# Toward Safe Dose Delivery in Plasma Medicine using Projected Neural Network-based Fast Approximate NMPC ⋆

**Angelo D. Bonzanini*, Joel A. Paulson**, David B. Graves*, and Ali Mesbah***

**Department of Chemical and Biomolecular Engineering, University of California, Berkeley, CA 94720, USA. Contact:* mesbah@berkeley.edu
***Department of Chemical and Biomolecular Engineering, The Ohio State University, Columbus, OH 43210, USA.*

**Abstract:** Atmospheric pressure plasma jets (APPJs) are increasingly used for biomedical applications. Reproducible and effective operation of APPJs hinges on controlling the nonlinear effects of plasma on a target substrate in the face of intrinsic variabilities of the plasma as well as exogenous disturbances. This paper presents a low-memory fast approximate nonlinear model predictive control (NMPC) strategy for an APPJ with prototypical applications in plasma medicine. The NMPC objective is to regulate the delivery of the cumulative thermal effects of plasma to a substrate, while adhering to constraints pertaining to a patient's safety and comfort. Deep neural networks are used to approximate the implicit NMPC law with a cheap-to-evaluate explicit control law that has low memory requirements. Robust constraint satisfaction is guaranteed by projecting the output of the neural network onto a set that ensures the state stays within an appropriately defined invariant set. Closed-loop simulations and real-time control experiments indicate that the proposed approximate NMPC strategy is effective in handling nonlinear control costs at fast sampling times, while guaranteeing satisfaction of safety-critical system constraints. This work takes a crucial step toward fast NMPC of safety-critical plasma applications using resource-limited embedded control hardware.

*Keywords*: Nonlinear model predictive control; Robust constraint satisfaction; Deep neural networks; Cold atmospheric plasma

## 1. INTRODUCTION

Atmospheric pressure plasma jets (APPJs) are a class of cold atmospheric plasma devices, which are increasingly used in biomedical applications. APPJs are shown to be effective in combating antibiotic-resistant bacteria, shrinking cancerous tumors, and accelerating the healing rate in chronic wounds (Fridman et al., 2008). These therapeutic effects are postulated to arise due to thermal, chemical, and electrical effects acting synergistically. However, APPJs are challenging to control. In particular, they exhibit distributed dynamics across multiple length and time scales. Furthermore, they are multivariable and highly nonlinear systems, which exhibit run-to-run variability even when the operating conditions are close to identical (Shin and Raja, 2007). APPJs are also sensitive to exogenous disturbances due to sharp spatial gradients in temperature and concentration of reactive species.

The importance of predictive control for safe, reproducible, and effective operation of APPJs has recently been shown in Gidon et al. (2017) and Gidon et al. (2018). The primary control objective in biomedical applications of APPJs is to regulate the delivery of nonlinear plasma effects (i.e., "plasma dose") to a complex substrate, while honoring safety-critical constraints during treatment to avoid inflicting damage to patients (Gidon et al., 2019). Nonetheless, an important computational challenge in nonlinear model predictive control (NMPC) of APPJs arises from fast system dynamics and, thus, the need for fast sampling times for feedback control. Furthermore, point-of-care applications of APPJs would require control implementation on resource-limited (i.e., low power and memory) embedded systems.

We aim to develop a fast and safe NMPC strategy for APPJs to enable plasma dose delivery at sufficiently fast sampling times, while guaranteeing satisfaction of safety-critical constraints of the plasma in the presence of uncertainty. Significant work has been done on the development of fast MPC strategies, which can be broadly categorized into the development of: (i) tailored solvers and efficient optimization methods (Richter et al., 2009; Wang and Boyd, 2009); and (ii) *explicit* MPC (Alessio and Bemporad, 2009). Explicit MPC leverages the fact that the solution to a MPC problem reduces to a multiparametric problem that can be pre-computed offline, provided that the control cost function is linear or quadratic, the model is linear, and the constraints are affine (Bemporad et al., 2002). However, explicit MPC based on nonlinear models or cost functions remains a largely open area of research. Furthermore, the number of polytopic regions in which the optimal inputs lie grows exponentially with the number of constraints, which can significantly increase the evaluation time and memory requirements of explicit MPC in control problems with long prediction horizons and/or high state

⋆ This work was supported by the National Science Foundation under Grant 1839527.

dimensions. Considerable work has been done to address these challenges by simplifying the partitioning of the state space (Summers et al., 2011; Jones and Morari, 2010).

Alternatively, there has been increasing interest in so-called *approximate* MPC strategies, which seek to learn a cheap-to-evaluate explicit expression for the controller using data generated from offline solution of an MPC problem. A variety of function approximators have been proposed to derive such control laws, including polynomials (Chakrabarty et al., 2016), radial basis functions (Csekő et al., 2015), and artificial neural networks (Parisini and Zoppoli, 1995; Karg and Lucia, 2018; Chen et al., 2018). Even though approximate MPC strategies have been shown to provide good closed-loop performance in simulations, they do not guarantee important theoretical properties by design, such as stability, constraint satisfaction, and/or recursive feasibility.

