

Machine learning-enhanced model-based scenario optimization for DIII-D

S. Morosohk^{1,*} , B. Leard² , T. Rafiq²  and E. Schuster²

¹ Oak Ridge Associated Universities, San Diego, CA, United States of America

² Lehigh University, Bethlehem, PA 18015, United States of America

E-mail: morosohks@fusion.gat.com

Received 2 January 2024, revised 20 February 2024

Accepted for publication 20 March 2024

Published 28 March 2024



CrossMark

Abstract

Scenario development in tokamaks is an open area of investigation that can be approached in a variety of different ways. Experimental trial and error has been the traditional method, but this required a massive amount of experimental time and resources. As high fidelity predictive models have become available, offline development and testing of proposed scenarios has become an option to reduce the required experimental resources. The use of predictive models also offers the possibility of using a numerical optimization process to find the controllable inputs that most closely achieve the desired plasma state. However, this type of optimization can require as many as hundreds or thousands of predictive simulation cases to converge to a solution; many of the commonly used high fidelity models have high computational burdens, so it is only reasonable to run a handful of predictive simulations. In order to make use of numerical optimization approaches, a compromise needs to be found between model fidelity and computational burden. This compromise can be achieved using neural networks surrogates of high fidelity models that retain nearly the same level of accuracy as the models they are trained to replicate while reducing the computation time by orders of magnitude. In this work, a model-based numerical optimization tool for scenario development is described. The predictive model used by the optimizer includes neural network surrogate models integrated into the fast Control-Oriented Transport simulation framework. This optimization scheme is able to converge to the optimal values of the controllable inputs that produce the target plasma scenario by running thousands of predictive simulations in under an hour without sacrificing too much prediction accuracy.

Keywords: tokamak control, scenario optimization, neural network surrogate model

(Some figures may appear in colour only in the online journal)

* Author to whom any correspondence should be addressed.



Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

1. Introduction

Both feedforward and feedback approaches to plasma control are routinely used in tokamaks: feedback control, calculated in real time based on the measured plasma state, is always necessary for applications such as shape control and vertical stabilization [1]; feedforward control, determined ahead of time and not adjusted once the plasma shot starts, is often used with actuators such as the auxiliary heating systems to try to achieve a specific plasma state. This feedforward approach requires a significant amount of effort to determine what actuator trajectories are needed to achieve the desired state. This can, and historically has, been done using experimental trial and error, but the limited availability and high cost of experimental run time on a tokamak motivates the need for other solutions. High fidelity integrated predictive simulations (e.g. the physics-oriented transport code TRANSP [2, 3]) can be used to predict the results of a set of proposed actuator trajectories to determine whether scenario goals are obtained. There has been a recent push to use these type of tools in a predict-first approach [4–6], or to tune the feedforward actuator requests through a series of predictive simulations in order to minimize the experimental resources necessary to achieve the desired plasma. Going a step further, it would be very desirable to have the capability to have an optimization tool predict ideal actuator trajectories according to the model instead of requiring a physicist to do a manual tuning. Unfortunately, this type of automatic optimization can require hundreds or even thousands of predictions using different perturbations of the actuator trajectories; high fidelity predictive models such as predictive TRANSP may take hours to tens of hours to run a single predictive simulation, making it too computationally-intensive to work well with this type of optimizer.

In order to develop this type of optimization algorithm, a computationally lighter predictive model is needed. Control-oriented predictive codes such as RAPTOR [7] or the Matlab-based transport code Control-Oriented Transport Simulator (COTSIM) [8] have significantly lower calculation times that make them reasonable options for a numerical optimization scheme. This type of offline numerical actuator trajectory optimization has previously been conducted for NSTX-U [9, 10], EAST [11], DIII-D [12–14], ITER [15, 16], ASDEX-U and TCV [17]. However, optimization using a predictive model has the caveat that the optimized actuator trajectories are only as good as the model used to generate them. Any simplifications used to improve the calculation speed of the model have the potential to limit the relevance of the optimizer results. In order for this type of optimization scheme to produce actuator trajectories that are compatible with experimental operation, a balance needs to be found between model fidelity and calculation speed. One approach to achieving both high prediction accuracy and fast calculation speed that has gotten significant attention in recent years is using machine learning surrogates for high fidelity predictive models [18–20]. With this method, the machine learning algorithm is trained on simulation data generated by the high fidelity model, and is thus able to closely replicate that calculation with an inference time potentially orders of magnitude faster.

A model-based optimization tool is presented here for DIII-D using the nonlinear transport code COTSIM as the predictive model. As shown in figure 1, COTSIM is a modular code that allows the user to choose between a variety of modules and configurations to customize the predictive simulation. The configuration of COTSIM used here evolves transport equations for the poloidal stream function, electron temperature, and toroidal rotation profiles and assumes a prescribed equilibrium, including the plasma shape and position. After evolving each transport equation, it calculates values such as the safety factor profile and β_N that are used to define different scenarios. A variety of different models are available to calculate plasma properties such as the density, resistivity, transport coefficients, and pedestal location, as well as the auxiliary heating and non-inductive current sources. Specifically, two neural network surrogate models are available that will be used in this work. NubeamNet [21] replicates the calculation of the effects of neutral beam injection from NUBEAM [22], including the heating, current, and torque drive profiles, and has been integrated into COTSIM as an alternative to a simple empirical beam model. MMMnet [8] replicates the calculation of certain anomalous diffusivity coefficients from MMM [23], and is available as an option in COTSIM along with the Bohm/gyro-Bohm [24] and Coppi-Tang [25] models. Both of these neural network surrogates produce similar results to the models they were trained to replicate while achieving significantly shorter inference times. This allows COTSIM access to higher fidelity predictions while maintaining its fast calculation speed [26]. The optimizer presented in this work takes advantage of both the speed and the higher fidelity of COTSIM when these models are used to determine actuator trajectories that produce a target plasma scenario in a reasonable amount of time and with a high level of accuracy.

This paper is organized as follows. In section 2, the optimization problem is defined in terms of the target plasma scenario and the constraints on both the plasma state and the available actuators. In section 3, the optimizer is tested using plasma scenario targets that are specifically generated to ensure that they are feasible for the predictive model to achieve. It is tested for cases using NubeamNet as the beam model and for cases using MMMnet as the turbulent transport model. In section 4, the optimizer is tested using plasma scenario targets that are taken from an experimental shot, and therefore it is not guaranteed that the predictive model will be able to perfectly match the target with any available set of actuator trajectories. This test is intended to evaluate the capabilities of the optimizer to produce real experimental plasma scenarios of interest, and thus the relevance of this tool for future scenario development efforts. In section 5, conclusions are discussed and future plans for this tool are proposed.

