Control of the tokamak safety factor profile with time-varying constraints using MPC

E. Maljaars[1], **F. Felici**[1], **M.R. de Baar**[1,2], **J. van Dongen**[2], **G.M.D. Hogeweij**[2], **P.J.M. Geelen**[1], **M. Steinbuch**[1]

[1] Eindhoven University of Technology, Mechanical Engineering, Control Systems Technology,
P.O. Box 513, 5600MB Eindhoven, The Netherlands
2 FOM Institute DIFFER, P.O. Box 1207, 3430BE Nieuwegein, The Netherlands

E-mail: `e.maljaars@tue.nl`

Abstract. A controller is designed for the tokamak safety factor profile that takes real-time-varying operational and physics limits into account. This so-called *model predictive controller* (MPC) employs a prediction model in order to compute optimal control inputs that satisfy the given limits. The use of linearized models around a reference trajectory results in a quadratic programming problem that can easily be solved online. The performance of the controller is analyzed in a set of ITER L-mode scenarios simulated with the nonlinear plasma transport code RAPTOR. It is shown that the controller can reduce the tracking error due to an overestimation or underestimation of the modeled transport, while making a trade-off between residual error and amount of controller action. It is also shown that the controller can account for a sudden decrease in the available actuator power, while providing warnings ahead of time about expected violations of operational and physics limits. This controller can be extended and implemented in existing tokamaks in the near future.

1. Introduction

Successful high performance tokamak operation simultaneously requires plasma stability at high values of the normalized pressure β_N as well as long energy confinement times τ_E. Both confinement and stability are associated with the distribution of the current density in the plasma, equivalent to the safety factor profile $q(\rho)$. Achieving and maintaining a reference q-profile during a tokamak discharge in the presence of disturbances and uncertainties is one of the prospective main areas of research in currently operational tokamaks, ITER, and in other future long pulse devices such as JT60SA and Tore Supra WEST [1, 2, 3, 4].

In present-day practice, actuator trajectories to achieve approximately the desired q-profile are chosen by the tokamak operator based on extensive experience. Those actuator trajectories are provided in open-loop to the actuators (for instance the time traces of auxiliary heating and current drive and the desired plasma current). Recently, model-based optimization of the actuator trajectories has been proposed to achieve a target q-profile at the beginning of the flat-top phase [5, 6, 7]. In practice, the desired q-profile cannot be obtained or maintained without feedback control, due to the presence of model mismatches and disturbances.

Feedback controllers for the q-profile can either be designed using first principle models or using models obtained by system identification procedures. The models obtained by system identification (such as applied in [8]) are valid in the vicinity of a single operating point only. First principle models can be used in essence in an arbitrary range of operating points, including the ramp-up and ramp-down phase [9, 10, 11].

In the literature, a large number of first principle model based feedback control approaches to control the q-profile are proposed of which we mention here the most recent [12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22]. Most of these contributions take fixed actuator constraints passively into account, meaning that they are not included in the controller design, but a posteriori imposed by the use of e.g. an anti-windup loop. Handling of actuator constraints in the q-profile controller design itself is done in [13, 14, 16], where actuator constraints are fixed.

Dynamic actuator sharing is required for the simultaneous control of magnetic and kinetic profiles/variables, Neoclassical Tearing Modes (NTMs) etc. using a common set of actuators [23]. A common example is electron cyclotron heating and current drive, that can be used for many of these tasks [24]. A supervisory controller may optimize the allocation of actuators to simultaneously satisfy a number of control and plant protection tasks. This implies that the available actuator power for profile control (and the distribution thereof) varies in time and hence the controller should take those real-time-varying limits actively into account.

Besides actuator limits, it would be desirable for a controller to ensure simultaneously that physics operational limits (e.g. limits on β_N and $l_i^{(3)}$) and desired physics limits (e.g. $q(\rho) > 1$ in the case of hybrid scenarios) are satisfied, as argued in e.g. [25]. Active handling of both actuator and plasma physics constraints simultaneously is not yet reported for the control of the q-profile in tokamak plasmas and is shown in this paper for the first time using a model predictive controller.

Model predictive control (MPC) is a general optimal control method which uses a predictive model to compute the control action and can deal routinely with actuator and state constraints [26, 27, 28]. MPC has already been proposed for q-profile control in [13, 14]. The approach used in these references resulted in a non-linear optimization problem (which is computationally demanding). Moreover, the used models contained only the magnetic flux evolution equation and only static actuator constraints were considered.

The MPC-controller in this work uses locally linearized models at each time step around a reference trajectory to minimize the tracking error in the presence of real-time-varying actuator and plasma physics constraints, model uncertainties and disturbances. The controller uses a quadratic optimization which is computationally less demanding and the locally linearized models contain the dynamic response of the poloidal magnetic flux and electron temperature.

The proposed feedback controller is designed to improve normal well-prepared operation of currently operational tokamaks by complementing feedforward actuator trajectories with feedback control around a predefined reference trajectory. The profile controller is intended to work under normal plasma behavior with modest uncertainty and when very significant and unexpected

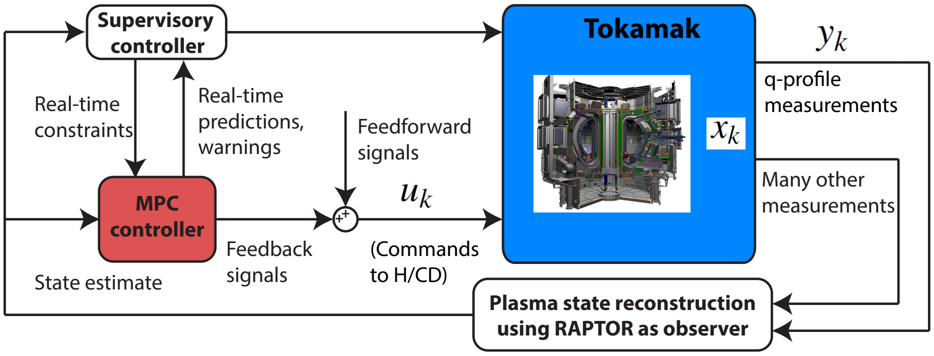

Figure 1. Envisioned implementation of the MPC-controller in a tokamak. A supervisory controller receives real-time predictions and warnings of the MPC-controller. It provides the MPC-controller with real-time constraints. The MPC-controller calculates the optimal feedback signal which is provided with the feedforward signal to the actuators in the tokamak. The plasma state is reconstructed from the available measurements using a state observer and fed to the MPC-controller. A state observer based on RAPTOR is presented in [29].

changes occur in the plasma behavior, e.g. high confinement loss or large impurity accumulation etc., than the priority is not in controlling the profiles but the supervisory system will switch to exception handling. Therefore the controller has not been tested for such off-normal situations.

Simulations using the nonlinear plasma transport simulator RAPTOR [7] show the effectiveness of this approach. An L-mode plasma with ITER-like parameters and a low shear q-profile is simulated during the ramp-up and flat-top phase with the desired plasma current I_p, NBI and ECCD as actuators. We show effective control of the q-profile in the presence of model mismatches and time-varying actuator and plasma physics constraints. It is shown that real-time prediction of the q-profile evolution enables early warnings of plasma physics constraint violations. While the most important dynamics and underlying nonlinearities from the 1D transport physics are taken into account, the method is fast enough to be implemented on the timescale of currently operational tokamaks.

This paper is organized as follows. In section 2 we present the methodology of the MPC-controller design. The simulation setup is presented in section 3. In section 4, we show performance and merits of this MPC-controller for a number of typical cases. In section 5 we discuss the results and possible extensions of the current MPC-design. The conclusion is provided in section 6.

2. Methodology

In this section we describe the MPC-controller design. First we introduce the context in which the controller will function and then the RAPTOR code, from which we obtain local linearized models used for the controller design, is briefly explained. The reader will subsequently be introduced to the concept of MPC and to the details of the control design up to the implementation in an algorithm.

2.1. Controller context

The envisioned implementation of the MPC-controller in the larger context of a tokamak plasma control system is given in Figure 1. A supervisory controller provides real-time constraints to the MPC-controller. With these constraints the MPC-controller calculates the optimal feedback action which is provided in combination with the feedforward signal to the actuators in the tokamak. From the available measurements the plasma state is reconstructed (either by a real-time observer [29] or by constrained equilibrium reconstruction) and fed to the MPC-controller.

To systematically develop the MPC-controller towards the envisioned implementation, the

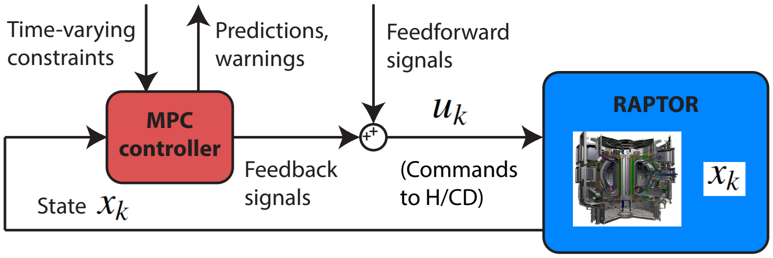

Figure 2. Simplified implementation as used in this work. The simulated plasma state is directly available to the MPC-controller.

MPC-controller is as a first step interfaced with the nonlinear plasma transport simulator RAPTOR. This is illustrated in Figure 2. This allows for a evaluation of the proposed control method in the nonlinear plasma transport simulation of RAPTOR. The MPC-controller is a state feedback controller [27], requiring access to the actual state of the plasma, in this case, full knowledge of the plasma profiles. In this work we have direct access to the state of RAPTOR, which simplifies the implementation. The supervisory controller is not considered in this paper, instead time-varing constraints are manually provided to the MPC-controller.

2.2. RAPTOR: nonlinear plasma transport simulator

In this work we employ linearized models from RAPTOR inside the controller and use RAPTOR as simulator to test the controller. RAPTOR [7, 11] is a control-oriented, physics-based 1D transport code that solves the simplified non-linear coupled transport of the electron temperature T_e and the poloidal magnetic flux ψ as a function of the normalized square-root toroidal flux ρ, represented by partial differential equations (PDEs) [30]. Bootstrap current and neoclassical conductivity are calculated using the Sauter-Angioni equations [31, 32]. Sources and sinks of thermal energy are modeled, including Ohmic heating, and simple models for Electron Cyclotron (EC) and Neutral Beam (NB) heating and current drive. Losses from Bremsstrahlung, line radiation, and electron-ion heat exchange are included. The electron thermal diffusivity is computed using an empirical model, in this case the Bohm-gyroBohm transport model [33]. Impurities are not considered other than their effect via the Z_{eff} profile, which is also prescribed. Another important assumption in RAPTOR is that the geometric terms in the transport equations, i.e. those terms that depend on the flux surface geometry, are chosen for one particular equilibrium and kept fixed thereafter. The total plasma current is imposed as boundary condition for the poloidal flux equation, while the electron temperature at the boundary is prescribed. In [34] is shown that by careful choices of what to exclude, RAPTOR is able to approach CRONOS [35] simulations of ITER in [36] within $\sim 15\%$ or better, with < 3ms per time step.

Actuators considered in this work are the inductive and non-inductive heating and current drives (H&CD). The controlled variables are the inverse q-profile ($\iota = 1/q$) at several locations ρ in the plasma.A state space description of the 1D plasma transport is used in which the internal state variables describe the electron temperature and the magnetic flux profiles at a particular moment in time. By discretization in both space and time, the following input, state and controlled variables vectors can be introduced (as used in the simulations in this work):

$$\underbrace{u_k = \begin{bmatrix} I_p(t_k) \\ P_{ec,1}(t_k) \\ P_{ec,2}(t_k) \\ P_{ec,3}(t_k) \\ P_{ec,4}(t_k) \\ P_{nbi}(t_k) \end{bmatrix}}_{\substack{\text{input vector} \\ u_k \in \mathbb{R}^{n_u}}}, \qquad \underbrace{x_k = \begin{bmatrix} \hat{T}_{e\rho=0}(t_k) \\ \vdots \\ \hat{T}_{e\rho=1}(t_k) \\ \hline \hat{\psi}_{\rho=0}(t_k) \\ \vdots \\ \hat{\psi}_{\rho=1}(t_k) \end{bmatrix}}_{\substack{\text{state vector} \\ x_k \in \mathbb{R}^{n_x}}}, \qquad \underbrace{z_k = \begin{bmatrix} \iota_{\rho=0}(t_k) \\ \vdots \\ \iota_{\rho=1}(t_k) \end{bmatrix}}_{\substack{\text{controlled variables vector} \\ z_k \in \mathbb{R}^{n_z}}} .$$

The hats in the state vector denote that these are the finite element basis function coefficients instead of the actual values of T_e and ψ [11]. In the simulations we used $n_u = 6$, $n_x = 64$, $n_z = 16$, although $n_x = 44$ would have been sufficient. Note that we calculate the internal state variables at a more fine ρ-grid than the controlled variables. This allows for more accurate solving of the transport while limiting the number of controlled variables. Using the discretization in both space and time the PDEs are formulated into a nonlinear state update equation used to yield the next state x_{k+1} from knowledge of the current state x_k and actuator commands u_k [7]:

$$\tilde{f}_k \equiv \tilde{f}(x_{k+1}, x_k, u_k) = 0. \tag{1}$$

This equation is solved in RAPTOR for each time instance t_k. The controlled variables are related to the state by the following controlled variable equation:

$$z_k = h(x_k), \tag{2}$$

which may provide many different controlled variables as a function of the state (e.g. inverse q-profile, β_N, stored energy, magnetic shear). In this work we restrict ourselves to the inverse q-profile at different locations.

