IEEE Control Systems Letters paper presented at
2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

Mixed-Integer MPC Strategies for Fueling and Density Control in Fusion Tokamaks

Christopher A. Orrico, Matthijs van Berkel, *Member, IEEE*, Thomas O. S. J. Bosman,
W. P. M. H. Heemels, *Fellow, IEEE*, Dinesh Krishnamoorthy, *Member, IEEE*

***Abstract*— Model predictive control (MPC) is promising for fueling and core density feedback control in nuclear fusion tokamaks, where the primary actuators, frozen hydrogen fuel pellets fired into the plasma, are discrete. Previous density feedback control approaches have only approximated pellet injection as a continuous input due to the complexity that it introduces. In this letter, we model plasma density and pellet injection as a hybrid system and propose two MPC strategies for density control: mixed-integer (MI) MPC using a conventional mixed-integer programming (MIP) solver and MPC utilizing our novel modification of the penalty term homotopy (PTH) algorithm. By relaxing the integer requirements, the PTH algorithm transforms the MIP problem into a series of continuous optimization problems, reducing computational complexity. Our novel modification to the PTH algorithm ensures that it can handle path constraints, making it viable for constrained hybrid MPC in general. Both strategies perform well with regards to reference tracking without violating path constraints and satisfy the computation time limit for real-time control of the pellet injection system. However, the computation time of the PTH-based MPC strategy consistently outpaces the conventional MI-MPC strategy.**

I. Introduction

Nuclear fusion is a prospective source of renewable energy that has made significant progress in recent years. Nuclear fusion power is produced when deuterium (D) and tritium (T), two hydrogen isotopes, fuse into helium. When maintained at sufficiently high density and temperature, a D-T plasma will achieve "ignition," producing net energy from fusion [1]. However, the plasma density and temperature necessary for ignition are extremely challenging to reach and maintain.

The fusion power produced by the D-T reaction scales with the square of the plasma electron density (n_e) [1]. Because the torus-shaped magnetic topology confining the plasma is difficult to penetrate with particles for high density core plasma fueling, large tokamaks, such as ITER, will need to use frozen hydrogen pellets fired at a velocity sufficient to penetrate into the plasma edge for fueling (cf. Fig. 1) [3]–[5]. The particle flux from fired pellets must maintain the desired fueling rate without exceeding the safety-critical edge density limit (cf. Fig. 2b) [6]. Violating this limit leads to a plasma disruption that can severely damage the tokamak. The control objective is to maximize the core density (and thus fusion power) without violating the edge density limit [3].

Plasma core density and temperature can be modeled by continuous nonlinear coupled drift-diffusion partial differential equations (PDEs) [7], [8], with hybrid system dynamics arising when the controller makes a binary decision to launch a pellet or not [9]. If the controller can choose from a set of pellet frequencies, sizes, or fuel compositions, this binary set extends to a larger combinatorial set of discrete actuation decisions. Moreover, the controller cannot exceed the maximum computation time, determined by the pellet injection system's maximum repetition rate [5]. Due to the complexity that pellet fueling introduces, literature on tokamak core fueling and density control has only integrated pellet fueling as a discrete feedforward signal [10]–[12] or as a continuous actuator for feedback control [13]–[15]. We seek to progress core density control by integerating pellet fueling as a discrete actuator.

Hybrid model predictive control (MPC) [9], [16] is a promising strategy for controlling systems with hybrid dynamics arising from discrete actuators [9]. Moreover, by

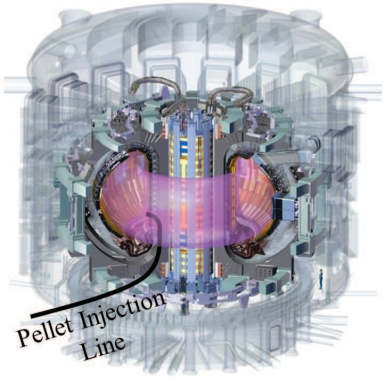

Fig. 1. The pellet injection line, which fires pellets into the plasma from inner side of the plasma torus, overlaid on a cutaway diagram of the ITER tokamak. Image courtesy of the ITER Organization, with modifications [2]

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No. 101052200—EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them. This publication is part of the project *Balls to the Wall* (project no. 19695) of the research programme NWO Talent Programme VIDI, financed in part by the Dutch Research Council (NWO). DIFFER is an institute of the NWO.

C. A. Orrico, M. van Berkel, T. O. S. J. Bosman, and D. Krishnamoorthy are with the Dept. of Mechanical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands and DIFFER - Dutch Institute for Fundamental Energy Research, 5612 AJ Eindhoven, The Netherlands (e-mail: c.a.orrico@tue.nl, m.vanberkel@differ.nl, t.o.s.j.bosman@differ.nl, d.krishnamoorthy@tue.nl).

