Model predictive profile control and actuator management in tokamaks

Bert Maljaars
Model predictive profile control
and actuator management
in tokamaks

Bert Maljaars

2017
This work has been performed within the framework of the NWO-RFBR Center of Excellence (grant 047.018.002) on Fusion Physics and Technology, as well as within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. This work is also supported by the Netherlands Organisation for Scientific Research (NWO) via the Innovational Research Incentives Scheme.

The work described in this thesis was carried out at the Eindhoven University of Technology and experiments are performed on the TCV tokamak at the Swiss Plasma Center in Lausanne, Switzerland.

A catalogue record is available from the Eindhoven University of Technology Library. ISBN: 978-90-386-4254-3

Typeset using \LaTeX.

Reproduction: Ipskamp Drukkers B.V., Enschede, the Netherlands.
Cover: Illustration of magnetic field lines at last closed magnetic flux surface. The figure is created using Matlab3Dtools [Albergante and Felici, 2011].

©2017 by E. Maljaars. All rights reserved.
Model predictive profile control and actuator management in tokamaks

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op donderdag 4 mei 2017 om 16:00 uur

door

Bert Maljaars

geboren te Vlissingen
Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter: prof.dr. L.P.H. de Goey  
1ᵉ promoter: prof.dr. M.R. de Baar  
2ᵉ promoter: prof.dr.ir. M. Steinbuch  
copromotor: dr.ir. F.A.A. Felici  
leden: prof.dr.ir. S. Weiland  
        prof.dr.ir. M. Mattei (Universitá della Campania ‘Luigi Vanvitelli’)  
        prof.dr.ir. B. de Schutter (Technische Universiteit Delft)  
        dr. J.A. Snipes (ITER Organization)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.
Controlled nuclear fusion is a potential sustainable energy source to complement other energy sources in meeting society’s ever growing energy demand. The furthest developed nuclear fusion device is the tokamak, in which a plasma (ionized gas) is confined in a axisymmetric donut-shaped vacuum chamber using magnetic fields and heated to achieve fusion conditions.

Control of key plasma parameters in tokamaks such as the safety factor profile and plasma pressure is important to ensure reliable high-performance operation while remaining in stable plasma conditions by satisfying operational limits. As the highest plasma performance is often achieved close to operational and actuator limits in tokamaks, profile controllers are required that can effectively deal with these limits and these are not yet available.

In Part I of this thesis, a model predictive control (MPC) approach is exploited to control key plasma parameters in tokamak plasmas that can deal effectively with actuator and operational limits. First an MPC controller is designed using multiple linear models to track a predefined reference safety factor profile evolution. Its performance is demonstrated in closed-loop simulations with parameters representative of the ITER tokamak, where successful tracking is obtained in the presence of under- or overestimated heat transport while satisfying an operational limit. It is also shown that the controller can provide warnings if operational limit violations are expected in the near future.

Next an MPC controller with extended capabilities is presented where, among others, disturbance estimation is included to improve steady state tracking and constraint handling. Analysis of the controller performance in closed-loop simulations with parameters representative of high performance plasmas in the ASDEX-Upgrade tokamak demonstrates successful tracking of the safety factor profile and plasma thermal stored energy with minimum steady state error.
Real-time changes in references, actuator and operational limits are effectively handled.

Finally the performance of an input-constrained MPC controller is demonstrated in both simulations and experiments on the TCV tokamak. Successful tracking of the estimated safety factor profile as well as plasma pressure is achieved under uncertain plasma conditions and disturbances. The controller exploits the knowledge of the time-varying actuator limits in the actuator input calculation such that fast transitions between targets are achieved without overshoot. A controller development and implementation environment is also presented that facilitates thorough off-line preparation and experimental implementation of profile controllers in the TCV tokamak.

In addition to this plasma profile control task, other control tasks may need to be executed during a tokamak experiment that often require the same set of actuators. This calls for a systematic integration of multiple plasma control tasks that share a set of actuators in tokamaks.

Part II of this thesis starts with evaluating possible architectures for the plasma control system (PCS) to achieve integrated control. First a PCS architecture with active management of actuators is used to demonstrate integrated control of the safety factor profile and neoclassical tearing modes (magnetic instabilities) in simulations with parameters that are representative of the ITER tokamak. Secondly, multiple architectures for integrated control with shared actuators are evaluated and recommendations are given to choose a specific scheme dependent on the size and complexity of the actuator system involved.

Finally an efficient actuator allocation algorithm for Heating and Current Drive (H&CD) actuators is developed. The actuator allocation problem is formulated as a Mixed-Integer Programming optimization problem, allowing to efficiently search for the best allocation option without the need to compute all allocation options. The desired allocation behavior and allocation feasibility can be easily specified to meet user specific needs. The algorithm’s performance is demonstrated in examples involving the full proposed ITER H&CD system, where the desired allocation behavior is clearly achieved and the required computational time allows for real-time implementation in ITER.

This thesis contributes to the reliable high performance operation of tokamaks within operational limits, where multiple control tasks share a set of actuators, by developing control algorithms based on state-of-the-art control engineering practices. It encourages further extension and exploration of the developed model predictive profile control and actuator management tools for the operation of (future) tokamaks.
Model predictive profile control and actuator management in tokamaks

The ever increasing energy demand of mankind requires sustainable energy sources that can complement those presently existing. Controlled nuclear fusion may be a viable alternative energy source that has the potential to produce no carbon dioxide at a virtually unlimited supply of the required fuels. Nuclear fusion is the process inside our sun that provides its energy and the same process can be achieved on earth in a reactor. The tokamak is the furthest developed nuclear fusion device where a hot gas (called a plasma) is confined by magnetic fields and heated by heating sources to temperatures of millions degrees Celsius.

Achieving and maintaining an optimal magnetic field structure and pressure inside this plasma is important to ensure reliable operation of tokamaks at maximum performance. In the first part of this thesis controllers are designed to accomplish this by intelligently adjusting the various heating powers during an experiment. Comparing it to a far less complex control problem, a room temperature control system at home also adjusts the heating to achieve the desired room temperature. The successful performance of the designed controllers is demonstrated in simulations for three tokamaks across Europe and also in experiments on the TCV tokamak in Switzerland.

In addition to this control task, other control tasks may need to be performed during a tokamak experiment that often require the same set of heating sources. In the second part of this thesis actuator management is considered to make optimal use of the available heating sources to achieve the various control task goals. The most striking contribution here is an efficient algorithm that can quickly select the best assignment of heating sources in the future ITER tokamak from about 100,000,000,000,000 feasible assignments.