2.3. Model Predictive Control

The principle of MPC is illustrated and explained in Figure 3. MPC solves at each moment in time an optimization problem to find the future actuator inputs up to a prediction horizon. This optimization problem involves a cost function (that represents e.g. the future tracking error) and constraints on actuators and states. MPC uses a process model of the dynamics to predict the future process behavior. Only the computed actuator commands for the next time instant are implemented and the controller solves on the next step a new optimization problem with the updated state.

An MPC-controller requires a model that relates the states and controlled variables at the next time instant to the current state and future actuator commands. The type of model required depends on which formulation of MPC that will be used. Nonlinear MPC is available [37], but has a number of drawbacks with respect to linear MPC. Linear MPC is more established, and the use of a linear model in combination with a quadratic cost function and linear constraints results in a Quadratic Programming (QP) problem, while nonlinear MPC requires the optimization of a nonlinear optimization problem which is computationally more demanding. Moreover, the solution of a QP-problem is a global constrained minimum whereas nonlinear optimization problems may have local minima.

The controller should run about 5 times faster than the fastest process time scale that dominates the influence of inputs on the controlled variable evolution or the evolution of constrained quantities. For profile control that includes also kinetic quantities, this is the energy confinement time, which in medium-sized tokamaks such as ASDEX-Upgrade is ∼ 50 ms. This implies that for implementation on a device such as ASDEX-Upgrade, the computational time of the MPC-controller is limited to approximately 10 ms. It is therefore necessary to reduce the online computational cost as much as possible, and this motivates the use of linear MPC techniques and hence the formulation of linear models.

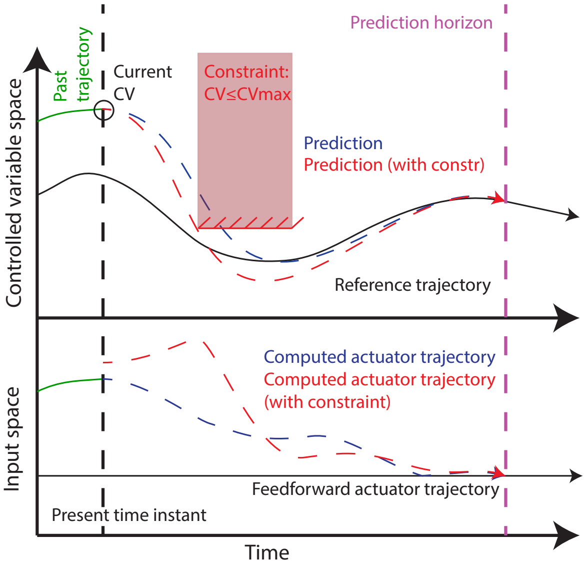

Figure 3. Illustration of MPC with single input and single controlled variable. Top: controlled variable trajectories. Bottom: actuator trajectories. Feedforward actuator trajectory and its resulting nominal CV trajectory (that functions as reference) are given in black. Past trajectories of actuator input and controlled variable are given in green. MPC-controller predicts until a given prediction horizon (magenta) what actuator trajectory (blue, bottom) is required in order to bring the controlled variable back to the reference trajectory (blue, top). In case of the presence of a controlled variable constraint, the MPC-controller computes a future actuator trajectory (red, bottom panel) to handle this constraint (red, top panel).

We benefit from the fact that tokamak discharges follow a predefined sequence of current ramp-up, flat-top and current ramp-down with the corresponding predefined auxiliary actuator trajectories. These actuator trajectories will nominally (i.e. in the absence of disturbances and model mismatches) result in a particular feedforward state evolution. We therefore assume that profiles in RAPTOR also evolve nominally along precalculated trajectories $(u_k^o, x_k^o, z_k^o, \forall k)$. An actuator trajectory with its corresponding state and controlled variable trajectory can thus be specified as a nominal trajectory, where the nominal controlled variable trajectory may function as time-varying reference. The nonlinear dynamics can then be linearized at each time instant t_k resulting in a sequence of linear time-varying models that can be used in a linear MPC-controller to track this reference trajectory.

Hereafter we will derive the linearizations, define the prediction model, the cost function and constraints and apply some strategies to reduce the computational cost.

2.4. Linearizations around trajectory

Linearizations around the nominal trajectory are obtained offline by defining an infinitesimally small perturbation in the state $\tilde{x}_k = x_k - x_k^o$ and input $\tilde{u}_k = u_k - u_k^o$. The dynamics of $\tilde{x}_k$ can then be derived using the Taylor expansion of (1):

$$0 = \tilde{f}(x_{k+1}^o, x_k^o, u_k^o) = \frac{\partial \tilde{f}}{\partial x_{k+1}}\tilde{x}_{k+1} + \frac{\partial \tilde{f}}{\partial x_k}\tilde{x}_k + \frac{\partial \tilde{f}}{\partial u_k}\tilde{u}_k. \quad (3)$$

A so-called linear time-varying (LTV) state space model [38] can now be derived by solving (3) for $\tilde{x}_{k+1}$ and linearizing the controlled variables equation (2):

$$\tilde{x}_{k+1} = A_k\tilde{x}_k + B_k\tilde{u}_k, \quad (4)$$

$$\tilde{z}_k \quad = C_k\tilde{x}_k + D_k\tilde{u}_k, \quad (5)$$

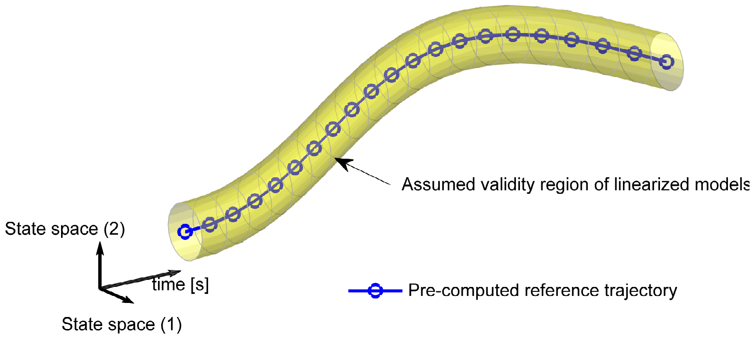
Assumed validity region of linearized models
State space (2)
time [s]
State space (1)
Pre-computed reference trajectory

Figure 4. Illustration of using linearizations around precalculated trajectory in state space. At each moment in time (denoted by dots) a linearized state space model is derived. The yellow contour around the precalculated trajectory illustrates that each linearized state space model has a limited validity region (each indicated by circle). Together, linearized models with their validity region allow for an accurate description of the dynamics around a nominal trajectory.

with the state space matrices defined as:

$$A_k = \left(\frac{\partial \tilde{f}}{\partial x_{k+1}}\right)^{\text{-1}} \frac{\partial \tilde{f}}{\partial x_k}, \quad B_k = \left(\frac{\partial \tilde{f}}{\partial x_{k+1}}\right)^{\text{-1}} \frac{\partial \tilde{f}}{\partial u_k}, C_k = \frac{\partial h}{\partial x_k}, D_k = \frac{\partial h}{\partial u_k} = 0.$$

RAPTOR provides the underlying Jacobians at each time step around the nominal trajectory. Practically, the Jacobian $(\partial \tilde{f}/\partial x_{k+1})$ is invertible for all physically relevant values of the state x_k.

Figure 4 illustrates the linearizations around a predefined trajectory and shows how the linearized models allow for an accurate description of the dynamics around the nominal trajectory within a validity region. The applicability of this approach will be verified in section 3.4.

2.5. Prediction model

The MPC-controller uses a prediction model to relate the future states and controlled variables to the current state and future actuator commands. We can define a prediction model to predict N steps ahead, where N is called the prediction horizon. First we introduce the following stacked vectors:

$$\underbrace{\tilde{U}_{k,N} = \begin{bmatrix} \tilde{u}_k \\ \tilde{u}_{k+1} \\ \vdots \\ \tilde{u}_{k+N} \end{bmatrix}}_{\text{Future inputs}}, \quad \underbrace{\tilde{X}_{k+1,N} = \begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{x}_{k+2} \\ \vdots \\ \tilde{x}_{k+N} \end{bmatrix}}_{\text{Future states}}, \quad \underbrace{\tilde{Z}_{k+1,N} = \begin{bmatrix} \tilde{z}_{k+1} \\ \tilde{z}_{k+2} \\ \vdots \\ \tilde{z}_{k+N} \end{bmatrix}}_{\text{Future controlled variables}}.$$

Using the same notation we can express:

$$\begin{aligned} U_{k,N} &= U^0_{k,N} + \tilde{U}_{k,N}, \\ X_{k+1,N} &= X^0_{k+1,N} + \tilde{X}_{k+1,N}, \\ Z_{k+1,N} &= Z^0_{k+1,N} + \tilde{Z}_{k+1,N}. \end{aligned} \tag{6}$$

By using the obtained linearized state space models at each moment in time, we can write the future state deviations $\tilde{X}_{k+1,N}$ and future controlled variable deviations $\tilde{Z}_{k+1,N}$ as:

$$\begin{aligned} \tilde{X}_{k+1,N} &= \Gamma_{A_{k,N}} \tilde{x}_k + \Gamma_{B_{k,N}} \tilde{U}_{k,N}, \\ \tilde{Z}_{k+1,N} &= \Gamma_{C_{k,N}} \tilde{x}_k + \Gamma_{D_{k,N}} \tilde{U}_{k,N}. \end{aligned} \tag{7}$$

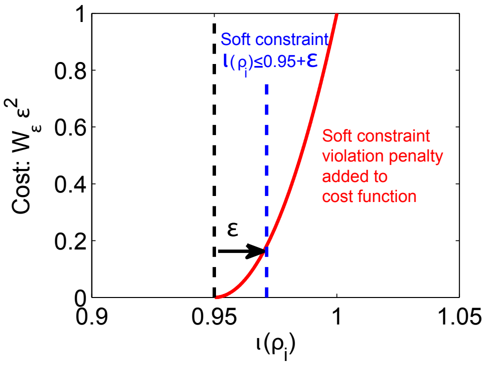

Figure 5. Illustration of using a soft state constraint $\iota(\rho) \le 0.95 + \varepsilon$ instead of a hard state constraint $\iota(\rho) < 1$. Exceeding $\iota(\rho) = 0.95$ is allowed but the term $W_\varepsilon \varepsilon^2$ penalizes entering this soft constraint region.

The matrices $\Gamma_{A_{k,N}}$, $\Gamma_{B_{k,N}}$, $\Gamma_{C_{k,N}}$ and $\Gamma_{D_{k,N}}$ can be constructed from N different linearized models from RAPTOR and are given in Appendix A.1. The matrices are computed offline to reduce the online computational cost. The reader is referred to [39] for a full derivation of this prediction model. In practice, the time interval between predictions of future states and controlled variables is increased by deleting some of the rows of the prediction matrices (see section 2.8).

2.6. Actuator and state constraints

Constraints can be imposed on the individual actuators (for example the desired plasma current I_p^{des} is both constrained in amplitude and in ramp-rate) and on combinations of actuators (for example the available EC-power to the different beams). As shown in Appendix A.2.1, all the actuator constraints can be formulated as time-varying linear inequality constraints on the future input deviations $\tilde{U}_{k,N}$:

$$A_{\text{inp}} \tilde{U}_{k,N} \le b_{\text{inp},k}. \tag{8}$$

The state constraints limit a certain function of the states. An example of a state constraint is that if we want to avoid sawteeth, we ask to ensure $q(\rho) > 1$ ($\iota(\rho) < 1$) at all times. Since we have $D = 0$ in our model, constraints on controlled variables can be rewritten as linear inequality constraints on the states.