This paper presents a low-memory, approximate NMPC strategy for an APPJ with prototypical applications in plasma medicine. The control objective is to regulate the delivery of thermal nonlinear effects of plasma to a target substrate, while ensuring safety-critical constraints are satisfied in the presence of system uncertainty (i.e., plant-model mismatch and exogenous disturbances). Deep neural networks (DNNs) are used to approximate the NMPC law, since DNNs are particularly effective in capturing the piecewise nature of MPC laws with a limited memory footprint (Karg and Lucia, 2018). To ensure input and state constraints are satisfied during closed-loop operation, the control inputs computed by the DNN are projected onto a *robust admissible input* (RAI) set (Paulson and Mesbah, 2020). The RAI set not only guarantees that input constraints are satisfied, but also that the system state remains within a robust invariant subset of the output constraints. Moreover, we propose to obtain a fully offline solution to the projection problem using multiparametric programming, so that online optimization is completely avoided in our approach. The efficacy of the projected neural network-based approximate NMPC strategy is demonstrated in both closed-loop simulations and real-time control experiments. To the best of our knowledge, this work represents the first real-time experimental demonstration of a safe neural network-based NMPC strategy.

## 2. THERMAL DOSE DELIVERY USING APPJS

### 2.1 Atmospheric Pressure Plasma Jet

We investigate predictive control of a kHz-excited atmospheric pressure plasma jet (APPJ) in Helium under safety-critical constraints. An illustration of the experimental setup is shown in Fig. 1. The manipulated variables are the Helium flowrate ($q$) and the power applied to the jet ($P$), while the measured outputs are the substrate temperature ($T$) and the total optical emission intensity ($I$) of the plasma. The substrate temperature is measured using an infrared camera (FIR Lepton) and the emission intensity is measured with a spectrometer (OceanOptics USB2000+, 0.375 nm resolution). Data acquisition and communication between the APPJ, custom-built software, and an Arduino UNO microcontroller is coordinated in Python. A detailed description of the testbed in Fig. 1 is given in Gidon et al. (2018).

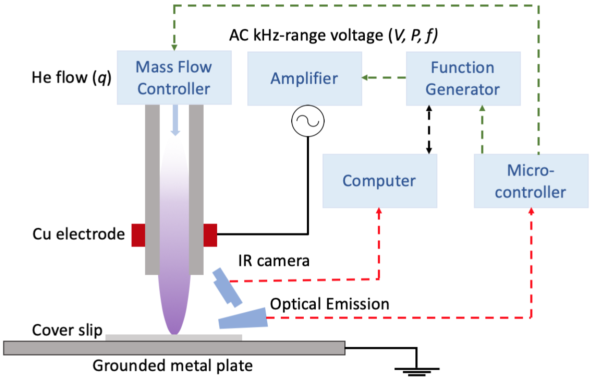


Fig. 1. Schematic of the APPJ testbed. The red dashed lines indicate measured outputs, whereas the green dashed lines indicate manipulated inputs.

First-principles models of plasmas are generally not amenable to real-time control due to the distributed nature of the plasma and plasma-substrate interaction dynamics that span multiple length and times scales. Here, we adopt the subspace identification method to construct a data-driven model of the APPJ operation within a desired operating window (Gidon et al., 2018). The resulting state-space model takes the form

$$x_{k+1} = Ax_k + Bu_k + w_k, \tag{1a}$$

$$y_k = Cx_k + Du_k, \tag{1b}$$

with states $x_k \in \mathbb{R}^{n_x}$ ($n_x = 2$), inputs $u_k = [q\ P]^\top \in \mathbb{R}^2$, outputs $y_k = [T\ \ I]^\top \in \mathbb{R}^2$, and additive disturbance $w_k \in \mathbb{W} \subseteq \mathbb{R}^{n_x}$. The disturbance $w$ is meant to describe the plant-model mismatch and exogenous disturbances acting on the APPJ. The set $\mathbb{W} := \{w_k \mid |w_k|_\infty \le w_\mathrm{b}\}$, where $w_b$ is a fixed uncertainty bound, is estimated by considering the maximum difference between the nominal model predictions (with $w_k = 0$) and the respective outputs in an "unseen" dataset, i.e., a dataset that has not been used in model identification.

### 2.2 Control Problem Formulation

Safe, reproducible, and therapeutically effective operation of APPJs in biomedical applications is particularly challenging due to the intrinsically variable and nonlinear dynamics of the plasma interactions with a target substrate (Gidon et al., 2017). Furthermore, APPJs are very sensitive to exogenous disturbances such as ambient temperature and tip-to-surface separation distance. In addition, there are safety-critical constraints that must be adhered to at all times during a plasma treatment; such as constraints on the maximum substrate temperature that, if exceeded, may cause various undesired effects, ranging from patient discomfort to irreversible cell damage.

Here, the control objective is to deliver a specified thermal dose to a target substrate at the end of a treatment time. In plasma medicine, shorter treatment times are highly desired. We use the thermal dose metric of Cumulative Equivalent Minutes (CEM) at a reference temperature of 43°C to quantify the nonlinear thermal effects of the plasma on a substrate, which is informed by hyperthermia applications (Sapareto and Dewey, 1984). The nonlinear $\mathrm{CEM}_{t_0:t_f}$ dose metric is defined as

$$\mathrm{CEM}_{t_0:t_f} = \int_{t_0}^{t_f} K^{(43-T(t))} dt, \tag{2}$$

where $K$ is a temperature-dependent constant that describes the response of the substrate to thermal stress, and $t_0$ and $t_f$ are the initial and final treatment times, respectively. In practice, both the reference temperature and $K$ can be chosen based on the substrate properties. In this work, we define $K$ as

$$K = 0.5, \text{ if } T \geq 35^\circ C, \quad K = 0, \text{ otherwise.}$$

At the same time, the constraints on the plasma outputs have to be robustly satisfied in the presence of system uncertainties and intrinsic variabilities, which is essential to ensuring the safety and comfort of a patient. The input and output constraints are defined as

$$\begin{aligned} \mathbb{U} &= \{[q\ P]^\top \mid 0.8 \text{ slm} \leq q \leq 10 \text{ slm},\ 0.5 \text{ W} \leq P \leq 5 \text{ W}\}, \\ \mathbb{Y} &= \{[T\ I]^\top \mid 33^\circ\text{C} \leq T \leq 41^\circ\text{C},\ 0 \text{ a.u.} \leq I \leq 250 \text{ a.u.}\}. \end{aligned} \tag{3}$$