W. P. M. H. Heemels is with the Dept. of Mechanical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands (e-mail: w.p.m.h.heemels@tue.nl).

using MPC, we explicitly prevent the controller from violating safety-critical path constraints [14], [15]. Given a model of system dynamics, the MPC controller solves an optimal control program (OCP) subject to state and input constraints over a finite prediction horizon [17]. If states or inputs must take integer values, the OCP becomes a mixed-integer programming (MIP) problem that may be $\mathcal{NP}$-hard and, consequently, intractable for real-time control [18]. In this letter, we implement hybrid MPC as a strategy for the pellet fueling and density control problem in nuclear fusion tokamaks in a manner suitable for real-time control.

To this end, we model particle transport in a fusion plasma with discrete fuel pellet injection as a hybrid dynamical system. We then propose two real-time MI-MPC setups to solve the density control problem. The conventional strategy for solving an MI-MPC problem is to use a mixed-integer solver. Here, we compute a mixed-integer quadratic programming (MIQP) problem at each discrete-time step using the branch and bound (B&B) algorithm of the `Gurobi` solver. Next, we develop a novel version of the penalty term homotopy (PTH) algorithm and apply it to the MI-MPC problem, such that it is solvable using a quadratic programming (QP) solver (such as `qpOASES`), rather than an MIP solver. The PTH algorithm [19] transforms a MI-MPC problem into a series of continuous optimization problems by relaxing the integer requirements, instead penalizing non-integer solutions in the objective function [19]. However, as originally designed [19], the PTH algorithm may not be able to satisfy both integer requirements and path constraints. By adding a logarithmic barrier term, we modify the PTH algorithm in a novel way to be viable for path-constrained hybrid MPC problems. We then compare the performance of the two strategies in terms of their reference tracking error and computation time.

II. Problem Formulation

During a tokamak plasma discharge, we plan to maintain high density in the core of the plasma using hydrogen fuel pellets [5]. In Section II-A, we model the tokamak plasma core temperature and density using a system of coupled 1D drift-diffusion equations, integrating discrete pellet actuation as a particle source. We then translate this model into a form appropriate for control. In Section II-B, we formulate the control problem, introducing key parameters and constraints.

A. Drift-Diffusion Model

In MPC, the model of the system dynamics is crucial for predicting the state trajectory. Fusion plasma transport dynamics across nested flux surfaces (cf. Fig. 2a) are modeled as a system of continuous coupled nonlinear PDEs [1]. As computing the evolution of these equations in 3D space is impractical for real-time control, we approximate transport in a toroidal plasma with a simplified 1D drift-diffusion model by adopting several realistic assumptions. First, we assume that the plasma particles (electrons and ions) travel much faster along the nested flux surfaces in Fig. 2a than they travel perpendicular to the flux surfaces [1]. Consequently, the density and temperature are constant along the flux surfaces,

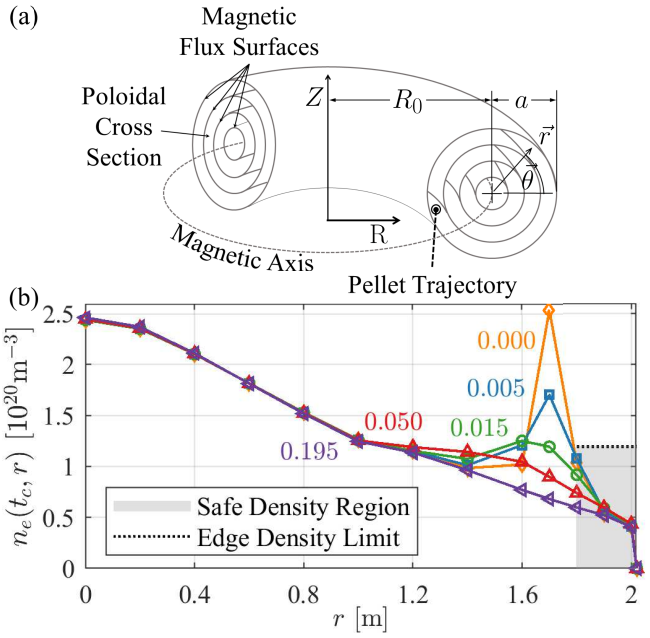

Fig. 2. (a) Diagram showing nested flux surfaces of the cylindrical tokamak plasma approximation, where a fuel pellet fired along the pellet injection line in Fig. 1 will ablate near the plasma edge. Here, R_0 is the major radius, a is the minor radius, and θ is the angle from the midplane. (b) The 1D plasma response to a fuel pellet ablating at $r = 1.7$ m, for $t_c = \{0.000, 0.005, 0.015, 0.050, 0.195\}$ s after the pellet penetrates the plasma.