Hard state constraints are discrete (i.e. they are satisfied or violated). It is important to note that operating in the vicinity of a hard state constraint can be dangerous. Being in the vicinity of the hard constraint, a disturbance may result in violating the constraint and this also implies that the controller may find no actuator trajectory to stay within the constraint.

A common approach in MPC is to use soft state constraints in which state constraint violation is allowed, but the violation is penalized in the controller cost function. The parameter ε indicates the soft state constraint violation and $\varepsilon > 0$ when the soft state constraint is violated. The penalty in the cost function is added as $W_\varepsilon \varepsilon^2$. The scalar W_ε sets the softness or stiffness of the soft state constraint.

In Figure 5 it is illustrated how a soft state constraint $\iota(\rho) \le 0.95 + \varepsilon$ can be applied to avoid operation in the vicinity of the hard constraint $\iota(\rho) \le 1$. The soft constraint violation ε can be monitored in real-time and reported to a supervisory controller that may anticipate on this information. In Appendix A.2.2 it is shown that all soft state constraints can be cast as time-varying linear inequality constraints on the future input deviations $\tilde{U}_{k,N}$ and the soft constraint violation ε:

$$A_{\text{state}} \begin{bmatrix} \tilde{U}_{k,N} \\ \varepsilon \end{bmatrix} \le b_{\text{state},k}. \tag{9}$$

2.7. Cost function

The controller has the objective to minimize the future error while avoiding too aggressive control actions and entering the soft constraint region. This can be expressed in the cost function J_k as follows:

$$J_k = \tilde{Z}_{k+1,N}^T Q \tilde{Z}_{k+1,N} + \Delta\tilde{U}_{k,N}^T R_{\Delta\tilde{U}} \Delta\tilde{U}_{k,N} + W_\varepsilon \varepsilon^2, \tag{10}$$

where

$$\Delta\tilde{U}_{k,N} = \begin{bmatrix} \tilde{u}_{k+1} - \tilde{u}_k \\ \vdots \\ \tilde{u}_{k+N} - \tilde{u}_{k+N-1} \end{bmatrix} = \Gamma_\Delta \tilde{U}_{k,N} \tag{11}$$

and Γ_Δ being a difference matrix operator. In the cost function (10) we identify the following weights:

- Q is a diagonal performance weight on the future error norm
- $R_{\Delta U}$ is a diagonal input difference weight to avoid aggressive control actions
- W_ε is a weight that defines the flexibility of the soft constraint

Note that by choosing this structure of the cost function, the actuator trajectories remain unchanged in the absence of model mismatches, disturbances and more strict constraints. This is desirable as more complete knowledge is available offline in the optimization of the feedforward actuator trajectories than online in the MPC-controller.

Tuning of the weights in an MPC-controller in the absence of model-mismatches and active constraints is straight forward and intuitive. However, it is known from literature (e.g. [26]) that the presence of (many) active constraints and model mismatches makes the effect of tuning parameters less clear. Therefore performing a set of representative simulations is required prior to experiments in order to obtain the optimal settings of the controller as a compromise between performance and robustness under the circumstances (e.g. model mismatches and active constraints) that can be expected in the actual experiments. The chosen control settings are discussed in Section 3.5.

2.8. Strategies to reduce the online computational cost

In order to meet the computational requirements, the optimization problem can be made more compact (less optimization variables and linear constraints) using concepts from literature (e.g. [26, 28, 40]). We will discuss here the use of input parameterization and the reduction of the number of predicted states, controlled variables and constraints via the concept of coincidence points.

We parameterize each future actuator input sequence in $\tilde{U}_{k,N}$ by a relative small number of unknown parameters. In fact, the parameters $\tilde{p}_{k,N}$ are the inputs at some specified time instants (nodes), where we linearly interpolate between those nodes for the remaining time instants. In addition, we keep all inputs constant after a so called control horizon N_c, which is common in MPC practice.

The linear mapping between a future input parameter sequence $\tilde{p}_{k,N}$ and the future input sequence $\tilde{U}_{k,N}$ can be written as:

$$\tilde{U}_{k,N} = P_{\text{map}} \tilde{p}_{k,N}, \tag{12}$$

where the parameterization mapping matrix $P_{\text{map}} \in \mathbb{R}^{(N+1)\cdot n_u \times n_p}$ is fixed and chosen offline.

To reduce the number of predicted states, controlled variables and constraints, we use the concept of coincidence points. Herein, the reference and predicted controlled variables are desired to coincide only at a limited number of time instants in the prediction horizon. We choose to compute the predicted states and controlled variables only at this subset of time instants and additionally state constraints are only imposed at these time instants. By choosing the time between coincidence points not larger than the time between the nodes in the input

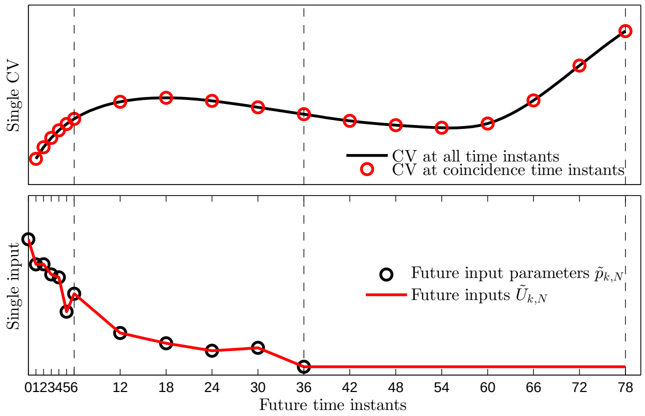

Figure 6. Illustration of reducing the computational cost using concepts of input parametrization and coincidence points as used in the simulation results. Figure illustrates for one actuator and one controlled variable. Input parameterization (bottom panel): a small number of parameters $\tilde{p}_{k,N}$ (o) are used to define a complete future input sequence $\tilde{U}_{k,N}$ (-). Coincidence points (top panel): the states and controlled variables are only predicted at a subset of time instants in the prediction horizon and desired to coincide with a reference trajectory only at these coincidence time instants. The CV at all time instants (-) is shown together with the CV at the coincidence time instants (o). The settings are explained in Section 3.5.

parameterization, using linear interpolation and considering the diffusive nature of the plasma profile dynamics, we can safely assume that no significant excursions of the states and controlled variables will occur in between those coincidence points.

The concept of input parameterization and coincidence points is illustrated in Figure 6 for one actuator and one controlled variable, using the settings as will be discussed in Section 3.5.

2.9. Quadratic programming

The future input deviations $\tilde{U}_{k,N}$ can be found by minimizing the cost function (10) subjected to the constraints (8) and (9). Note that the first input $\tilde{u}_k$ is the currently active input on the system and hence fixed in the optimization via equality constraints. Together, this can be rewritten using the prediction model (7) and CVP (12) as an online Quadratic Programming (QP) problem at each time instant t_k:

$$\begin{aligned} \underset{\xi_k}{\text{minimize}} \quad & \tfrac{1}{2}\xi_k^T H_k \xi_k + \tilde{x}_k^T F_k \xi_k, \\ \text{subject to} \quad & A_{\text{ineq},k}\xi_k \leq b_{\text{ineq},k}, \\ & A_{\text{eq}}\xi_k = b_{\text{eq},k}, \end{aligned} \tag{13}$$

where $\xi_k^T = [\tilde{p}_{k,N}^T \quad \varepsilon]$. The matrices H_k, F_k, $A_{\text{ineq,k}}$ and A_{eq} are computed offline and the vectors $b_{\text{ineq,k}}$ and $b_{\text{eq,k}}$ are externally provided by the user or set in real-time by a supervisory controller. All are defined in Appendix A.4.

QP is computationally cheap and has a unique constrained solution. Many algorithms exist to solve the QP problem (13) online, that can handle large number of free parameters and abundant linear constraints, e.g. [41, 42]. Using QPC [41], results in a computational time per time instant of less than 8 ms in the simulation cases of section 4 on a standard off-the-shelf laptop. This is fast enough for implementation on an existing tokamak, even more when one considers using dedicated hardware.

2.10. Summary of implementation

The implementation of the MPC-controller consists of an offline part and an online (real-time) part. To reduce the online computational burden, as many steps as possible are done offline. Offline, the steps are performed as given in Algorithm 1.

Algorithm 1 Offline algorithm to prepare for online QP problems

```
Run RAPTOR simulation with the (optimized) nominal actuator trajectory to obtain:
    • Linearized state space models: (4-5)
    • Nominal state and controlled variable (reference) trajectories: $(x_k^o, y_k^o)$
Choose prediction horizon $N$
Choose controller weights in (10): $Q$, $R_{\Delta U}$ and $W_\varepsilon$
Choose a future input sequence parametrization in (12): $P_{\text{map}}$
Compute the prediction matrices at each time instance in (7):
$\Gamma_{A_{k,N}}$, $\Gamma_{B_{k,N}}$, $\Gamma_{C_{k,N}}$ and $\Gamma_{D_{k,N}}$
Compute the pre-computable part of the matrices and vectors in (13)
```

Once the offline steps have been done, the online part can start during the simulation or experiment. Inside the MPC-controller, at each time step, Algorithm 2 is used.

Algorithm 2 Online algorithm inside MPC-controller

```
for all $t_k$ do
    Load the necessary pre-computed quantities for time step $t_k$, computed in Algorithm 1
    Receive from real-time observer or simulator: state estimate $x_k$
    Receive from supervisory controller (or user): constraint vectors for actuators and states
    Receive currently active input $\tilde{u}_k$ to fix in optimization
    Compute $b_{\text{ineq},k}$ and $b_{\text{eq},k}$ using (A.18) and (A.19)
    Solve QP (13) for $\xi_k^T = [\tilde{p}_{k,N}^T \quad \varepsilon]$
    Compute $\tilde{U}_{k,N}$ using (12)
    Compute $U_{k,N}$, $X_{k,N}$ and $Z_{k,N}$ using (6) and (7)
    Provide actuator command $u_{k+1}$ from $U_{k,N}$ to actuators
    Provide to supervisory controller (or user):
        • The future actuator commands $U_{k,N}$
        • The predicted evolution of states $X_{k,N}$ and controlled variables $Z_{k,N}$
        • The soft constraint violation
        • The active constraints
end for
```

3. Simulation setup

In this section we describe the simulation setup we developed to test the MPC-controller. A plasma scenario is set up in RAPTOR with ITER parameters [34]. It is not the objective of these simulations to provide quantitative estimates of the ITER performance or controllability of a particular scenario, but to illustrate the potential of an MPC-controller for profile control. First we introduce the physics model in RAPTOR, then an optimized nominal trajectory will be explained, after which we will investigate the performance of the linearized models as used in the controller and end with the controller settings as used in the simulations section.

3.1. Physics model and settings in RAPTOR

A CRONOS simulation of an ITER L-mode ramp-up in [36] is used to supply the initial profiles of T_e and ψ, the prescribed profile evolutions of n_e, n_i and Z_{eff} as well as those quantities in the transport equations depending on the 2D magnetic equilibrium. Although the controller design is in principle applicable to any operating mode, we restrict ourselves in this work to L-mode simulations. The L-H transition is not modeled in these simulations. It is assumed that we remain always in L-mode, although we slightly exceed the predicted L-H transition power threshold for this density evolution as reported in [36].

In the present work, the 2D magnetic equilibrium is assumed to be fixed in time. The simulation starts at 20 s, when the plasma current equals 4.7 MA. The ion temperature profile evolution is a fixed scaling of the electron temperature evolution, i.e.: $T_i(\rho, t) = f_{T_i}(\rho)T_e(\rho, t)$. As most heating in the core is provided to the electrons, rather than the ions, $f_{T_i}(\rho)$ is chosen as a linear function, increasing from 0.6 at the center until 1 at the edge. As RAPTOR is using an implicit time-discretization scheme, large time steps are allowed while remaining numerically stable at the expense of some loss of accuracy for larger time steps. For this plasma all dominant time scales in the input - controlled variable behavior are larger than 1 s. Therefore we chose in the first second a time step of 5 ms to simulate the initial phase, for the remainder a time step of 1 s was chosen. Open-loop simulations with smaller time steps produced the same results up to a small error, confirming that the chosen time steps are small enough.

