrate the advantage of the proposed adaptive scheme with respect to a widely used control strategy, simulation results are also obtained for the nominal MPC approach formulated in Appendix C (that is, MPC without model correction or re-identification, but with online output sampling from the true system) using CasADi [57], which are shown in Fig. 5. Fig. 5 shows that, even in the ideal case of no measurement noise and no disturbance, MPC reaches a cost (i.e., minimum treatment time) of 111.56 s, which is not a significant improvement with respect to the performance obtained by solving the OCP offline using the nominal model, for which a cost of 111.72 s is obtained. In contrast, recall that the proposed adaptive scheme reaches a cost of 109.72 s, which is significantly closer to the optimal cost of 109.45 s for the true system. Furthermore, the costly computation time of MPC (on the order of several seconds) is significantly greater than the proposed path constraint tracking (on the order of milliseconds or less). In this case, we would require that the OCP is solved every several seconds, which is insufficient for the plasma application. While Fig. 3 and Fig. 4 show the results for $T_0 = 310.15$ K, it is also important to assess whether the proposed method is robust to changes in the initial conditions. Hence, Fig. 6 and Fig. 7 present the corresponding results for $T_0 = 312.15$ K. These figures show that the conclusions of the simulation results are also valid for different initial conditions.

The fact that the solution computed after 5 iterations of the proposed adaptive approach is nearly optimal for the true system is further supported by the following facts: the arc sequence and the control strategy ensure that the temperature increases as quickly as possible in the first arc, remains at the upper bound in the second arc, and decreases as quickly as possible in the third arc, which ensures that the final time that corresponds to the terminal cost is minimal for the final CEM that is achieved; and the terminal constraints for the final temperature and final CEM are marginally satisfied, which indicates that these terminal constraints are nearly active and confirms that the solution is nearly optimal. In addition, the computational time needed by the proposed algorithm for real-time implementation is very small owing to the simplicity of the proposed control strategy, and even the computational time in the offline part of the algorithm, that is, between consecutive runs of the APPJ, is not significant. Hence, the proposed approach appears to have benefits over alternative methods for the application under study, which validates the applicability of the proposed approach. In summary, the closed-loop simulation results indicate that, despite the existence of disturbances and structural mismatch between the true system and its model, the combination of the adaptive algorithm with an appropriate control strategy for tracking of active path constraints enforces convergence of the true system to the optimal solution to the OCP with

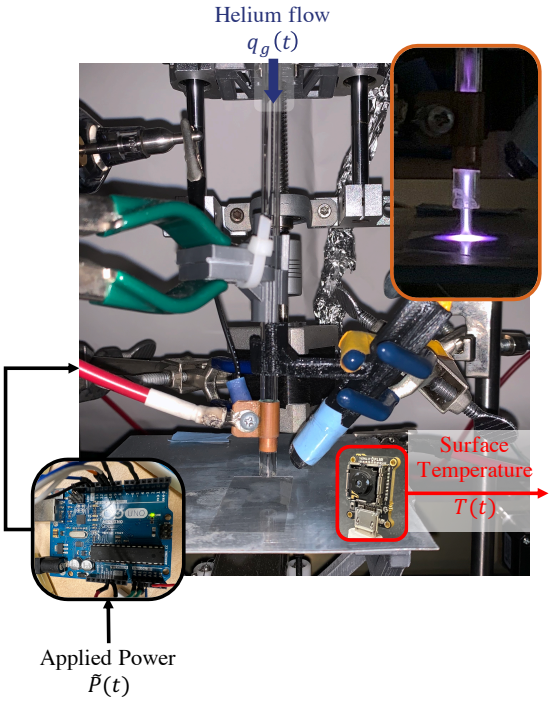

Fig. 8. Labeled (corresponding with Fig. 1) image of the kHz-excited APPJ in He used in the real-time control experiments. As described in Section II-A, the setup has the following configuration: A standard laptop CPU (2.4 GHz Quad-Core Intel i5 processor) is connected to the Arduino UNO (outlined in black) to send the computed input (applied power, $\tilde{P}(t)$) values to the setup. The applied power is used to compute an applied voltage via an embedded PI controller. The applied voltage is sent to a function generator (integrated circuit, XR-2602CP; not pictured) and amplifier (TREK 10/40A-HS; not pictured). The high-frequency, high voltage signal is used to excite Helium gas via a copper electrode. Helium gas flows ($q_g(t)$) downward from an external source through the transparent quartz tube. Surface temperature is measured via a thermal camera (outlined in red) placed above the plasma incidence with the surface. A subimage (outlined in orange) of the plasma jet when powered on (emitting a plasma beam) is located in the upper right corner.

