ACCEPTED MANUSCRIPT

Toward active disruption avoidance via real-time estimation of the safe operating region and disruption proximity in tokamaks

To cite this article before publication: Mark D Boyer *et al* 2021 *Nucl. Fusion* in press https://doi.org/10.1088/1741-4326/ac359e

Toward active disruption avoidance via real-time estimation of the safe operating region and disruption proximity in tokamaks

M. D. Boyer[1], C. Rea[2], M. Clement[1]

[1]Princeton Plasma Physics Laboratory, Princeton, NJ, USA
[2]Massachusetts Institute of Technology, Cambridge, MA, USA

E-mail: mboyer@pppl.gov

January 2021

Abstract. This paper describes a real-time capable algorithm for identifying the safe operating region around a tokamak operating point. The region is defined by a convex set of linear constraints, from which the distance of a point from a disruptive boundary can be calculated. The disruptivity of points is calculated from an empirical machine learning predictor that generates the likelihood of disruption. While the likelihood generated by such empirical models can be compared to a threshold to trigger a disruption mitigation system, the safe operating region calculation enables active optimization of the operating point to maintain a safe margin from disruptive boundaries. The proposed algorithm is tested using a random forest disruption predictor fit on data from DIII-D. The safe operating region identification algorithm is applied to historical data from DIII-D showing the evolution of disruptive boundaries and the potential impact of optimization of the operating point. Real-time relevant execution times are made possible by parallelizing many of the calculation steps and implementing the algorithm on a graphics processing unit (GPU). A real-time capable algorithm for optimizing the target operating point within the identified constraints is also proposed and simulated.

1. Introduction

Next step fusion experiments, like ITER, and future fusion reactors will require very high performance, generally corresponding to operation near the limits of plasma stability and device constraints, e.g., forces or heat loads. At the same time, operation of these devices must be reliable - discharges must be sustained for long periods of time, and the machine must be operational a large fraction of the year. Operation beyond passive stability limits and near engineering limits will set a requirement for a high performance, active control system that is robust to off-normal events. For off-normal events with a potential for severe device damage, the control system must be capable of actively avoiding these events and reliably predicting when they become unavoidable in order to initiate mitigation strategies.

One of the main concerns for tokamak reliability is a disruption: when the operating point moves (due to the complex interaction of actuators, sensors, the control system and its settings, device conditions, and the physics of plasma stability) to a region of operating space in which the active feedback system can no longer stabilize the plasma, resulting in an eventual collapse of plasma current and stored energy. While the collapse may be preceded by a chain of precursors, the loss of current occurs very rapidly, leading to large induced forces on machine components and potentially damaging heat loads on plasma facing components.

In order to maintain the operating point within the safe operating space, a control algorithm must be able to: estimate the present state of the system, predict the future state given candidate actuator trajectories, quantify the stability and performance of the system, and optimize the performance while maintaining a suitable distance from the boundaries of safe operating space. Control-oriented models have recently been developed that can provide real-time estimates and predictions of the state of the system and the effect of various actuators on future behavior [1, 2]. Furthermore, machine learning approaches have recently been used to accelerate computationally intensive physics modules, like NUBEAM [3] and transport models [4, 5], enabling additional physics fidelity in real-time capable models. While there are many possible model-based control strategies that can be applied [6, 7, 8], model predictive control (MPC) [9, 10] is a particularly promising approach to optimizing performance while actively avoiding constraints on the operating space [3, 11, 12, 13]. The missing component for applying MPC to the problem of disruption avoidance is a description of the safe operating space in a form that is suitable for use in a model-predictive control framework.

Ideally, physics models would be used to identify the boundaries of safe operation, and there has been a great deal of work in this direction, e.g, [14, 15]. However, the complex physics involved has motivated the application of data-driven techniques to the years of historical data available for existing fusion devices. Neural networks, support vector machines, and random forest algorithms have been applied and demonstrated high success rates [16, 17, 18, 19, 20, 21]. For a given input, these approaches typically provide an indicator of 'disruptivity' between 0 and 1, and a threshold is chosen to achieve desired true positive and false positive rates. Implemented in a real-time control system, these have been demonstrated to provide a trigger to initiate a soft or hard shutdown of the device to avoid a disruption.

These results are an important step toward reliable tokamak operation. However, shutting down the device and triggering mitigation strategies like shattered pellet injection are a last-resort option. Preferably, disruptions would be avoided instead by either recognizing and responding to any disruption precursors or, ideally, by maintaining the operating point within the safe operating space. Applications of techniques that make machine learning algorithms more interpretable, like sensitivity analysis and feature importance studies, are a step toward the former goal. The feature contributions identified in [21] provide important information for diagnosing the cause or type of imminent disruption event. This could provide operators with the

information needed to adjust operating points between shots, or, an expert-system could be developed to respond appropriately to different types of causes. However, in order to facilitate active optimization of the operating point to maintain a safe distance from the boundaries of safe operating space, these boundaries must be made available in a suitable form and rate for real-time control. This was explored successfully in [22] in which a real-time capable neural network was developed to predict the high-β limit on DIII-D. A similar approach, in which a neural network vertical growth rate estimation was used to adjust elongation to avoid vertical displacements, was recently demonstrated in [23].

This work aims to build upon the approaches in [22, 23] to account for additional disruption types, take advantage of the recent advances in machine learning disruption prediction, and to enable multi-dimensional optimization of operating points in real-time using approaches like model predictive control. An algorithm is developed to identify a convex set of linear inequalities in real-time that bound the safe operating region around the current operating point. The safety of operating points can be determined by machine learning disruption predictors and/or physics-based predictors. These inequalities provide a convenient calculation of the proximity of an operating point to a limit and the direction in state-space the system should move to increase the margin of safety. The use of a convex set of linear constraints to locally approximate the disruption boundary is ideal for application of efficient MPC algorithms [10] or other constrained optimization based control schemes. The algorithm is tested using a random forest disruption predictor, and is shown to accurately identify the safe operating region throughout a discharge with real-time relevant execution time. Real-time relevant execution times are made possible by parallelizing many of the calculation steps and implementing the algorithm on a graphics processing unit (GPU).

Figure 1 shows how the models and algorithms proposed in this work fit into a larger disruption and avoidance scheme. The contributions of this work, indicated in blue in the diagram, are envisioned to be used in nominal control scenarios to optimize the operating point. A high reliability disruption predictor (like the one developed in [21]) would be used to determine when to switch from the nominal control scenario to a soft landing or mitigation scenario.