yielding a 1D model of the density and temperature gradients normal to said surfaces. Next, we approximate the plasma as a cylindrical torus, further simplifying the density and temperature gradients as functions of the radius r (cf. Fig. 2a). The 1D coupled nonlinear PDEs between the plasma density and temperature are written as

$$\frac{\partial n_e}{\partial t_c} = S_{n_e}(r,t_c) + \frac{1}{r}\left\{\frac{\partial}{\partial r} r D_n \left[\frac{\partial n_e}{\partial r} + \frac{n_e}{T_e}\frac{\partial T_e}{\partial r} + \right.\right. \\ \left.\left. + \frac{D_B}{4\eta_\parallel}\frac{B_p}{T_e r}\left(B_p + r\frac{\partial B_p}{\partial r}\right)\right] + v\frac{\partial}{\partial r}(r n_e)\right\}, \tag{1a}$$

$$\frac{\partial T_e}{\partial t_c} = \frac{1}{3n_e}\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r n_e \chi \frac{\partial T_e}{\partial r}\right) + S_{T_e}(r,t_c)\right], \tag{1b}$$

where n_e is the electron density, T_e is the electron temperature, B_p is the poloidal magnetic field, r is the radius from the plasma core, t_c is continuous time, $\eta_\parallel$ is the Spitzer resistivity, and the magnetic diffusion coefficient is $D_B = \eta_\parallel/\mu_0$ [1], [7]. The device parameters (including minor radius a) are set to those of the ITER tokamak [20]. The particle diffusion, the particle drift, and the thermal diffusion coefficients are D_n, v, and χ, respectively [21], [22]. Here, a fired fuel pellet is a density source $S_{n_e}(r,t_c)$ at the radius of deposition over the time period of ablation $\mathcal{O}(1)$ ms [4]. $S_{T_e}(r,t_c)$ is the plasma heat source from fusion reactions and auxiliary heating, which we assume to be constant in r and t_c. For a detailed discussion on modeling fusion plasma transport, the reader is referred to [1]. While

(1) simplifies fusion plasma dynamics, it is straightforward to analyze. Consequently, (1) is ideal as a proof-of-concept model against which to test control strategies.

Discretizing (1) with respect to r, we construct a piecewise smooth ODE for n_e and T_e using the first-order finite difference method. Boundary conditions are set as $n_e|_{r=a+\lambda} = 0$, $T_e|_{r=a+\lambda} = 0$, $\nabla n_e|_{r=0} = 0$, and $\nabla T_e|_{r=0} = 0$. Here the edge is a perfect sink at the virtual domain node $r = a + \lambda$ [7], where the minor radius $a = 2$ m and λ is the "scrape-off layer" width [20].

To be consistent with control conventions, we translate (1) and nuclear fusion specific terminology to a state-space model, constraints, and a state reference. We begin by defining the continuous time (t_c) state space description of the dynamics as

$$\dot{x}(t_c) = f(x(t_c), u(t_c)), \tag{2a}$$

$$x(t_c) = [n_e(t_c, r_1)\ n_e(t_c, r_2)\ \dots\ n_e(t_c, r_{n/2}),\ T_e(t_c, r_1)\ T_e(t_c, r_2)\ \dots\ T_e(t_c, r_{n/2})]^\top, \tag{2b}$$

$$u(t_c) = [u_1(t_c)\ u_2(t_c)\ \dots\ u_m(t_c)]^\top. \tag{2c}$$

The nonlinear, piecewise smooth ODE discretized in r from (1) is given as the vector field $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, and is a function of the state vector $x(t_c) \in \mathbb{R}^n$ for $n/2$ discrete radial nodes $[r_1, r_2, \dots, r_{n/2}]$ and input vector $u \in \mathbb{R}^m$ for m pellet launchers. We take $n = 26$ corresponding to 13 radial nodes $r = \{0, 0.2, 0.4, ..., 1.6, 1.7, ..., 2.0\}$ m. For simplicity, we take $m = 1$ for a single pellet launcher depositing pellets at $r = 1.7$ m (cf. Fig. 2b). The pellets have a fixed size of 6×10^{21} D atoms, which is the fuel pellet size planned for ITER [5].

To simulate the plasma behavior during the "flat-top" of an ITER 15 MA plasma discharge [20], we discretize the hybrid system dynamics (2) in time using the Runge-Kutta 4th order (RK4) method with a sampling time of 5 ms, which is sufficiently fast to ensure numerical stability and is consistent with the rate of pellet ablation in the plasma [4], [23]. The controller supplies an input $u(t)$, computed by the controllers designed in Section III, in feedback to the simulation.

B. Control Problem Description

The objective of pellet fueling and density control is to track a reference density without violating safety-critical path constraints. For instance, because fusion power scales $\propto n_e^2$, the reference may seek to maximize the core density. In this section, we outline the general control problem to be solved in Section III.

For the MPC prediction model, which is distinct from the non-linear plasma dynamics simulation above, we take the Jacobian linearization of (2) around a desired local equilibrium point [17], [23]. We compute the exact discretization of the Jacobian linearization using the zero-order hold (ZOH) method [23], selecting $T_{s,\mathrm{ZOH}} = 5$ ms both for consistency with the simulated system RK4 time discretization and that a fired pellet ablating for 5 ms results in a constant density source over $T_{s,\mathrm{ZOH}}$. Thus, the linearization approximates (1) as a linear time-invariant (LTI) state space model with state and input matrices A and B in discrete time t.