negligible constraint violation.

C. Real-time Control Experiments

We now demonstrate the adaptive optimal control approach on the kHz-excited APPJ in He described in Section II-A and shown in Fig. 8. We use the same OCP formulation and APPJ model (10). However, for the experimental study (i) the bounds for the applied power were adjusted to $\underline{P} = 1.5$ W to prevent accidental shutoff of the APPJ and $\overline{P} = 4.5$ W to allow for switching between the first and second arcs; and (ii) the feedback linearization control was tuned to better control the physical system. The solution to the OCP is computed in MATLAB first, once again in less than one second, which yields the optimal values of the decision variables $\bar{t}_1^* = 85.8$ s and $t_f^* = 101.7$ s. The constraints are inactive and feasible with $T^*(t_f^*) - T_0 = -1.8$ K, $CEM_{sp} - CEM^*(t_f^*) = -0.22$ min. These values of the decision variables become the first point $\boldsymbol{\tau}_1$ evaluated by the adaptive algorithm. For five iterations of the adaptive algorithm, besides an experimental run at the nominal point $\boldsymbol{\tau}_k$, two additional experiments are run for the $N = 2$ auxiliary points $\boldsymbol{\tau}_k^j$. Again, the step away from each nominal point to obtain an auxiliary point corresponds to $+1.0$ s for $\bar{t}_1$ and $+1.0$ s for t_f. To mitigate run-to-run variations, experiments are procedurally performed with the same initial conditions ($\tilde{P} = 2$ W, $q_g = 3$ slm) to reach an initial surface temperature $T = 310.15$ K before collecting experimental data and applying the control approach.

As in the simulation case study, at each point $\boldsymbol{\tau}_k^j$ evaluated by the adaptive algorithm, the control strategy via feedback linearization and derivative estimation is used to track the constraint $T(t) - \overline{T} = 0$ in the second arc without assuming any knowledge of the rate $\varphi\big(T(t)\big)$. The design parameters were chosen as $\tau_c = 3$ s and $\Delta t = 5$ s to improve control performance under the limitations of the physical system (i.e., restricted measurement frequency). Fig. 9 shows the resulting evolution of $T(t)$, $CEM(t)$, and $\tilde{P}(t)$ over time for the nominal point of iterations $k = 1, 3, 5$. The experimental results suggest that the control strategy adequately tracks the constraint $T(t) - \overline{T} = 0$ in the second arc despite the existence of measurement noise and the lack of knowledge about the true system. Run-to-run variations appear to affect the initial stages of control significantly. Nonetheless, the proposed adaptive optimal control approach is able to converge to the desired CEM setpoint in as few as five iterations despite the simple physics-based model (10) used in the OCP. In practice, back-off parameter tuning to mitigate the temperature constraint violation should be addressed according to the specifications of the application. The solution computed after 5 iterations of the proposed adaptive approach is nearly optimal for the true system for the reasons mentioned for the simulation case study, and the computational time is again small.

V. Conclusions

This paper proposed an adaptive optimal control approach for controlling the delivery of thermal effects of plasma while minimizing the plasma treatment time. This approach relies on: (i) Pontryagin's maximum principle and parsimonious input parameterization to convert the OCP into a numerical optimization problem with only a few decision variables; (ii) modifier adaptation to update the OCP such that it converges to the optimal solution for the true APPJ system after few iterations; and (iii) feedback linearization and derivative estimation for online tracking of some active path constraints in the solution to the OCP. The proposed approach is a relatively simple and intuitive way to optimize the delivery of plasma effects to a surface, and overcomes the loss of optimality due to plant-model mismatch. Closed-loop simulations and real-time control experiments demonstrated that the approach enables convergence to the optimal solution for the true plasma jet despite the existence of measurement noise and plant-model mismatch. The approach proposed in this paper is potentially applicable to other nonlinear, uncertain systems other than cold atmospheric plasmas. Future work will focus on extending the proposed approach to more complex models of APPJs and controlling nonthermal effects of plasma.