The remainder of the paper is organised as follows. In section 2 the disruption predictor used for demonstration of the safe operating region identification algorithm is described. Section 3 describes the approach used to identify linear inequality constraints from the disruption prediction algorithm. The implementation of the algorithm for a GPU is described in Section 4. An initial approach to real-time optimization of an operating point within the identified boundaries is described in Section 5 and the algorithms are demonstrated in Section 6. Conclusions and future work are discussed in Section 7.

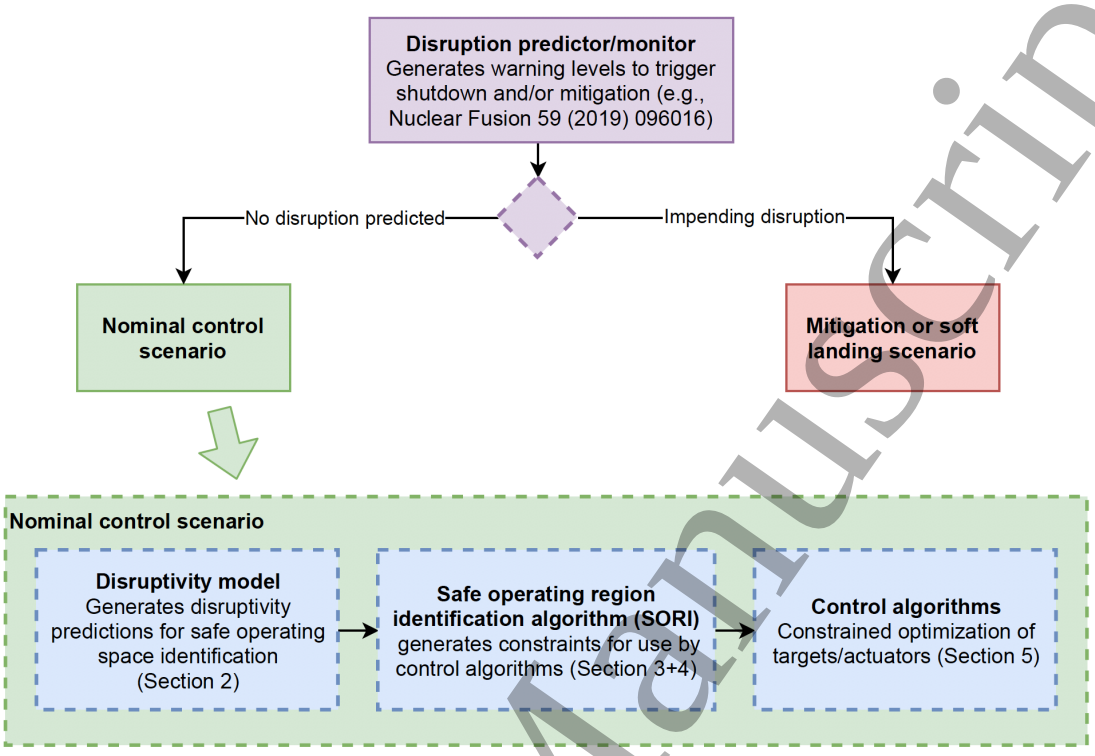

Figure 1: Simplified illustration of a disruption avoidance and mitigation scheme utilizing the models and algorithms proposed in this work (indicated in blue boxes). Two different disruption prediction models are considered, one tailored to triggering mitigation or soft landings, the other tailored to identification of the safe operating region for control/optimization.

2. Disruption predictor for safe operating region identification

2.1. Unique considerations for safe operating region identification

The safe operating region identification algorithm presented in this work requires a model to determine whether a given operating point is safe or unsafe. The algorithm is independent of the model chosen, and could use a physics-based model, any variety of machine learning model (decision tree, neural network, etc.), or an ensemble of models to classify points. While the structure of the model is not restricted by the method, the inputs to the model must be selected carefully to enable meaningful exploration of the dependence of disruptivity on the operating point.

For the purpose of triggering shutdown scenarios and measures to mitigate disruptions, a disruption detector can make use of as many signals as are available in real-time. However, for the problem of identifying the safe operating space, additional constraints must be placed on the plasma descriptors used as inputs to the model. Since the goal in this case is essentially to identify whether particular changes to manipulated variables will or will not lead to a disruption, it is critical that the model identifies

conditions that *cause* disruptions, as opposed to focusing on immediate *precursors* or *symptoms* of disruptions. For example, rapid vertical motion of the plasma is a symptom of a vertical displacement events and is often a precursor of a disruption. Monitoring this motion is particularly useful for early detection and mitigation of such disruptions, however, the root cause of the disruption in this situation is often a change in plasma parameters leading to an increase in the vertical growth rate beyond the value the control system can stabilize. A disruption predictor that uses internal inductance, elongation, and pressure of the plasma, key factors in determining the vertical instability, would enable the algorithm proposed in this work to determine actionable changes to manipulated variables (shaping, heating) that could maintain the growth rate within safe limits. This idea applies to other symptom/precursor signals, like plasma current target tracking error and locked mode detector signals. Furthermore, for the purpose of identifying (and optimizing operation within) the safe operating space, it is preferable to restrict the disruption prediction model to use plasma descriptors that are routinely controlled (enabling independent manipulation via available actuators) or not significantly correlated with other inputs (enabling independent evaluation of their impact on stability).

Limiting the inputs used for training will inevitably reduce the accuracy of the model when compared to models that include precursors and symptoms of disruptions. It is important to keep in mind that a complete disruption avoidance and mitigation strategy would make use of both types of models: lower accuracy models for approximating the safe operating region for operating point optimization, and higher accuracy models that would trigger shutdown or mitigation strategies in the event of an imminent disruption.

2.2. Predictor used for case study: Disruption Prediction via Random Forest algorithm

The safe operating region identification algorithm is demonstrated in this work using an adaptation of the existing Disruption Prediction via Random Forest (DPRF) algorithm, which is already installed in DIII-D PCS [21] and has recently been used in closed-loop experiments to test its integration with asynchronous and emergency response to off-normal events. DPRF is a supervised binary classifier based on the Random Forest algorithm [24], therefore the definition of proper class labels is mandatory. The two classes are defined as 'non-disruptive', i.e., all of the flattop data from non-disruptive discharges and the samples from stable regions shots that eventually disrupt, and 'unstable', i.e., the unstable time slices of disruptive discharges prior to the disruption. With respect to previous work from the authors [20, 21], the class label definition relies on the manual identification of the first disruptive precursor. Such a procedure is detailed in [25], and is aimed at refining the predictive capabilities of the model by further isolating the unstable operational space, based also on previous statistical studies by De Vries et al. [26].