Finally, the maximum repetition rate of the pellet injection system must be built into the prediction model. We base our model off of the ITER tokamak pellet injection system, which has 2 pellet injection lines operating with a maximum repetition rate of 4 Hz per injector [5]. As the system requires a control decision every ≈ 100 ms, we set the control sampling time $T_s = 100$ ms and define discrete time $t \in \mathbb{N}$ with respect to continuous time as $t_c = tT_s$. The prediction model now must include the first 5 ms of zero-order hold actuation and subsequent 95 ms of state evolution with $u_{k|t} = 0$ to account for the time between decisions. Accordingly, we extend the LTI state prediction model as

$$x(t+1) = \bar{A}x(t) + \bar{B}u(t), \tag{3}$$

where $\bar{A} = A^\tau$ and $\bar{B} = A^{\tau-1}B$, defining $\tau = T_s/T_{s,\mathrm{ZOH}} \in \mathbb{N}$, such that the prediction model now incorporates one control decision every $T_s = 100$ ms [17].

The edge density limit of a fusion plasma, given as $n_{Gw} = I_p/(\pi a^2)$ (where I_p [MA] is the total plasma current) [6], places hard path constraints on the controller (cf. Fig 2b) and is central to the controller design. We construct an element-wise inequality with matrix G and vector h that MPC stricly enforces $n_e(t, r = 1.8\ \mathrm{m}) \le n_{Gw}$ as

$$Gx - h \le 0. \tag{4}$$

For reference tracking, the line-averaged electron density $\bar{n}_e(t)$ [24] is the key diagnostic measurement, calculated as

$$\bar{n}_e = \frac{1}{a}\int_0^a n_e dr\,. \tag{5}$$

We set the state reference $x^r(t) = \bar{n}_e^r(t)$ for a desired core density [25]. The control objective is now to design a controller capable of tracking $x^r(t)$ without violating constraints (4) in real-time (given as the computation time limit T_s).

III. Model Predictive Control Strategies

To solve the problem formulated in the previous section, we propose to use MPC. This method solves a receding horizon optimization problem based on the measured state online and applies the first control action to the plant, repeating this process for each discrete-time step [17]. The optimization problem seeks to minimize an objective function with respect to the optimization variable $u_{k|t}$. The objective function (see (6a), (7a) below) consists of stage and terminal costs, which are functions of the current state $x(t)$, the inputs $u_{k|t}$, and the predicted state trajectory $x_{k|t}$ for $k = \{0, 1, \dots, N\}$ discrete-time steps after t. Solving the resulting receding horizon optimization problem yields the optimal $u^*_{k|t}$.

A. Mixed-Integer MPC

MI-MPC is a special case of MPC, where the set of admissible states or inputs must take integer values [18]. In the case of pellet injection control, we enforce a binary

integer constraint (6d) on all predicted inputs $u_{k|t}$. At $k|t$, no pellet is fired if $u_{k|t}=0$, whereas a pellet is fired if $u_{k|t}=1$. The MI-MPC problem is defined at t for the current state $x(t)$ as

$$\min_{u_{k|t}} \|(x_{N|t}-x^r(t))\|_P^2 + \sum_{k=0}^{N-1} \|(x_{k|t}-x^r(t))\|_Q^2 + \|u_{k|t}\|_R^2 \tag{6a}$$

s.t.

$$x_{k+1|t} = \bar{A}x(t) + \bar{B}u(t), \quad k \in \{0,1,\dots,N-1\} \tag{6b}$$
$$Gx_{k|t} - h \le 0, \quad k \in \{0,1,\dots,N\} \tag{6c}$$
$$u_{k|t} \in \{0,1\}, \quad k \in \{0,1,\dots,N-1\} \tag{6d}$$
$$x_{0|t} = x(t). \tag{6e}$$

Above, the norm $\|a\|_B^2 := a^\top Ba$ yields the convex objective function (6a), wherein $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{m\times m}$, and $P \in \mathbb{R}^{n\times n}$ are diagonal tuning matrices that penalize the state error, the input cost, and the terminal state error, respectively. Q and P scale the density state to 1 and are weighted to account for non-uniform radial discretization, while the temperature states are weighted to 0. Lastly, we let $R=0$ as we are not concerned with penalizing control actuation. We implement the MI-MPC problem using `CasADi` and the `Yalmip` toolbox [26], [27]. We warm-start $u_{k|t}$ with $u^*_{k|t-1}$. The B&B algorithm (computed online with the MIQP solver `Gurobi`) searches a tree of possible candidates for $u^*_{k|t}$ to find the optimal control sequence [28]. For further details on the B&B algorithm, we direct the reader to [28].