2. Optimization problem definition

2.1. Target state and cost function

The optimization algorithm is designed to find the set of actuator trajectories that lead to the plasma state that most closely matches a target plasma state. This target state can be defined

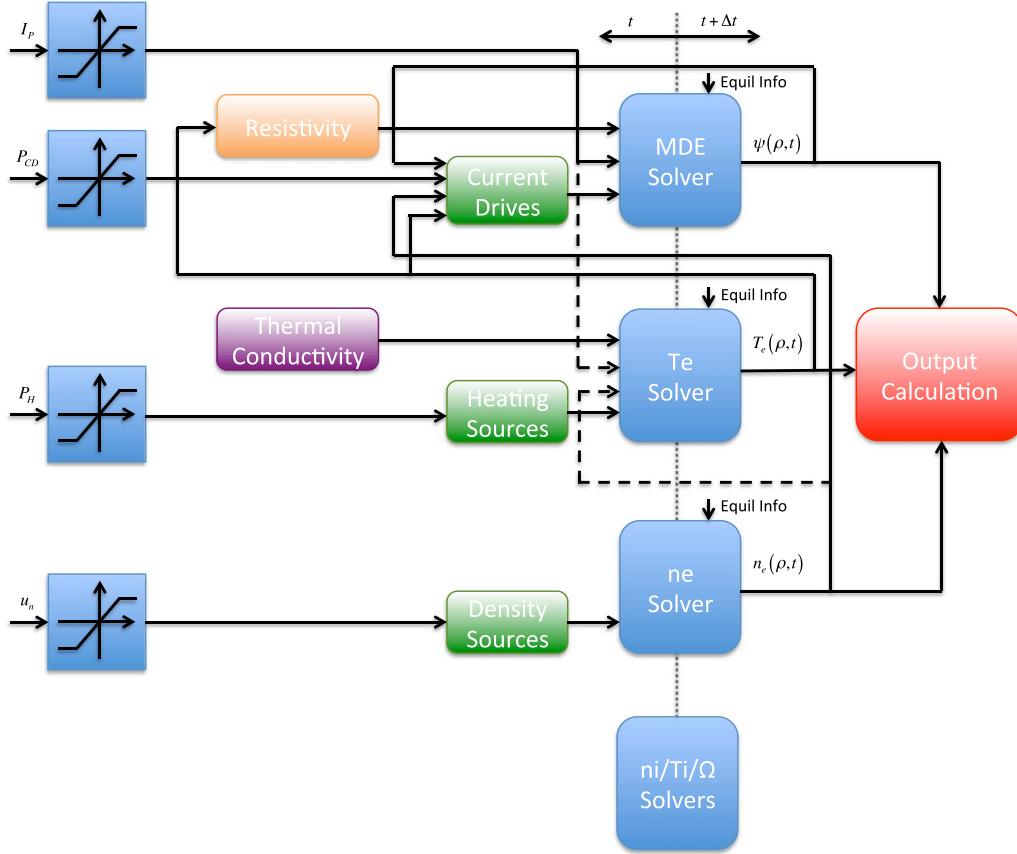


Figure 1. Schematic of the Control-Oriented Transport Simulator (COTSIM). Inputs to the simulation are plasma current (I_p), power to auxiliary current drive sources (P_{CD}), power to auxiliary heating sources (P_H), and line-average density (u_n). The magnetic diffusion equation (MDE) is solved for the evolution of the poloidal stream function (ψ), and a combination of other transport equations and simplified solvers are available to predict the evolution of electron temperature (T_e), electron density (n_e), ion temperature (T_i), ion density (n_i), and rotation (Ω).

by a variety of both profile (e.g. electron temperature T_e , safety factor q) and scalar (e.g. β_N) properties. The scalar properties are defined at a specific subset of times throughout the shot (t_l), and the profile properties are defined at the final optimization time (t_f). After this final time, the goal is for the plasma to be in a stationary state, so the final optimized values of the actuator trajectories can be maintained and the final target values of the scalars and the target profiles will be preserved.

In order to quantify how close the optimized output gets to the target state, a cost function is devised. This cost function contains terms that measure proximity to the scalar targets at each time point for which the scalar targets are defined, proximity to the profile targets across the spatial range of the profile at the final optimization time, and a measure of the stationarity of the final plasma state. The cost function is defined as,

$$J(t_s, t_f) = k_{ss} J_{ss}(t_f) + \sum_s \sum_l k_s J_s(t_l) + \sum_p k_p J_p(t_f), \quad (1)$$

where s represents the set of scalars used to characterize the target plasma state, p represents the set of profiles used to characterize the target plasma state, and the weights k_{ss} , k_s , and k_p define the relative weight placed on the stationary term, each

scalar term, and each profile term. Allowing for certain aspects of the target to be weighted higher than others adds a layer of flexibility that can be tuned to align with the needs of the scenario developer. The different terms in the cost function are defined as,

$$\begin{aligned} J_{ss}(t_f) &= \int_0^1 W_{ss}(\hat{\rho}) [g_{ss}(\hat{\rho}, t_f)]^2 d\hat{\rho}, \\ J_s(t_l) &= [s^{tar}(t_l) - s(t_l)]^2, \\ J_p(t_f) &= \int_0^1 W_p(\hat{\rho}) [p^{tar}(\hat{\rho}) - p(\hat{\rho}, t_f)]^2 d\hat{\rho}. \end{aligned} \quad (2)$$

where the weights W_{ss} and W_p define the relative weights placed on different spatial regions of the plasma for the stationary term and each profile term and the spatial dependence is given in term of $\hat{\rho} = \rho/\rho_b = \sqrt{\Phi/\pi B_{\phi,0}}/\rho_b$, where $B_{\phi,0}$ is the reference magnetic field at the geometric major radius R_0 , Φ is the toroidal magnetic flux, and ρ_b is the value of ρ at the plasma boundary. The stationary term comes from the time derivative of the plasma parameter that is generally the slowest to evolve, and is therefore expected to be the last parameter to reach a stationary condition; in this case, safety factor profile is assumed to evolve on the slowest time scale. The safety

factor is related to the spatial gradient of the poloidal flux (see section 2.3), so the stationary condition is written as,

$$g_{ss}(\hat{\rho}, t) = \frac{\partial^2 \psi}{\partial t \partial \hat{\rho}}. \quad (3)$$

2.2. Simulation constraints

The optimizer algorithm obeys a variety of constraints that fall into two general categories: constraints on the actuator trajectories that are being optimized; and constraints on the predicted plasma state. The first set of constraints on the actuator trajectories provide a minimum and maximum absolute value for each actuator at any point in time based on the hardware capabilities of the DIII-D tokamak. For each of the eight individual neutral beams, the injected power is limited to 3 MW, and the sum of the injected power across all neutral beams is constrained to fall between 3 and 12 MW. These limits can easily be fine-tuned or adjusted based on changes in the availabilities of different beams. The total ECH power injected is limited to 3 MW based on the assumption of five available gyrotrons; as more gyrotrons are expected to become available at DIII-D in the coming experimental campaigns, this constraint can be easily adjusted. The total auxiliary power injected into the plasma is constrained to remain above 5 MW; this is to ensure that the plasma remains in H-mode. The total plasma current is restricted to remain between 0.5 and 1.5 MA, and the line average electron density is constrained to fall between 2×10^{19} and $6 \times 10^{19} \text{ m}^{-3}$. These constraints on the actuator values at any point in time are written as,

$$\begin{aligned} P_{nbi,i} &\geq 0 \text{ MW} \cap P_{nbi,i} \leq 3 \text{ MW}, \\ &\text{for } i \in [30L, 30R, 150L, 150R, 210L, 210R, 330L, 330R] \\ P_{nbi,\text{total}} &\geq 3 \text{ MW} \cap P_{nbi,\text{total}} \leq 12 \text{ MW} \\ P_{ec,\text{total}} &\geq 0 \text{ MW} \cap P_{ec,\text{total}} \leq 3 \text{ MW} \\ P_{\text{total}} &\geq 5 \text{ MW}. \\ I_p &\geq 0.5 \text{ MA} \cap I_p \leq 1.5 \text{ MA} \\ \bar{n}_e &\geq 2 \times 10^{19} \text{ m}^{-3} \cap \bar{n}_e \leq 6 \times 10^{19} \text{ m}^{-3}. \end{aligned} \quad (4)$$