This thesis contributes to establishing the reliable operation of tokamaks at maximum performance that may supply coming generations with a sustainable and environmentally friendly energy source.
Samenvatting

Model predictive profile control and actuator management in tokamaks

De wereldwijd stijgende energievraag vereist de ontwikkeling van duurzame energiebronnen die de bestaande energiebronnen kunnen aanvullen. Kernfusie is een duurzame energiebron die de potentie heeft om geen CO2 te produceren en waarvoor een vrijwel oneindige hoeveelheid grondstoffen beschikbaar is. Kernfusie is het proces dat onze zon voorziet van energie en datzelfde proces kan ook bereikt worden op aarde in een reactor. De tokamak is de verf ontwikkelde kernfusie reactor waarin een heet gas (plasma) opgesloten wordt met magneetvelden en verhit wordt door verschillende verhittingsbronnen tot temperaturen van miljoenen graden Celsius.

Het bereiken en behouden van de optimale magneetveldstructuur en de druk in het plasma is belangrijk voor het betrouwbaar functioneren van tokamaks bij maximale prestaties. In het eerste deel van dit proefschrift zijn regelaars ontworpen die dit kunnen bereiken door op een slimme manier het verhittingsvermogen aan te passen gedurende een experiment. Hoewel het een veel eenvoudiger regelprobleem is, kan dit principe vergeleken worden met de regeling van de kamertemperatuur thuis waarbij de CV-installatie voortdurend het verhittingsvermogen aanpast om de gewenste kamertemperatuur te bereiken. Het succesvol functioneren van de in dit proefschrift ontworpen regelaars is aangetoond in simulaties voor drie verschillende tokamaks in Europa en daarnaast in experimenten op de TCV tokamak in Zwitserland.

Naast deze regeltaak moeten vaak ook andere regeltaken worden uitgevoerd tijdens een tokamak experiment, waarbij die regeltaken tegelijkertijd dezelfde set van verhittingsbronnen moeten gebruiken. In het tweede deel van dit proefschrift is daarom onderzocht hoe optimaal gebruik gemaakt kan worden van de beschikbare verhittingsbronnen (actuator management) zodat de verschillende doelen van de regeltaken worden bereikt. Een belangrijke bijdrage in dit tweede deel is een algoritme dat in de toekomstige ITER tokamak efficiënt de beste
toewijzing van verhittingsbronnen aan regeltaken kan selecteren uit ongeveer 100.000.000.000.000.000 toewijzingsmogelijkheden.

Dit proefschrift draagt bij aan de ontwikkeling van het betrouwbare functioneren van tokamaks bij maximale prestaties om toekomstige generaties te kunnen voorzien van een duurzame en milieuvriendelijke energiebron.
Contents

1 Introduction
  1.1 Nuclear fusion for sustainable energy supply .......................... 1
  1.2 Tokamak ........................................................................ 2
  1.3 Tokamak control problems involving H&CD actuators .............. 7
  1.4 A control-oriented summary of the plant and control problem ... 9
  1.5 Research objectives and contributions ................................. 10

I Model predictive profile control 17

2 Control of the tokamak safety factor profile with time-varying constraints using MPC 19
  2.1 Introduction .................................................................... 20
  2.2 Methodology ................................................................... 22
  2.3 Simulation setup .............................................................. 33
  2.4 Simulation results ............................................................ 42
  2.5 Discussion ....................................................................... 47
  2.6 Conclusions .................................................................... 49
  Appendices ........................................................................... 51
  2.A Definitions and derivations ............................................... 51

3 Model predictive control of the current density distribution and stored energy in tokamaks using trajectory linearizations 55
  3.1 Introduction ..................................................................... 56
  3.2 Process modeling and controller requirements ....................... 58
  3.3 Controller design ............................................................ 61
  3.4 Performance analysis in simulations .................................... 68
  3.5 Conclusions and outlook .................................................. 72
4 Profile control simulations and experiments in TCV: a controller test environment and results using model predictive controller 75
  4.1 Introduction ........................................... 76
  4.2 Modeling plasma transport using RAPTOR ............... 78
  4.3 Plasma scenario and control problem definition .......... 79
  4.4 Controller development and implementation environment .. 85
  4.5 Model Predictive Controller design ..................... 90
  4.6 Simulation results ..................................... 94
  4.7 Experimental results on TCV ........................... 96
  4.8 Qualitative comparison of MPC controller performance to other control methods. .............................. 104
  4.9 Conclusions and outlook ................................. 107
Appendices .................................................... 109
  4.A Available information to controllers ..................... 109
  4.B Details MPC controller design ......................... 110
  4.C Analysis aspects experimental results ................... 114

II Actuator management for integrated control 119

5 Simultaneous control of plasma profiles and neoclassical tearing modes with actuator management in tokamaks 121
  5.1 Introduction ........................................... 122
  5.2 Control architecture .................................. 123
  5.3 Simulation results ..................................... 123
  5.4 Conclusion and outlook ................................ 127

6 Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm. 129
  6.1 Introduction ........................................... 130
  6.2 Overview and brief evaluation of architectures for integrated control 131
  6.3 H&CD actuator allocation problem formulation .......... 138
  6.4 Performance of H&CD actuator allocation algorithm ...... 152
  6.5 Conclusions and outlook ................................ 163
  6.6 Acknowledgements ..................................... 164
Appendices .................................................... 165
  6.A Current drive parametrization .......................... 165
  6.B Additional cost penalties ............................... 165
  6.C Details regarding constraints ........................... 167
  6.D Remarks on alternative problem formulations ......... 170
  6.E Practical issues ....................................... 171
7 Conclusions, discussion and recommendations
  7.1 Conclusions ................................................................. 173
  7.2 Discussion and recommendations ..................................... 176

Bibliography ........................................................................... 180
Dankwoord ............................................................................ 191
List of publications .................................................................. 193
Curriculum vitae ....................................................................... 195
Chapter 1

Introduction

Controlled nuclear fusion is considered as a potential sustainable energy source to complement other energy sources in meeting society’s ever growing energy demand. The furthest developed nuclear fusion device is the tokamak, in which a plasma is confined using magnetic fields and heated to achieve fusion conditions. Control of key plasma parameters in tokamak plasmas is important to ensure reliable high-performance operation while not exceeding operational limits.

In this thesis, contributions to two distinct challenges in the control of nuclear fusion are made. The first part aims at developing controllers for a number of key plasma parameters that can deal effectively with actuator and operational limits. The second part aims at evaluating architectures and developing an algorithm to systematically integrate multiple plasma control tasks that share a set of actuators.