B. Penalty Term Homotopy MPC

When the B&B algorithm solves (6) for the optimal control sequence, the size of the search tree (and thus the number of QPs the MIQP solver computes) increases exponentially with N. Sager [19] proposes the PTH algorithm as a means to compute the optimal solution to a MIQP problem using a QP solver rather than a MIQP solver. By relaxing (6d) to $u_{k|t} \in [0,1]$, (6a) transforms into a continuous function of $u_{k|t}$ in (7a). To implicitly enforce integer requirements, the concave penalty term $u_{k|t}\beta_j(1-u_{k|t})$ in (7a) replaces the binary constraint on $u_{k|t}$ in (6d). Each iteration of Algorithm 1 solves (7), resulting in a series of QPs. Consequently, the computation time scales $\mathcal{O}(2N)$ rather than $\mathcal{O}(2^N)$ (note that the value 2 here indicates a binary decision variable).

Algorithm 1 Modified Penalty Term Homotopy

1: Set $j \leftarrow 0$, $\gamma_j \leftarrow 0$, and $\beta_j \leftarrow 0$
2: **while** $\left(u_{k|t} > \varepsilon^\dagger \text{ **or** } 1-u_{k|t} > \varepsilon^\dagger\right)\ \forall k \in \{0,\dots,N-1\}$ … **and** $j < j_{\max}{}^\dagger$ **do:**
3: Solve the relaxed optimization problem (7)
4: Set $j \leftarrow j+1$
5: Set $\beta_j \leftarrow \beta_{\text{init}}\beta_{\text{inc}}^{j-1}$, $\gamma_j \leftarrow \gamma_{\text{init}}\gamma_{\text{inc}}^{j-1}{}^\ddagger$
6: **end while**

$^\dagger\varepsilon$, a user-defined tolerance, is sufficiently small such that the rounding error on $u_{k|t}$ will not result in violation of state constraints and j_{max} guarantees that the algorithm is finite. If $j = j_{max}$, Algorithm 1 terminates and sets $u_{k|t} \leftarrow 0$. Sager provides additional stopping criteria to ensure sufficiently refined control discretization [19], which we omit here (see Section III-C).

‡Setting $\gamma_{\text{init}} = 0$ results in the unmodified PTH algorithm.

At discrete-time step t, `qpOASES` [29] solves the warm-started QP for each iteration j of Algorithm 1,

$$\begin{aligned}\min_{u_{k|t}} \|(x_{N|t}-x^r(t))\|_P^2 \ - \ \gamma_j \ln\left(h - Gx_{N|t}\right) \ + \\ + \sum_{k=0}^{N-1}\Big[\|(x_{k|t}-x^r(t))\|_Q^2 \ + \ \|u_{k|t}\|_R^2 \ + \\ + \ u_{k|t}\beta_j(1-u_{k|t})^\top - \ \gamma_j \ln\left(h - Gx_{k|t}\right)\Big],\end{aligned} \tag{7a}$$

s.t.

$$x_{k+1|t} = \bar{A}x(t) + \bar{B}u(t), \quad k \in \{0,1,\dots,N-1\} \tag{7b}$$
$$Gx_{k|t} - h \le 0, \quad k \in \{0,1,\dots,N\} \tag{7c}$$
$$u_{k|t} \in [0,1], \quad k \in \{0,1,\dots,N-1\} \tag{7d}$$
$$x_{0|t} = x(t), \tag{7e}$$

in which $\beta_j = \beta_{\text{init}}\beta_{\text{inc}}^{j-1}$ and increases for each j until $u_{k|t} \in \{0,1\}$ for all $k \in \{0,1,\dots,N-1\}$ (or another stopping criterion is met) [19]. We tune β_{init} and β_{inc} for performance, starting with the initial values that [19] proposes of $\beta_{init} = 1\times10^{-4}$ and $\beta_{inc} = 2$ for systems with variables scaled to 1. Increasing β_{init} and β_{inc} reduces the average algorithm iterations at each time step at the cost of reference-tracking performance. Note that in the original PTH algorithm, $\gamma_j = 0$. The logarithmic barrier term $-\gamma_j \ln(h - Gx_{k|t})$ is a novel addition to the PTH algorithm and will be addressed in Section III-C.

C. Penalty Term Homotopy MPC with Constraints

In its original form [19], the PTH algorithm is not guaranteed to produce feasible solutions for systems with path constraints. By relaxing the integer requirement, the optimal solution $u^*_{k|t}$, obtained by minimizing (7a) in the presence of path constraints (7c), may not converge to the same solution as the antecedent MIQP (6). When the descent direction of (7a) with the concave $u_{k|t}\beta_j(1-u_{k|t})$ is towards path constraints, (7) may get "stuck" on an active path constraint with $u^*_{k|t} \notin \{0,1\}$, even as $\beta_j \to \infty$. Table I demonstrates this behavior for the control trajectory in Fig. 3.