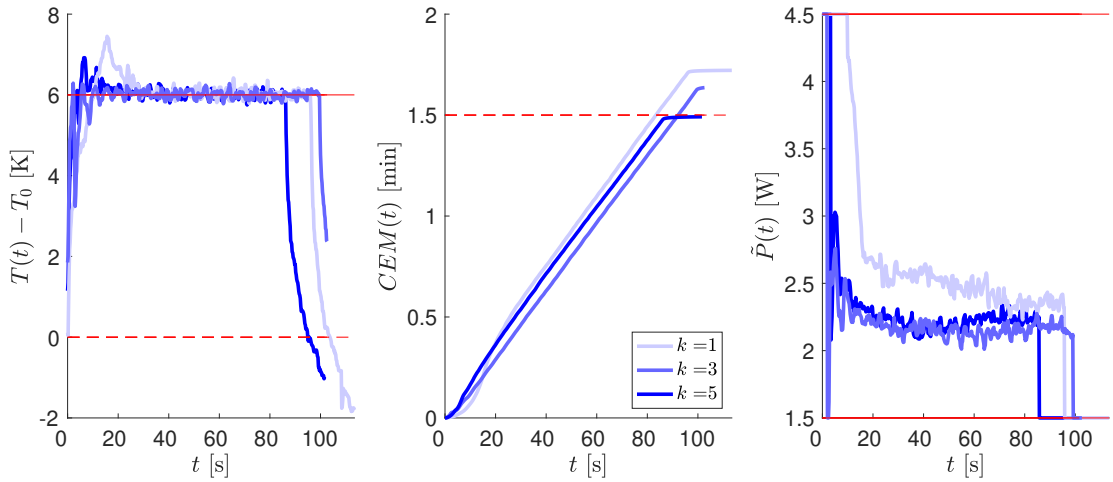

Fig. 9. Real-time control experiments on the kHz-excited APPJ in He. Evolution of $T(t)$, $CEM(t)$, and $\tilde{P}(t)$ over time for the nominal point of iterations $k = 1, 3, 5$, with the setpoint in the second arc and upper bound $\overline{T}$ for $T(t)$ and the lower and upper bounds $\underline{P}$ and $\overline{P}$ for $\tilde{P}(t)$ represented by solid red lines and the terminal constraints for $T(t)$ and $CEM(t)$ represented by dashed red lines.

APPENDIX

A. Continuous-time Derivative Estimation

An infinite impulse response (IIR) filter can also be applied to $\tilde{y}_c(t)$ and $\tilde{y}_a(t)$, which results in a linear observer [58], [59]. However, the following results about derivative estimation correspond to the FIR filter. The key advantages of the FIR filter with respect to an IIR filter are the property of deadbeat estimation of the unknown derivative after the window size Δt, instead of asymptotic convergence, and the minimization of the effect of measurement noise $w_c(t)$ for a given Δt without the need to tune the observer parameters.

Since $r(t)$ may be discontinuous, the same is valid for $\tilde{u}(t)$ and $\tilde{y}_a(t)$ and, consequently, for $u(t)$, $\dot{y}_a(t)$, and $\dot{y}_c(t)$. On the other hand, since $\mathbf{x}(t)$ is absolutely continuous, the same is valid for $y(t)$ and $\dot{y}_u(t)$. We assume that $w(t)$ is absolutely continuous, thus the same is valid for $\tilde{y}(t)$ and $\tilde{y}_c(t)$. The fact that these functions are absolutely continuous is used in some technical results in this paper. We propose to compute the estimate $\hat{y}_u(t)$ as the quantity that minimizes the mean squared error between the sensor output $\tilde{y}_c(t-\Delta t+\tau)$ and the predicted output $\hat{y}_c(t-\Delta t+\tau)$ for $\tau \in [0, \Delta t]$, with the goal of minimizing the effect of $w_c(t)$ for a given Δt. This mean squared error is defined as