At each sampling time $k$, we use the nominal model of (1) to formulate the following optimal control problem (OCP)

$$\begin{aligned} \min_{\mathbf{u}_k} \quad & V_N(\mathbf{u}_k; x_k, p_k) & & \text{(4a)} \\ \text{s.t.} \quad & x_{0|k} = x_k, & & \text{(4b)} \\ & x_{j+1|k} = Ax_{j|k} + Bu_{j|k}, & j \in \{0, \cdots, N-1\}, & \text{(4c)} \\ & y_{j|k} = Cx_{j|k} + Du_{j|k}, & j \in \{0, \cdots, N\}, & \text{(4d)} \\ & y_{j|k} \in \mathbb{Y}, & j \in \{1, \cdots, N\}, & \text{(4e)} \\ & u_{j|k} \in \mathbb{U}, & j \in \{0, \cdots, N-1\}, & \text{(4f)} \end{aligned}$$

where $\mathbf{u}_k := [u_{0|k}^\top \cdots u_{N-1|k}^\top]^\top$ is the vector of decision variables, and $V_N(\mathbf{u}_k; x_k, p_k)$ is a nonlinear thermal dose delivery cost given at each time $k$ given by

$$V_N(\mathbf{u}_k; x_k, p_k) = (p_k + \mathrm{CEM}_{k:k+N} - \mathrm{CEM}_{\mathrm{sp}})^2, \tag{5}$$

where $\mathrm{CEM}_{\mathrm{sp}}$ is the CEM setpoint at the end of the treatment time, $p_k = \mathrm{CEM}_{0:k}$ is the estimated CEM delivered up until the current time that acts as a parameter in our formulation, and the discretized CEM is defined by

$$\mathrm{CEM}_{k:k+N} = \sum_{j=1}^{N} K^{(43-T_{j|k})} \Delta t.$$

The optimal solution to the OCP (4) is denoted by $\mathbf{u}_k^\star(x_k, p_k)$, implemented in a receding-horizon fashion as

$$\kappa_{\mathrm{mpc}}(x_k, p_k) = u_{0|k}^\star(x_k, p_k), \tag{6}$$

where $\kappa_{\mathrm{mpc}}$ denotes the NMPC control law.

The control law (6) is implicitly defined, meaning that the OCP (4) must be solved in real-time given the currently observed $x_k$ and $p_k$. The fast dynamics of APPJs necessitate solving the OCP (4) on fast timescales (on the order of milliseconds). On the other hand, due to the nonlinear control cost (5), we cannot derive an offline solution to (4) using multiparametric programming as in explicit MPC. Thus, we use deep neural networks (DNNs) to learn an explicit control law offline with limited online evaluation cost and low-memory footprint for implementation on resource-limited embedded systems. Given the safety-critical nature of this application, we propose to project the output of the learned explicit controller onto a "safe" input set. This ensures that the explicit controller is recursively feasible and robustly satisfies constraints during closed-loop operation. Note that the proposed approach is independent of the structure of system model (1) or the MPC approach chosen to formulate (4), as shown in Paulson and Mesbah (2020). The projected neural network-based NMPC strategy is presented in the next section.

## 3. APPROXIMATE NMPC WITH GUARANTEED CONSTRAINT SATISFACTION

We first present the deep learning approximation to the NMPC problem (4), which yields an explicit representation of the implicit feedback control law (6). We then discuss how recursive feasibility and robust constraint satisfaction of the resulting explicit control law can be ensured via projection of its outputs onto an appropriately designed invariant set.

### *3.1 Deep Learning-based Approximate NMPC*

Several works have studied the functional approximation of MPC laws (Chakrabarty et al., 2016; Karg and Lucia, 2018; Chen et al., 2018). The idea dates back to the work of Parisini and Zoppoli (1995), in which they proposed to use shallow neural networks (with only one hidden layer) to approximate a NMPC law. This approximation relied on the universal function approximation theory that indicates that a neural network with only one layer can approximate any function to any desired accuracy level under mild conditions (Barron, 1993).

We select DNNs as function approximators in this work for two main reasons. First, recent theoretical results suggest that neural networks with several hidden layers have superior representation power when compared to classical shallow neural networks. Second, promising results for approximating MPC laws with DNNs have recently been demonstrated in, e.g., Karg and Lucia (2018); Chen et al. (2018) among others. The NMPC problem (4) is a parametric optimization problem that depends on the current state and current CEM estimate. To construct a DNN approximation, a finite number of $N_s$ samples of the state $x^{(i)}$ and current CEM $p^{(i)}$ are selected and then $N_s$ optimization problems are solved to obtain the corresponding control inputs $\kappa_{\mathrm{mpc}}(x^{(i)}, p^{(i)})$ as in (6).