The controlled actuators considered are the desired plasma current I_p^{des} and the power to four EC-beams deposited at different locations as given below:

Beam	Location ρ	Type EC	Gaussian full width in ρ []
1	0	co	0.2
2	0	counter	0.2
3	0.2	co	0.1
4	0.4	co	0.1

The combination of equatorial launcher and the upper port launchers as given in the ITER design allows for EC-deposition in the region between $\rho = 0$ and $\rho \approx 0.6$, where we use here deposition up to $\rho = 0.4$. The EC-beam power and current density deposition is modeled by Gaussian profiles where the current drive efficiency scales with T_e/n_e. The nominal flat-top value is $\sum_{i=1}^{4} P_{\mathrm{EC,i}} = 20\,\mathrm{MW}$. It is explicitly assumed that the power will be distributed over the equatorial launcher and the upper port launchers by a dedicated low level controller that is outside the scope of this contribution. The off-axis NBI heating and current drive is modeled using a pencil beam model [34] and the NBI-power is fully prescribed, ramping between 60 s and 70 s from zero to its flat-top value of 16.5 MW, i.e. half of the available NBI-power.

3.2. Constraints

We define here a set of actuator and plasma physics constraints that is imposed in the design of the nominal trajectory and in the MPC-controller. The actuator constraints are:

- $0 \leq I_p \leq 10\,\mathrm{MA}$, during flat-top $6 \leq I_p \leq 8\,\mathrm{MA}$
- $-0.1\,\mathrm{MA/s} \leq \frac{dI_p}{dt} \leq 0.1\,\mathrm{MA/s}$
- $P_{\mathrm{EC,i}} \geq 0\,\mathrm{MW}, \forall i \in [1, 2, 3, 4]$
- $\Sigma P_{\mathrm{EC}} \leq 25\,\mathrm{MW}$

A soft state constraint is added as $\iota(\rho) \leq 0.95 + \varepsilon$. In the design of the nominal trajectory the hard constraint $\iota(\rho) < 0.95$ is added together with the requirement $I_p(t_{\mathrm{end}}) \geq 7\,\mathrm{MA}$.

Large and steep variations of the plasma current I_p during the flat-top phase are seen to induce MHD-activity and problems with the vertical stability control system, both related to exceeding limits on $l_i^{(3)}$. Therefore the plasma current I_p is in present day operation not often varied during the flat-top phase. However, imposing strict requirements on I_p during the flat-top phase maintains the plasma within the limits on $l_i^{(3)}(t)$ and allows for using I_p as a feedback actuator. Alternatively, (linearized) soft state constraints may be imposed directly on $l_i^{(3)}(t)$.

3.3. Optimized nominal trajectory

In principle, the nominal feedforward actuator trajectories can be determined by tokamak operators. Here, the feedforward actuator trajectories and the corresponding state and controlled variables trajectories are calculated following the numerical optimization described in [7].

The nominal trajectory optimization aims at achieving any ι-profile which is *stationary* at the beginning of the flat-top phase, while staying within the actuator and plasma physics constraints as defined in section 3.2. The flat-top phase starts after 120 s. The loop voltage profile $U_{\mathrm{pl}}(\rho)$ must be as flat as possible after 120 s, meaning that the ohmic current redistribution has almost stopped and therefore the ι-profile is stationary. The optimization cost function contains therefore solely a penalty on the gradient of the final loop voltage profile $U_{\mathrm{pl}}(\rho)$.

In our simulations, it was not possible to obtain a stationary q-profile at the beginning of the flat-top within the constraints with a flat-top plasma current I_p of more than 7 MA. This is mainly caused by the lack of the bootstrap current and low current drive efficiency due to low temperature in L-mode. We noticed that the obtainable flat-top plasma current I_p is also sensitive to the settings of the NBI actuator model, that was not yet benchmarked against other NBI models.

The optimized nominal trajectories for the current and EC-power distribution are shown in 7(a) and 7(e-h) respectively. The prescribed NBI-power and the total EC-power are shown in 7(b). The plasma current I_p is ramped up quickly until 40 s, after which the numerical optimization calculates a small current overshoot at 100 s. Such overshoots have been reported to be beneficial for achieving a stationary flat ι-profile [7, 36, 43].

The optimization uses the flexibility in the EC-power allocation. At most times, the counter-ECCD power (Figure 7(f)) significantly exceeds the co-ECCD power (Figure 7(e)). This implies that during such phases, a resultant negative central current is driven by these two EC-beams. This helps to flatten the U_{pl}-profile and prevents violating the constraint $\iota \leq 0.95$.

The EC-beam at $\rho = 0.2$ (Figure 7(g)) provides significant heating and current drive during the ramp-up phase. Its peak-power occurs just before the NBI-power (Figure 7(b)) is ramped up between 60 s and 70 s. The EC-power deposition at $\rho = 0.4$ (Figure 7(h)) is ramped-up until finally all available EC-power is provided at this location (20 MW) at 120 s. The EC-power is reduced during and just after the ramp-up of the NBI-power as shown in 7(b). The prescribed ramp-up of central electron and ion densities is given in Figure 7(d). The resulting central ion and electron temperatures $T_{i,0}$ $T_{e,0}$ are shown in 7(c). Note that central temperatures reach their maxima after 40 s.

At the times indicated by the gray dotted lines in the top panels (60,70,100 and 120 s), the resulting profiles of ι, T_e, T_i, n_e, U_{pl} and the components of the parallel current density $j_{\|}$ are shown in the bottom panels of Figure 7. The ι-profile increases smoothly to the final profile at 120 s, which is close to the constraint $\iota(\rho) < 0.95$. The temperature profile peaks at the center after the ramping-up of the NBI-beam (70 s) and the increasing power to the counter EC-beam at $\rho = 0$. The temperature profile afterwards (100 s and 120 s) widens and drops in the center after the increasing concentration of all EC-power at $\rho = 0.4$. The U_{pl}-profile evolves towards the nearly flat profile at 120 s. Finally, the evolution of the cumulative current density profile is shown. The inward diffusion of the ohmic current density can be noticed which follows after 120 s the typical conductivity profile (corresponding to the flat U_{pl}-profile). The NBI current density appears after 70 s. Also the EC current density active at different locations can be noticed clearly. The bootstrap current is hardly visible and is only 6% of the plasma current in flat-top.

3.4. Validity of linearized models

The MPC-controller approximates the nonlinear plasma behavior as a linearized system at each moment in time. Therefore, we now analyze the validity of the linearized models around the designed nominal trajectory by open-loop simulations. Special emphasis is put on the time evolution of the differences between the simulated outputs of the linearized and non-linear models as this is indicative for the expected prediction error in the MPC-controller. A comparison with

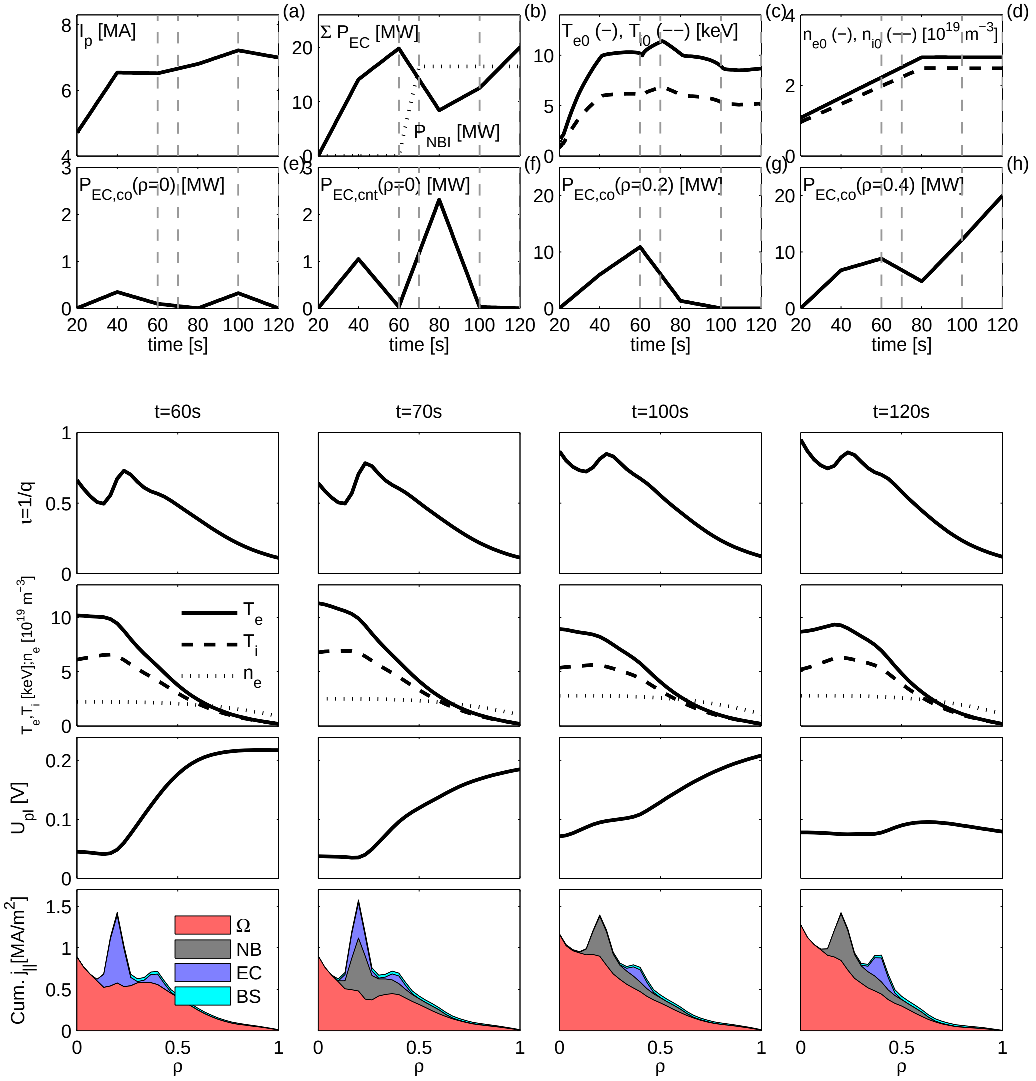

Figure 7. Optimized nominal trajectories for achieving stationary ι-profile at the beginning of the flat-top phase in L-mode, while satisfying actuator and plasma physics constraints. Top panels show the time evolution of optimal actuator commands of the plasma current I_p (a), four EC-beams (e-h) and corresponding total EC-power (b), central temperatures (c) and central densities (d). The prescribed evolution of NBI-power is also shown in (b). Bottom panels show the profiles ι,$T_{e,i}$, n_e, U_{pl} and components of $j_{\|}$ at 60, 70, 100 and 120 s. Note $I_p(120\,\mathrm{s}) = 7\,\mathrm{MA}$, that is the maximum achievable plasma current in our simulations for achieving a stationary ι-profile within the given constraints in L-mode.

a single linear model is also included.

For this purpose the responses of the linearized and nonlinear models to a perturbation input vector are compared in the following cases:

- Nonlinear model (RAPTOR): the perturbation input vector $\tilde{u}_k$ is added to the nominal input vector u_k^{ref} and fed to RAPTOR. Comparison of the outputs of RAPTOR for the case with perturbation and without perturbation yields $\Delta T_{\mathrm{e,nonlin}}$ and $\Delta \iota_{\mathrm{nonlin}}$.
- Multiple linearized models: the perturbation input vector $\tilde{u}_k$ is fed into the models locally linearized at each moment in time which yields $\Delta T_{\mathrm{e,multilin}}$ and $\Delta \iota_{\mathrm{multilin}}$.
- Single linear models: the perturbation input vector $\tilde{u}_k$ is fed into a single linear model (taken at end of ramp-up) which yields $\Delta T_{\mathrm{e,singlelin}}$ and $\Delta \iota_{\mathrm{singlelin}}$.

The results are given in Figure 8. Random sum of sinusoids with a limited frequency band (often used in system identification [44]) are used for the verification and are shown in column 1 (- -), (-··-) and (-··-) on top of the nominal actuator inputs (—). The amplitude of the perturbation inputs scale with the nominal actuator trajectory. The same perturbation inputs are used for the nonlinear (- -) case and the cases with multiple linearizations (-··-) and a single linear model (-··-). The effect of the perturbation inputs can be appreciated in column 2 (for $\Delta\iota$) and column 3 (for ΔT_e) at several locations in the plasma.