An additional set of constraints are applied to the rate of change of two of the available actuators. There is no need to impose a rate of change constraint on the auxiliary powers because it is possible to change the magnitude of power being injected into the system very quickly. However, it can take more time to safely and effectively adjust the values of the plasma current and line average density. Because of this, maximum rate of change constraints are placed on these two actuators in both the positive and negative directions. Note that the values chosen here for these rate limits came from qualitative inspection of previous DIII-D plasmas, so the maximum rate limits may not be achievable in every scenario. The magnitude of these rate limits can easily be adjusted by the scenario developer based on expectations of what rate of change is realistically achievable. The rate limit constraints used in the optimization cases presented in the work are written as,

$$\begin{aligned} \dot{I}_p &\leq |6| \text{ MA s}^{-1} \\ \dot{\bar{n}}_e &\geq -1 \times 10^{19} \text{ m}^{-3} \text{ s}^{-1} \cap \dot{\bar{n}}_e \leq 3 \times 10^{19} \text{ m}^{-3} \text{ s}^{-1}. \end{aligned} \quad (5)$$

A final set of constraints is applied to the plasma state as determined by the predictive model. These constraints are based on well-known stability limits and are designed to keep the plasma in a stable regime. The first stability limit considered is the Greenwald density limit, defined as $n_G = I_p / \pi a^2$, where n_G is in 10^{20} m^{-3} , I_p is in MA, and a is in m [27]. The line average density is constrained to remain below the Greenwald limit. The second stability limit taken into account is related to the normalized pressure ratio β_N . Very high values of β_N are associated with a variety of MHD instabilities, and have the potential to cause a disruption [28]. In order to avoid this, a maximum value of β_N is enforced. Finally, a minimum value of the safety factor q is imposed at any point in the spatial range of the profile at any time in the simulation. Too low values of q are known to cause sawtooth instabilities, or a sudden drop in temperature in the center of the plasma [29], which have the potential to seed other, more dangerous instabilities. This set of constraints is intended to keep the plasma state away from basic stability limits, and is written as,

$$\begin{aligned} \bar{n}_e &\leq n_G \\ \beta_N &\leq 4 \\ q(\hat{\rho}) &\geq 1.05. \end{aligned} \quad (6)$$

The values chosen for these constraints for the cases presented in this work are approximate; for some applications they may not be strict enough to guarantee plasma stability, and in other applications they may be overly restrictive. For example, certain plasma scenarios are intended to operate with $q < 1$. The values of the constraints in equation (6) are used for the optimization cases presented here, but they can easily be adjusted on a case-by-case basis. This allows the user to take into account more detailed knowledge of the specific plasma scenario they are working with and set the values of the constraints accordingly.

2.3. Optimization problem statement

The trajectories of each individual actuator are discretized in time using a temporal grid defined as t_i , $i = 1, 2, \dots, n$, where t_n is equal to the final simulation time t_f . The discrete actuator trajectories are then written in a single vector of parameterized input to the simulation as,

$$\begin{aligned} \alpha = [&P_{nbi,30L}(t_1), \dots, P_{nbi,30L}(t_n), P_{nbi,30R}(t_1), \dots, P_{nbi,30R}(t_n), \\ &\dots, P_{nbi,330R}(t_1), \dots, P_{nbi,330R}(t_n), P_{ec}(t_1), \dots, P_{ec}(t_n), \\ &I_p(t_1), \dots, I_p(t_n), \bar{n}_e(t_1), \dots, \bar{n}_e(t_n)]. \end{aligned} \quad (7)$$

The evaluation of the cost function defined in (1) and (2) depends on the values of the stationary state term $g_{ss}(\hat{\rho}, t_f)$ and the scalars and profiles that are chosen to be included in the target. In this case, the only scalar is chosen as $\beta_N(t)$, so $s = [\beta_N]$. The two profiles chosen for the target plasma state are

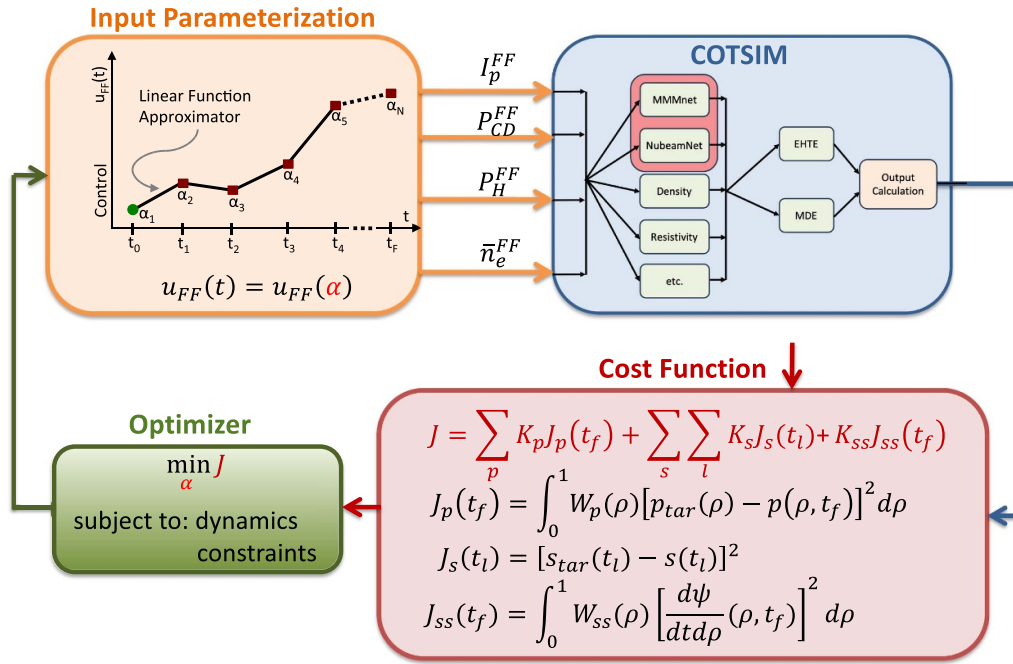


Figure 2. Visual representation of the workflow of the optimization method. This diagram includes the input parameterization, predictive model of the plasma evolution (COTSIM), evaluation of the cost function, and minimization of the cost function with respect to the parameterized inputs.

the electron temperature $T_e(\hat{\rho}, t_f)$ and the safety factor $q(\hat{\rho}, t_f)$, so $p \in [T_e, q]$. The variable $\theta = \frac{\partial \psi}{\partial \hat{\rho}}$ is introduced and defined as the spatial gradient of the poloidal stream function. Using this definition, the safety factor can be shown to depend on θ as $q = \frac{-B_{\phi,0} \rho_b^2 \hat{\rho}}{\theta}$. In addition, the stationary state term $g_{ss}(\hat{\rho}, t_f)$ depends on θ as $\dot{\theta} = \frac{\partial}{\partial t}(\theta) = \frac{\partial}{\partial t} \left(\frac{\partial \psi}{\partial \hat{\rho}} \right)$, as shown in (3). The nonlinear constrained optimization problem can be expressed in terms of the parameterized inputs α defined as in (7) and the cost function (1) as,

$$\min_{\alpha} J(t_f) = J(\dot{\theta}(t_f), \beta_N(t), T_e(t_f), \theta(t_f)), \quad (8)$$

subject to,

$$\begin{aligned} \begin{bmatrix} \dot{T}_e \\ \dot{\theta} \end{bmatrix} &= z(T_e, \theta, u) \rightarrow \text{dynamics of the plasma state,} \\ A_u(t) \alpha &\leq b_u \rightarrow \text{actuator constraints,} \end{aligned} \quad (9)$$

where the predictive model is defined by z and the plasma state constraints (6), the actuator magnitude limits (4), and the actuator rate of change limits (5) are represented by A_u and b_u .