A DNN with fully connected layers is defined as a function of the form

$$\mathcal{N}(x, p; \lambda) = \alpha_{L+1} \circ \beta_L \circ \alpha_L \circ \cdots \circ \beta_1 \circ \alpha_1(s), \tag{7}$$

where $s = [x^\top\ p^\top]^\top \in \mathbb{R}^{n_x+1}$ is the input to the network, $\lambda$ are the unknown network parameters, and $L$ is the number of hidden layers. Each hidden layer connects a preceding affine function of the form $\alpha_l(\xi_{l-1}) = W_l \xi_{l-1} + b_l$, where $\xi_{l-1} \in \mathbb{R}^H$ is the output of the previous layer with a nonlinear activation function $\beta_l$, and $H$ denotes the number of nodes per hidden layer. Common choices for the activation function are rectified linear units (ReLUs) and the hyperbolic tangent function.The parameters of all layers are grouped into $\lambda = \{\lambda_1, \ldots, \lambda_{L+1}\}$ with $\lambda_l = \{W_l, b_l\}$, where $W_l$ and $b_l$ are the weights and biases of the affine functions $\alpha_l(\cdot)$, respectively. The best data-driven approximation of (6) for fixed network dimensions $L$ and $H$ can then be defined as the one that minimizes a given loss function, such as the mean squared error (MSE), of a given training dataset

$$\lambda^\star = \underset{\lambda}{\operatorname{argmin}} \ \frac{1}{N_s}\sum_{i=1}^{N_s}(\kappa_{\mathrm{mpc}}(x^{(i)},p^{(i)}) - \mathcal{N}(x^{(i)},p^{(i)};\lambda))^2.$$

The resulting approximate NMPC law is denoted by

$$\kappa_{\mathrm{dnn}}(x,p) = \mathcal{N}(x,p;\lambda^\star).$$

*Remark 1.* We define relevant operating ranges for the state and parameter values in order to generate training data. Whenever these ranges are not known from prior knowledge, they can be estimated from closed-loop simulations, i.e., $x_{k+1} = Ax_k + B\kappa_{\mathrm{mpc}}(x_k,p_k) + w_k$ given a collection of random disturbance values $w_k \in \mathbb{W}$.

### *3.2 Projection-based Robust Constraint Satisfaction*

The approximate NMPC law $\kappa_{\mathrm{dnn}}(x,p)$ in general differs from the exact control law, i.e.,

$$\|\kappa_{\mathrm{mpc}}(x,p) - \kappa_{\mathrm{dnn}}(x,p)\| \le \epsilon_{\mathrm{approx}}, \tag{8}$$

where $\epsilon_{\mathrm{approx}}$ is the error due to approximating (6) by a DNN of fixed size that was trained on a finite dataset. If we could identify tight upper bounds on $\epsilon_{\mathrm{approx}}$, then we could appropriately tighten constraints in (4) in a similar fashion to tube MPC. Methods for deriving such "backoff" parameters have been discussed in Paulson and Mesbah (2018). To deal with the safety-critical nature of the plasma application, we instead present a projection-based method for guaranteed robust constraint satisfaction.

To this end, we leverage robust control invariant (RCI) sets, which play a fundamental role in control. A set $\mathcal{C} \subseteq \mathbb{X}$ is a RCI set if and only if it satisfies the condition

$$\forall x \in \mathcal{C} \Rightarrow \exists u \in \mathbb{U} : Ax + Bu + w \in \mathcal{C}, \ \forall w \in \mathbb{W},$$

where $\mathbb{X} = \{x \in \mathbb{R}^{n_x} \mid \exists u \in \mathbb{U} : Cx + Du \in \mathbb{Y}\}$ is the set of states that satisfy output constraints. Given an RCI set, we can define the corresponding robust admissible input (RAI) set for any state $x \in \mathcal{C}$ as

$$\mathcal{C}_u(x) = \{u \in \mathbb{U} \mid Ax + Bu + w \in \mathcal{C}, \ \forall w \in \mathbb{W}\}.$$

The set $\mathcal{C}_u(x)$ represents a "safe" input set in the sense that any input $u \in \mathcal{C}_u(x)$ ensures the states will evolve within $\mathcal{C} \subseteq \mathbb{X}$. That is, there always exists at least one input that keeps the states within $\mathcal{C}$, which by design lies inside the state/output constraints. Therefore, we can ensure constraints are robustly satisfied in closed-loop by projecting $\kappa_{\mathrm{dnn}}(x,p)$ onto the RAI set

$$\kappa_{\mathrm{pnn}}(x,p) = \underset{u \in \mathcal{C}_u(x)}{\operatorname{argmin}} \ \|u - \kappa_{\mathrm{dnn}}(x,p)\|^2. \tag{9}$$

We refer to the structure of the function $\kappa_{\mathrm{pnn}}(x,p)$ as a projected neural network (PNN).

*Theorem 1.* For any $x_0 \in \mathcal{C}$, the uncertain closed-loop system (1) with $u_k = \kappa_{\mathrm{pnn}}(x_k,p_k)$ must satisfy constraints $y_k \in \mathbb{Y}$ and $\kappa_{\mathrm{pnn}}(x_k,p_k) \in \mathbb{U}$ for all possible disturbances $w_k \in \mathbb{W}$, $p_k \in \mathbb{R}$, and future time steps $k \in \mathbb{N}_0$.

*Proof.* See Paulson and Mesbah (2020).

Note that a similar projection strategy has recently been proposed in Karg and Lucia (2018) and Chen et al. (2018) for deterministic linear systems, as well as in Bonzanini et al. (2020) for the case of linear systems with state-dependent uncertainty.

*Remark 2.* The maximal RCI set $\mathcal{C}^\infty \subseteq \mathbb{X}$ is the RCI set that contains all other RCI sets in $\mathbb{X}$ and represents the largest region of the state space for which an admissible control law exists for all future time steps. Although any RCI set can be used to define (9), it is advantageous to select $\mathcal{C}^\infty$ when computable in order to enable safe operation in the largest possible state space region.