The evolution of ΔT_e using the multiple linearized models is in excellent agreement with the fully nonlinear model for all shown locations. The evolution of $\Delta\iota$-profile shows a small difference in the center of the plasma at $\rho = 0$, $\rho = 0.2$ and $\rho = 0.4$. These differences are caused by the nonlinear effect of the electron temperature T_e on the conductivity $\sigma_{\|}$, which influences the dynamics of the ι-profile and cannot be contained accurately inside the multiple linearized state space models. While the quantitative estimates are not entirely correct for this region, the sign of the response is the same in both cases, which is essential for feedback control.

Prediction is in essence an open-loop simulation. For effective and reliable state constraint handling in MPC, prediction errors should be small, even at the end of the prediction horizon. Longer prediction horizons introduce increasing prediction errors due to model mismatches. Using the multiple linearizations results here in an error on the ι-profile of less than 0.01 in magnitude.

The response of the single linear model shows that the ΔT_e evolution is still in fair agreement with the nonlinear model, except for the center. However, a large deviation with respect to the nonlinear case occurs in the $\Delta\iota$ evolution, especially at $0 \leq \rho \leq 0.6$. Note that the sign of the response is not correct at many times. Although the contribution of the bootstrap current in L-mode plasmas is small, the remaining nonlinear terms (e.g. nonlinear coupling of kinetic and magnetic profiles via plasma conductivity) prevent that a single linear model is not sufficient.

These specific results (not necessarily worst-case) indicate that the multiple linearized models describe accurately enough the dynamics in the broad vicinity of the nominal trajectory to justify the usage of these models in the MPC-controller.

3.5. Controller settings

In this section we discuss the controller settings. The MPC-controller can be easily tuned as only a few parameters have to be set with a clear definition of the impact on the controller performance. The controller settings to be chosen are:

Reference. The nominal controlled variable evolution will function as a time-varying reference during the ramp-up phase, during the flat-top phase (after 120 s) the reference is kept constant.

Prediction horizon N. The prediction horizon N should be chosen such that the dominant effect of the actuators on the controlled variables is taken into account. From analysis of the eigenvalues of the system matrix A_k that are dominant in the input-controlled variable behavior (analyzed via the so called Hankel singular values [45]), we obtain that the slowest eigenmode in the flat-top of the nominal simulation has a timescale of 90 s. This is the resistive diffusion time scale. We choose $N = 78$ (equals 78 s), being a compromise between on the one hand capturing crucial dynamics and on the other hand increasing prediction errors due to model uncertainties and higher computational cost.

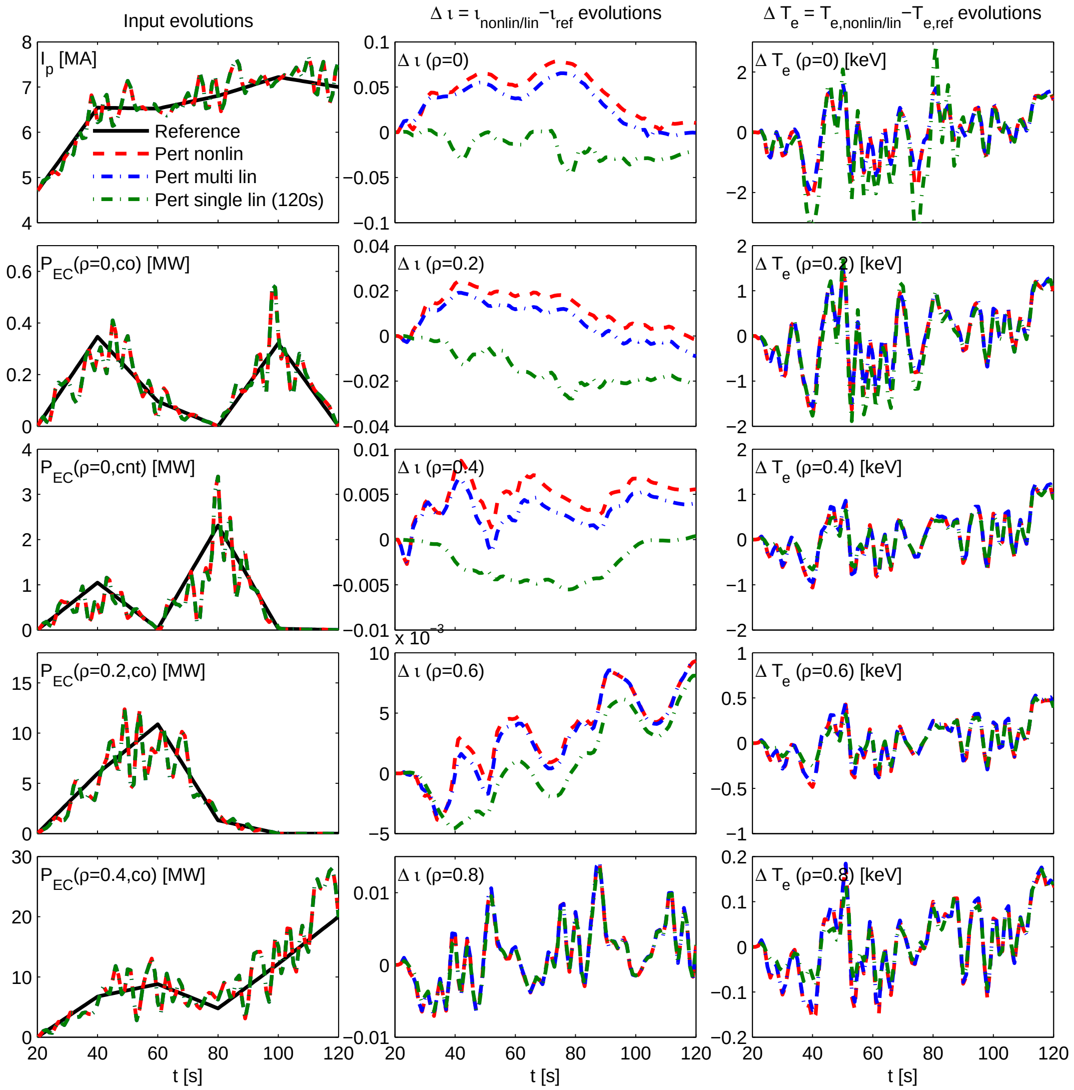

Figure 8. Comparison responses of linearized and nonlinear models. Column 1: actuator evolutions of plasma current I_p and four EC-powers for nominal case (—) and sum of sinusoids inputs on top of nominal trajectory for nonlinear (- -) and cases using linearizations at each moment in time (-·-) and a single linearization (-·-). Column 2: response to perturbation inputs of $\Delta\iota$-profile. Column 3: effect of added perturbation inputs on $\Delta T_{\rm e}$-profile. Note the far more accurate description by multiple linearized models of both the T_e-profile and ι-profile with respect to a single linear model.

Weights Q, $R_{\Delta U}$ and W_ε. In this work we obtained the settings of the weights Q, $R_{\Delta U}$ and W_ε in the controller cost function by simulating the provided scenarios with varying these parameters around an initial intuitive guess and comparing robustness and performance. We noticed that especially the tight constraints on the plasma current (amplitude and ramp rate) and the available EC-power in combination with a significant model mismatch in the electron heat transport reduced the clear meaning of the tuning parameters. However, using the fast simulator RAPTOR, it was possible to obtain a satisfying tradeoff in a reasonable time.
The coefficients on the diagonal of Q are chosen such that the error is only penalized in the region $\rho \leq 0.6$. $\iota(\rho = 1)$ (and hence the total current) is therefore free.

First the weight matrices are defined as a matrix having matrix norm equal to one multiplied by a scalar. By choosing $\frac{\|R_{\Delta U}\|_2}{10^{-12}} = 1$ (normalizing for units in W or A), we can define the scalar $W_Q = \|Q\|_2$. Varying W_Q and W_ε and simulating the provided scenarios led to the choice of $W_Q = 5 \cdot 10^3$ and $W_\varepsilon = 2 \cdot 10^7$.

Input parameterization P_{map}, including control horizon N_c. This is chosen as shown in Figure 6 for all actuators. During the first six time instants every input is also a parameter, allowing for much control freedom in the beginning of the prediction horizon. Afterwards for each sixth time instant the input is a parameter up to the control horizon $N_c = 36$ and the inputs in between are linearly interpolated. A spacing of six time instants allows still reasonable control freedom, while reducing the number of optimization parameters significantly. After the control horizon, the inputs are kept constant. Using this input parameterization reduces the number of free variables in the optimization from 396 to 61.

Coincidence time instants. The coincidence time instants are chosen as in Figure 6. These overlap with the time instants at which the input parameters are defined, where additionally coincidence time instants are added after the control horizon $N_c = 36$ at each sixth time instant. The number of time instants at which the states and controlled variables are predicted (and the state constraints are imposed) is this way reduced from 78 to 19.

4. Simulation results

In this section we demonstrate the effectiveness of the MPC-controller in simulations. First we illustrate tracking in the flat-top phase. Secondly we show handling of a sudden drop in available EC-power by the MPC-controller and finally we illustrate the tracking performance and constraint handling under plant-controller model mismatch caused by underestimated or overestimated thermal transport.

4.1. Tracking the reference in flat-top phase

In the first simulation case, tracking of the reference ι-profile during the flat-top phase will be considered. The reference ι-profile at the beginning of the flat-top phase is not yet fully in stationary state, since the U_{pl}-profile in Figure 7 was not completely flat. Therefore the ι-profile is expected to drift away from its reference in the absence of feedback control (only feedforward). The question is to what extent the feedback control will compensate for this drift. For this purpose, all feedforward actuator trajectories and prescribed profiles and quantities are extended from the end of the ramp-up until 400 s. The reference ι-profile during the flat-top phase is also taken at the end of the ramp-up phase.

The tracking error will be expressed by the normalized 2-norm of the vector of the error in the ι-profile: $\|\iota - \iota_{\text{ref}}\|_2 / \|\iota_{\text{ref}}\|_2$. This error is only taken in the region $0 \leq \rho \leq 0.6$, consistent with the part of the ι-profile which is weighted in the controller cost function.

The results for tracking the reference ι-profile in the flat-top phase are presented in Figure 9. In the feedforward only case (- -), the actuator commands (Figure 9 (a,e-h) are kept constant after the end of the ramp-up phase. This yields also a constant total EC-power (Figure 9(b)). The resulting error norm (Figure 9(c)) increases as the ι-profile drifts away from the reference ι-profile and violates also the soft constraint to a large extent (Figure 9(d)). The resulting ι-profile is given in the bottom panels (Figure 9(i-l)) for the time steps also indicated by the gray dotted lines in Figure 9(c): 120, 150, 200 and 399 s. In Figure 9(l) can be observed clearly that the ι-profile has evolved away from the reference (black) and has even entered the soft constraint region.

The feedback case (- · -) shows the performance of the MPC-controller to track a reference ι-profile during the flat-top phase. The controller reduces the tracking error (Figure 9(c)) significantly and avoids the soft constraint violation (Figure 9(d)) by lowering the plasma current I_p (Figure 9(a)) and adjusting the powers to the co-EC-beam at $\rho = 0$ (Figure 9(e)) and the co-EC-beam at $\rho = 0.4$ (Figure 9(h)). The reduction of the tracking error is also clearly shown in Figure 9(l), where the ι-profile is almost on top of its reference.