Table 1: List of signals used for the development of the random forest-based disruption predictor for the application reported in this manuscript. The first column shows the relative feature importance in discriminating the unstable operational space.

Importance	Signal description	Source
0.242	Electron density, n_e [m^{-3}]	Interferometer
0.193	Plasma current, I_p [A]	Rogowski Coil
0.146	Squareness, ζ	EFIT
0.098	Plasma minor radius, a	EFIT
0.092	Normalised internal inductance, ℓ_i	EFIT
0.089	Stored plasma energy, W_{MHD} [J]	EFIT
0.075	Elongation, κ	EFIT
0.065	Triangularity, δ	EFIT

2.2.1. Dataset The dataset used for the application reported in this manuscript consists of a subset of disruptions belonging to the 2015 and 2016 DIII-D experimental campaigns: in particular, 198 disruptive discharges of different nature, and manually tagged to identify a $t_{unstable}$ - the beginning of the unstable phase leading to the disruption‡. Intentional disruptions and the ones caused by hardware failures or emergency shutdowns were not included in the dataset. Together with disruptive data, the training set is complemented by 927 non disruptive discharges from the same experimental campaigns. The algorithm is then trained using only data during the plasma current flattop of these 1125 discharges.

2.2.2. Inputs While the training dataset is consistent with the one used for the existing real-time implementation of DPRF, the input features differ substantially. As previously discussed, we aim at exploring the dependence of the disruptivity on directly controllable quantities, while the existing version was intended to be as descriptive as possible of relevant plasma instabilities that could potentially lead to a disruption. Specifically, the input features for the existing real-time DPRF included non-axisymmetric magnetic field amplitude, peaking factors from plasma profiles, and mostly dimensionless quantities, such as the Greenwald density fraction (instead of the electron density itself as listed in Table 1). The inputs used in this work are all either routinely controlled or are controllable using existing algorithms in the DIII-D PCS.

2.2.3. Predictor performance Offline performances for a binary classifier can be evaluated via the cost-sensitive F_γ-score optimization in the cross-validation procedure, where:

$$F_\gamma \text{ score} = \frac{(1+\gamma^2)\ TP}{(1+\gamma^2)\ TP + \gamma^2\ FN + FP} \quad (1)$$

‡ Here defined as $\max(\mathrm{d}I_p/\mathrm{dt})$, typically half-way down the final current quench.

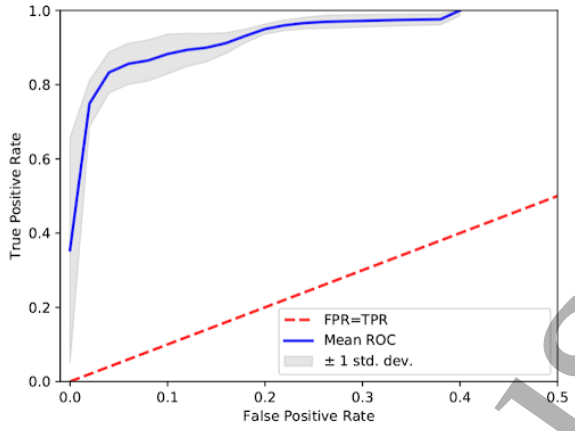

Figure 2: ROC curve for the disruption predictor used in this work illustrating the trained model's performance during the validation procedure. The thick blue line is the average of the ROCs from the five folds.

In the formula above, the True Positives (TP) and False Positives (FP) metrics refer to the correct or mistaken detection of disruptive discharges ahead of time, while True Negatives (TN) and False Negatives (FN) refer to the correct/incorrect identification of non-disruptive data. If we choose $\gamma > 1$, we attribute more importance to the high accuracy on the positive class detection (usually, the minority class), while trying to minimize losses due to missed warnings. The definition of a cost function (in particular the F2-score), further allows the identification of an optimal threshold on the disruptivity, taking also into account possible class imbalance in the training data: The optimal value was found to be $F2 = 0.829$, corresponding to disruptivity thresholds between 0.4-0.45. The performance metrics are found to deteriorate slightly with respect to the existing DPRF algorithm, where the F2-score is approximately 0.855, for similar values of thresholds on disruptivity; nevertheless, in both cases these F2 numbers correspond to true positive rates greater than 80%.

Figure 2 shows the receiver operating characteristic (ROC) curve for the validation set. The ROC shows the relative trade-off between correctly detected fraction of flattop disruptions (true positive rate) as a function of the false alarms flagged (false positive rate) as the threshold of the predicted probability used to discriminate between a positively and a negatively labelled sample is varied. The thick blue line in Figure 2 represents the average of the ROCs from the K-fold (with K=5) validation folds. Comparing with the ROC for the predictor from [21], the model used in this work can obtain the same true positive rate as the model with additional inputs, but at a higher false positive rate. This implies that the safe operating region approximated using this model is likely to be conservative. While using overly conservative estimates of the safe operating region could artificially limit plasma performance, having moderately conservative estimates is likely desirable as it will reduce the likelihood of triggering a disruption (or mitigation system) while operating within the identified boundaries.

3. Safe Operating Region Identification (SORI) algorithm

Machine learning algorithms are attractive solutions to the disruption prediction problem because they are flexible enough to classify safe and unsafe operating points despite the complexity of the controllability boundary (arising due to the highly nonlinear and coupled nature of tokamak dynamics). However, for the purpose of real-time optimization of operating points (e.g., model predictive control [9]), it is desirable to have a simply defined boundary, ideally defined by a convex set of linear inequalities, that is accurate for the local operating space.

We therefore consider the problem of finding, for a given point in operating space, a convex set of linear constraints bounding a local region of operating space over which the disruptivity is less than a specified threshold d_{max}. First, input features are separated into two groups: *active* and *context* features. The active group represents the subset of N_{active} inputs features within which the operating point will be actively optimized in real-time. The context features are the remaining $N_{context}$ features, which will not be actively changed for the purposes of disruption avoidance. This separation reduces the computational complexity of the problem by projecting the safe-operating space onto the space defined by the active dimensions. The optimal choice of active dimensions depends on several factors. There are engineering considerations (which dimensions are easiest to adjust in real-time and can be varied a useful amount with the available actuators?), physics considerations (which dimensions have the most influence on the disruptivity at the current operating point?), and operational considerations (which dimensions are the operator or experimental session leader willing to allow the real-time system to adjust away from the pre-programmed target values?). For illustration purposes, the active dimensions in the examples shown in this work were chosen based on former criteria. Future work will explore systematic methods of optimizing the choice.