$$J(t) = \tfrac{1}{\Delta t}\int_0^{\Delta t} \hat{\varepsilon}(t-\Delta t+\tau)^2 \mathrm{d}\tau, \tag{64}$$

where the prediction error is

$$\begin{aligned}\hat{\varepsilon}(t-\Delta t+\tau) &= \tilde{y}_c(t-\Delta t+\tau) \\ &\quad - \hat{y}_c(t) - \textstyle\int_{\Delta t}^{\tau} \hat{\dot{y}}_c(t-\Delta t+\zeta)\mathrm{d}\zeta,\end{aligned} \tag{65}$$

with

$$\hat{\dot{y}}_c(t-\Delta t+\tau) = \hat{\dot{y}}_u(t) + \tilde{\dot{y}}_a(t-\Delta t+\tau). \tag{66}$$

The predicted output $\hat{y}_c(t)$ that minimizes the mean squared error $J(t)$ is given by

$$\hat{\dot{y}}_u(t) = \tfrac{1}{\Delta t^2}\textstyle\int_0^{\Delta t} \left(12\frac{\tau}{\Delta t} - 6\right) \upsilon(t-\Delta t+\tau)\mathrm{d}\tau, \tag{67}$$

where

$$\upsilon(t-\Delta t+\tau) = \tilde{y}_c(t-\Delta t+\tau) - \textstyle\int_{\Delta t}^{\tau} \tilde{\dot{y}}_a(t-\Delta t+\zeta)\mathrm{d}\zeta. \tag{68}$$

Hence, we define $\mathscr{D}(\tilde{y}_c, t)$ as

$$\mathscr{D}(\tilde{y}_c, t) := \tfrac{1}{\Delta t^2}\textstyle\int_0^{\Delta t} \left(12\frac{\tau}{\Delta t} - 6\right) \tilde{y}_c(t-\Delta t+\tau)\mathrm{d}\tau. \tag{69}$$

The function $\mathscr{D}(\tilde{y}_c, t)$ corresponds to the continuous-time version of the differentiation Savitzky-Golay filter of order 1, which was developed originally for the discrete-time case [60] and is known as algebraic time-derivative estimation in the continuous-time case [48]. Also, the use of a smaller window size Δt of the differentiation filter amplifies the effect of measurement noise in $w_c(t)$.

B. Parameter Values

Parameter values for the adaptive control approach are given in Table I and Table II.

TABLE I
SIMULATED APPJ MODEL PARAMETERS

Parameter	Description	Value
K	Base of exponential dependence of variation of CEM on the surface temperature	0.5
T_{ref}	Reference temperature for variation of CEM	316.15 K
T_∞	Ambient temperature	298.15 K
T_b	Surface bulk temperature	308.15 K
ρ	Surface density	2800 kg m^{-3}
c_p	Surface heat capacity	795 J kg^{-1} K^{-1}
r	Radius of the plasma jet	1.5×10^{-3} m
d	Surface thickness	0.2×10^{-3} m
k	Heat transfer coefficient	1.43 W m^{-2} K^{-1}
β	Dimensionless constant of heat transfer	90.82
μ	Dimensionless constant of heat capacity	9.84×10^{-4}

C. Nominal Model Predictive Control Formulation

The optimal control problem (OCP) for nominal model predictive control (MPC) utilizes the same OCP formulation to minimize the treatment time as described in general in (12)