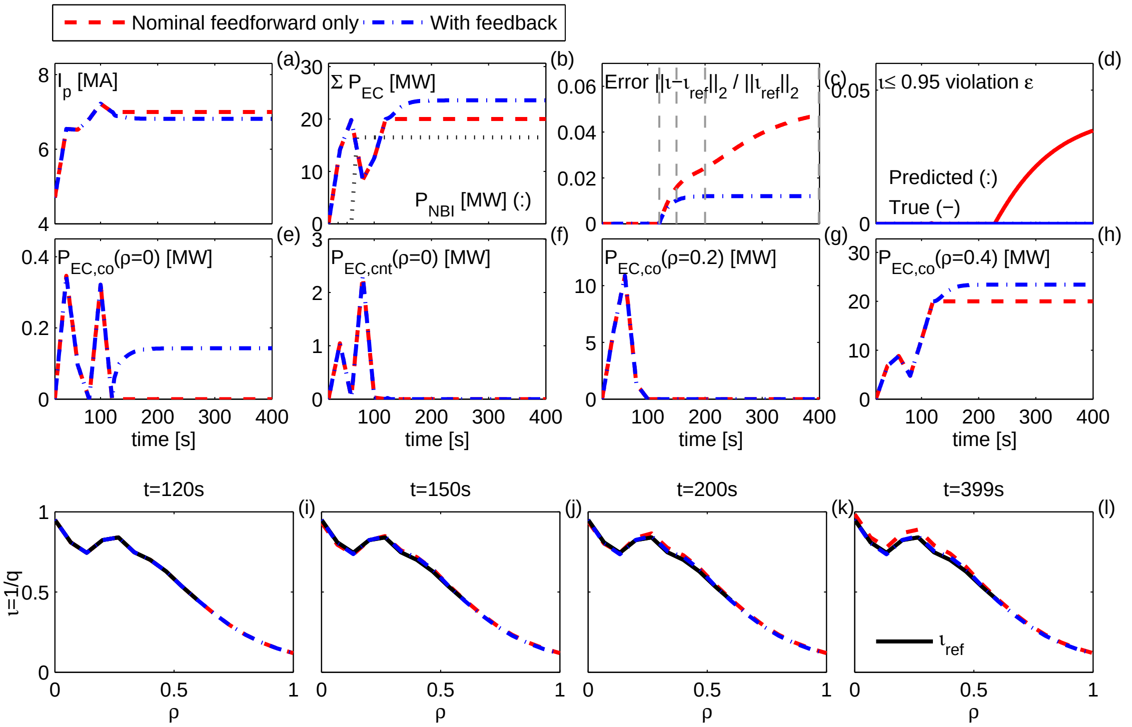

Figure 9. Tracking of reference ι-profile in flat-top phase. Case with feedforward only (- -) is shown together with feedback case (-··-). Top panels show time evolution of actuator commands of plasma current I_p (a), four EC-beams (e-h) and corresponding total EC-power (b) error norm (c) and soft constraint violation (d). Bottom panels reveal ι-profiles at time steps also indicated by gray dotted lines in top panels (120, 150, 200 and 399 s). Reference ι-profile is also shown for comparison (-). Feedback controller reduces the tracking error (c) and avoids the soft constraint violation (d) by adjusting the actuators (a,e,h).

It can be concluded that the MPC-controller can significantly reduce the error during tracking a fixed reference ι-profile in the flat-top phase, despite the fact that the ι-profile was not yet fully in stationary state at the start of the flat-top.

4.2. Tracking with time-varying EC-power limit

The next simulation case illustrates the handling of time-varying power limits. The goal is the same as in the previous case: tracking the reference ι-profile during the flat-top phase. However, after 200 s we limit the maximum available EC-power to 14 MW in stead of 25 MW. In Figure 10 the response of the plasma to this limit will be compared between simply setting the available EC-power to its maximum value (14 MW) and the case in which the MPC-controller is aware of the time-varying limits.

The unconstrained EC-power case from Figure 9 is repeated (-) and shown together with the saturated case (- -) and the case in which the MPC-controller automatically takes the time-varying limits into account (-··-).

In the saturated case (- -), the actuator trajectories are taken from the unconstrained EC-power case and applied in open-loop, where at 200 s the EC-power to the beam at $\rho = 0.4$ is reduced to 14 MW. In the absence of feedback, this results in an increasing error (Figure 10(c)) and a soft constraint violation exceeding early $\iota(\rho) \leq 1$ (Figure 10(d)). In Figure 10(j-l),it noted that this violation happens at $\rho = 0$ where $\iota(\rho)$ is largest.

In the case in which the MPC-controller is aware of the time-varying limit (-··-), the controller

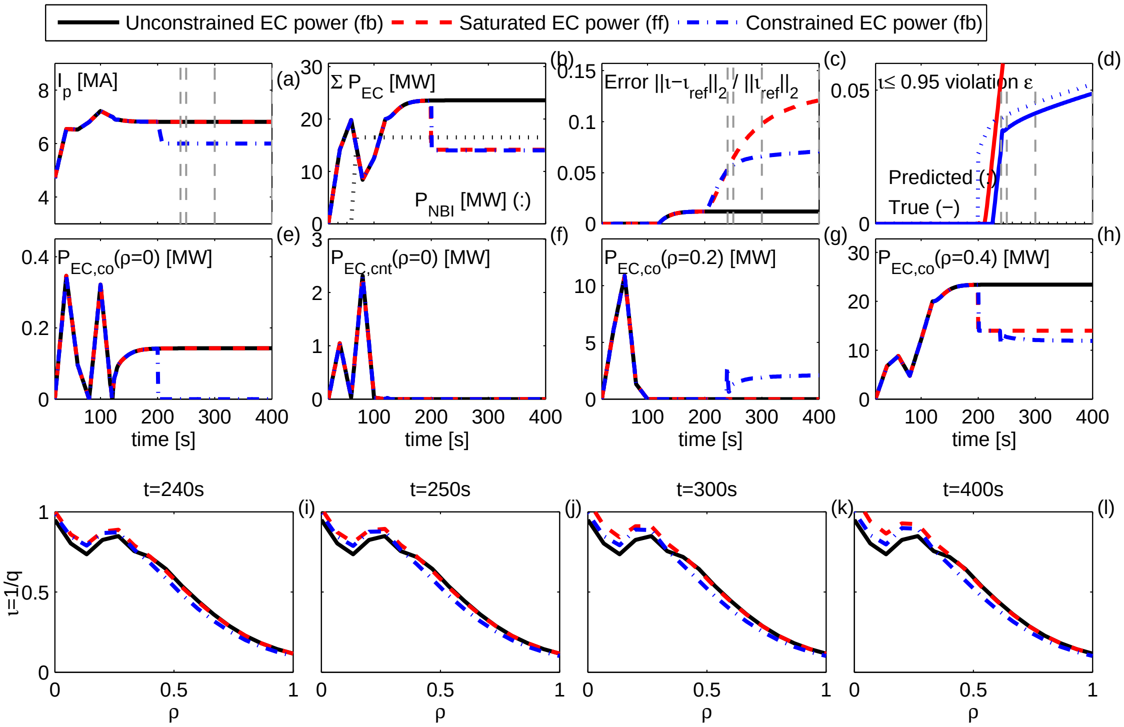

Figure 10. Saturation and handling of time-varying available EC-power constraint. The unconstrained feedback case from Figure 9 is repeated (-). In the saturated case (- -), the actuator trajectories are copied from the unconstrained EC-power feedback case and applied in open-loop, where the EC-power to the beam at $\rho = 0.4$ is reduced to 14 MW after 200 s (h). This results in an increasing error (c, i-j) and soft constraint violation (d). The MPC-controller (-··-) limits the soft constraint violation (d) at a smaller error norm then with manual saturation(c), by adjusting the actuators (a,e,g,h), resulting in a different ι-profile (i-l).

aims at avoiding the soft constraint at the expense of a higher control error then without the strict limit (c). At 200 s, the maximum available EC-power is indeed reduced to 14 MW and the power to the EC-beam at $\rho = 0.4$ subsequently decreases further, the co-EC-beam at $\rho = 0$ is shut off, while the EC-power to the beam at $\rho = 0.2$ is ramped up such that the maximum available EC-power is used. This means an inward shift of the EC-power deposition. Notice the maximum ramp-down in the plasma current I_p immediately after 200 s, after which the plasma current is kept at its minimum of 6 MA. The resulting change in the ι-profile can be observed in Figure 10(i-l).

A detailed view of Figure 10 (d) in the time interval between 200 s and 400 s is given in Figure 11. Immediate after the available power drops, the controller predicts that the soft state constraint will be violated significantly within the prediction horizon. The actual soft state constraint violation reaches this point only 35 s later.

The prediction of the expected soft constraint violation together with the expected actuator and profile evolution can be provided to a supervisory controller. This would enable the supervisory controller to take adequate actions. In this case adequate actions may be (temporally) modifying the constraints on the ι-profile and also modifying the reference ι-profile towards a target which is achievable with the limited actuators. Alternatively, the supervisory controller can decide to switch to an entirely different plasma scenario depending on the experimental program in case of ITER. By those actions the controller would be able to compute desirable actuator inputs such that the plasma is maintained within the new limits.

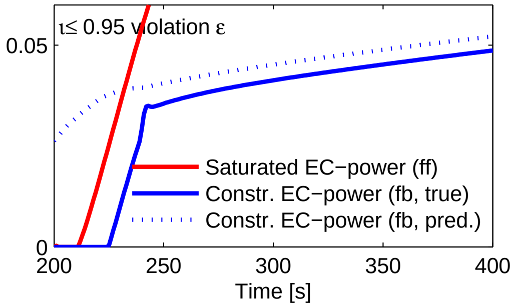

Figure 11. Detail of resulting soft constraint violation for saturation and automatic handling of time-varying available EC-power constraint. Note that already at 200 s a soft constraint violation is predicted to occur sometime within the prediction horizon. Note that the prediction is 35 s before the actual violation reaches the predicted point.

4.3. Tracking with model mismatches

Model mismatches between the real system and the model in the controller are expected and the controller needs to compensate for them. Two simulation cases are therefore intended to show the performance of the MPC-controller under a significant model mismatch during the ramp-up. In this example we change the transport model of the simulator model, while maintaining the same settings for the controller model as used in the previous cases.

4.3.1. Case with reduced thermal transport In the first case with model mismatch, the empirical Bohm-gyroBohm transport coefficients are reduced by 30% with respect to the nominal case. This reduces the thermal transport significantly and hence higher temperatures (about 2 keV difference in $T_e(\rho = 0)$ in flat-top) are achieved for the same heating power, affecting the conductivity profile and therefore also the current density evolution.

Figure 12 reveals the results. The reference (-) is shown for comparison. When these feedforward actuator trajectories are used in open-loop (- -) in the simulator with model mismatch, this model mismatch causes a substantial deviation from the reference ι-profile (c, i-l).

In the feedback controlled case (-·-) the tracking error can be significantly reduced by the MPC-controller (c,i-l). The controller changes the actuator commands to the EC-beams especially during the ramp-up (e-h) at finally a slightly lower plasma current I_p (a).

The soft state constraint is slightly violated, but not predicted. This is caused by the error in the prediction model, leading to less accurate predictions and constraint handling. The fact that the EC-power is still increased at reduced transport with respect to the feedforward case results in even higher (central) temperatures and hence even larger model mismatches in the current diffusion dynamics.

4.3.2. Case with increased thermal transport In this second case with model mismatch the transport is increased, leading to a reduction of the central temperature of more than 1 keV for the same heating power during the flat-top. We will show that this requires even more challenging control actions in which many actuator constraints become active.

Figure 13 reveals the results with increased thermal transport. Using only feedforward, results in a large error in the ι-profile (c,i-l), while the ι-profile also violates the soft state constraint (d). This even results in $\iota(\rho) > 1$ after 97 s.

Feedback control avoids the soft state constraint violation (d) and reduces the error significantly during the ramp-up, whereas a significant error remains in the flat-top close to the error in the feedforward only case (c). The controller pushes the actuators to the limits in ramp-rate (I_p, a) and amplitude (a,b), while distributing the EC-power over the different beams (e-h). A minor soft constraint violation is predicted, whereas a much smaller violation occurs temporally.