For the purposes of finding the safe operating space, each of the context features are assumed to remain within a prescribed range $\pm s_{variation,i}$ around the current operating point. This range could be defined by noise levels, estimation/tracking errors, or expected future values, e.g., based on model predictions or trends. Each of the active feature values are constrained within a range $\pm s_{scan,i}$, based on the reachable range of these values or a more conservative range chosen by the machine operator. Indeed, since MPC is a finite-horizon trajectory optimization, there is no need to consider a range of values beyond those that could be realized within the prediction horizon.

The disruptivity is evaluated at a random sample of N_{sample} points within the local region defined above, and each point is labeled as either safe ($d < d_{max}$) or unsafe ($d \geq d_{max}$). The points provide a map of disruptivity. As an illustration, the results of this scan at three times during DIII-D shot 180808 are plotted in Figure 3. For this example, the active dimensions were taken to be plasma current and stored energy. The points are colored by disruptivity value and the plot is shaded based on interpolation between the points (interpolation is not part of the real-time SORI algorithm; it is only used for illustration). An orange line is included indicating the contour of

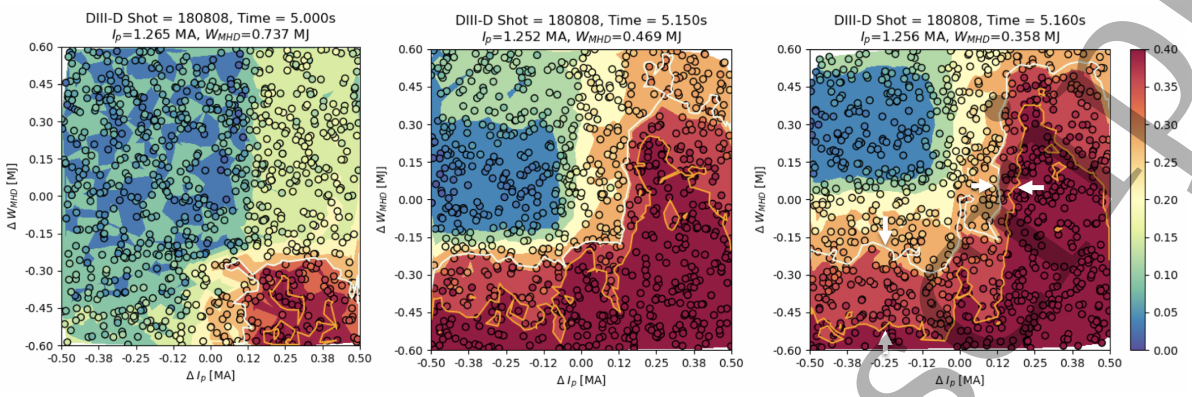

Figure 3: Contours of disruptivity evaluated around the experimental operating point at three times in DIII-D shot 180808. The orange line indicates the contours of $d = d_{max} = 0.4$ while the white line indicates the contours of $d = 0.75 \times d_{max}$

.

$d = d_{max} = 0.4$, while the white line indicates the contour of $d = 0.75 \times d_{max}$. The complexity and non-convexity of these contours motivate identifying an approximate boundary based on linear constraints. Furthermore, these plots illustrate the importance of controlling distance from a disruptive boundary rather than disruptivity itself. For example, in the right plot, two pairs of white arrows illustrate areas where the distance between the $d = 0.75 \times d_{max}$ and $d = d_{max}$ boundaries differ significantly. As a result, disruption avoidance based on control of the calculated disruptivity may not be very robust to measurement and process noise. By controlling distance from the disruptive boundary, rather than the predicted disruptivity value, a control algorithm can ensure that the operating point remains far enough from the boundary to robustly avoid causing a disruption (or the need to trigger a mitigation system).

To define a convex set of linear constraints, each constraint is taken to be the hyperplane perpendicular to a line extending from the current operating point to a point with coordinates (in the space of active dimensions) written as $\bar{a}_i$. Each constraint can be written as

$$\bar{a}_i \bar{x} < \|\bar{a}_i\|_2^2, \tag{2}$$

where $\bar{a}_i \in \mathbb{R}^{1 \times N_{active}}$ and $\bar{x} \in \mathbb{R}^{N_{active} \times 1}$ is a point in operating space in normalized relative (to the reference point) coordinates. In this work, the coordinates were normalized by the range of the constraints placed on the active dimensions, i.e.,

$$\bar{x}_i = (x_i - x_{ref,i})/s_{scan,i} \tag{3}$$

The task then becomes the selection of $N_{constraints}$ points in active feature space such that the convex region defined by the constraints is as large as possible while containing no (or at least very few) disruptive points. This is achieved by maximizing the value function

$$J = N_{safe,inside} - w_{unsafe} N_{unsafe,inside}, \tag{4}$$

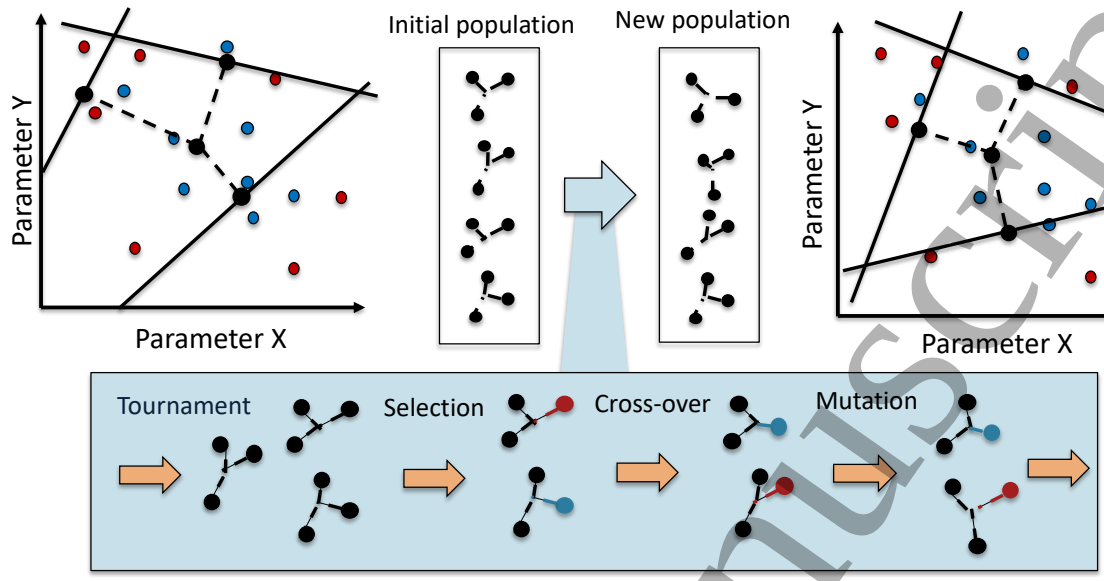

Figure 4: Genetic algorithm for optimizing constraints. A fraction of the initial population is used to produce variations on the fittest individuals, which are passed on to the next generation along with a small number of 'elite' individuals that continue into the next generation unmodified.