TABLE II

CONTROL PARAMETERS

Parameter	Description	Value
T_0	Initial surface temperature	310.15 K
CEM_{sp}	Desired thermal effect in CEM	1.5 min
$\underline{P}$	Lower bound for applied power	1.0 W
$\overline{P}$	Upper bound for applied power	5.0 W
$\overline{T}$	Upper bound for surface temperature	316.15 K

and specifically in (13)–(16). The specific OCP is replicated here for clarity,

$$\min_{P(\cdot),t_f} \quad t_f, \tag{70a}$$

$$\text{s.t.} \quad \begin{bmatrix} T(t_f)-T_0 \\ CEM_{sp}-CEM(t_f) \end{bmatrix} \leq \mathbf{0}_2, \tag{70b}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), u(t)\big), \quad \forall t \in [t_0, t_f], \tag{70c}$$

$$\mathbf{x}(t_0) = \mathbf{x}(t), \tag{70d}$$

$$\begin{bmatrix} \underline{P}-P(t) \\ P(t)-\overline{P} \end{bmatrix} \leq \mathbf{0}_2, \quad \forall t \in [t_0, t_f], \tag{70e}$$

$$T(t) - \overline{T} \leq 0, \quad \forall t \in [t_0, t_f], \tag{70f}$$

where, $P(\cdot)$ is the piecewise-constant applied power, and, as a reminder to the reader, (70c) represents the nonlinear state-space model describing the plasma (11). We assume the piecewise-constant applied power $P(\cdot)$ has a shrinking length, starting with 50 discrete steps. The optimal input trajectory is obtained by solving this OCP (70), and the first input is applied to the true system, as defined by the following control law,

$$\kappa(\mathbf{x}) = P^\star(0; \mathbf{x}(t)). \tag{71}$$

We assume that the solution to the OCP is only recomputed after the duration of the application of this first optimal input. We assume the ideal case of no measurement noise and no input disturbance, which may otherwise require the use of a smoothing filter to return precise output measurements at each computation of the MPC.

REFERENCES

[1] L. Minati, C. Migliaresi, L. Lunelli, G. Viero, M. Dalla Serra, and G. Speranza, "Plasma assisted surface treatments of biomaterials," *Biophys. Chem.*, vol. 229, pp. 151–164, 2017.

[2] J. Friedrich, "Mechanisms of plasma polymerization–reviewed from a chemical point of view," *Plasma Processes and Polym.*, vol. 8, no. 9, pp. 783–802, 2011.

[3] S.-j. Lee, D. Yan, X. Zhou, H. Cui, T. Esworthy, S. Y. Hann, M. Keidar, and L. G. Zhang, "Integrating cold atmospheric plasma with 3D printed bioactive nanocomposite scaffold for cartilage regeneration," *Mater. Sci. Eng., C*, vol. 111, p. 110844, 2020.

[4] E. Stoffels, A. Flikweert, W. Stoffels, and G. Kroesen, "Plasma needle: a non-destructive atmospheric plasma source for fine surface treatment of (bio) materials," *Plasma Sources Sci. Technol.*, vol. 11, no. 4, p. 383, 2002.

[5] R. B. Gadri, J. R. Roth, T. C. Montie, K. Kelly-Wintenberg, P. P.-Y. Tsai, D. J. Helfritch, P. Feldman, D. M. Sherman, F. Karakaya, Z. Chen *et al.*, "Sterilization and plasma processing of room temperature surfaces with a one atmosphere uniform glow discharge plasma (OAUGDP)," *Surf. Coat. Technol.*, vol. 131, no. 1-3, pp. 528–541, 2000.

[6] J. Y. Jeong, S. E. Babayan, V. J. Tu, J. Park, I. Henins, R. F. Hicks, and G. S. Selwyn, "Etching materials with an atmospheric-pressure plasma jet," *Plasma Sources Sci. Technol.*, vol. 7, no. 3, pp. 282–285, 1998.

[7] M. Laroussi, M. Kong, G. Morfill, and W. Stolz, *Plasma Medicine*. New York: Cambridge University Press, 2012.

[8] A. Schutze, J. Y. Jeong, S. E. Babayan, J. Park, G. S. Selwyn, and R. F. Hicks, "The atmospheric-pressure plasma jet: a review and comparison to other plasma sources," *IEEE Trans. Plasma Sci.*, vol. 26, no. 6, pp. 1685–1694, 1998.