This optimization problem can be approached using the sequential quadratic programming (SQP) [30, 31] algorithm. This method minimizes a nonlinear program by solving a sequence of quadratic subprograms using constraints that are linearized around the current estimate. For this application, the nonlinear program is defined by (8) and (9). An overview of this optimization process is presented graphically in figure 2.

3. Numerical testing of the optimization algorithm on feasible targets

In this section, the optimization problem defined in section 2.3 is solved for targets that are guaranteed to be feasible. In this case, feasible is taken to mean that there exists a set of actuator trajectories for which the predictive model exactly reproduces all of the scalar and profile targets. In order to guarantee this feasibility, the targets are taken from the outputs of a COTSIM simulation using the same configuration of COTSIM with all the same settings that the optimizer will be using. This is not a guarantee that the optimizer will achieve a cost function value of zero. The feasible targets are not necessarily taken from a time point where the predicted plasma is in a stationary state, so there may not be a solution where a zero value of the stationary state term is consistent with a perfect match to the scalar and profile targets. In this situation, the optimizer will use the applied weights to determine which terms to place the highest priority on minimizing.

For the optimization cases presented in this section, the initial guess of actuator trajectories used by the optimizer is taken to be 1 MW at all simulation times for the 30°, 210°, and 330° neutral beams, 0 MW at all simulation times for the 150° neutral beams, 1 MW at all simulation times of total EC power, an initial I_p value of 0.6 MA at $t = 0.6$ s ramping up to 1 MA for the rest of the shot, and a line-average density of $4 \times 10^{19} \text{ m}^{-3}$ at all simulation times. These values were chosen only because they are well within the applied actuator constraints, so they are unlikely to produce a plasma state that is a good match to the target. It is then up to the optimizer to figure out what changes need to be made to the actuator trajectories to achieve the target plasma state.

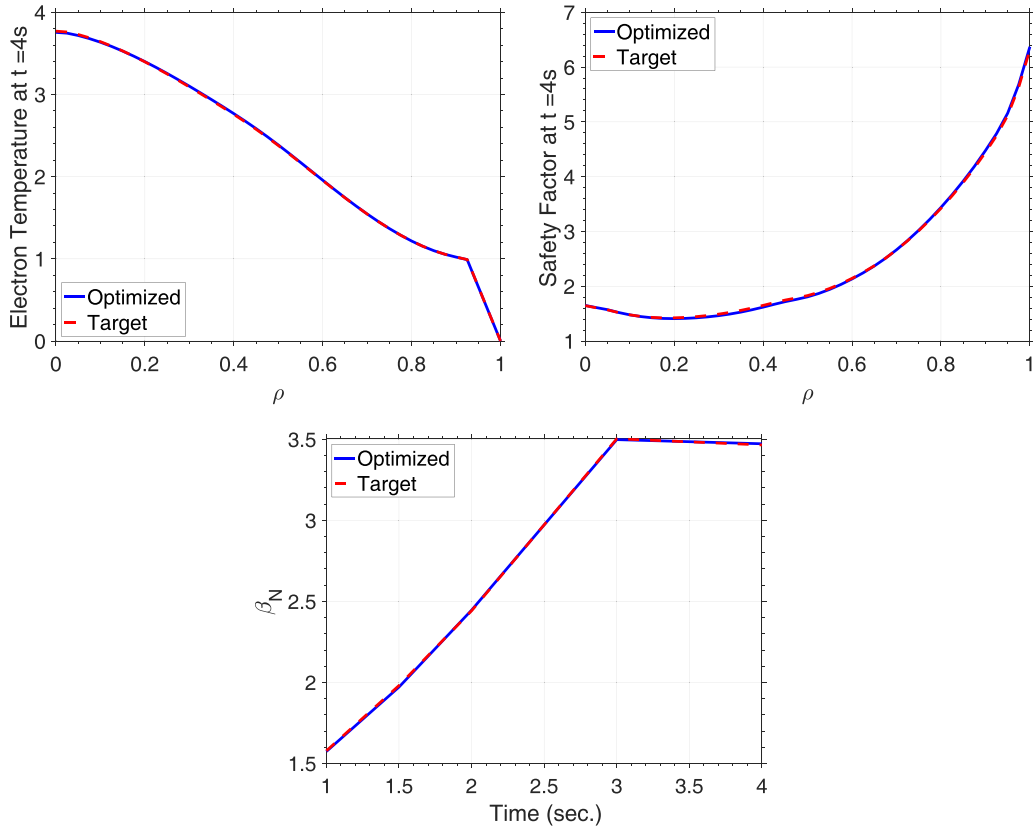


Figure 3. Simulated electron temperature, safety factor, and β_N achieved by the optimized actuator trajectories compared to target parameters generated by COTSIM using NubeamNet and using actuator trajectories from DIII-D shot 147634.

In all the cases shown in this work, the weights applied to the cost function in (1) and (2) are chosen as,

$$\begin{aligned}
 k_{ss} &= 1, \quad k_q = 1000, \quad k_{T_e} = 1000, \quad k_{\beta_N} = 1000, \\
 W_{ss} &= \begin{cases} 250 & 0 \leq \hat{\rho} \leq 0.4 \\ 500 & 0.4 < \hat{\rho} \leq 0.8, \\ 1250 & 0.8 < \hat{\rho} \leq 1 \end{cases}, \\
 W_q &= 10, \\
 W_{T_e} &= 10.
 \end{aligned} \tag{10}$$

This works out to roughly an order of magnitude higher weight on the profile targets than on the scalar or the stationary state target. In addition, this places equal weight across the spatial range of each of the profiles, but a significantly higher weight on the outer portion of the spatial range of the stationary state target. The plasma current is proportional to the boundary condition of the transport equation for the poloidal flux profile, so this spatial variation is applied in an attempt to force the plasma current to reach a steady state by the end of the simulation time. The values of these weights are very easy to manipulate, and in the future can be tuned to match the priorities of the user.