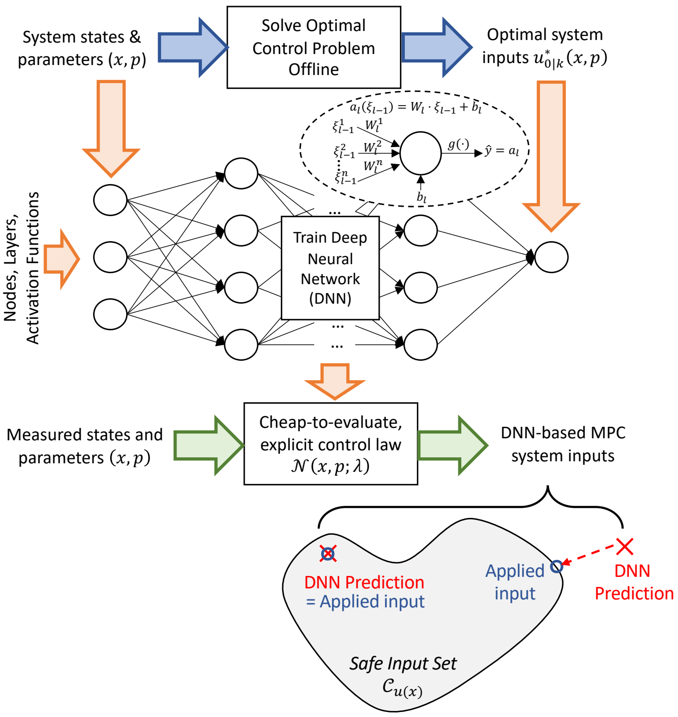

System states & parameters $(x,p)$
Solve Optimal Control Problem Offline
Optimal system inputs $u^*_{0|k}(x,p)$
$a_l(\xi_{l-1}) = W_l \cdot \xi_{l-1} + b_l$
$\xi^1_{l-1}$
$W^1_l$
$\xi^2_{l-1}$
$W^2_l$
$\xi^n_{l-1}$
$W^n_l$
$g(\cdot)$
$\hat{y} = a_l$
$b_l$
Nodes, Layers, Activation Functions
Train Deep Neural Network (DNN)
Measured states and parameters $(x,p)$
Cheap-to-evaluate, explicit control law $\mathcal{N}(x,p;\lambda)$
DNN-based MPC system inputs
DNN Prediction = Applied input
Applied input
DNN Prediction
Safe Input Set $\mathcal{C}_{u(x)}$

Fig. 2. The proposed fast approximate NMPC strategy with guaranteed closed-loop constraint satisfaction.

*Remark 3.* The PNN-based NMPC strategy has been extended to guarantee closed-loop input-to-state stability through the use of polyhedral Lyapunov functions (Paulson and Mesbah, 2020).

### *3.3 Explicit Projected Neural Network Control Laws*

For linear systems with polytopic constraints, the maximal RCI set is a polytope, i.e., $\mathcal{C}^\infty = \{x \in \mathbb{R}^n \mid C_c x \le d_c\}$ that can be computed using standard toolboxes, such as MPT3 (Herceg et al., 2013). Since in this work $\mathbb{U}$ and the Pontryagin difference $\mathcal{C}^\infty \ominus \mathbb{W}$ are polytopes, the projection problem (9) can be written as a standard quadratic program (QP). Instead of solving this optimization problem online, we leverage well-known multiparametric quadratic programming (mpQP) methods to find an explicit solution to the projection (9) fully offline. We note that both $x$ and $\kappa_{\mathrm{dnn}}(x,p)$ can be thought of as parameters that are fed to the projection problem, denoted by $\theta(x,p) = [x^\top \ \kappa_{\mathrm{dnn}}(x,p)^\top]^\top$ for simplicity. As shown in Paulson and Mesbah (2020), the solution to this mpQP is of the form

$$\kappa_{\mathrm{pnn}}(x,p) = K_i\theta(x,p) + h_i, \quad \text{if } E_i\theta(x,p) \le e_i,$$

where the polyhedral sets $\{\theta \mid E_i\theta \le e_i\}_{i=1}^R$ are a partition of $\Theta = \mathcal{C}^\infty \times \mathbb{R}^{n_u}$ composed of $R$ critical regions. An illustrative overview of the proposed PNN-based NMPC strategy is shown in Fig. 2.

*Remark 4.* Even for linear systems, algorithms for RCI set construction based on backward reachability are known to be computationally demanding in high dimensions. When

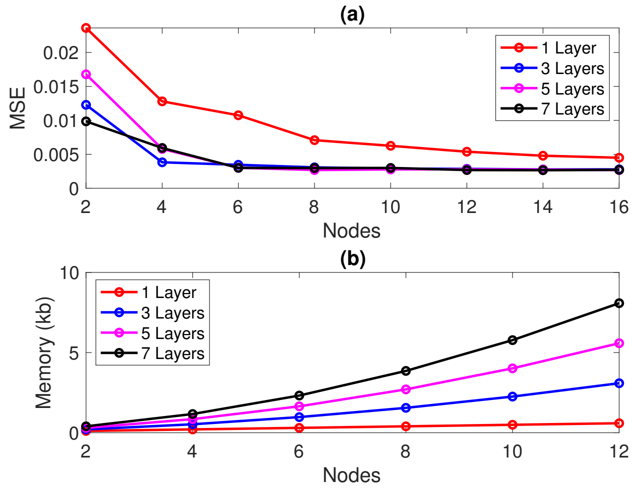


Fig. 3. (a) Mean squared error (MSE) and (b) memory footprint of the DNN approximation of the NMPC law as a function of number of nodes and layers.

this is the case, alternative methods can be employed. For example, Mirko and Mazen (2017) constructed invariant inner approximations to $\mathcal{C}^\infty$ for systems with more than 10 states and inputs.