[9] J. Park, I. Henins, H. Herrmann, G. Selwyn, J. Jeong, R. Hicks, D. Shim, and C. Chang, "An atmospheric pressure plasma source," *Appl. Phys. Lett.*, vol. 76, no. 3, pp. 288–290, 2000.

[10] X. Lu, M. Laroussi, and V. Puech, "On atmospheric-pressure non-equilibrium plasma jets and plasma bullets," *Plasma Sources Sci. Technol.*, vol. 21, no. 3, p. 034005, 2012.

[11] M. W. Dewhirst, B. L. Viglianti, M. Lora-Michiels, M. Hanson, and P. J. Hoopes, "Basic principles of thermal dosimetry and thermal thresholds for tissue damage from hyperthermia," *Int J. Hyperthermia*, vol. 19, no. 3, pp. 267–294, 2003.

[12] G. Fridman, A. D. Brooks, M. Balasubramanian, A. Fridman, A. Gutsol, V. N. Vasilets, H. Ayan, and G. Friedman, "Comparison of direct and indirect effects of non-thermal atmospheric-pressure plasma on bacteria," *Plasma Processes Polym.*, vol. 4, no. 4, pp. 370–375, 2007.

[13] J. L. Roti Roti, "Cellular responses to hyperthermia (40-46 degrees C): cell killing and molecular events," *Int J. Hyperthermia*, vol. 24, no. 1, pp. 3–15, 2008.

[14] D. Staack, B. Farouk, A. Gutsol, and A. Fridman, "Stabilization of the ionization overheating thermal instability in atmospheric pressure microplasmas," *J. Appl. Phys.*, vol. 106, no. 1, p. 013303, 2009.

[15] J. J. Liu and M. G. Kong, "Sub-60°C atmospheric helium-water plasma jets: modes, electron heating and downstream reaction chemistry," *J. Phys. D: Appl. Phys.*, vol. 44, no. 34, p. 345203, 2011.

[16] A. D. Bonzanini, D. B. Graves, and A. Mesbah, "Learning-based SMPC for reference tracking under state-dependent uncertainty: An application to atmospheric pressure plasma jets for plasma medicine," *IEEE Trans. Control Syst. Technol.*, 2021.

[17] A. Mesbah and D. B. Graves, "Machine learning for modeling, diagnostics, and control of non-equilibrium plasmas," *J. Phys. D: Appl. Phys.*, vol. 52, no. 30, p. 30LT02, 2019.

[18] O. Baranov, K. Bazaka, H. Kersten, M. Keidar, U. Cvelbar, S. Xu, and I. Levchenko, "Plasma under control: Advanced solutions and perspectives for plasma flux management in material treatment and nanosynthesis," *Appl. Phys. Rev.*, vol. 4, no. 4, p. 041302, 2017.

[19] D. Gidon, D. B. Graves, and A. Mesbah, "Effective dose delivery in atmospheric pressure plasma jets for plasma medicine: a model predictive control approach," *Plasma Sources Sci. Technol.*, vol. 26, no. 8, p. 085005, 2017.

[20] D. Gidon, B. Curtis, J. A. Paulson, D. B. Graves, and A. Mesbah, "Model-based feedback control of a kHz-excited atmospheric pressure plasma jet," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 2, no. 2, pp. 129–137, 2018.

[21] D. Gidon, H. S. Abbas, A. D. Bonzanini, D. B. Graves, J. M. Velni, and A. Mesbah, "Data-driven lpv model predictive control of a cold atmospheric plasma jet for biomaterials processing," *Control Eng. Pract.*, vol. 109, p. 104725, 2021.

[22] D. Gidon, X. Pei, A. D. Bonzanini, D. B. Graves, and A. Mesbah, "Machine learning for real-time diagnostics of cold atmospheric plasma sources," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 3, no. 5, pp. 597–605, 2019.

[23] D. Gidon, D. B. Graves, and A. Mesbah, "Spatial thermal dose delivery in atmospheric pressure plasma jets," *Plasma Sources Sci. Technol.*, vol. 28, no. 2, p. 025006, 2019.