where $N_{safe,inside}$ is the number of safe points that satisfy each of the constraints (2), and $N_{unsafe,inside}$ is the number of disruptive points that satisfy the constraints. The first term encourages identification of a region that is as large as possible, while the latter term limits the region's size to exclude unsafe points. A large enough value of the weight w_{unsafe} helps ensure the identified region does not include many unsafe points; a value of 40 was used for the results in this work.

A genetic algorithm, illustrated in Figure 4, is used to solve the optimization problem. An initial population of N_{pop} candidate sets of $N_{constraints}$ constraints is randomly generated and the cost function (4) is evaluated for each. A fraction f_{elite} of individuals with the lowest cost are marked as elite and retained for the next generation. The remainder of the new population is formed by applying a tournament selection algorithm. The selected individuals are then paired, and a crossover function is applied. Specifically, the crossover function loops over each of the $N_{constraints}$ constraints, exchanging the constraint among the pair of individuals with a probability p_{cross}. Finally, each dimension of each constraint has a probability p_{mutate} of being mutated. If chosen for mutation, the dimension is perturbed by a value randomly selected from a normal distribution. The effect of this process is to create a new population of random combinations and variations of the fittest individuals while retaining the elite individual unchanged. The new generation is therefore at least as good as the previous (since the elite individuals are retained), while exploring alternative candidate solutions. This process is repeated for N_{gen} generations, at which point the individual with the highest value is chosen as the approximate solution to the problem constraint identification.

An additional $2 \times N_{active}$ constraints are added to the identified constraint set to represent the boundaries defining the local operating space that was considered by the SORI algorithm (i.e., the box defined ranges $\pm s_{scan,i}$).

3.1. Disruption margin calculation

Once boundaries are identified using the SORI algorithm, the distance of a point from each disruptive boundary can be calculated. The boundary that is closest to the point is considered to be the disruption margin. While the disruption predictor described in Section 2 provides a scalar likelihood of disruption that can be used to trigger a transition to a safe device shutdown procedure, the disruption margin indicates the distance of the point from disruption boundaries in terms of the dimensions of the operating space. Furthermore, the direction normal to the closest boundary represents the most effective direction to move away from the boundary.

Since each of the active dimensions have different physical scales, it is necessary to scale each dimension by a metric to make a physically meaningful disruption margin calculation. Suitable metrics may include the standard deviation of measurement noise or tracking errors. We define a scaling matrix T for the active dimensions, with diagonal elements defined as

$$T_{i,i} = s_{scan,i}/s_{variation,i}, \forall i \in N_{active} \tag{5}$$

For the i-th constraint, the normalized disruption margin can then be calculated as

$$p_i = \frac{||\bar{a}_i||_2^2 - \bar{a}_i \bar{x}}{||\bar{a}_i T||_2} \tag{6}$$

4. Algorithm implementation on a GPU

Due to the complexity of tokamak dynamics, control is often developed hierarchically, with dedicated fast control loops devoted to active control of unstable modes, like $n = 0$ vertical instability or resistive wall modes, or dedicated loops relating an actuator to a specific physics quantity, e.g., gas valve feedback for density control. Higher level control loops aim at achieving particular scenario characteristics. For example, the profile control algorithm on DIII-D [27] manipulates the target density and plasma current, among other actuators, to achieve desired profile evolution. These types of algorithms have slower cycle times than the lower level control algorithms due to the relatively slow time scales of energy confinement and current diffusion. Since the constraints identified by the SORI algorithm will be used to optimize the targets of these higher level control loops, the SORI algorithm should have a similar cycle time. Since the profile control algorithm on DIII-D typically runs with a 10-20ms cycle time, we take this as a target cycle time for the SORI algorithm.

In order to meet this target cycle time, it was found to be necessary to implement the SORI algorithm on a graphics processing unit (GPU). The GPU implementation, written in CUDA (NVIDIA's Compute Unified Device Architecture), allows massive

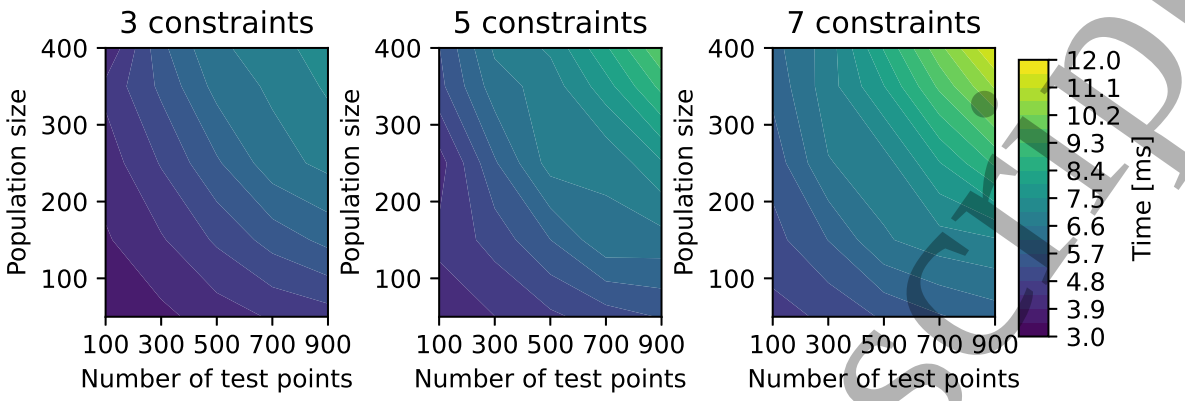

Figure 5: Timing of algorithm for various parameters. The choice of 3 constraints provided satisfactory representation of the safe operating space for shots studied, and choosing a population size of 400 and 900 test points provided reliable optimization results within 10ms.

parallelization of calculations, which can be exploited in this case to greatly accelerate the execution time of the algorithm over a serial implementation on a CPU. There are multiple levels of parallelization that are exploited for acceleration: for the disruptivity scan, each point can be evaluated in parallel, and for each point, each tree in the random forest can be evaluated in parallel. For the first step of the genetic algorithm, the value function for each individual can be evaluated in parallel. For each evaluation of the value, violation of each constraint for each test point can be checked in parallel. Each individual in the new generation can also be generated in parallel (tournament selection, cross-over, mutation).