3.1. Optimization results using NubeamNet

For this first optimization case, the targets were generated from a COTSIM simulation using NubeamNet to calculate

the heat and current deposition from the neutral beams; the Bohm/gyro-Bohm model to calculate the anomalous component of electron thermal diffusivity; no neoclassical component of electron thermal diffusivity; a simplified 0.5D model for the electron density evolution; a simplified Spitzer resistivity model; the Sauter [32] bootstrap current model; and the PEDESTAL [33] model to calculate the width and height of the electron temperature pedestal. The same configuration of COTSIM was used by the optimization algorithm. To generate the targets, COTSIM used actuator trajectories based on those used for DIII-D shot 147634. This shot is a high q_{\min} scenario shot, and demonstrates a high non-inductive current fraction that is appealing as a starting point for developing steady-state plasma scenarios. The empirical models in COTSIM are tuned to this same experimental shot. Figure 3 shows results from the optimization case using these targets, and demonstrates a nearly perfect match between the all of the profile and scalar targets and the results of the optimization. The actuator trajectories that the optimizer found to produce this nearly perfect match are presented in figure 4, compared to the actuator trajectories that were used by COTSIM to generate the target plasma state. The actuator values found by the optimizer are generally reasonable and obey all of the constraints placed upon them. In addition, the final two values of the plasma current are essentially identical, indicating that the high weight on the outer spatial range of the stationary state term is indeed forcing the plasma current to converge to a steady state. For plasma current and line-average density, the

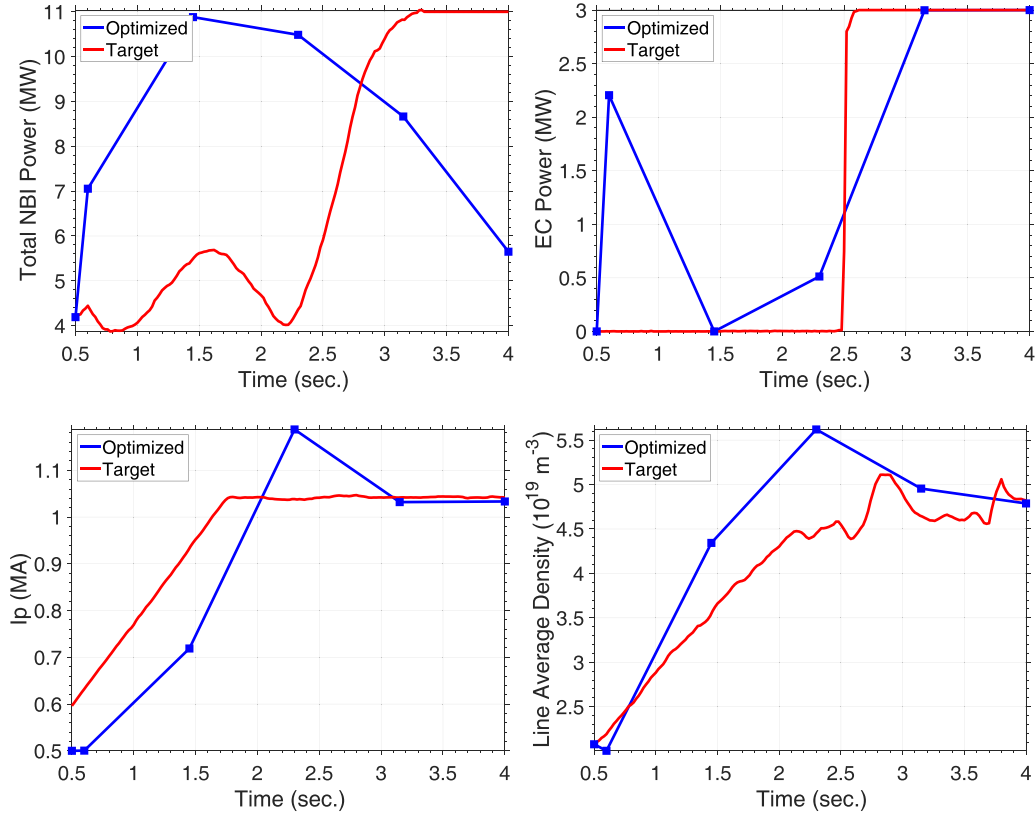


Figure 4. Actuator trajectories determined by optimizer compared to the actuator trajectories that generated the target profiles NubeamNet turned on.

optimizer produces a reasonable approximation of the trajectories that generated the target. The optimizer also figures out that the EC power is low for the first half of the simulation and high for the second half. However, it does not come very close to matching the NBI power. The COTSIM simulation that generated the target took its actuator trajectories from a real DIII-D shot that was not constrained to reach a stationary state; the additional objective of attaining a stationary state imposed on the optimized solution could be a source of this discrepancy in the neutral beam power trajectory. The optimizer could be making changes to the beam powers to control the evolution of the current to drive the safety factor profile to a stationary state.

3.2. Optimization results using MMMnet

For the second optimization case, the targets were generated from a COTSIM simulation using an empirical beam model to calculate the heat and current deposition from the neutral beams; MMMnet to calculate the anomalous component of electron thermal diffusivity; no neoclassical component of electron thermal diffusivity; a simplified 0.5D model for the electron density evolution; a simplified Spitzer resistivity model; the Sauter [32] bootstrap current model; and a fixed width and height of the electron temperature pedestal. The same configuration of COTSIM was used by the optimization algorithm. To generate the targets, COTSIM again used the actuator trajectories from DIII-D shot 147634. Figure 5 shows

results from the optimization case using these targets; like in the NubeamNet case, a nearly perfect match is seen between the all of the profile and scalar targets and the results of the optimization. The actuator trajectories that the optimizer found to produce this result are presented in figure 6. Once again, the optimizer is generally able to roughly recover the trajectories that produced the target for the on-axis beam powers, EC power, plasma current, and line-average density, but reaches a different solution for the beam powers.

4. Numerical testing of the optimization algorithm on experimental targets

As this optimizer is intended to aid in scenario development activities, it was tested to see if it could find actuator trajectories that achieve a plasma scenario that is currently of interest to the tokamak community. DIII-D shot 155543 is a high- β_N hybrid scenario [34] discharge utilizing off-axis neutral beam injection, which is a very promising scenario for future reactors. If the optimizer can figure out how to achieve the profiles from this shot, that is a good indication that it will be useful for future scenario development work. In this section, the optimizer was given a set of targets taken from an analysis mode TRANSP run for shot 155543, meaning these profiles were originally generated by fitting experimental data. If the optimizer can match these experimental profiles, that is a good indication that it will be a useful tool for further developing the hybrid and potentially and other scenarios.