In this chapter we will first introduce the reader to nuclear fusion, the tokamak and a set of tokamak control problems that require heating and current drive actuators. Next, the research objectives of this thesis are introduced and the contributions in this thesis to achieve these objectives are summarized. Finally the outline of this thesis is presented.

1.1 Nuclear fusion for sustainable energy supply

The ever increasing energy demand of mankind requires sustainable energy sources that can complement those existing [International Energy Agency, 2016]. Controlled nuclear fusion in a reactor on earth may be such a viable alternative energy source [Freidberg, 2007]. This energy source has the potential to produce no carbon dioxide and relatively low amounts of short-lived nuclear waste and it also has a high power per land surface area. In addition, there exists a virtually
unlimited supply of the required fuel.

In nuclear fusion hydrogen nuclei fuse into a larger helium nucleus [Freidberg, 2007]. To fuse the reacting elements, the Coulomb barrier between the elements needs to be overcome by the attracting strong nuclear force and hence high temperatures and densities are required to bring the elements close enough together. As the reactivity of this process has a maximum well above 100 MK, the fuel will be in the - fully ionized - plasma state. A point can be reached where the plasma energy can be sustained sufficiently long against energy losses by means of the power released in the fusion reactions [Freidberg, 2007]. This point is known as ignition and is described in the Lawson criterion that states that the triple product of density $n$, temperature $T$ and energy confinement time $\tau_E$ should exceed a threshold:

$$nT\tau_E \geq 3 \cdot 10^{21} \text{keV m}^{-3} \text{s}.$$ (1.1)

The tokamak is the furthest developed nuclear fusion device and closest to realize this ignition condition. The ITER tokamak [ITER Organization, 2017], currently under construction, is designed to exceed this threshold by operating at low density ($n \approx 10^{20} \text{m}^{-3}$), high temperatures ($T \approx 10 \text{keV}$) and low thermal losses ($\tau_E \approx 3.5 \text{s}$).

1.2 Tokamak

The tokamak is the furthest developed nuclear fusion device using magnetic confinement [Wesson, 2011]. As this is the considered device in this thesis, we will next introduce the reader briefly to the tokamak. A more extensive control-oriented introduction to tokamaks is given in [Pironti and Walker, 2005, 2006].

1.2.1 Magnetic equilibrium and plasma profiles

In a tokamak an ionized hot gas (plasma) is confined in an axisymmetric donut-shaped vacuum chamber using a helical magnetic field. As electrons and ions are separated in a hot plasma, these will feel the influence of magnetic and electrical fields. This concept is used in magnetic confinement fusion to keep the hot plasma away from the wall. An overview of the coils that produce the confining magnetic field in a tokamak is given in Figure 1.1.

The confining magnetic field is produced by a set of coils that consists of the inner poloidal field coils (called the central solenoid), toroidal field coils and outer poloidal field coils. At fusion-relevant temperatures, the plasma is a good electrical conductor. A tokamak plasma carries an electrical current, which is primarily induced by a varying current in the central solenoid. This solenoid forms the primary circuit of a transformer and the plasma forms the secondary circuit. Toroidal field magnets are used to create a magnetic field in the toroidal
1.2 Tokamak

Figure 1.1. Configuration of magnets in a tokamak producing the confining magnetic field. Courtesy of EUROFusion (http://www.euro-fusion.org/)

direction (cyan arrow). Moreover a poloidal magnetic field (green arrows) is necessary for good confinement of the plasma, creating an equilibrium in which the plasma pressure is balanced by the magnetic forces. The poloidal field is produced by the toroidal plasma current and also the outer poloidal magnets. The combination of the poloidal and toroidal magnetic fields produces the helical magnetic field lines.

Within the tokamak vessel, loci of constant poloidal magnetic flux form nested closed surfaces, on which the plasma kinetic pressure is approximately constant. For the purposes of this thesis, we consider the tokamak plasma to be axisymmetric, such that the plasma may be visualized in a 2D cross-section as seen in Figure 1.2 (left). Note that, within the plasma, the projection of the surfaces of constant flux onto this 2D cross section form closed contours.

The flux surfaces can be indexed by a radial flux label $\rho$, where at the magnetic axis $\rho = 0$ (plasma center) and at the last closed flux surface (LCFS) $\rho = 1$. This LCFS determines the edge of the confined plasma, while outside this flux surface the fieldlines will reach the plasma facing wall.

Due to this specific topology of the magnetic field, many plasma quantities can be expressed as a function of this flux label $\rho$ and these are known as plasma profiles. Two examples of typical profiles are given in Figure 1.2, the electron temperature ($T_e(\rho)$) profile and the so-called safety factor profile ($q$-profile).

The twist of the helical magnetic field determines the value of the $q$-profile, which is defined as the number of toroidal turns ($m$) divided by the number
of poloidal turns ($n$) after which a field line closes on itself. Since the spatial distribution of current in the plasma and the poloidal field are linked, the $q$-profile is also closely related to the current profile. As explained next, this $q$-profile plays an important role in the performance and stability of the tokamak plasma.

The evolution of these and other plasma profiles can be described by a number of nonlinear coupled partial differential equations (fluid transport equations together with the Maxwell equations). These equations are known as the magnetic-hydrodynamic (MHD) description of the plasma and yield a 1-D distributed parameter system, which is actually coupled to the evolution of the 2D magnetic equilibrium.

1.2.2 Tokamak operating scenarios, performance and stability

A tokamak can be used in different operating points in the plasma parameter space. Classes of operating points are known as tokamak scenarios that are characterized by specific $q$-profile ranges, as these determine largely the performance and MHD-stability of these plasmas [Gormezano et al., 2007]. Before some of these tokamak operating scenarios are introduced, we will first briefly discuss plasma performance and plasma MHD-stability.

Plasma performance can be defined following the Lawson criterion (1.1) as requiring a long energy confinement time $\tau_E$. A high $\tau_E$ requires low radial thermal transport, resulting in high temperatures for the same input power (power balance). Radial thermal transport has been shown to be dependent on the $q$-profile shape [Doyle et al., 2007; Gormezano et al., 2007].