The acceleration is highly dependent on the specific parameters chosen for the algorithm and GPU hardware used. Figure 5 shows the total time for identifying the optimal constraints for a range of parameters. The scan was performed with 2 active dimensions, 25 generations, 4 individuals per tournament, and 10 elite individuals carried over from generation to generation. The number of test points was scanned between 100 and 900. Increasing the number of test points enables a more complete sample of the context dimensions, improving the accuracy of the results. The population size was varied from 50 to 400. Increasing the population size improves the genetic optimization by enabling a wider variety of individuals to be evaluated in each generation. For the results shown in the following sections the choice of 3 constraints provided satisfactory representation of the safe operating space and choosing a population size of 400, 900 test points, and 25 generations provided reliable optimization results within a target evaluation time of 10ms. Figure 5 shows that more complex representations using more constraints can be identified with similar settings within 12ms, however, this was not found to be necessary for the shots studied.

5. Constrained optimization of target variables

As an initial step towards a more sophisticated model predictive control strategy to actively optimize the plasma state while avoiding the identified boundaries, we consider the problem of modifying target values such that they remain close to a target operating point and within the safe operating space.

An interior-point optimization algorithm is used to find the solution to the constrained minimization problem:

$$\min_{\bar{x}} \quad \frac{1}{2}(\bar{x} - \bar{x}_{ref})^t Q(\bar{x} - \bar{x}_{ref}) + w_p P(\bar{x}) \tag{7}$$

where $\bar{x}_{ref} \in \mathbb{R}^{N_{active}}$ is the normalized operator defined target, Q is a positive semi-definite weight matrix used to adjust relative importance of each target value. The term P is weighted by scalar w_p and used to penalize solutions that do not maintain a normalized disruption margin larger than a selected minimum, m, i.e., solutions violating the constraint $d_i > m$. The term is defined as

$$P(\bar{x}) = \sum_{i}^{N_{constraints}+2\times N_{active}} P_i(\bar{x}) \tag{8}$$

where

$$P_i(\bar{x}) = \min(0, ||\bar{a}_i||_2^2 - \bar{a}\bar{x} - m||\bar{a}_i T||_2)^2 \tag{9}$$

The interior-point method iteratively arrives at an approximate solution $\bar{x}_{opt}$ starting from an initial guess $\bar{x}_0$ by calculating the search direction from the gradient of the cost function. A line search is done to find the best solution (lowest cost) along that direction. The value of $\bar{x}_{opt}$ is updated to the best solution (if it is an improvement over the current iterate). The steps are repeated with an increased weight on the penalty term until a satisfactory solution is found.

6. Simulation of SORI algorithm

In this section, the disruption predictor and SORI algorithm are demonstrated using historical data from DIII-D. The input variation ranges considered for the simulation results are shown in Table 2. For the purposes of demonstration, the values were manually selected by observing typical variations in the inputs around their moving average. Figure 6 shows the time history of normalized disruptivity (as predicted by the random forest) and normalized disruption margin (as identified by the SORI algorithm using stored energy and plasma current as the active dimensions) for DIII-D shot 180808, which disrupted at around $t = 5.18$s as a result of impurity accumulation and eventual mode locking. In generating the results the stored energy and plasma current were scanned over the ranges ± 0.6MJ and ± 0.5MA, respectively. The values in Table 2 corresponding to these variables were used as $s_{variation,i}$ in the calculation of normalized disruption margin (6). In Figure 6, the disruptivity remains roughly constant at a very low value until around $t = 4.9$s, then increases slightly before a sharp increase at $t = 5.1$s.

Table 2: Variation ranges used in simulations. These values are also used as $s_{variation}$ in calculating normalized disruption margin.

Signal	Variation range
Electron density, n_e [m^{-3}]	4×10^{18}
Plasma current, I_p [A]	2×10^{4}
Squareness, ζ	0.0025
Plasma minor radius, a	0.005
Normalised internal inductance, ℓ_i	0.05
Stored plasma energy, W_{MHD} [J]	2×10^{4}
Elongation, κ	0.005
Triangularity, δ	0.01

The threshold for triggering a disruption warning ($d_{max} = 0.4$) is reached by $t = 5.2$s. The normalized disruption margin begins to decrease slowly at $t = 4.8$s and drops rapidly at $t = 5.1$s. Since the disruption margin is normalized based on the expected variance of the active dimensions, values less than 1 are indicative of particularly dangerous operating points. The operating point remains a safe distance from disruption until around 4.5s, at which point the distance from the disruptive boundary decreases rapidly, eventually dropping well below 1.

The change in disruption margin prior to reaching the disruption warning threshold is illustrated in Figure 7. This figure shows, for three times near the end of the discharge, the points at which the random forest was evaluated (blue markers are non-disruptive, red markers are disruptive), and the identified constraints (black lines). The identified safe region is shaded in blue. The current operating point is centered at (0, 0) for each time, and the axes represent deviations from the current operating point. The plots show that, for each time, the constraints provide a good approximation of the largest convex safe region. It is evident that, as the shot progresses, the current operating point moves closer to the unsafe region. The location of the identified boundaries change over time as a result of changes in both the active and the context dimensions.