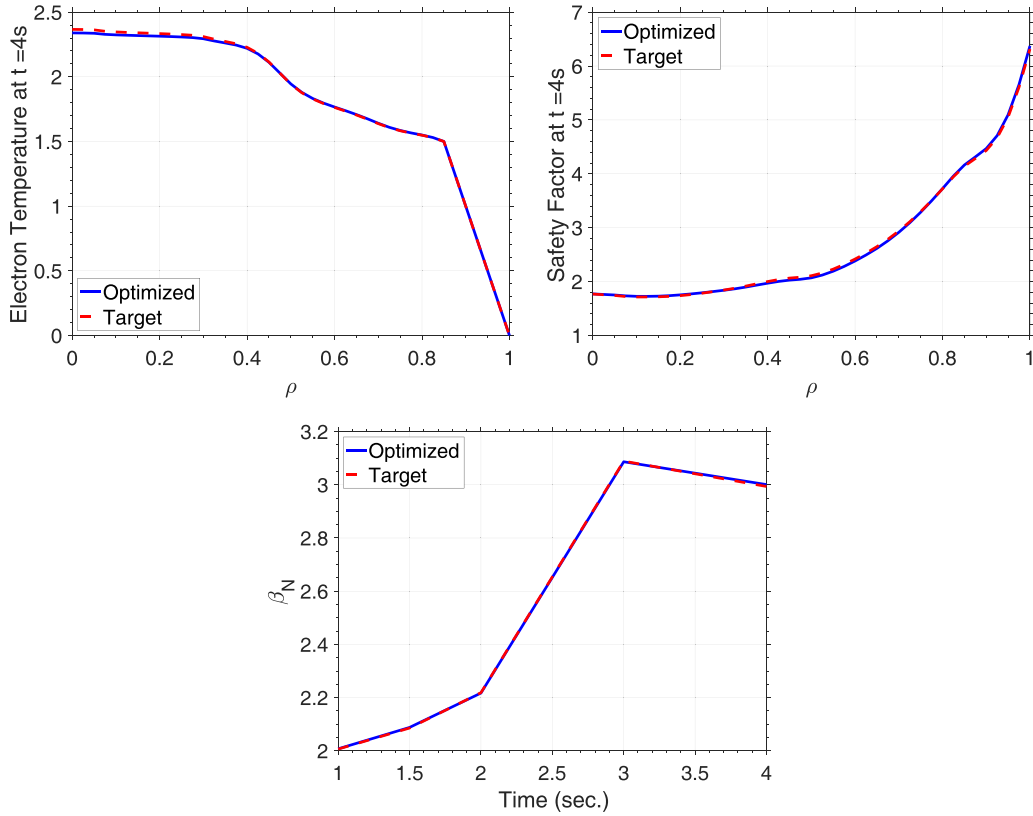


Figure 5. Simulated electron temperature, safety factor, and β_N achieved by the optimized actuator trajectories compared to target parameters generated by COTSIM with MMMnet turned on and using actuator trajectories from DIII-D shot 147634.

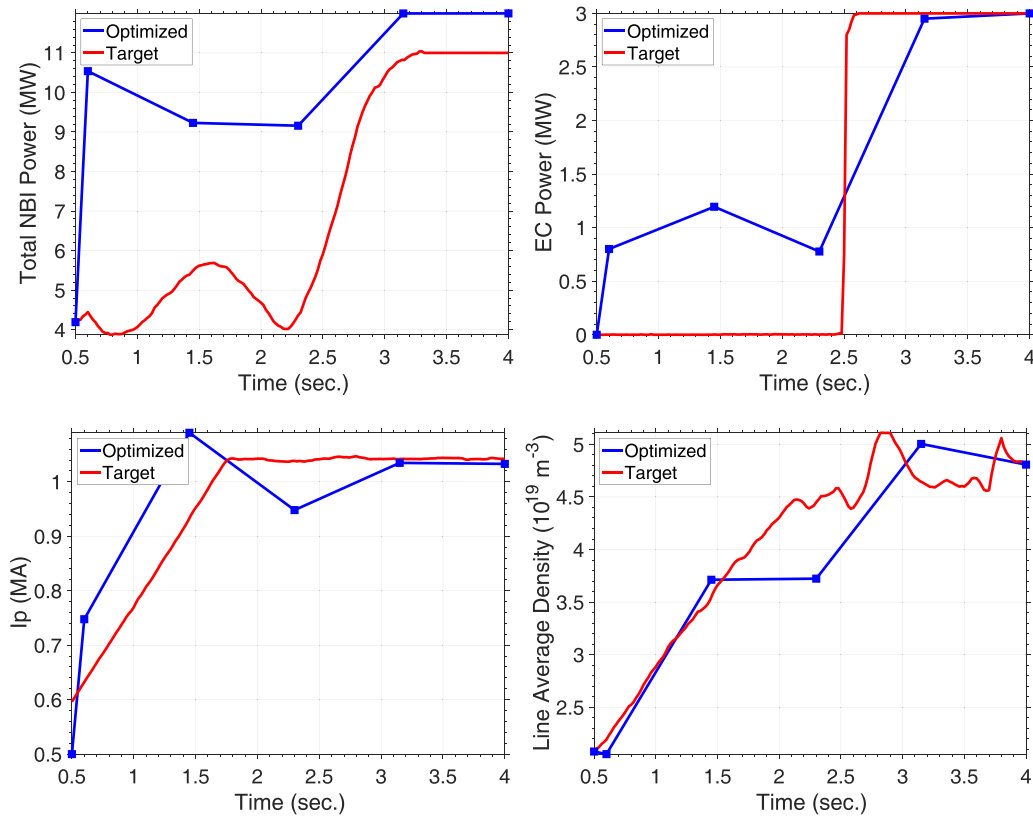


Figure 6. Actuator trajectories determined by optimizer compared to the actuator trajectories that generated the target profiles MMMnet turned on.

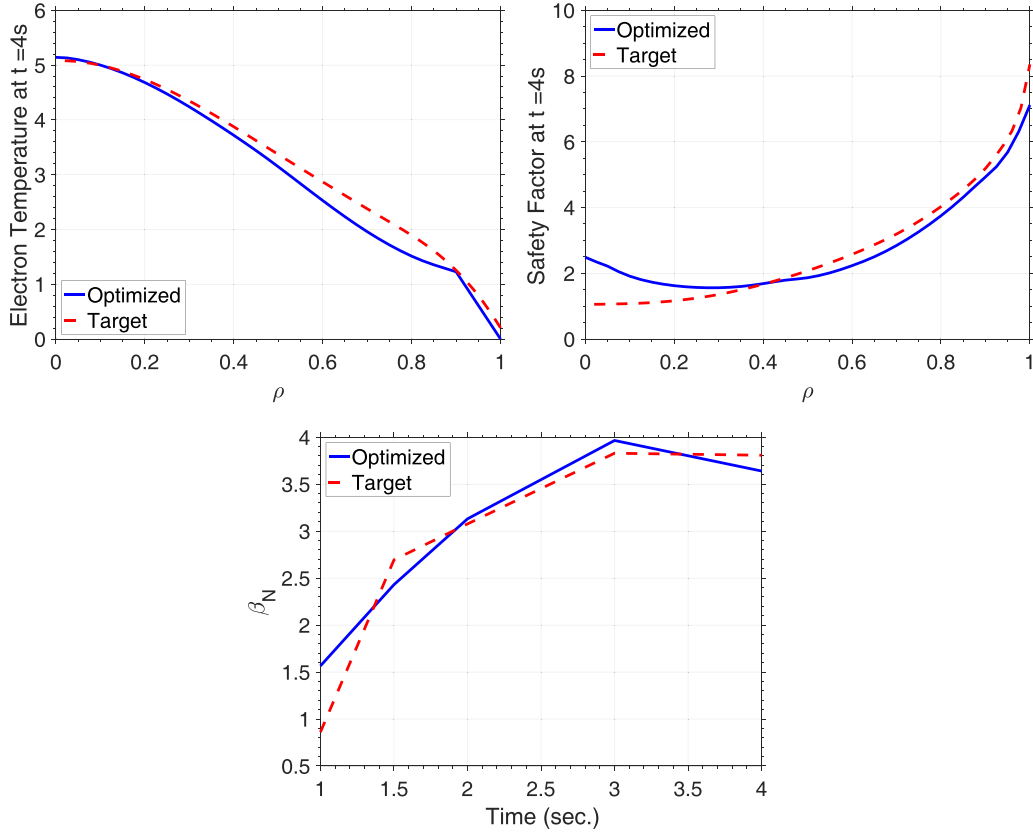


Figure 7. Simulated electron temperature, safety factor, and β_N achieved by the optimized actuator trajectories compared to target parameters taken from experimental shot 155543.