Unstable and performance degrading plasma behavior is observed in specific regions of the operation state space and are caused by MHD-instabilities. Several MHD-instabilities are associated with specific rational $q$-values and are
characterized by the reconnection of magnetic field lines occurring under certain conditions [Zohm, 2014]. Two important MHD-instabilities are the sawtooth instability and neoclassical tearing modes (NTM) that both can be controlled (see Section 1.3.2). The sawtooth instability occurs in a region where \( q < 1 \) and results in a periodic redistribution of particles in the plasma, flattening the temperature profile inside the \( q = 1 \) surface. Sawteeth instabilities with large amplitude are also seen to trigger NTMs. NTMs are magnetic islands that may develop at rational \( q \)-surfaces (e.g. at \( q = \frac{3}{2} \) or \( q = \frac{3}{2} \)) resulting in increased heat transport over the island and hence confinement degradation (lower \( \tau_E \)). If these NTM-instabilities are allowed to grow unboundedly, these may lead to a sudden plasma termination (disruption). NTMs are more likely to develop in conditions of high plasma pressure [Sauter et al., 1997, 2002b], especially at high normalized plasma pressure \( \beta_N \sim p/I_p \).

A set of typical tokamak operating scenarios with their corresponding \( q \)-profile regions are given in Figure 1.3 [Gormezano et al., 2007]:

- **Inductive scenario**, where a region of the \( q \)-profile has \( q < 1 \). This scenario has typically sawteeth.

- **Hybrid scenario**, where \( q > 1 \) and a broad flat region of the \( q \)-profile in the center has been shown to lead to higher confinement [Doyle et al., 2007].

- **Advanced scenarios**, where \( q > 1 \) and the nonmonotonic \( q \)-profiles have an off-axis minimum (called reverse shear), where reduced thermal transport leads to the formation of a transport barrier [Doyle et al., 2007].
1.2.3 Heating and current drive actuators

The plasma temperature and current profiles can be affected by spatially distributed heating and current drive (H&CD) actuators. The plasma current driven in the plasma provides an ohmic contribution to the current profile and can also resistively heat the plasma. However, as the plasma resistivity scales as $\eta \sim T_e^{-3/2}$, the ohmic heating efficiency quickly drops for high temperature plasmas. To heat the plasma to temperatures required for fusion, auxiliary H&CD actuators have been developed that are briefly introduced here:

Neutral Beam Injection (NBI) H&CD. Highly energetic neutral particles (e.g. 1MeV in ITER) can be injected into the plasma where they will transfer their kinetic energy in collisions with ions. NBI can be used to provide bulk H&CD.

Electron Cyclotron (EC) H&CD. The electrons and ions in a plasma spiral along the magnetic fieldlines with the gyro-frequency. By injecting electromagnetic radio-frequency (RF) waves into the plasma, these waves can resonate with the rotation of the particles. Electron Cyclotron H&CD uses this concept to couple power into the plasma. These RF-waves are generated in a gyrotron and directed via waveguides and launchers into the plasma, where they are absorbed in the vicinity of a resonant magnetic surface where the electron cyclotron frequency coincides with the frequency of the injected waves. There they locally transfer heat to the electrons and additionally drive a current in the plasma. By steering the launchers, control over the deposition location in the plasma is available.

Ion Cyclotron (IC) H&CD. The same concept as for EC can be used to transfer heat to the plasma ions, but here the RF waves need to be emitted in an antenna close to the plasma at the ion cyclotron frequency. These waves are then locally absorbed in the plasma at the resonant magnetic surface with the same resonance frequency as the injected RF waves.

Although not actuators themselves, the plasma can also have two sources of self-generated heating and current drive. The plasma in fusion conditions can heat itself by high energetic charged $\alpha$-particles produced in fusion reactions between the reactants Deuterium and Tritium. In addition, a significant amount of so-called bootstrap current can be self-driven in the plasma in the presence of a pressure gradients [Peeters, 2000].

1.2.4 Considered tokamaks in this thesis

Many tokamaks have been in operation and some are under construction, all for scientific research [Wesson, 2011]. The tokamaks considered in this thesis are the ITER tokamak presently under construction in St. Paul-lez-Durance [ITER.
1.3 Tokamak control problems involving H&CD actuators

In nuclear fusion reactors, the plasma needs to be actively monitored and controlled by automated control systems. The Plasma Control System (PCS) inside the tokamak needs to ensure that references for controlled variables are achieved and maintained throughout the discharge. These references represent the desired plasma operating point in terms of plasma quantities. While some processes in a tokamak are open-loop unstable such that the tokamak cannot be operated without any feedback control, most processes are open-loop stable. Present tokamak operation often relies on feedforward-control, which has been sufficient for most plasma physics research. However, without feedback control the quantities of interest will likely vary between experiments due to varying hardware and plasma conditions, disturbances and plasma events during experiments. Reliable high performance operation of tokamaks requires feedback control to ensure good reference tracking performance to achieve and maintain the desired plasma performance.

In this thesis we consider only control problems that require the use of the H&CD actuators. Therefore we introduce only these control problems here, with an emphasis on the control of plasma profiles and parameters as that control
problem constitutes the main part of this thesis.

1.3.1 Control of plasma profiles and parameters

Control over plasma profiles and parameters is important in order to maximize plasma performance while ensuring MHD-stability in hybrid and advanced plasma scenarios [Gribov et al., 2007]. As discussed in Section 1.2.2, plasma performance and MHD-stability in a tokamak plasma are related to the $q$-profile and operational limits. Therefore, it is desired to have control over plasma profiles, in particular the $q$-profile. Also control over other plasma parameters is important, such as the plasma pressure or stored energy as these, together with the $q$-profile, define an operating point in the plasma parameter space. The $q$-profile can be controlled by localized current drive by the H&CD actuators, while the temperature and density profile shapes are hard to control, and at most bulk quantities can be controlled such as volume averages of these profiles.

The highest plasma performance in tokamaks is often achieved close to actuator limits and in addition close to operational limits. To minimize costs, the installed power of H&CD actuators often does not greatly exceed the nominal required power at an operating point. Even if these actuator and operational limits may not be reached at the target operating point, these are often still restrictive during transient phases (e.g. current ramp-up, current ramp-down or transitions between operating points). Therefore, it is desirable that a profile controller should not only track a reference for plasma profiles and parameters that defines the operating point with the desired plasma performance, but, at the same, time deal with these actuator limits and ensure operation within the stable plasma parameter space.

Feedback control of plasma profiles and parameters requires detailed knowledge about the state of these quantities in real-time. This is not trivial in a tokamak, as measuring the internal distribution of plasma profiles is challenging. Therefore, state reconstruction techniques are employed in control of tokamak plasmas to reconstruct these profiles and parameters using real-time equilibrium reconstruction (e.g. [Moret et al., 2015]) or observers [Felici et al., 2014a]. These state reconstructions are constrained as much as possible by the available measurements and complemented by carefully chosen parameterizations (equilibrium reconstruction) or model-based predictions (observers).