## 4. RESULTS AND DISCUSSION

This section first discusses the effect of the DNN hyperparameters on the approximation accuracy of the explicit control law of the approximate NMPC strategy. This is followed by demonstrating the performance of the proposed control strategy using closed-loop simulations and real-time control experiments on the APPJ in Fig. 1.

### *4.1 DNN Approximation of NMPC Law*

To ensure adequate approximation of the NMPC law (6), the impact of the two main hyperparameters of the DNN, namely, the number of layers and number of nodes per layer, was explored for a fixed number of training samples $N_s = 5000$. Fig. 3(a) shows the average mean squared error (MSE) as a function of the number of nodes for different number of layers. Due to the stochasticity of the training process, Fig. 3 is generated by repeating the training 5 times for each combination of hyperparameters and averaging the results. Clearly, there is an initial decrease in the MSE as the number of nodes is increased, which eventually plateaus after 6 to 8 nodes. A similar trend is observed when increasing the number of layers. The initial increase from 1 to 3 layers results in a substantial reduction in MSE, while increasing the number of layers further has a negligible impact.

The choice of hyperparameters does not only affect the accuracy of the approximation, but also the corresponding memory footprint. Fig. 3(b) illustrates how the memory footprint of the DNNs increases both with the number of nodes and the number of layers. For example, if one wishes to implement the approximate NMPC law using embedded systems, then the memory footprint must be kept as low as possible, which comes at the expense of higher MSE. Here, we chose to construct the DNN approximation of the NMPC law (6) by setting $L = 5$ and $H = 6$.

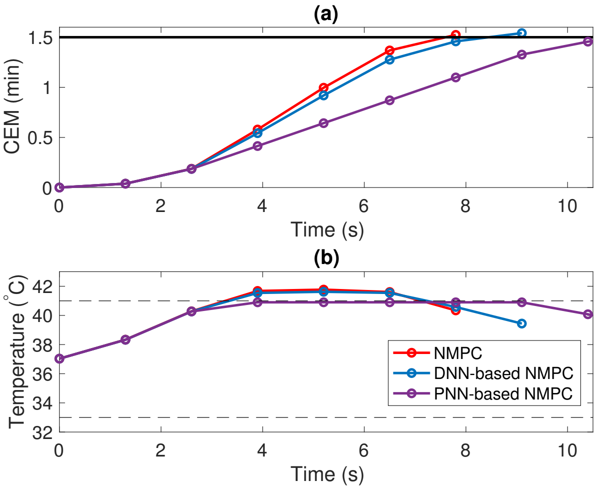


Fig. 4. Closed-loop simulation profiles of (a) CEM dose metric and (b) substrate temperature. The black solid line is the CEM setpoint while the dashed lines are the temperature constraints.

### *4.2 Closed-Loop Simulations*

We compare the performance of three controllers: NMPC whereby the OCP (4) is solved online; DNN-based NMPC whereby the OCP (4) is substituted with the DNN but does not include the projection step; and PNN-based NMPC whereby the projection step is included to guarantee robust constraint satisfaction. The OCP (4) is solved in MATLAB using the CasADi library (Andersson et al., 2019) and the IPOPT algorithm (Wächter and Biegler, 2006). Closed-loop simulations are carried out by introducing a plant-model mismatch through the uncertainty $w_k$ in (1). The value of $w_k$ is fixed to its upper (worst-case) bound, which is chosen to be 1.25°C based on the data used for system identification. The prediction horizon is fixed to $N = 10$, such that the NMPC can be deployed online to achieve a fair comparison with the DNN-based and PNN-based NMPC controllers.

The closed-loop simulation results for the three controllers are shown in Fig. 4. Note that the plasma is shut off as soon as the CEM setpoint is reached to avoid exceeding the target thermal dose delivery and thus potentially inflicting damage to the substrate. Since the CEM is a cumulative thermal dose metric, it is not possible to reduce it. Fig. 4 suggests that the NMPC and DNN-based NMPC exhibit very similar control performance, indicating that