This optimization case was run using the same configuration of COTSIM as the case presented in section 3.1. The optimizer was originally tested for these experimental targets using the weights given in (10); although a decent match was achieved to the T_e and q profile targets, the predicted β_N evolution was still fairly far off from the target. Because of this, the optimization was rerun with a higher weight on the scalar β_N target; the weights used for the optimization case presented in this section are,

$$\begin{aligned}
 k_{ss} &= 1, k_q = 1000, k_{T_e} = 1000, k_{\beta_N} = 10000, \\
 W_{ss} &= \begin{cases} 250 & 0 \leq \hat{\rho} \leq 0.4 \\ 500 & 0.4 < \hat{\rho} \leq 0.8, \\ 1250 & 0.8 < \hat{\rho} \leq 1 \end{cases}, \\
 W_q &= 10, \\
 W_{T_e} &= 10.
 \end{aligned} \tag{11}$$

Results of this optimization are shown in figure 7. Although there is a dip in the shape of the T_e profile around $\hat{\rho} = 0.8$, the optimized T_e values at the top of the pedestal and in the core of the plasma are a very close match to the target profile. A decent match for the q profile is produced in the spatial range from $0.3 \leq \hat{\rho} \leq 0.9$, with a small discrepancy in q_{95} potentially caused by a mismatch in the plasma shape assumptions and a noticeable over-prediction of the value of q_0 ; this seems to be the part of the target that the optimizer had the most trouble

matching. The predicted evolution of β_N over time is again a decent match to the target, not perfect but close enough to be clear that the optimizer is functioning as intended. The targets shown in figure 7 are not necessarily possible to achieve exactly with the predictive model being applied; even so, the optimization algorithm has managed to find a set of actuator trajectories that provide at least a decent match to all three scalar and profile targets.

The actuator trajectories that produced these results are shown in figure 8, and again appear to be reasonable requests and show a constant value of plasma current between the final two parameterized time points. Unlike in figures 4 and 6, the actuator trajectories in this case are not compared to the actuator trajectories that generated the target. In section 3, the relationship between the actuator trajectories and the targets they produce is fully characterized by COTSIM. It is theoretically possible for the optimizer to recover the trajectories that generated the targets, making the comparisons in figures 4 and 6 more meaningful to discuss. In this section, the relationship between the experimental actuator trajectories and the experimental targets is only approximately characterized by COTSIM; it is therefore possible for there to be a considerable mismatch between the experimental trajectories and the trajectories derived by the optimizer, and such a mismatch would not indicate any failure of the optimization tool. There are known limitations in the predictive model, such as the assumption of a prescribed equilibrium that matches the equilibrium

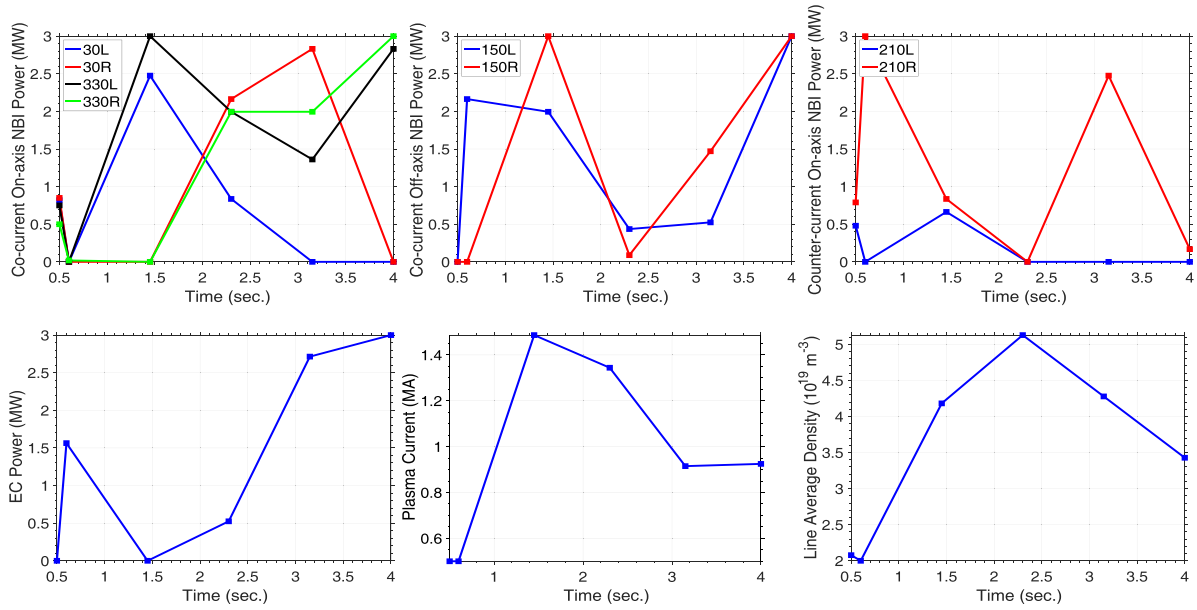


Figure 8. Actuator trajectories determined by optimizer that achieve the closest match to experimental target profiles with NubeamNet turned on.

from shot 147634. Next steps will include minimizing the effects of the known limitations of COTSIM model, including running this optimization algorithm for experimental targets using a prescribed equilibrium that better matches the equilibrium associated with the target profiles and scalars and eventually using an equilibrium solver. This could produce a more meaningful comparison with the experimental actuator trajectories as well as potentially a better match to the targets, specifically the target q profile. However, the fact that the optimizer has been shown to be capable of producing a decent match to the experimental plasma state still represents significant progress in the development of this optimization tool, and demonstrates its relevance for future scenario development work.

5. Conclusions and future work

In this work, an optimization algorithm has been developed to find actuator trajectories that produce a plasma state that most closely matches a target scenario. The optimizer uses the COTSIM code as its predictive model, including neural network surrogates that are available to increase the accuracy of predictions. The target plasma scenario is defined by the final values of a set of plasma profiles (electron temperature and safety factor) as well as the evolution of scalar values (β_N) over the time span of the simulation. How close any one predictive simulation gets to the target scenario is measured using a cost function that contains terms for each of the target plasma parameters as well as a measure of the stationarity of the final plasma state. The optimizer is able to manipulate the values of the total plasma current, the line-average electron density, the ECH/ECCD system, and the eight neutral beams that are available on DIII-D; these actuator values are constrained to

be in a range that is physically reasonable for DIII-D. The plasma state is also constrained to ensure it does not violate a set of basic stability limits. Minimization of the cost function with respect to the actuator trajectories, subject to the dynamic model and the set of constraints, is performed using the SQP algorithm.