1.3.2 Other control problems involving H&CD actuators

Various other control problems also require to use the H&CD actuator systems [Henderson et al., 2015; Snipes et al., 2012]. These are very briefly introduced here:

- NTM control: Neoclassical Tearing Modes (NTMs) are magnetic islands that may develop at rational $q$-surfaces resulting in increased heat trans-
port and hence confinement degradation. These can be controlled or suppressed by depositing EC current drive within the magnetic island.

- Sawtooth control: the period of this periodic MHD-instability at the $q = 1$ surface (known as the sawtooth instability) can be modified by using both EC and IC, hereby changing the probability to trigger NTMs.

- Impurity control: plasma impurities tend to accumulate in the center of the plasma. This can lead to reduced confinement by their thermal radiation, but can be expelled from the plasma center by increasing temperature gradients, for example by using central heating with EC.

## 1.4 A control-oriented summary of the plant and control problem

In this thesis, for control purposes, the tokamak plant will be treated as a nonlinear dynamical system with inputs, state and outputs. This is visualized in Figure 1.5.

![Figure 1.5. Tokamak plant description as nonlinear dynamical system with inputs $u(t)$, states $x(t)$ and outputs $y(t)$.](image)

The considered actuator systems in this thesis are the spatially distributed Heating and Current Drive (H&CD) actuators that receive actuator inputs $u(t)$. The most relevant part of the tokamak plant dynamics $\dot{x}(t) = f(x(t), u(t))$ related to these actuators is the evolution of radial profiles of electron temperature $T_e(\rho)$ and magnetic flux $\psi(\rho)$ (where $\rho$ is a radial label). For this plasma transport process, the process state $x(t)$ may be expressed in terms of these two profiles that are highly nonlinearly coupled. Relevant outputs $y(t) = h(x(t), u(t))$ in this thesis are the safety factor profile $q(\rho)$ (related to $\psi(\rho)$), and the scalar...
thermal stored energy in the plasma $W_{th}(\rho)$ or the scalar plasma pressure $\beta$ (both related to $T_e(\rho)$).

The plant dynamics show two distinct time scales:

- Fast dynamics for electron temperature evolution ($T_e$): energy confinement time $\tau_E$.
- Slow dynamics for the magnetic field evolution: resistive diffusion time $\tau_R$.

The ratio between these time scales is typically similar ($\tau_R/\tau_E \approx 25$) for the tokamaks considered in this thesis, but their absolute value varies with the size of the tokamak (see Figure 1.4). The fast energy confinement time $\tau_E$ ranges for the considered tokamaks (see Figure 1.4) between 5s (ITER) to 5ms (TCV).

The plant is subjected to actuator and operational limits. Parts of the operational state space are prone to unstable and performance degrading behavior that may eventually stop the plasma, although damage to the machine is not possible as other protection systems are in place to protect people and machinery. Therefore it is important to ensure operation within the stable plasma parameter range.

Control over the plasma profiles and parameters (given in outputs $y_k$) is required using the actuators $u_k$ to achieve and maintain the desired high performance operating point while satisfying the actuator and operational limits. This requires the use of control methods that can take both limits into account. The fast time scales in currently operational tokamaks (e.g. the mentioned 5ms in TCV) impose additional challenges on the design of such a controller.

### 1.5 Research objectives and contributions

Here we introduce the two research objectives and the contributions in this thesis to achieve these objectives.

#### 1.5.1 Part I: Model predictive profile control

As discussed in Section 1.3.1, profile control is required to achieve and maintain desired plasma profiles for high performance (advanced) tokamak operation, while ensuring the tokamak is operated within actuator limits and the stable plasma parameter space. Therefore, control methods are required that can effectively deal with these actuator limits and and physics limits.

Recently (and mainly throughout the course of this thesis work), many model-based profile controllers have been developed using a wide variety of controller models and control methods. The applied methods include adaptive control [Kim and Lister, 2012], backstepping control [Boyer et al., 2014], passivity-based control [Vu et al., 2014], Lyapunov control [Bribiesca Argomedo et al., 2013], linear-quadratic-integral control [Boyer et al., 2013; Moreau et al., 2013]
1.5 Research objectives and contributions

and robust control [Barton et al., 2014a]. Some of these have been implemented in experiments, others only in simulations. Some earlier work on profile control using linear multiple-input-multiple-output (MIMO) methods can be found in [Joffrin et al., 2003; Laborde et al., 2004; Mazon et al., 2002; Moreau et al., 2003].

The control methods mentioned above impose only actuator limits on the actuator signal after it has been computed by the controller. This approach has the important limitation that the controller is not aware of the actuator limits in the actuator input calculation itself and cannot anticipate for these. None of these methods can ensure satisfaction of operational limits.

Therefore the research objective for Part I of this thesis is formulated as:

**Research objective 1:**
Design a controller that ensures that desired plasma profile and parameters are achieved in the presence of disturbances while time-varying actuator and operational limits are satisfied.

In this thesis a model predictive control (MPC) approach is chosen to control the plasma profiles and parameters. MPC is a well-established advanced control method that has been used for decades to control MIMO processes in industry that are subjected to time-varying input and state constraints [Camacho and Bordons, 2004; Maciejowski, 2002; Mayne, 2014; Rossiter, 2013]. MPC uses a process model to predict the future evolution of states and controlled variables up to a horizon. This prediction model is used in an optimization problem to find the future actuator input sequence that minimizes a cost function penalizing the tracking error on the controlled variables. Time-varying actuator and state constraints can be imposed in this optimization problem. MPC will be explained in more detail in Section 2.2.3. Preliminary applications of MPC to profile control in simulations are presented in the literature [Ou et al., 2011; Ouarit et al., 2011], but these imposed only fixed actuator constraints. Very recently input-constrained MPC is also successfully applied in profile control experiments at the DIII-D tokamak [Wehner et al., 2016].

The specific contributions in this thesis to achieve the first research objective are:

**Contribution 1: Demonstration of MPC control of the safety factor profile in ITER simulations.** A model predictive controller is designed that uses multiple linearized models to control the safety factor profile while dealing with actuator and physics limits. The performance of the controller is analysed in a set of simulations with parameters representative of the ITER tokamak, where the controller successfully tracks the nominal inverse safety factor profile evolution under model variations and effectively acts on changes in actuator limits. The controller provides also warnings ahead of time about expected violations of operational and physics limits.
Contribution 2: Demonstration of MPC control of the safety factor profile and stored energy in ASDEX-Upgrade simulations. Here an MPC profile controller is designed that extends and improves the previous controller design such that real-time varying references can be tracked with minimum steady state error and an important physics limit can be taken into account. The controller performance is demonstrated in simulations with parameters representative of ASDEX-Upgrade H-mode experiments where effective handling of changes in references, actuator constraints and mixed input-output constraints is shown.