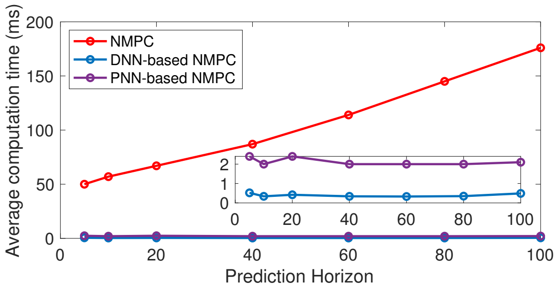


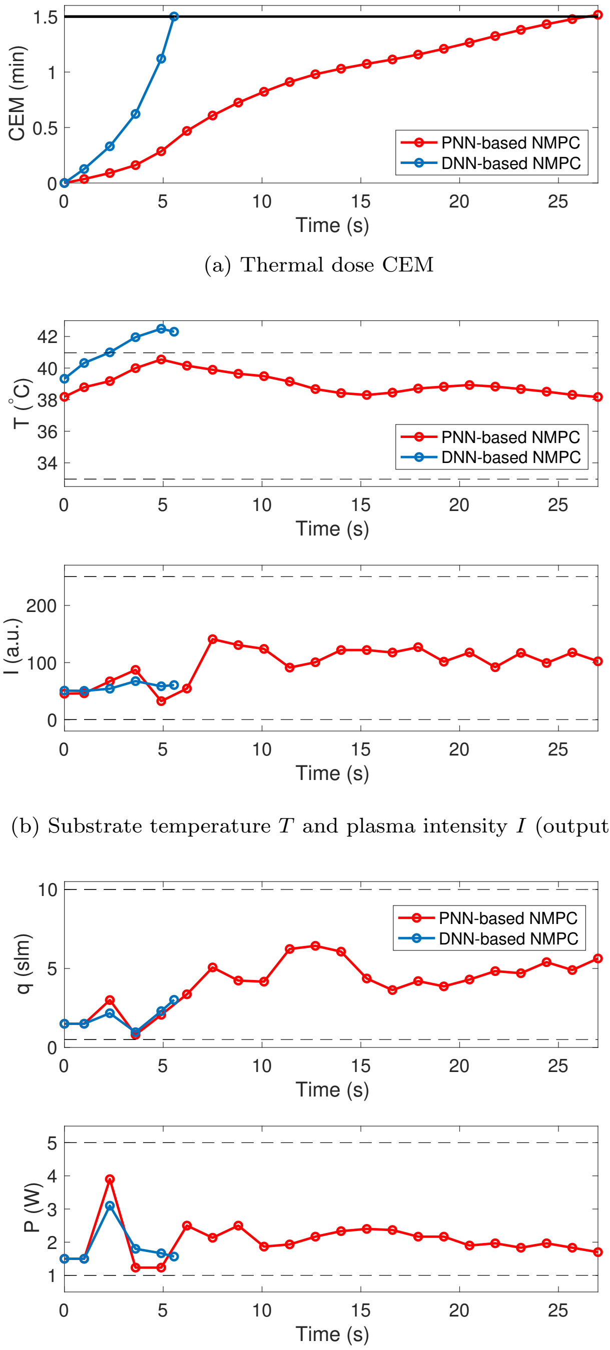
Fig. 5. Average online computation time of the NMPC, DNN-based NMPC, and PNN-based NMPC per iteration as a function of the prediction horizon.

the trained DNN-based control law provides an acceptable approximation of the implicit NMPC law. However, the temperature constraint is violated (see Fig. 4(b)). Such a constraint violation can be unacceptable, particularly in medical applications where cells can be sensitive to temperature differences of just a few degrees. In contrast, the PNN-based NMPC ensures constraint satisfaction by modifying the inputs resulting from the DNN such that they lie in a “safe” input set. As a result, the substrate temperature is guaranteed to stay within its constraints, as long as the uncertainty respects the assumed bounds. As expected, guaranteed constraint satisfaction is achieved at the expense of worse control performance in that the CEM reaches its setpoint slower, prolonging the treatment time by approximately 35%.

Fig. 5 shows the average online computation time of the three controllers. The computation time of NMPC scales with the prediction horizon, whereas it is almost independent of the prediction horizon for the both approximate NMPC controllers. The DNN-based approximation of the implicit NMPC law drastically reduces its online computation time to approximately 0.5 ms, without even optimizing our numerical implementations. The projection step slightly increases the computation time of the PNN-based NMPC to about 2 ms. The significant reduction in computation times of the DNN- and PNN-based NMPC relative to the NMPC will become even more pronounced as these approximate NMPC controllers are implemented on embedded systems in our future work.

### *4.3 Real-Time Control Experiments*

Fig. 6 shows the results of the real-time control of the APPJ using the DNN- and PNN-based NMPC. Experiments were repeated three times to ensure reproducibility and minimize the effects of intrinsic process noise; the average closed-loop profiles are shown in Fig. 6. The PNN-based NMPC exhibits loss of performance in thermal dose delivery (i.e., in terms of prolonged treatment time) due to the incorporation of the projection step (Fig. 6(a)). This is expected since the substrate temperature is not allowed to increase as quickly or to the same extent as in the DNN-based NMPC case. Since the thermal dose CEM has an exponential dependence on temperature (see (2)), even slight variations in temperature can cause drastic changes to the CEM. This is further corroborated by Fig. 6(b), where the blue temperature profile is consistently higher than the red one, causing the CEM to increase more rapidly and consequently reaching the setpoint faster. Nevertheless, the DNN-based NMPC suffers from the problem of temperature constraint violation, which may compromise the safety and comfort of a patient undergoing plasma treatment. As a result, when dealing with safety-critical biomedical applications, it is prudent to sacrifice some performance in order to guarantee safe system operation.



(a) Thermal dose CEM

(b) Substrate temperature $T$ and plasma intensity $I$ (outputs)

(c) Helium flowrate $q$ and applied power $P$ (inputs)

Fig. 6. Real-time control experiments. Experiments were repeated three times to ensure reproducibility of results in the face of the intrinsic stochasticity of plasma dynamics. Average profiles are shown here.

Since the plasma dynamics are intrinsically stochastic, even repeating the control experiments three times still results in small variations in the initial temperature between the experiments with the DNN- and PNN-based NMPC, as can be seen in Fig. 6(b). While starting from a slightly higher temperature puts the DNN-based NMPC temperature profile closer to the upper temperature constraint, it is evident that this difference is not what causes the constraint to be violated. A closer look at the CEM profiles (Fig. 6(a)) reveals that in the first few seconds the DNN-based NMPC drives the CEM closer to the setpoint than the PNN-based NMPC, while still respecting the temperature constraint. However, because the DNN-based NMPC is blind to the model uncertainty, it ends up violating the temperature constraint. Fig. 6(c) illustrates the input profiles to ensure that the observed closed-loop output and CEM profiles are not due to input saturation.