[24] E. Gjika, S. Pal-Ghosh, A. Tang, M. Kirschner, G. Tadvalkar, J. Canady, M. A. Stepp, and M. Keidar, "Adaptation of operational parameters of cold atmospheric plasma for in vitro treatment of cancer cells," *ACS Appl. Mater. Interfaces*, vol. 10, no. 11, pp. 9269–9279, 2018.

[25] Y. Lyu, L. Lin, E. Gjika, T. Lee, and M. Keidar, "Mathematical modeling and control for cancer treatment with cold atmospheric plasma jet," *J. Phys. D: Appl. Phys.*, vol. 52, no. 18, p. 185202, 2019.

[26] L. Lin, Z. Hou, X. Yao, Y. Liu, J. R. Sirigiri, T. Lee, and M. Keidar, "Introducing adaptive cold atmospheric plasma: The perspective of adaptive cold plasma cancer treatments based on real-time electrochemical impedance spectroscopy," *Phys. Plasma*, vol. 27, no. 6, p. 063501, 2020.

[27] M. Witman, D. Gidon, D. B. Graves, B. Smit, and A. Mesbah, "Sim-to-real transfer reinforcement learning for control of thermal effects

of an atmospheric pressure plasma jet," *Plasma Sources Sci. Technol.*, vol. 28, no. 9, p. 095019, 2019.

[28] A. D. Bonzanini, J. A. Paulson, D. B. Graves, and A. Mesbah, "Toward safe dose delivery in plasma medicine using projected neural network-based fast approximate NMPC," in *Proc. 21st IFAC World Congress*, Berlin, Germany, 2020, pp. 5353–5359.

[29] A. D. Bonzanini, J. A. Paulson, G. Makrygiorgos, and A. Mesbah, "Fast approximate learning-based multistage nonlinear model predictive control using Gaussian processes and deep neural networks," *Comput. Chem. Eng.*, vol. 145, p. 107174, 2021.

[30] A. D. Bonzanini, K. Shao, A. Stancampiano, D. B. Graves, and A. Mesbah, "Perspectives on machine learning-assisted plasma medicine: Towards automated plasma treatment," *IEEE Transactions on Radiation and Plasma Medical Sciences*, 2021.

[31] S. Tong, K. Sun, and S. Sui, "Observer-based adaptive fuzzy decentralized optimal control design for strict-feedback nonlinear large-scale systems," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 2, pp. 569–584, 2018.

[32] Y. Li, K. Sun, and S. Tong, "Adaptive fuzzy robust fault-tolerant optimal control for nonlinear large-scale systems," *IEEE Transactions on Fuzzy Systems*, vol. 26, no. 5, pp. 2899–2914, 2018.

[33] T. Wang and S. Tong, "Observer-based fuzzy adaptive optimal control for nonlinear continuous-time saturation interconnected systems," *Optimal Control Applications and Methods*, vol. 39, no. 4, pp. 1273–1290, 2018.

[34] Y. Li, K. Sun, and S. Tong, "Observer-based adaptive fuzzy fault-tolerant optimal control for SISO nonlinear systems," *IEEE Transactions on Cybernetics*, vol. 49, no. 2, pp. 649–661, 2019.

[35] Y. Li, T. Yang, and S. Tong, "Adaptive neural networks finite-time optimal control for a class of nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4451–4460, 2020.

[36] Y. Zhang, B. Zhao, and D. Liu, "Deterministic policy gradient adaptive dynamic programming for model-free optimal control," *Neurocomputing*, vol. 387, pp. 40–50, 2020.

[37] T. Zhang, H. Xu, X. Xia, and Y. Yi, "Adaptive neural optimal control of uncertain nonlinear systems with output constraints," *Neurocomputing*, vol. 406, pp. 182–195, 2020.

[38] X. Yang and H. He, "Self-learning robust optimal control for continuous-time nonlinear systems with mismatched disturbances," *Neural Networks*, vol. 99, pp. 19–30, 2018.

[39] A. Wang, X. Liao, and T. Dong, "Event-driven optimal control for uncertain nonlinear systems with external disturbance via adaptive dynamic programming," *Neurocomputing*, vol. 281, pp. 188–195, 2018.