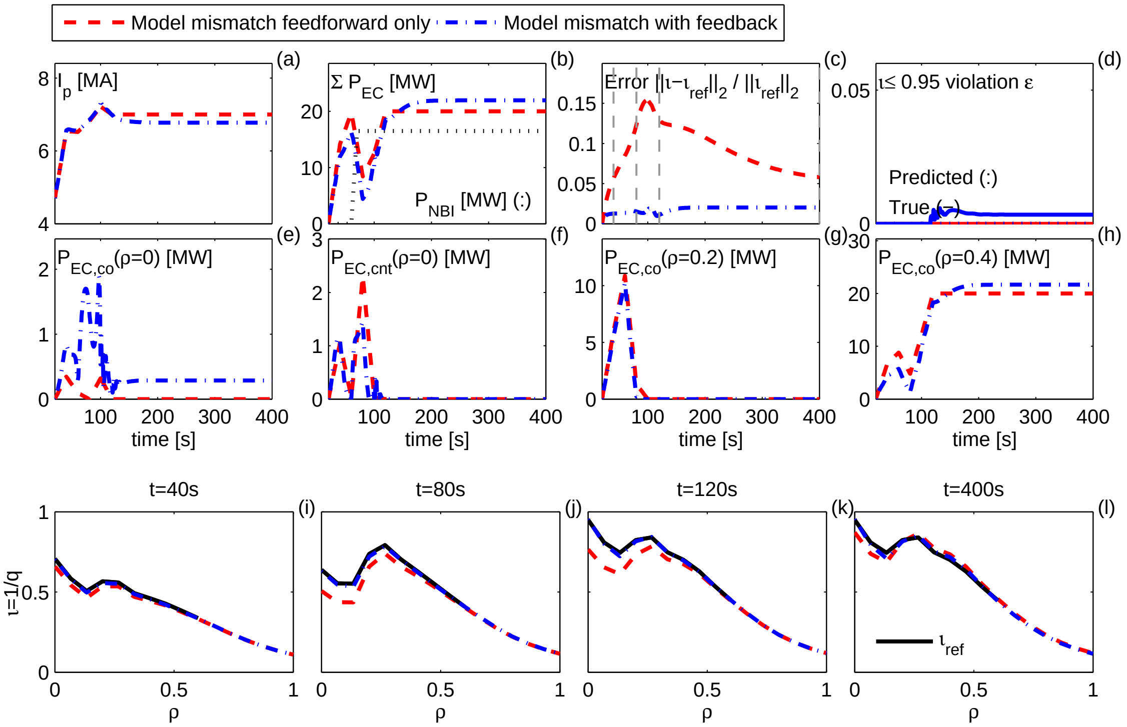

Figure 12. Tracking with decreased electron heat transport. The transport coefficients are reduced by 30% with respect to nominal case. Evolution subjected to model mismatch for feedforward only case (- -) is shown together with feedback controlled case (-·-). With only feedforward, the model mismatch results in a significant deviation from the reference ι-profile (c,i-l), while ι-profile slightly violates (unpredicted) the soft state constraint (d). Feedback control reduces the tracking error significantly (c,i-l), by adjusting the EC-power to the several beams (e-h) while while ι-profile slightly violates (unpredicted) the soft state constraint (d).

This indicates again that the prediction is not fully accurate (as expected), due to the model mismatch.

Both simulation cases with model mismatch indicate that the MPC-controller can reduce the tracking error due to a realistic model mismatch significantly, while handling the actuator and soft state constraints simultaneously. Taking the actuator limits actively into account allows for exploiting the full capabilities of the actuators.

5. Discussion

The performance is evaluated of an MPC-controller for the control of the safety factor profile that actively takes the time-varying actuator and state constraints into account. In this section we discuss the results and provide directions to improve or extend this work. Implementing this MPC-controller on an existing tokamak requires further validation in a more complete closed-loop simulation environment including plasma state reconstruction.

In the simulations we note a steady-state tracking error for which the current controller design cannot compensate. While a significant amount of the error is due to the fact that the reference profile is not achievable with the given set of model mismatches, disturbances and constraints, an other part is because of the current design of the controller. The tracking error due to model mismatches and disturbances may be further reduced by including real-time disturbance estimations in the controller, a concept that is often used in MPC. This allows for (near) offset-free tracking performance in the flat-top phase and more accurate real-time profile evolution

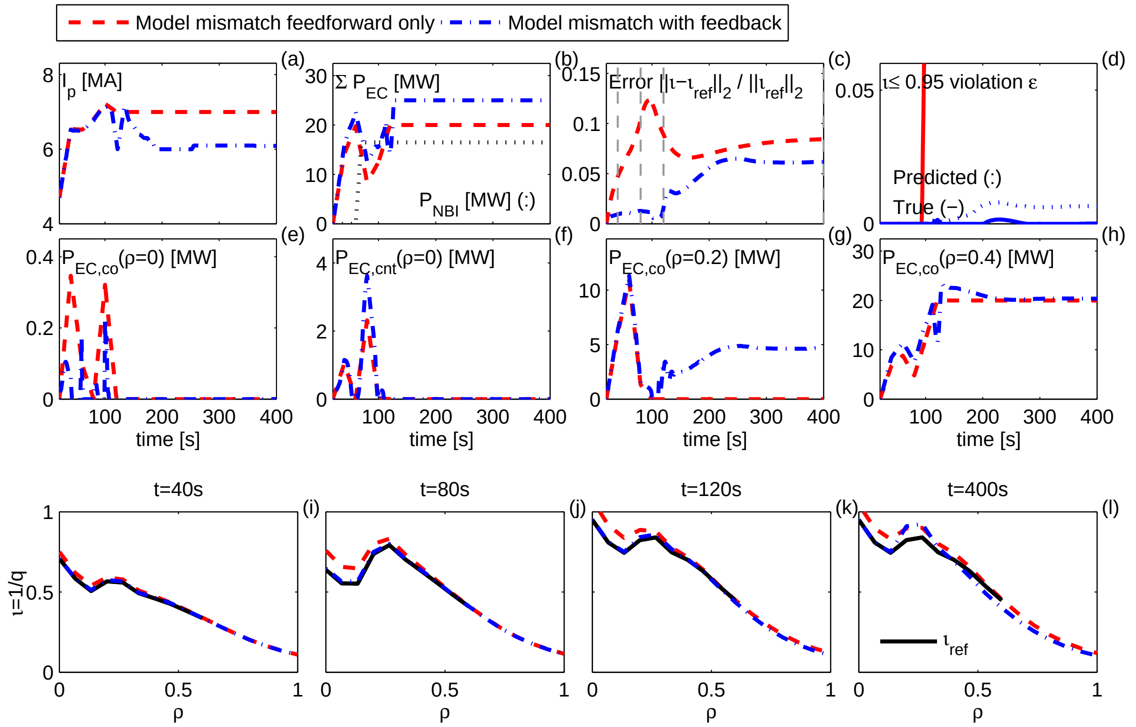

Figure 13. Tracking with increased electron heat transport. The transport coefficients are in this case increased by 30% with respect to nominal case. With only feedforward, the model mismatch results in a significant deviation from the reference ι-profile (c,i-l), while ι-profile violates the soft state constraint such that even $\iota(\rho) > 1$ after 97 s (d). Feedback control reduces the tracking error significantly during the ramp-up (c,i-l), by adjusting the EC-power to the several beams (e-h) and avoiding the soft state constraint violation. At the beginning of the flat-top, the EC-power is increased to its maximum (b) and distributed over the beams at $\rho = 0.2$ (g) and $\rho = 0.4$ (h), while the plasma current I_p (a) is decreased and increased at its ramp-rate limits and afterwards again decreased to close to its lower limit.

predictions. Disturbance estimations can be provided online, by using e.g. RAPTOR as observer [29].

The use of offline linearizations around a nominal trajectory limits excursions from these offline computed trajectories. However, the current design can be modified to include real-time varying references that are still within the vicinity of the offline computed nominal trajectory, but are more tractable depending on e.g. the real-time actuator constraints.

The controller does predict the true soft constraint violation only 35 s ahead instead of the prediction horizon of 78 s. This indicates that the prediction is not fully accurate on those timescales (as mentioned before). Nevertheless, these 35 s warning time is already valuable information. The proposed extension of the controller for including disturbance estimates is expected to result in more accurate prediction of the soft constraint violations.

The MPC-controller design (including disturbance estimates) is expected to satisfy the needs of profile control in case of normal well-prepared operation with well-behaved plasmas under modest uncertainty. Dealing with less well-behaved plasmas may be enabled by nonlinear MPC-approaches that are likely to be feasible on future larger scale tokamaks like ITER, having longer timescales of the profile evolution that provides more computational time. These nonlinear MPC-approaches may allow for dealing with real-time varying references or control objectives, larger deviations from the nominal feedforward evolution and larger disturbances and constraint changes.

The flexibility of the proposed MPC-controller to include time-varying constraints set by a supervisory controller is worth to be explored in future work, for instance for the simultaneous control of profiles and NTMs. The supervisory controller may use the real-time information of predictions of actuator and plasma state evolutions for allocating the shared actuator sources to the different control tasks, modifying references and state constraints and applying preemptive mitigation actions if disruptions cannot be avoided.

The required computational time can be further reduced by using faster dedicated algorithms and hardware to solve the online QP problem. This improvement may allow for implementation on faster time scales, e.g. smaller tokamaks or other control problems in fusion. One particular interesting case would be the simultaneous constraint handling for the plasma boundary control and the profile control. However, this requires the development of an integrated control-oriented model which solves self-consistently the nonlinear coupled evolution of shape and profiles.

6. Conclusions

Active handling of time-varying actuator and plasma physics constraints is desired in the control of the safety factor profile in advanced operation of tokamaks. MPC is the only control method which can handle both time-varying actuator and state constraints. This work presented the design of an MPC-controller and its application in closed-loop simulations.

The proposed MPC-controller uses a prediction model, based on linearizations around a reference trajectory, to relate the future evolution of states and controlled variables to the present plasma state and future actuator commands. The MPC-controller minimizes a cost function which penalizes the future error norm while avoiding too aggressive actuator inputs and violating soft state constraints. The minimization is subjected to time-varying actuator and plasma physics constraints. The resulting QP-problem can be solved using dedicated solvers on a normal laptop within 8 ms, which is fast enough to be implemented on currently operational tokamaks.

The potential of the MPC-controller is demonstrated in closed-loop simulations using RAPTOR for an L-mode scenario with ITER parameters and a q-profile with $q > 1$. Simulation examples show tracking of a reference q-profile evolution during the ramp-up and flat-top, even in cases of underestimated or overestimated transport in the transport model. Another simulation case reveals that the controller can handle a sudden reduction of the available EC-power and in case that the constraints become too stringent, the MPC-controller predicts that it cannot stay within the (soft) constraints in the near future. This allows one to provide a supervisory controller with warnings of these expected (soft) constraint violations and moreover with real-time predictions of the expected state and controlled variable evolution.

Finally, it is discussed that the MPC-controller can be extended and improved to allow for (near) offset-free tracking in the flat-top phase and more accurate real-time profile evolution predictions and hereby more reliable state constraint handling. The MPC-controller is proposed to allow for dynamic actuator sharing for the simultaneous control of profiles and NTMs. Reduction of the required computational time may open up new applications in fusion.

Acknowledgements

The first author gratefully acknowledges the discussions with Hugo van den Brand and Jonathan Citrin of FOM DIFFER and Olivier Sauter of CRPP-EPFL. This work has been performed in the framework of the NWO-RFBR Center of Excellence (grant 047.018.002) on Fusion Physics and Technology. This work is supported in part by the Netherlands Organization for Scientific Research and the European Communities. It was carried out within the framework of the European Fusion Programme. The views and opinions expressed herein do not necessarily reflect those of the European Commission.

Appendix A. Definitions and derivations

In this appendix we provide the definitions and derivations as used in section 2.

Appendix A.1. Prediction matrices

For the use in the prediction equations (7), the following notation is introduced:

$$\coprod_{l=m}^{n} A_{k+l} \equiv \begin{cases} A_{k+n}A_{k+n-1}...A_{k+m} & \text{if} \quad m \leq n \\ I & \text{if} \quad m > n \end{cases} \tag{A.1}$$

The prediction matrices $\Gamma_{A_{k,N}}$, $\Gamma_{B_{k,N}}$, $\Gamma_{C_{k,N}}$ and $\Gamma_{D_{k,N}}$ can now be written as follows:

$$\Gamma_{A_{k,N}} = \begin{bmatrix} \coprod_{l=1}^{0} A_{k+l} \\ \coprod_{l=1}^{1} A_{k+l} \\ \vdots \\ \coprod_{l=1}^{N-1} A_{k+l} \end{bmatrix} A_k, \tag{A.2}$$

$$\Gamma_{B_{k,N}} = \begin{bmatrix} \left[\coprod_{l=1}^{0} A_{k+l}\right] B_k & 0 & \cdots & 0 & 0 \\ \left[\coprod_{l=1}^{1} A_{k+l}\right] B_k & \left[\coprod_{l=2}^{1} A_{k+l}\right] B_{k+1} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ \left[\coprod_{l=1}^{N-1} A_{k+l}\right] B_k & \left[\coprod_{l=2}^{N-1} A_{k+l}\right] B_{k+1} & \cdots & \left[\coprod_{l=N}^{N-1} A_{k+l}\right] B_{k+N-1} & 0 \end{bmatrix}, \tag{A.3}$$

$$\Gamma_{C_{k,N}} = \begin{bmatrix} C_{k+1} \coprod_{l=1}^{0} A_{k+l} \\ C_{k+2} \coprod_{l=1}^{1} A_{k+l} \\ \vdots \\ C_{k+N} \coprod_{l=1}^{N-1} A_{k+l} \end{bmatrix} A_k \tag{A.4}$$

and

$$\Gamma_{D_{k,N}} = \begin{bmatrix} C_{k+1} \left[\coprod_{l=1}^{0} A_{k+l}\right] B_k & D_{k+1} & \cdots & 0 & 0 \\ C_{k+2} \left[\coprod_{l=1}^{1} A_{k+l}\right] B_k & C_{k+2} \left[\coprod_{l=2}^{1} A_{k+l}\right] B_{k+1} \quad D_{k+2} & & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ C_{k+N} \left[\coprod_{l=1}^{N-1} A_{k+l}\right] B_k & C_{k+N} \left[\coprod_{l=2}^{N-1} A_{k+l}\right] B_{k+1} & \cdots & C_{k+N} \left[\coprod_{l=N}^{N-1} A_{k+l}\right] B_{k+N-1} & D_{k+N} \end{bmatrix}. \tag{A.5}$$