To resolve this issue, we add the logarithmic barrier term $-\gamma_j \ln(h - Gx_{k|t})$ to (7a) (with $\gamma_j = \gamma_{\text{init}}\gamma_{\text{inc}}^{j-1}$ defined in Algorithm 1). As γ_j increases, the logarithmic barrier

TABLE I
PTH-MPC STICKING BEHAVIOR

j	β_j	$u^*_{0\|0}$	$u^*_{1\|0}$	$u^*_{2\|0}$	$u^*_{3\|0}$	$u^*_{4\|0}$
0	0	0.9364	0.5382	0	0.2498	0.3086
1	0.1	0.9364	0.9239	0	0	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
$j_{\max}$	$\beta_{j_{\max}}$	0.9364	0.9239	0	0	0

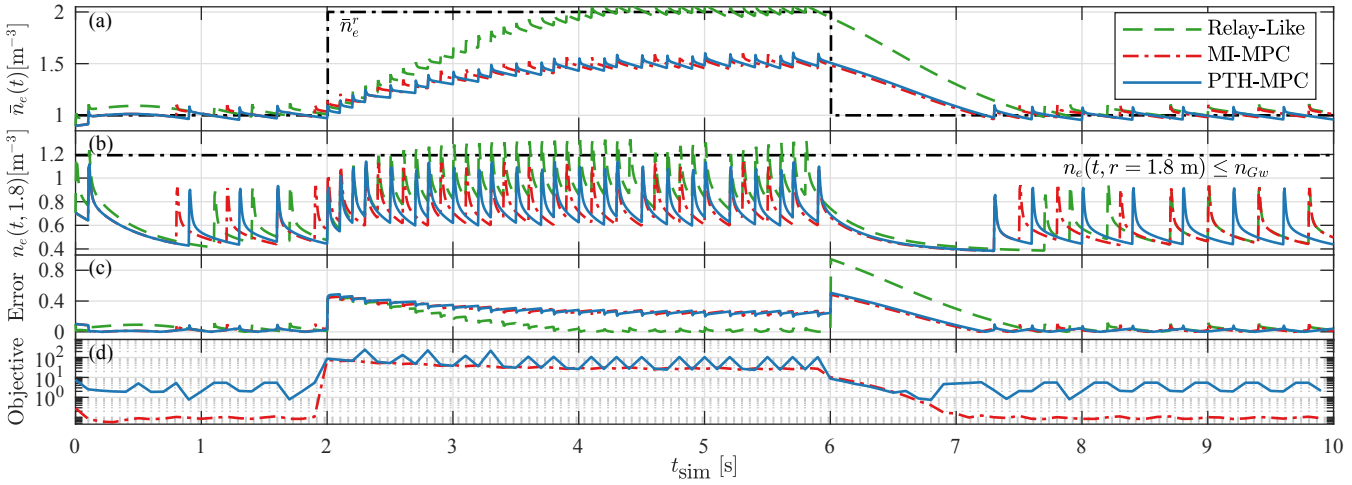

Fig. 3. (a) The line-averaged density $\bar{n}_e(t)[\mathrm{m}^{-3}]$ (calculated from $x(t)$ using (5)) for $N = 5$. The grey dashed line indicates $x^r(t) = \bar{n}_e^r(t)$. (b) The edge density $n_e(t, r = 1.8\ \mathrm{m})[\mathrm{m}^{-3}]$ with the density limit (4) indicated as a black dashed line. Each pellet actuation is visible as a jump in density. (c) The reference tracking error of $\bar{n}_e(t)$ relative to $\bar{n}_e^r(t)$. (d) The value of the objective function of each MPC strategy.

term drives the QP solver solution for $u^*_{k|t}$ away from the path constraints and towards the MIQP solution. To the best of our knowledge, this modification of the PTH algorithm, which makes it broadly viable for path-constrained MI-MPC problems, is novel. For the various N values in Section IV, we tune γ_j relative to β_j as $\beta_{init} = 0.5/N$, $\beta_{inc} = 3$, and $\gamma_j = \beta_{j+1}$ such that $j < j_{max} \forall t$.

IV. Results

The MPC strategies are evaluated for real-time performance with two metrics: the reference tracking error and the computation time. The controllers, tested with prediction horizons $N \in \{2, 5, 10, 20\}$ in a feedback loop, are also compared to a relay-like control law: $\{u(t) \leftarrow 1$ **if** $\bar{n}_e(t) < \bar{n}_e^r(t)$**, else** $u(t) \leftarrow 0\}$. The RK4 discretization of (1) simulates the nonlinear system dynamics over a period $t_{\mathrm{sim}} = [0, 10]$ s.

A. Reference Tracking

The state reference $x^r(t)$ is initially set at $\bar{n}_e(t) = 10^{20}$ m^{-3}, corresponding to the "flat-top" of a D-T H-mode 15 MA ITER plasma [20]. At $t_{\mathrm{sim}} = 2$ s, $x^r(t)$ steps up to $\bar{n}_e(t) = 2 \times 10^{20} \mathrm{m}^{-3}$. This reference is unreachable due to the edge density limit, forcing the MPC strategies to demonstrate their ability to drive the state towards an optimal trajectory without violating safety-critical path constraints, which is desirable for exploring maximum achievable plasma density [13].