6.1. Constrained optimization of target variables

The optimization approach described in Section 5 is demonstrated in this section by applying it to the disruptive shot 183246. In this case, the active dimensions were taken as stored energy and electron density, with the scan ranges set to ±0.6MJ and $\pm0.4 \times 10^{19}\mathrm{m}^{-3}$, respectively. The achieved values of W_{mhd} and n_e are taken as the pre-programmed target values, and the optimization approach is used to calculate modifications to these targets to remain, if possible, within the safety margins of the identified operating region. The safety margin is taken to be $m = 5$. The weight matrix was set to $Q = 0.01\mathbb{I}^{N_{active}\times N_{active}}$. $\bar{x}_{ref} = 0$ is chosen, which encourages solutions to

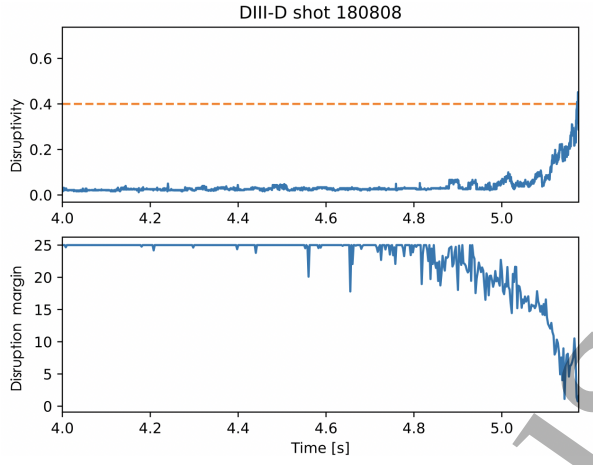

Figure 6: Disruptivity and normalized disruption margin (in stored energy - plasma current space; blue curves) for DIII-D shot 180808. Dashed red line indicates reference threshold of disruptivity = 0.4.

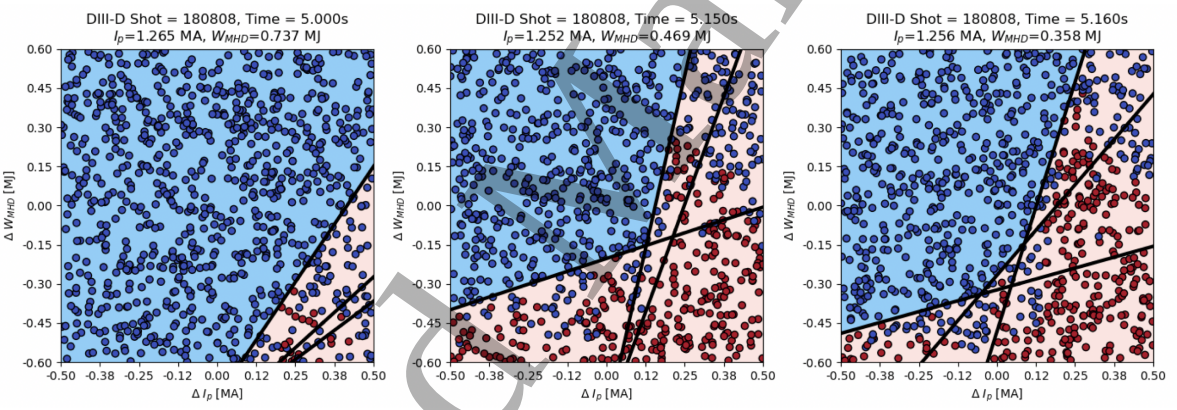

Figure 7: Identified safe operating regions (in stored energy - plasma current space) for DIII-D shot 180808. Points at which the random forest was evaluated are shown (blue markers are non-disruptive, red markers are disruptive), along with the identified constraints (black lines). The identified safe region is shaded in blue, while the unsafe region is shaded red.

remain close to the current experimental operating point at each sample.

Results are shown in Figure 8. The disruptivity, shown in the top left, increases slowly beginning around $t = 4.2$s, eventually reaching the disruption warning threshold around $t = 4.85$s. In the bottom left, the normalized disruption margin of the experimental input values is compared to the disruption margin for the optimized target values. The disruption margin begins to decrease at $t = 4.4$s and it is evident that the disruption margin of the optimized targets remains at 5 (the selected safety margin) after $t = 4.55$s, while the experimental value continues down to 0. The optimized and experimental stored energy and density are shown in the plots on the right. After a significant decrease in density and stored energy around $t = 4.3$s, the density slowly

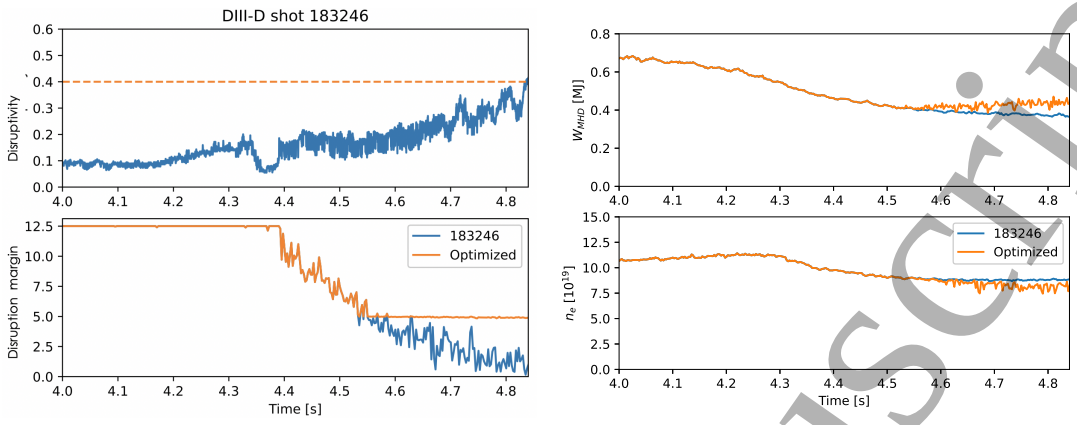

Figure 8: (left) Disruptivity and normalized disruption margin (in stored energy - electron density space) for DIII-D shot 183246. (right) Optimized targets compared to experimental values for DIII-D shot 183246.

decreases to a minimum before slowly increasing, while the stored energy continues to slowly decrease until the disruption warning threshold is reached. Though not shown, this trend continues after the disruption warning - stored energy continues to decrease while density increases until a disruption occurs at around $t = 5.2$s. The disruption margin optimization algorithm initially proposes little to no change to the optimized target values, but after $t = 4.55$s, the optimized stored energy is increased and density slightly decreased to maintain the selected safety margin.

Figure 9 shows the identified safe operating region at three times near the end of the discharge. The disruptive region is shaded red, the marginal region (distance to disruption less than $m = 5$) is shaded off-white, and the safe region is shaded blue. The optimized target values are indicated by white stars. As the experimental operating point (the point $(0, 0)$ on the plots) moves closer to the disruptive boundary, the optimized point is moved to higher stored energy and lower density values to maintain the prescribed safety margin.