Appendix A.2. Constraints

Here we derive the linear inequality constraints (8) and (9) representing the actuator and state constraints

Appendix A.2.1. Actuator constraints The actuator amplitude constraints are given as:

$$U_{\min,k} \leq U_{k,N} \leq U_{\max,k}. \tag{A.6}$$

Using (6) these constraints can be rewritten as:

$$\begin{bmatrix} I \\ -I \end{bmatrix} \tilde{U}_{k,N} \leq \begin{bmatrix} U_{\max,k} - U^0_{k,N} \\ -U_{\min,k} + U^0_{k,N} \end{bmatrix}. \tag{A.7}$$

Following (11), actuator ramp-rate constraints are given as:

$$\Delta U_{\min,k} \leq \Delta U_{k,N} \leq \Delta U_{\max,k}. \tag{A.8}$$

Using (6) and (11) these constraints can be rewritten as:

$$\begin{bmatrix} \Gamma_\Delta \\ -\Gamma_\Delta \end{bmatrix} \tilde{U}_{k,N} \leq \begin{bmatrix} \Delta U_{\max,k} - \Gamma_\Delta U^0_{k,N} \\ -\Delta U_{\min,k} + \Gamma_\Delta U^0_{k,N} \end{bmatrix}. \tag{A.9}$$

The mixed actuator constraints are given as:

$$A_{U,\mathrm{mix}} U_{k,N} \leq b_{U,\mathrm{mix}}. \tag{A.10}$$

Using (6) these constraints can be rewritten as:

$$A_{U,\mathrm{mix}} \tilde{U}_{k,N} \leq b_{U,\mathrm{mix}} - A_{U,\mathrm{mix}} U^0_{k,N}. \tag{A.11}$$

Combining now the equations (A.7), (A.9) and (A.11) we obtain:

$$\underbrace{\begin{bmatrix} I \\ -I \\ \Gamma_\Delta \\ -\Gamma_\Delta \\ A_{U,\mathrm{mix}} \end{bmatrix}}_{\mathrm{A}_{\mathrm{inp}},k} \tilde{U}_{k,N} \leq \underbrace{\begin{bmatrix} U_{\max,k} - U^0_{k,N} \\ -U_{\min,k} + U^0_{k,N} \\ \Delta U_{\max,k} - \Gamma_\Delta U^0_{k,N} \\ -\Delta U_{\min,k} + \Gamma_\Delta U^0_{k,N} \\ b_{U,\mathrm{mix}} - A_{U,\mathrm{mix}} U^0_{k,N} \end{bmatrix}}_{\mathrm{b}_{\mathrm{inp}},k}, \tag{A.12}$$

which defines the matrices as given in (8).

Appendix A.2.2. State constraints State constraints are defined as:

$$A_{X,\mathrm{mix}} X_{k+1,N} \leq b_{X,\mathrm{mix}} + \varepsilon I. \tag{A.13}$$

Rewriting (A.13) using (6) and (7) results in:

$$\underbrace{\begin{bmatrix} A_{X,\mathrm{mix}} \Gamma_{B_{k,N}} & -I \end{bmatrix}}_{A,\mathrm{state},k} \begin{bmatrix} \tilde{U}_{k,N} \\ \varepsilon \end{bmatrix} \leq \underbrace{\begin{bmatrix} b_{X,\mathrm{mix}} - A_{X,\mathrm{mix}} [X^0_{k+1,N} + \Gamma_{A_{k,N}} \tilde{x}_k] \end{bmatrix}}_{b_{\mathrm{state},k}}, \tag{A.14}$$

which defines the matrices as given in (9).

Appendix A.3. Cost function

The cost function J_k as given in (10) can be rewritten by using (7) and (11):

$$\begin{aligned} J_k = {} & \tilde{x}_k^T \Gamma^T_{C_{k,N}} Q \Gamma_{C_{k,N}} \tilde{x}_k \\ & + \tilde{x}_k^T \left[2\Gamma^T_{C_{k,N}} Q \Gamma_{D_{k,N}} \right] \tilde{U}_{k,N} \\ & + \tilde{U}^T_{k,N} \left[\Gamma^T_{D_{k,N}} Q \Gamma_{D_{k,N}} + R_{\Delta\tilde{U}} \right] \tilde{U}_{k,N} \\ & + w_\varepsilon \varepsilon^2. \end{aligned} \tag{A.15}$$

In the minimization of J_k with respect to the future input sequence $\tilde{U}_{k,N}$, the constant part depending not on $\tilde{U}_{k,N}$ can be neglected. The reduced cost function $\hat{J}_k$ reads then as follows:

$$\hat{J}_k = \tilde{x}_k^T \underbrace{\left[2\Gamma^T_{C_{k,N}} Q \Gamma_{D_{k,N}} \right]}_{\Gamma_{J_1,k}} \tilde{U}_{k,N} + \tilde{U}^T_{k,N} \underbrace{\left[\Gamma^T_{D_{k,N}} Q \Gamma_{D_{k,N}} + R_{\Delta\tilde{U}} \right]}_{\Gamma_{J_2,k}} \tilde{U}_{k,N} + w_\varepsilon \varepsilon^2. \tag{A.16}$$

Appendix A.4. Quadratic programming matrices and vectors

By implementing the input vector parameterization in (12) and using $\xi_k^T = [\tilde{p}_{k,N}^T \quad \varepsilon]$, then the cost function (A.16), actuator constraints (A.12) and state constraints (A.14) can be rewritten into the standard form of the QP problem (13) at each time instant t_k:

$$\hat{J}_k = \tilde{x}_k^T \underbrace{\begin{bmatrix} \Gamma_{J_1,k}P_{\mathrm{map}} & 0 \end{bmatrix}}_{F_k} \xi_k + \xi_k^T \underbrace{\begin{bmatrix} P_{\mathrm{map}}^T\Gamma_{J_2,k}P_{\mathrm{map}} & 0 \\ 0 & w_\varepsilon \end{bmatrix}}_{2H_k} \xi_k, \tag{A.17}$$

$$\underbrace{\begin{bmatrix} A_{\mathrm{inp}} & 0 \\ A_{\mathrm{state}} & \end{bmatrix}\begin{bmatrix} P_{\mathrm{map}} & 0 \\ 0 & 1 \end{bmatrix}}_{A_{\mathrm{ineq,k}}} \xi_k \leq \underbrace{\begin{bmatrix} b_{\mathrm{inp,k}} \\ b_{\mathrm{state},k} \end{bmatrix}}_{b_{\mathrm{ineq},k}}, \tag{A.18}$$

$$\underbrace{\begin{bmatrix} I_{n_u} & 0 \end{bmatrix}\begin{bmatrix} P_{\mathrm{map}} & 0 \end{bmatrix}}_{A_{\mathrm{eq}}} \xi_k = \underbrace{\begin{bmatrix} \tilde{u}_k \end{bmatrix}}_{b_{\mathrm{eq},k}}. \tag{A.19}$$

which defines the matrices F_k, H_k, $A_{\mathrm{ineq,k}}$ and A_{eq} and the vectors $b_{\mathrm{ineq},k}$ and $b_{\mathrm{eq},k}$ in (13).

References

[1] Gribov, Y. *et al.* 2007 *Nuclear Fusion* **47** S385
[2] Snipes, J. *et al.* 2014 *Fusion Engineering and Design* **89** 507
[3] Ishida, S. *et al.* 2011 *Nuclear Fusion* **51** 094018
[4] Grosman, A. *et al.* 2013 *Fusion Engineering and Design* **88** 497
[5] Ou, Y. *et al.* 2008 *Plasma Physics and Controlled Fusion* **50** 115001
[6] Xu, C. *et al.* 2010 *Plasma Science, IEEE Transactions on* **38** 163
[7] Felici, F. *et al.* 2012 *Plasma Physics and Controlled Fusion* **54** 025002
[8] Moreau, D. *et al.* 2013 *Nuclear Fusion* **53** 063020
[9] Witrant, E. *et al.* 2007 *Plasma Physics and Controlled Fusion* **49** 1075
[10] Ou, Y. *et al.* 2007 *Fusion Engineering and Design* **82** 1153
[11] Felici, F. *et al.* 2011 *Nuclear Fusion* **51** 083052
[12] Gaye, O. *et al.* 2011 in *50th IEEE CDC and ECC (CDC-ECC)* 2638–2643
[13] Ou, Y. *et al.* 2011 *Control Engineering Practice* **19** 22
[14] Ouarit, H. *et al.* 2011 *Fusion Engineering and Design* **86** 1018
[15] Xu, C. *et al.* 2011 *Automatica* **47** 418
[16] Bribiesca Argomedo, F. *et al.* 2013 *Nuclear Fusion* **53** 033005
[17] Barton, J.E. *et al.* 2012 *Nuclear Fusion* **52** 123018
[18] Boyer, M.D. *et al.* 2012 in *American Control Conference (ACC), 2012* 2996–3001
[19] Kim, S. *et al.* 2012 *Nuclear Fusion* **52** 074002
[20] Boyer, M.D. *et al.* 2013 *Plasma Physics and Controlled Fusion* **55** 105007
[21] Gaye, O. *et al.* 2013 *Automatica* **49** 2795
[22] Vu, N.T.M. *et al.* 2014 in *19th IFAC World Congress, Cape Town, South Africa*
[23] Winter, A. *et al.* 2012 in *Real Time Conference (RT), 2012 18th IEEE-NPSS* 1–6
[24] Snipes, J. *et al.* 2010 *Fusion Engineering and Design* **85** 461
[25] Kamada, Y. *et al.* 2013 *Nuclear Fusion* **53** 104010
[26] Camacho, E. *et al.* 2004 *Model Predictive Control* Springer London ISBN 9781852336943
[27] Maciejowski, J. 2002 *Predictive Control with Constraints* Prentice-Hall
[28] Rossiter, J. 2013 *Model-Based Predictive Control: A Practical Approach* Taylor & Francis ISBN 9780203503966
[29] Felici, F. *et al.* 2012 in *Proc. 24th Int. Conf. on Fusion Energy (San Diego, 2012), EX/P3-12*
[30] Hinton, F.L. *et al.* 1976 *Rev. Mod. Phys.* **48** 239
[31] Sauter, O. *et al.* 1999 *Physics of Plasmas (1994-present)* **6**
[32] Sauter, O. *et al.* 2002 *Physics of Plasmas (1994-present)* **9**
[33] Erba, M. *et al.* 1998 *Nuclear fusion* **38** 1013
[34] van Dongen, J. *et al.* 2014 *In preparation*
[35] Artaud, J. *et al.* 2010 *Nuclear Fusion* **50** 043001
[36] Hogeweij, G. *et al.* 2013 *Nuclear Fusion* **53** 013008
[37] Grüne, L. *et al.* 2011 *Nonlinear Model Predictive Control: Theory and Algorithms* Springer
[38] Hespanha, J.P. 2009 *Linear Systems Theory* Princeton Press, Princeton, New Jersey
[39] Grimble, M.J. *et al.* 2014 *Nonlinear Industrial Control Systems: Design and Applications* Springer, London
[40] Richalet, J. *et al.* 2009 *Predictive Functional Control* Springer ISBN 9781848824935
[41] Wills, A. 2007 *Quadratic Programming in C, Revision 2007* http://sigpromu.org/quadprog/
[42] Ferreau, H.J. *et al.* 2008 *International Journal of Robust and Nonlinear Control* **18** 816
[43] Hobirk, J. *et al.* 2012 *Plasma Physics and Controlled Fusion* **54** 095001

[44] Ljung, L. 1998 *System Identification: Theory for the User* Pearson Education

[45] Skogestad, S. *et al.* 2005 *Multivariable feedback control: analysis and design* John Wiley ISBN 9780470011676