TABLE II
Mean Reference Tracking Error

N	Reachable State % Error		Unreachable State % Error	
	MI-MPC	PTH-MPC	MI-MPC	PTH-MPC
2	2.34	2.13	29.4	30.2
5	2.34	2.09	29.4	29.7
10	2.34	2.12	29.4	30.3
20	2.34	2.11	29.4	29.8

The arithmetic mean error between $\bar{n}_e(t)$ relative to $\bar{n}_e^r(t)$ is calculated over $t_{\mathrm{sim}} = 0, \ldots, 2, 6, \ldots, 10$ s and $t_{\mathrm{sim}} = 2, \ldots, 6$ s for the reachable $\bar{n}_e^r(t)$ and unreachable $\bar{n}_e^r(t)$, respectively.

Fig. 3 compares the state trajectories for a prediction horizon of $N = 5$. Both the MPC strategies show good reference tracking when the state reference is reachable (cf. Fig. 3a) and, unlike the relay-like controller, do not violate path constraints (cf. Fig. 3b). The two MPC strategies compute slightly different control policies for the same prediction horizon (cf. Fig. 3a). This is a consequence of the differing objective costs (cf. Fig. 3d) that arise from the addition of the penalty terms in the PTH-MPC formulation. The controller performance in Fig. 3 is consistent across all tested values of N. Mean tracking error in Table II is nearly equal for both control approaches. Furthermore, increasing N has little impact on mean tracking error as the plasma transport is so rapid that the density profile settles within ≈ 200 ms of pellet injection (cf. Fig. 2b). Consequently, a prediction horizon as short as $N = 2$ can largely capture predicted plasma response to the next control decision.

B. Computation Time

For the purposes of real-time feedback control, we require the computation time of the controllers to not exceed 100 ms [5]. Table III lists the maximum computation times during the simulation in Fig. 3. For all N tested, the PTH-MPC control approach outperformed the MI-MPC method. The exponential growth of the MI-MPC computation time for increasing N is evident for $N = 20$, jumping an order of magnitude over the computation time of the PTH-MPC solution for the same N. For short N, the PTH-MPC method is sufficiently fast to not only accommodate the ITER pellet injection system, but also the up to 80 Hz repetition rates of the faster centrifugal pellet launching systems [30].

Across t_{sim}, the MI-MPC computation time remains relatively constant for all $x^r(t)$. However, when tracking an unreachable state reference, the PTH-MPC algorithm computation times fluctuate widely. This is due to Algorithm 1

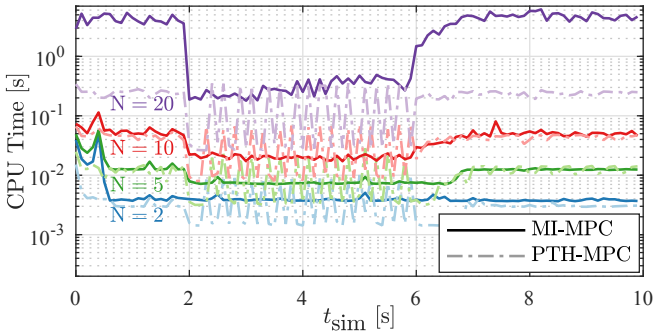

Fig. 4. Computation (CPU) time for the trajectory in Fig. 3a). The solid and dashed lines indicate the MI-MPC and the PTH-MPC controllers, respectively. The N labels identify the two lines directly above them. Simulations were performed on an Intel(R) Core™ i5 processor running at 2.40 GHz with 15.7 GB of useable RAM.

TABLE III
MAXIMUM COMPUTATION TIME

	N	2	5	10	20
CPU time [s]	MI-MPC	0.051	0.057	0.114	6.171
	PTH-MPC	0.015	0.028	0.071	0.351

requiring several iterations for the logarithmic barrier term to drive the calculated $u^*_{k|t}$ towards binary values. Increasing γ and β reduces computation time, but assessing the impact this has on controller performance requires further study.

V. CONCLUSION

In this investigation, we demonstrated that MI-MPC and PTH-MPC are promising candidates for core fueling and density control in fusion tokamaks. Future work will evaluate and improve MPC prediction model accuracy against an experimentally validated transport model [7], [14]. For pellet injection control in ITER, the control decision set will need to be extended to include multiple launchers, variable fuel composition, and actuation uncertainty. We will also investigate further methods to increase controller efficiency to cope with the extended combinatorial optimization problem.

REFERENCES

[1] J. P. Freidberg, *Plasma Physics and Fusion Energy*. Cambridge, UK: Cambridge University Press, 2007.

[2] (2020, August) Central solenoid. ITER Organization. [Online]. Available: http://www.ITER.org/

[3] L. R. Baylor, *et al.*, "Pellet fuelling and control of burning plasmas in ITER," *Nucl. fusion*, vol. 47, no. 5, p. 443, 2007.