[40] A. Marchetti, B. Chachuat, and D. Bonvin, "Modifier-adaptation methodology for real-time optimization," *Ind. Eng. Chem. Res.*, vol. 48, no. 13, pp. 6022–6033, 2009.

[41] A. G. Marchetti, T. Faulwasser, and D. Bonvin, "A feasible-side globally convergent modifier-adaptation scheme," *J. Process Control*, vol. 54, pp. 38–46, 2017.

[42] T. A. Ferreira, H. A. Shukla, T. Faulwasser, C. N. Jones, and D. Bonvin, "Real-time optimization of uncertain process systems via modifier adaptation and Gaussian processes," in *Proc. 2018 European Control Conference (ECC)*, Limassol, Cyprus, 2018, pp. 465–470.

[43] H. Nijmeijer and A. J. van der Schaft, *Nonlinear Dynamical Control Systems*. New York: Springer-Verlag, 1990.

[44] A. Isidori, *Nonlinear Control Systems*, 3rd ed. London: Springer-Verlag, 1995.

[45] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 2000.

[46] M. A. Henson and D. E. Seborg, "Critique of exact linearization strategies for process control," *J. Process Control*, vol. 1, no. 3, pp. 122–139, 1991.

[47] M. Mboup, C. Join, and M. Fliess, "Numerical differentiation with annihilators in noisy environment," *Numer. Algor.*, vol. 50, pp. 439–467, 2009.

[48] J. Reger and J. Jouffroy, "On algebraic time-derivative estimation and deadbeat state reconstruction," in *Proc. 48th IEEE Conference on Decision and Control (CDC)*, Shanghai, P.R. China, 2009, pp. 1740–1745.

[49] D. Rodrigues and H. Hjalmarsson, "Stability and performance analysis of control based on incomplete models," *IFAC-PapersOnLine*, vol. 52, no. 1, pp. 874–879, 2019.

[50] S. A. Sapareto and W. C. Dewey, "Thermal dose determination in cancer therapy," *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 10, no. 6, pp. 787–800, 1984.

[51] R. F. Hartl, S. P. Sethi, and R. G. Vickson, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM Rev.*, vol. 37, no. 2, pp. 181–218, 1995.

[52] B. Srinivasan, S. Palanki, and D. Bonvin, "Dynamic optimization of batch processes: I. Characterization of the nominal solution," *Comput. Chem. Eng.*, vol. 27, no. 1, pp. 1–26, 2003.

[53] D. Rodrigues and D. Bonvin, "On reducing the number of decision variables for dynamic optimization," *Optim. Control Appl. Meth.*, vol. 41, pp. 292–311, 2020.

[54] X. Xu and P. J. Antsaklis, "Optimal control of switched systems based on parameterization of the switching instants," *IEEE Trans. Autom. Contr.*, vol. 49, no. 1, pp. 2–16, 2004.

[55] J. A. Paulson and A. Mesbah, "Nonlinear model predictive control with explicit backoffs for stochastic systems under arbitrary uncertainty," *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 523–534, 2018.

[56] D. Rodrigues and A. Mesbah, "Multivariable control based on incomplete models via feedback linearization and continuous-time derivative estimation," *Int. J. Robust Nonlinear Control*, vol. 31, no. 18, pp. 9193–9230, 2021.

[57] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Math. Program.*, vol. 11, no. 1, pp. 1–36, 2019.

[58] M. Perrier, S. Feyo de Azevedo, E. C. Ferreira, and D. Dochain, "Tuning of observer-based estimators: theory and application to the on-line estimation of kinetic parameters," *Control Eng. Practice*, vol. 8, no. 4, pp. 377–388, 2000.

[59] L. B. Freidovich and H. K. Khalil, "Performance recovery of feedback-linearization-based designs," *IEEE Trans. Autom. Contr.*, vol. 53, no. 10, pp. 2324–2334, 2000.

[60] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Anal. Chem.*, vol. 36, no. 8, pp. 1627–1639, 1964.