7. Discussion and Future work

An algorithm for identifying a set of convex linear constraints approximating the safe, disruption-free region around an operating point of a tokamak plasma has been presented. Implementation of the algorithm on a GPU enables real-time relevant execution times. An approach to optimizing an operating point within the identified constraints is proposed. Simulations of the algorithm using data from DIII-D discharges demonstrates the ability to identify safe regions and optimize target operating points to maintain a safe distance from the disruptive boundary.

The approach is aimed at enabling active disruption avoidance by providing a direct connection between disruption probability predictors and real-time control algorithms capable of respecting state constraints (e.g., model predictive control). In future work,

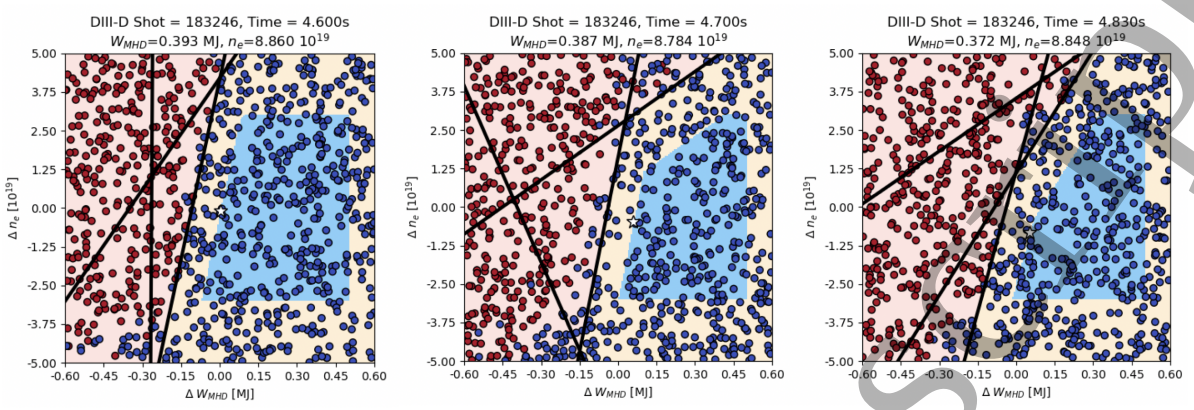

Figure 9: Identified safe operating regions (in stored energy - electron density space) for DIII-D shot 183246.

the identification and optimization algorithms proposed in this work will be implemented and tested in real-time. The success of the active disruption avoidance approach will depend on the accuracy of the disruption predictor and the reliability of the control algorithms that track the target plasma parameters. A key to the success of the approach will be ensuring that the inputs to the disruption predictor are sufficiently independent. If there is strong correlation between the inputs, then active changes made to a subset of inputs may change other inputs enough to lead to significant changes in the disruptivity boundaries. In some situations, this correlation between optimized targets and disruption boundaries could lead to a collapse of the safe operating space.

The implementation of the algorithm on a GPU exploited the parallelizability of many of the steps of the algorithm to achieve execution times of $<$12ms for the demonstrated settings. However, the use of more complex predictors (e.g., deep recurrent neural networks), the use of more active variables, or the use of more sampled points and/or more identified constraints (for higher resolution determination of boundaries) could increase execution times beyond the requirements for real-time. Aside from exploring hardware improvements and code profiling/optimization, algorithm improvements will be considered, e.g., dimensionality reduction techniques, optimization of the choice of active dimensions, or improving upon the naive uniform sampling of the operating space used in this work.

The proposed approach for identifying a set of convex constraints is general and not tied specifically to disruption events. In future work, the approach will be combined with predictors for other events, like confinement mode transitions, or destabilization of non-asymmetric modes. This will enable active avoidance, or triggering, of specific plasma events.

Acknowledgements

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 Award(s) DE-AC02-09CH11466, DE-SC0014264, and DE-FC02-04ER54698. Part of the data analysis reported in this paper was performed using the OMFIT integrated modelling framework [28].

References

[1] OU, Y. et al., Fusion Engineering and Design **82** (2007) 1153.
[2] FELICI, F. et al., Plasma Physics and Controlled Fusion **54** (2012) 025002.
[3] BOYER, M. D. et al., Nuclear Fusion **59** (2019).
[4] MENEGHINI, O. et al., Nuclear Fusion **57** (2017) 086034.
[5] CITRIN, J. et al., Nuclear Fusion **55** (2015) 092001.
[6] BOYER, M. D. et al., IEEE Transactions on Control Systems Technology **22** (2014) 1725.
[7] BOYER, M. D. et al., Plasma Physics and Controlled Fusion **55** (2013) 105007.
[8] BARTON, J. E. et al., Nuclear Fusion **52** (2012) 123018.
[9] MACIEJOWSKI, J. M., *Predictive control: with constraints*, Pearson Education, 2002.
[10] Wang, Y. et al., IEEE Transactions on Control Systems Technology **18** (2010) 267.
[11] WEHNER, W. et al., Proceedings of the American Control Conference (2017) 4872.
[12] ILHAN, Z. O. et al., Model predictive control with integral action for the rotational transform profile tracking in NSTX-U, in *2016 IEEE Conference on Control Applications (CCA)*, pp. 623–628, 2016.
[13] MALJAARS, E. et al., Nuclear Fusion **55** (2015).
[14] BERKERY, J. W. et al., Physics of Plasmas **24** (2017) 056103.
[15] STRAIT, E. et al., Nuclear Fusion **59** (2019) 112012.
[16] WINDSOR, C. G. et al., Nuclear Fusion **45** (2005) 337.
[17] MURARI, A. et al., Nuclear Fusion **49** (2009) 055028.
[18] CANNAS, B. et al., Plasma Physics and Controlled Fusion **57** (2015) 125003.
[19] KATES-HARBECK, J. et al., Nature **568** (2019) 526.
[20] MONTES, K. et al., Nuclear Fusion **59** (2019) 096015.
[21] REA, C. et al., Nuclear Fusion **59** (2019) 096016.
[22] WROBLEWSKI, D. et al., Nuclear Fusion **37** (1997) 725.
[23] SAMMULI, B. et al., Fusion Engineering and Design **169** (2021) 112492.
[24] BREIMAN, L., Machine Learning **45** (2001) 5.

[25] REA, C. et al., Fusion Science and Technology **76** (2020) 912.
[26] De Vries, P. C. et al., Nucl. Fusion **51** (2011) 53018.
[27] SCHUSTER, E. et al., Nuclear Fusion **57** (2017) 116026.
[28] MENEGHINI, O. et al., Nucl. Fusion **55** (2015) 083008.