[4] B. Pégourié, *et al.*, "Recent results on the fuelling and control of plasmas by pellet injection, application to ITER," *Plasma Phys. and Control. Fusion*, vol. 51, no. 12, p. 124023, 2009.

[5] L. R. Baylor, *et al.*, "Pellet injection technology and its applications on ITER," *IEEE Trans. on Plasma Sci.*, vol. 44, no. 9, pp. 1489–1495, 2016.

[6] M. Greenwald, *et al.*, "A new look at density limits in tokamaks," *Nucl. Fusion*, vol. 28, no. 12, p. 2199, 1988.

[7] T. Blanken, *et al.*, "Control-oriented modeling of the plasma particle density in tokamaks and application to real-time density profile reconstruction," *Fusion Eng. Des.*, vol. 126, pp. 87–103, 2018.

[8] M. G. Sevillano, I. Garrido, and A. J. Garrido, "Sliding-mode loop voltage control using ASTRA-Matlab integration in Tokamak reactors," *Int. J. Innov. Comput. Inf. Control.*, vol. 8, no. 9, pp. 6473–6489, 2012.

[9] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[10] L. Garzotti, *et al.*, "Simulations of density profiles, pellet fuelling and density control in ITER," *Nucl. Fusion*, vol. 52, no. 1, p. 013002, 2011.

[11] P. Vincenzi, *et al.*, "Fuelling and density control for DEMO," *Nucl. Fusion*, vol. 55, no. 11, p. 113028, 2015.

[12] T. Ravensbergen, *et al.*, "Density control in ITER: an ITERative learning control and robust control approach," *Nucl. Fusion*, vol. 58, no. 1, p. 016048, 2017.

[13] P. Lang, *et al.*, "Feedback controlled, reactor relevant, high-density, high-confinement scenarios at ASDEX Upgrade," *Nucl. Fusion*, vol. 58, no. 3, p. 036001, 2018.

[14] T. Bosman, M. van Berkel, and M. R. de Baar, "Model-based electron density profile estimation and control, applied to ITER," *J. Phys. Commun.*, vol. 5, no. 11, p. 115015, 2021.

[15] T. Bosman, M. van Berkel, and M. de Baar, "Constrained model-predictive control of the electron density profile in ITER using two pellet injectors," in *2022 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2022, pp. 812–818.

[16] M. Lazar, *et al.*, "Stabilizing model predictive control of hybrid systems," *IEEE Trans. on Automat. Contr.*, vol. 51, no. 11, pp. 1813–1818, 2006.

[17] J. Rawlings, D. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Santa Barbara: Nob Hill Publishing, 2017, pp. 5–172.

[18] C. Kirches, "Fast Numerical Methods for Mixed-Integer Nonlinear Model-Predictive Control," Ph.D. dissertation, Universität Heidelberg, Heidelberg, Germany, 2011.

[19] S. Sager, "Numerical methods for mixed-integer optimal control problems," Ph.D. dissertation, Interdisciplinary Center for Scientific Computing, Universität Heidelberg, Heidelberg, Germany, 2006.

[20] L. Garzotti, *et al.*, "Integrated core-SOL modelling of fuelling, density control and divertor heat loads for the flat-top phase of the ITER H-mode DT plasma scenarios," *Nucl. Fusion*, vol. 59, no. 2, p. 026006, 2018.

[21] Y.-S. Na, *et al.*, "On benchmarking of simulations of particle transport in ITER," *Nucl. Fusion*, vol. 59, no. 7, p. 076026, 2019.

[22] A. R. Polevoi, *et al.*, "ITER confinement and stability modelling," *J. Plasma Fusion Res.*, vol. 5, pp. 82–87, 2002.

[23] G. Franklin, J. Powell, and M. Workman, *Digital Control of Dynamic Systems*, 3rd ed. Menlo Park, USA: Addison-Wesley, 1997.

[24] M. A. Heald and C. B. Wharton, *Plasma Diagnostics with Microwaves*. Huntington, USA: R.E. Krieger Pub. Co., 1965, pp. 120–121.

[25] T. Bosman, *et al.*, "Kalman filter density reconstruction in ICRH discharges on ASDEX Upgrade," *Fusion Eng. and Des.*, vol. 170, p. 112510, 2021.

[26] J. A. Andersson, *et al.*, "CasADi – A software framework for nonlinear optimization and optimal control," *Math. Program. Comput.*, vol. 11, no. 1, pp. 1–36, 2019.

[27] J. Löfberg, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[28] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com

[29] H. J. Ferreau, *et al.*, "qpOASES: A parametric active-set algorithm for quadratic programming," *Math. Program. Comput.*, vol. 6, pp. 327–363, 2014.

[30] P. T. Lang, *et al.*, "Final design of the JT-60SA pellet launching system for simultaneous density and ELM control," *Fusion Eng. Des.*, vol. 146, pp. 91–95, 2019.