Contribution 3: Demonstration of MPC control of the safety factor profile and plasma pressure in TCV simulations and experiments. A linear input-constrained MPC controller is developed and tested in the TCV tokamak. Successful tracking of the inverse safety factor profile as well as the plasma pressure in the presence of uncertain plasma conditions and disturbances is demonstrated in TCV simulations and experiments. To enable the thorough testing and successful experimental implementation of profile controllers on the TCV tokamak, a profile controller development and implementation environment is designed, which will also be presented in this thesis.

The contributions 1,2 and 3 are summarized in Figure 1.6.

<table>
<thead>
<tr>
<th>Description</th>
<th>Contribution 1</th>
<th>Contribution 2</th>
<th>Contribution 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokamak</td>
<td>ITER</td>
<td>ASDEX Upgrade</td>
<td>TCV</td>
</tr>
<tr>
<td>Verification in</td>
<td>Simulations</td>
<td>Simulations</td>
<td>Simulations, experiments</td>
</tr>
<tr>
<td>Actuators</td>
<td>Plasma current, NBI, EC</td>
<td>Plasma current, NBI, EC</td>
<td>Plasma current, EC</td>
</tr>
<tr>
<td>Controlled Variables</td>
<td>q-profile</td>
<td>q-profile, stored energy</td>
<td>q-profile, plasma pressure</td>
</tr>
<tr>
<td>References for CV</td>
<td>Nominal trajectory of CV</td>
<td>RT-varying</td>
<td>RT-varying</td>
</tr>
<tr>
<td>Constraints on</td>
<td>Inputs, states</td>
<td>Inputs, states, combinations</td>
<td>Inputs</td>
</tr>
<tr>
<td>State space models</td>
<td>LTV</td>
<td>LTV</td>
<td>Single LTI</td>
</tr>
<tr>
<td>Disturbance estimation</td>
<td>None</td>
<td>State</td>
<td>State and output</td>
</tr>
<tr>
<td>Sample time</td>
<td>1s</td>
<td>10ms</td>
<td>1ms</td>
</tr>
</tbody>
</table>

Figure 1.6. Summary of the three contributions to research objective I.

1.5.2 Part II: actuator management for integrated control

The H&CD actuators in a tokamak are required for multiple control tasks, as was discussed in Section 1.3. The previous part of this thesis naturally raised questions such as which component of the Plasma Control System sets the time-varying actuator limits provided to the profile controller? How can actuator requests from a profile controller be realized by complex H&CD actuator systems
1.5 Research objectives and contributions

such as those proposed for the ITER tokamak? And if other control tasks that require the same actuators as used by the profile controller need to be executed simultaneously, how can conflicting requests to the actuators be avoided?

These questions have led to the second part of this thesis that we will introduce here.

In present-day devices, actuators are usually assigned to a single control task for an entire experiment. For instance some H&CD actuators may be assigned to control the plasma pressure, while some EC actuators are reserved to suppress an NTM if it appears. Executing multiple control tasks at the same time is sometimes performed in tokamaks and is known as integrated control in the controlled fusion community [Humphreys et al., 2005; Joffrin et al., 2007]. This is still an area of research and integrated control of all relevant phenomena is not performed routinely today.

New challenges arise for integrated control in view of future tokamaks such as ITER. There it will become increasingly important to use a limited set of H&CD actuators for multiple control tasks during plasma discharges in tokamaks. Also, during a plasma discharge the priority to execute a control task may vary in time due to plasma events and the availability of the actuators may change due to failure [Winter et al., 2014]. Therefore actuators can no longer be assigned for an entire experiment to a single control task, but may need to be reallocated in real-time. This is visualized in Figure 1.7.

The complexity of this integrated control problem with shared actuators is best illustrated for the proposed system of the ITER tokamak. This system consists of 24 gyrotrons that locally deposit power and current in the plasma via 11 steerable mirrors. This system may be used by at least 5 control tasks whose priority will change in time during the discharge [Henderson et al., 2015].

A PCS architecture defines the role of PCS components and the interfaces between these components. A coherent approach to the design of PCS architectures allowing for integrated control is presented in literature [Nouailletas et al., 2013; Ravenel et al., 2014; Treutterer et al., 2013, 2014a; Treutterer et al., 2014b]. A prominent role is given to a supervisory control layer that activates control tasks if necessary. Such a supervisor may also set control task priorities [Humphreys et al., 2015] that allow to allocate actuators in real-time to the control tasks according to these priorities.

One could think of several architectures to interface the prioritized control tasks on the one hand and actuator allocation on the other hand. However, it is not evident from literature what is the best architecture to interface the prioritized control tasks and actuator allocation for a given tokamak. For instance, it is not clear if the actuators should be assigned before a control task is executed, or afterwards based on the requests calculated by the control tasks.

Therefore the second research objective is formulated as:
Research objective 2: Evaluate Plasma Control System (PCS) architectures to integrate multiple control tasks requiring the same actuators.

The specific contributions in this thesis to achieve this second research objective are:

Contribution 4: Demonstration of integrated control of plasma profiles and NTMs using a PCS architecture including actuator management in ITER simulations A PCS architecture is designed that ensures that both control tasks (profile control and NTM-control) are aware of their assigned EC resources when executed. Closed-loop simulations for the ITER tokamak show that the proposed PCS design can effectively respond to the occurrence of an NTM by suppressing it, while at the same time the MPC profile controller (as designed in Chapter 5) maintains the profiles within operational limits, taking advantage of its supplied actuator limits.

Contribution 5: Evaluation of architectural designs for integrated control with shared actuators Possible architectures of the plasma control system for integrating multiple control tasks sharing actuators are
compared on several aspects. This evaluation confirms that hierarchical schemes are favorable and recommendations are given to choose a specific variant dependent on the scale and complexity of the actuator system and the number of control tasks involved.

An appropriate algorithm is required to allocate actuators in real-time according to the desired allocation behavior. Recently, an actuator allocation algorithm was developed to assign EC actuators to control tasks in the ASDEX-Upgrade tokamak [Rapson et al., 2015], and is successfully implemented in integrated control experiments [Rapson et al., 2016]. This algorithm computes in real-time the best actuator allocation option by evaluating a merit function for all options.

However, computing all allocation options for large and complex actuator systems like the one foreseen in ITER may not be feasible in real-time using the algorithm in [Rapson et al., 2015]. Hence an actuator allocation algorithm is required that can be executed sufficiently rapidly for real-time implementation on e.g. ITER.