Three different optimization cases are presented here. In the first two cases, the targets are generated by a simulation using the same configuration of COTSIM the optimizer is using, so the targets are guaranteed to be feasible for the optimizer to match. An optimization is performed using feasible targets when NubeamNet is used by COTSIM to calculate the beam heating and current drive, and another optimization is performed when MMMnet is used by COTSIM to calculate the anomalous electron thermal diffusivity coefficient. In both of these cases, the optimizer is able to achieve a nearly perfect match to all of the target plasma parameters. For the third case, the targets are taken from an experimental hybrid scenario DIII-D shot, so it is not guaranteed that the predictive model is capable of exactly reproducing the targets. However, the optimizer is still able to produce a decent match to the profile and scalar targets.

Current results indicate that this optimizer will be a useful tool for future scenario development work. It has the potential to help physicists figure out how to produce desired scenario targets that have not yet been reliably achieved in experiment. It could also be applied to achievable scenario targets to see if the same plasma state can be achieved using a different set of actuator trajectories, for example to take advantage of increased ECH power availability or to get around hardware maintenance issues. However, there are a number of tests still to be run, and further improvements that could be made. The configuration of COTSIM used in this work assumes the same prescribed equilibrium for all

of the cases shown; a more accurate prescribed equilibrium could be used for the experimental targets, or an equilibrium solver that has been recently implemented into COTSIM could be employed. In addition, the EPEDNN [19] neural network surrogate model is currently available in COTSIM, and it is planned to add other models such as a surrogate for GENRAY/CQL3D [20] in the future. Replacing more of the simplified, control-oriented models in COTSIM with higher-fidelity surrogate models should improve the accuracy of the predictive model, making this optimization tool even more appealing for scenario development and experimental planning.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, using the DIII-D National Fusion Facility, a DOE Office of Science user facility, under Awards DE-FC02-04ER54698, DE-SC0010661, DE-SC0013977, and by the National Science Foundation Graduate Research Fellowship Program under Grant No. 1842163.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORCID iDs

S. Morosohk  <https://orcid.org/0000-0002-3133-5095>
 B. Leard  <https://orcid.org/0000-0003-4141-080X>
 T. Rafiq  <https://orcid.org/0000-0002-2164-1582>

References

- [1] Hofmann F. and Jardin S. 1990 *Nucl. Fusion* **30** 2013
- [2] Hawryluk R. 1981 *Physics of Plasmas Close to Thermonuclear Conditions* (Pergamon)
- [3] Breslau J., Gorelenkova M., Poli F., Sachdev J. and Yuan X. 2018 TRANSP [computer software] (<https://doi.org/10.11578/dc.20180627.4>)
- [4] Poli F. et al 2018 How predict-first will change our approach to experimental planning *EPS Conf. on Plasma Physics (Prague, Czech Republic, 2–6 July 2018)*
- [5] Odstrcil T., Howard N., Sciortino F., Chrystal C., Holland C., Hollmann E., McKee G., Thome K.E. and Wilks T.M. 2020 *Phys. Plasmas* **27** 082503
- [6] Lyons B., Paz-Soldan C., Meneghini O., Lao L.L., Weisberg D.B., Belli E.A., Evans T.E., Ferraro N.M. and Snyder P.B. 2018 *Phys. Plasmas* **25** 056111
- [7] Felici F., Sauter O., Coda S., Duval B.P., Goodman T.P., Moret J.-M. and Paley J.I. 2011 *Nucl. Fusion* **51** 083052
- [8] Morosohk S.M., Pajares A., Rafiq T. and Schuster E. 2021 *Nucl. Fusion* **61** 106040
- [9] Ilhan Z.O. 2016 Model-based optimization and feedback control of the current density profile evolution in NSTX-U *PhD Thesis* Lehigh University
- [10] Wehner W., Schuster E., Boyer M. and Poli F. 2019 *Fusion Eng. Des.* **146** 547
- [11] Wang H., Schuster E., Rafiq T., Kritz A. and Ding S. 2017 *Fusion Eng. Des.* **123** 569
- [12] Ou Y., Xu C., Schuster E., Luce T.C., Ferron J.R., Walker M.L. and Humphreys D.A. 2008 *Plasma Phys. Control. Fusion* **50** 115001
- [13] Xu C., Ou Y., Dalessio J., Schuster E., Luce T.C., Ferron J.R., Walker M.L. and Humphreys D.A. 2010 *IEEE Trans. Plasma Sci.* **38** 163
- [14] Wehner W. et al 2017 *Fusion Eng. Des.* **123** 513
- [15] Van Dongen J., Felici F., Hogewij G., Geelen P. and Maljaars E. 2014 *Plasma Phys. Control. Fusion* **56** 125008
- [16] Mulders S.V., Felici F., Sauter O., Citrin J., Ho A., Marin M. and van de Plassche K.L. 2021 *Nucl. Fusion* **61** 086019
- [17] Teplukhina A., Sauter O., Felici F., Merle A. and Kim D. 2017 *Plasma Phys. Control. Fusion* **59** 124004
- [18] Citrin J., Breton S., Felici F., Imbeaux F., Aniel T., Artaud J.F., Baiocchi B., Bourdelle C., Camenen Y. and Garcia J. 2015 *Nucl. Fusion* **55** 092001
- [19] Meneghini O. et al 2017 *Nucl. Fusion* **57** 086034
- [20] Wallace G., Bai Z., Sadre R., Perciano T., Bertelli N., Shiraiwa S., Bethel E.W. and Wright J.C. 2022 *J. Plasma Phys.* **88** 895880401
- [21] Morosohk S.M., Boyer M.D. and Schuster E. 2021 *Fusion Eng. Des.* **163** 112125
- [22] Pankin A., McCune D., Andre R., Bateman G. and Kritz A. 2004 *Comput. Phys. Commun.* **159** 157
- [23] Rafiq T., Kritz A., Weiland J., Pankin A. and Luo L. 2013 *Phys. Plasmas* **20** 032506
- [24] Erba M., Aniel T., Basiuk V., Becoulet A. and Litaudon X. 1998 *Nucl. Fusion* **38** 1013
- [25] Jardin S., Bell M. and Pomphrey N. 1993 *Nucl. Fusion* **33** 371
- [26] Morosohk S. 2024 Enhanced plasma profile estimation and control in tokamaks via machine learning *PhD Thesis* Lehigh University
- [27] Greenwald M., Terry J.L., Wolfe S.M., Ejima S., Bell M.G., Kaye S.M. and Neilson G.H. 1988 *Nucl. Fusion* **28** 2199
- [28] Okabayashi M. et al 2009 *Nucl. Fusion* **49** 125003
- [29] Wesson J. 1990 *Nucl. Fusion* **30** 2545
- [30] Han S. 1977 *J. Optim. Theory Appl.* **22** 297–309
- [31] Nocedal J. and Wright S. 1999 Sequential quadratic programming *Numerical Optimization* (Springer) pp 526–73
- [32] Sauter O., Angioni C. and Lin-Liu Y. 1999 *Phys. Plasmas* **6** 2834
- [33] Onjun T., Bateman G., Kritz A. and Hammett G. 2002 *Phys. Plasmas* **9** 5018
- [34] Luce T. et al 2014 *Nucl. Fusion* **54** 013015