Interestingly, although the DNN-based NMPC exhibits smaller changes in control actions than the PNN-based NMPC, the corresponding CEM profile increases at a much faster rate. This is because the thermal dose CEM is exponential with respect to temperature, so regardless of the control actions, a higher temperature at each time instant will cause the CEM to grow more rapidly.

## 5. CONCLUSIONS

This paper presents a projected deep neural network-based approximate NMPC strategy for fast and safe predictive control of a plasma jet with prototypical safety-critical applications in plasma medicine. The fast approximate NMPC strategy can handle nonlinear cost functions while guaranteeing robust feasibility despite system uncertainties and the neural network approximation error. Real-time control experiments demonstrate that the proposed low-complexity controller guarantees robust satisfaction of the safety-critical system constraints at the expense of some loss in control performance. The approximate control law can be stored using only 2.5 kb of memory, and can be evaluated in under 2 ms in a non-optimized numerical setting. Future work will focus on implementation of the proposed low-memory and fast approximate NMPC strategy on a resource-limited embedded system using field-programmable gate arrays.

## REFERENCES

Alessio, A. and Bemporad, A. (2009). A survey on explicit model predictive control. In *Nonlinear Model Predictive Control*, 345–369. Springer, Berlin, Germany.

Andersson, J.A., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36.

Barron, A.R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3), 930–945.

Bemporad, A., Borrelli, F., and Morari, M. (2002). Model predictive control based on linear programming – the explicit solution. *IEEE Transactions on Automatic Control*, 47(12), 1974–1985.

Bonzanini, A.D., Paulson, J.A., and Mesbah, A. (2020). Safe learning-based model predictive control under state- and input-dependent uncertainty using scenario trees. In *Proceedings of the IEEE Conference on Decision and Control*. Jeju Island, Republic of Korea. Submitted.

Chakrabarty, A., Dinh, V., Corless, M.J., Rundell, A.E., Żak, S.H., and Buzzard, G.T. (2016). Support vector machine informed explicit nonlinear model predictive control using low-discrepancy sequences. *IEEE Transactions on Automatic Control*, 62(1), 135–148.

Chen, S., Saulnier, K., Atanasov, N., Lee, D.D., Kumar, V., Pappas, G.J., and Morari, M. (2018). Approximating explicit model predictive control using constrained neural networks. In *Proceedings of the American Control Conference*, 1520–1527. Milwaukee.

Csekő, L.H., Kvasnica, M., and Lantos, B. (2015). Explicit MPC-based RBF neural network controller design with discrete-time actual Kalman filter for semiactive suspension. *IEEE Transactions on Control Systems Technology*, 23(5), 1736–1753.

Fridman, G., Friedman, G., Gutsol, A., Shekhter, A.B., Vasilets, V.N., and Fridman, A. (2008). Applied plasma medicine. *Plasma Processes and Polymers*, 5, 503–533.

Gidon, D., Curtis, B., Paulson, J.A., Graves, D.B., and Mesbah, A. (2018). Model-based feedback control of a kHz-excited atmospheric pressure plasma jet. *IEEE Transactions on Radiation and Plasma Medical Sciences*, 2(2), 129–137.

Gidon, D., Graves, D.B., and Mesbah, A. (2017). Effective dose delivery in atmospheric pressure plasma jets for plasma medicine: a model predictive control approach. *Plasma Sources Science and Technology*, 26(8), 085005.

Gidon, D., Graves, D.B., and Mesbah, A. (2019). Predictive control of 2D spatial thermal dose delivery in atmospheric pressure plasma jets. *Plasma Sources Science and Technology*, 28(8), 085001.

Herceg, M., Kvasnica, M., Jones, C., and Morari, M. (2013). Multi-Parametric Toolbox 3.0. In *Proceedings of the European Control Conference*, 502–510. Zürich.

Jones, C.N. and Morari, M. (2010). Polytopic approximation of explicit model predictive controllers. *IEEE Transactions on Automatic Control*, 55(11), 2542–2553.

Karg, B. and Lucia, S. (2018). Efficient representation and approximation of model predictive control laws via deep learning. *arXiv preprint arXiv:1806.10644*.

Mirko, F. and Mazen, A. (2017). Computing control invariant sets is easy. *arXiv preprint arXiv:1708.04797*.

Parisini, T. and Zoppoli, R. (1995). A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31(10), 1443–1451.

Paulson, J.A. and Mesbah, A. (2018). Nonlinear model predictive control with explicit backoffs for stochastic systems under arbitrary uncertainty. *IFAC-PapersOnLine*, 51(20), 523–534.

Paulson, J.A. and Mesbah, A. (2020). Approximate closed-loop robust model predictive control with guaranteed stability and constraint satisfaction. *IEEE Control Systems Letters*, In Press.

Richter, S., Jones, C.N., and Morari, M. (2009). Real-time input-constrained MPC using fast gradient methods. In *Proceedings of the IEEE Conference on Decision and Control*, 7387–7393. Shanghai.

Sapareto, S.A. and Dewey, W.C. (1984). Thermal dose determination in cancer therapy. *International Journal of Radiation Oncology Biology Physics*, 10(6), 787–800.

Shin, J. and Raja, L.L. (2007). Run-to-run variations, asymmetric pulses, and long time-scale transient phenomena in dielectric-barrier atmospheric pressure glow discharges. *Journal of Physics D: Applied Physics*, 40(10), 3145–3154.

Summers, S., Jones, C.N., Lygeros, J., and Morari, M. (2011). A multiresolution approximation method for fast explicit model predictive control. *IEEE Transactions on Automatic Control*, 56(11), 2530–2541.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57.

Wang, Y. and Boyd, S. (2009). Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2), 267–278.