Therefore the third research objective is formulated as:

**Research objective 3:** Develop a real-time actuator allocation algorithm that can deal with large and complex H&CD systems such as proposed for ITER.

The specific contribution in this thesis to achieve this third research objective is:

**Contribution 6:** Efficient actuator allocation algorithm capable of allocating ITER H&CD system in real-time

An algorithm is developed for the allocation of H&CD actuator systems using a Mixed-Integer Programming (MIP) optimization problem formulation. This formulation allows to quickly search for the best allocation option without the need to compute all allocation options. The desired allocation behavior can be clearly defined in a cost function, whereas actuator availability and infeasible allocation options can be described in constraints. Examples involving the full planned ITER H&CD system size demonstrate the algorithm’s capability to perform the actuator allocation in real-time in correspondence to the desired allocation behavior.

### 1.5.3 Outline of this thesis

The remainder of this thesis is organized as follows. The Chapters 2, 3 and 4 present the several MPC controller designs and their performance analysis for the ITER, ASDEX-Upgrade and TCV tokamaks respectively (Contributions 1, 2 and 3). The integrated control of the safety factor profile and NTMs
using actuator management in ITER closed-loop simulations (Contribution 4) is given in Chapter 5. Chapter 6 presents the analysis of architectural designs for integrated control with shared actuators (Contribution 5) as well as the developed actuator allocation algorithm (Contribution 6). Finally, conclusions and recommendations are given in Chapter 7.
Part I

Model predictive profile control
Chapter 2

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

---

The work presented in this chapter is published:
Maljaars, E., et al. (2015), Control of the tokamak safety factor profile with time-varying constraints using MPC, Nuclear Fusion 55(2), 023001. The text has been revised after feedback of the thesis committee.
Chapter 2. Control of the tokamak safety factor profile with time-varying constraints using MPC

2.1 Introduction

Successful high performance tokamak operation simultaneously requires plasma stability at high values of the normalized pressure $\beta_N$ as well as long energy confinement times $\tau_E$. 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 JT60-SA and WEST [Gribov et al., 2007; Grosman et al., 2013; Ishida et al., 2011; Snipes et al., 2014].

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 [Felici and Sauter, 2012a; Ou et al., 2008; Xu et al., 2010]. 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 [Moreau et al., 2013] 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 [Felici et al., 2011b; Ou et al., 2007; Witrant et al., 2007].

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 [Barton et al., 2012; Boyer et al., 2012, 2013; Bribiesca Argomedo et al., 2013; Gaye et al., 2011, 2013; Kim and Lister, 2012; Ou et al., 2011; Ouarit et al., 2011; Vu et al., 2014; Xu et al., 2011]. 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 [Bribiesca Argomedo et al., 2013; Ou et al., 2011; Ouarit et al., 2011], 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 [Winter et al., 2012]. A common example is electron cyclotron heating and current drive, that can be used for many of these tasks [Snipes et al., 2010]. 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 $\beta_N$ and $I_i^{(3)}$) and desired physics limits (e.g. $q(\rho) > 1$ in the case of hybrid scenarios) are satisfied, as argued in e.g. [Kamada et al., 2013]. 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 chapter 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 [Camacho and Bordons, 2004; Maciejowski, 2002; Rossiter, 2013]. MPC has already been proposed for $q$-profile control in [Ou et al., 2011; Ouarit et al., 2011]. 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 chapter 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 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 [Felici and Sauter, 2012a] 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
This chapter is organized as follows. In Section 2.2 we present the methodology of the MPC-controller design. The simulation setup is presented in Section 2.3. In Section 2.4, we show performance and merits of this MPC-controller for a number of typical cases. In Section 2.5 we discuss the results and possible extensions of the current MPC-design. The conclusion is provided in Section 2.6.

2.2 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.

2.2.1 Controller context

The envisioned implementation of the MPC-controller in the larger context of a tokamak plasma control system is given in Figure 2.1. 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 [Felici et al., 2012b] or by constrained equilibrium reconstruction) and fed to the MPC-controller.

To systematically develop the MPC-controller towards the envisioned implementation, the MPC-controller is as a first step interfaced with the nonlinear plasma transport simulator RAPTOR. This is illustrated in Figure 2.2. 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 [Maciejowski, 2002], requiring access to the actual state of the plasma, in this case, full knowledge of the plasma profiles. In the work in this chapter we have direct access to the state of RAPTOR, which simplifies the implementation. The supervisory controller is not considered in this chapter, instead time-varying constraints are manually provided to the MPC-controller.

2.2.2 RAPTOR: nonlinear plasma transport simulator

In this chapter we employ linearized models from RAPTOR inside the controller and use RAPTOR as simulator to test the controller. RAPTOR [Felici and Sauter, 2012a; Felici et al., 2011b] 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 $\psi$ as a function of the
2.2 Methodology

Figure 2.1. 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 [Felici et al., 2012b].

Figure 2.2. Simplified implementation as used in this chapter. The simulated plasma state is directly available to the MPC-controller.

normalized square-root toroidal flux $\rho$, represented by partial differential equations (PDEs) [Hinton and Hazeltine, 1976]. Bootstrap current and neoclassical conductivity are calculated using the Sauter-Angioni equations [Sauter et al., 1999, 2002a]. 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 [Erba et al., 1998]. Impurities are not considered other than their effect via the $Z_{\text{eff}}$ 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 [van Dongen et al., 2014] is shown that by careful choices of what elements to exclude from more comprehensive plasma transport simulations, RAPTOR is able to approach CRONOS [Artaud et al., 2010] simulations of ITER in [Hogeweij et al., 2012] within $\sim 15\%$ or better, with $< 3\text{ms}$ per time step.

Actuators considered in this chapter are the inductive and non-inductive heating and current drives (H&CD). The controlled variables are the inverse $q$-profile ($t = 1/q$) at several locations $\rho$ 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 chapter):

$$u_k = \begin{bmatrix} I_p(t_k) \\ P_{ec,1}(t_k) \\ P_{ec,2}(t_k) \\ P_{ec,3}(t_k) \\ P_{ec,4}(t_k) \\ P_{nbi}(t_k) \end{bmatrix}$$

$$x_k = \begin{bmatrix} \hat{T}_{e,\rho=0}(t_k) \\ \vdots \\ \hat{T}_{e,\rho=1}(t_k) \\ \hat{\psi}_{\rho=0}(t_k) \\ \vdots \\ \hat{\psi}_{\rho=1}(t_k) \end{bmatrix}$$

$$z_k = \begin{bmatrix} t_{\rho=0}(t_k) \\ \vdots \\ t_{\rho=1}(t_k) \end{bmatrix}$$

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 $\psi$ [Felici et al., 2011b]. 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 $\rho$-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 $u_k$ [Felici and Sauter, 2012a]:

$$\tilde{f}_k \equiv \tilde{f}(x_{k+1}, x_k, u_k) = 0. \quad (2.1)$$

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

$$z_k = h(x_k), \quad (2.2)$$
2.2 Methodology

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

2.2.3 Model Predictive Control

The principle of MPC is illustrated and explained in Figure 2.3. 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 [Grüne and Pannek, 2017], 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 $\sim 50$ 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.

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 $(u^o_k, x^o_k, z^o_k, \forall k)$. 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
Figure 2.3. 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).

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.

2.2.4 Linearizations around trajectory

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

(2.1):
\[ 0 = \tilde{f}(x^o_{k+1}, x^o_k, u^o_k) = \frac{\partial \tilde{f}}{\partial x_{k+1}} \tilde{x}_{k+1} + \frac{\partial \tilde{f}}{\partial x_k} \tilde{x}_k + \frac{\partial \tilde{f}}{\partial u_k} \tilde{u}_k. \]

(2.3)

A so-called linear time-varying (LTV) state space model [Hespanha, 2009] can now be derived by solving (2.3) for \( \tilde{x}_{k+1} \) and linearizing the controlled variables equation (2.2):
\[
\tilde{x}_{k+1} = A_k \tilde{x}_k + B_k \tilde{u}_k, \quad \tilde{z}_k = C_k \tilde{x}_k + D_k \tilde{u}_k,
\]

with the state space matrices defined as:
\[
A_k = \left( \frac{\partial \tilde{f}}{\partial x_{k+1}} \right)^{-1} \frac{\partial \tilde{f}}{\partial x_k}, \quad B_k = \left( \frac{\partial \tilde{f}}{\partial x_{k+1}} \right)^{-1} \frac{\partial \tilde{f}}{\partial u_k}, C_k = \frac{\partial h}{\partial x_k}, D_k = \frac{\partial h}{\partial u_k} = 0.
\]

RAPTOR provides the underlying Jacobians at each time step around the nominal trajectory. The Jacobian \( \frac{\partial \tilde{f}}{\partial x_{k+1}} \) is based on nonzero plasma profiles and hence practically invertible for all physically relevant values of the state \( x_k \).

Figure 2.4 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 2.3.4.

2.2.5 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:
\[
\hat{U}_{k,N} = \begin{bmatrix} \hat{u}_k \\ \hat{u}_{k+1} \\ \vdots \\ \hat{u}_{k+N-1} \end{bmatrix}, \quad \hat{X}_{k+1,N} = \begin{bmatrix} \hat{x}_{k+1} \\ \hat{x}_{k+2} \\ \vdots \\ \hat{x}_{k+N} \end{bmatrix}, \quad \hat{Z}_{k+1,N} = \begin{bmatrix} \hat{z}_{k+1} \\ \hat{z}_{k+2} \\ \vdots \\ \hat{z}_{k+N} \end{bmatrix}.
\]

Using the same notation we can express:
\[
U_{k,N} = U^0_{k,N} + \hat{U}_{k,N}, \quad X_{k+1,N} = X^0_{k+1,N} + \hat{X}_{k+1,N}, \quad Z_{k+1,N} = Z^0_{k+1,N} + \hat{Z}_{k+1,N}.
\]
Chapter 2. Control of the tokamak safety factor profile with time-varying constraints using MPC

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:

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

(2.7)

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 2.A.1. The matrices are computed offline to reduce the online computational cost. The reader is referred to [Grimble and Majecki, 2015] 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 2.2.8).

2.2.6 Actuator and state constraints

Constraints can be imposed on the individual actuators (for example the desired plasma current $I_p^{\text{des}}$ 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 2.A.2, all the actuator constraints can be formulated as time-varying linear inequality constraints on the future input deviations $\tilde{U}_{k,N}$:

$$
A_{\text{inp}} \tilde{U}_{k,N} \leq b_{\text{inp},k}.
$$

(2.8)
2.2 Methodology

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 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 \( \varepsilon \) 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 \( W\varepsilon^2 \). The scalar \( W \) sets the softness or stiffness of the soft state constraint and needs to be chosen carefully as it determines the controller behavior upon reaching these limits.

In Figure 2.5 it is illustrated how a soft state constraint \( \iota(\rho) \leq 0.95 + \varepsilon \) can be applied to avoid operation in the vicinity of the hard constraint \( \iota(\rho) \leq 1 \). The soft constraint violation \( \varepsilon \) can be monitored in real-time and reported to a supervisory controller that may anticipate on this information. In 2.A.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 \( \varepsilon \):

\[
A_{\text{state}} \begin{bmatrix} \tilde{U}_{k,N} \\ \varepsilon \end{bmatrix} \leq b_{\text{state},k}. \tag{2.9}
\]

**Figure 2.5.** Illustration of using a soft state constraint \( \iota(\rho) \leq 0.95 + \varepsilon \) instead of a hard state constraint \( \iota(\rho) < 1 \). Exceeding \( \iota(\rho) = 0.95 \) is allowed but the term \( W\varepsilon^2 \) penalizes entering this soft constraint region.
2.2.7 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:

$$J_k = ˜Z^T_{k+1,N}Q ˜Z_{k+1,N} + \Delta ˜U^T_{k,N} R_{\Delta U} \Delta ˜U_{k,N} + W_{\epsilon} \epsilon^2,$$

where

$$\Delta ˜U_{k,N} = \begin{bmatrix} ˜u_{k+1} - ˜u_k \\ \vdots \\ ˜u_{k+N-1} - ˜u_{k+N-2} \end{bmatrix} = \Gamma_{\Delta} ˜U_{k,N}$$

and $\Gamma_{\Delta}$ being a difference matrix operator. In the cost function (2.10) 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 action’s.
- $W_{\epsilon}$ 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 model knowledge (e.g. a fully nonlinear plasma transport model) 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 straightforward and intuitive. However, it is known from literature (e.g. [Camacho and Bordons, 2004]) 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 2.3.5.

2.2.8 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. [Camacho and Bordons, 2004; Richalet et al., 2009; Rossiter, 2013]). 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 relatively 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:

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

(2.12)

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 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 2.6 for one actuator and one controlled variable, using the settings as will be discussed in Section 2.3.5.

### 2.2.9 Quadratic programming

The future input deviations $\tilde{U}_{k,N}$ can be found by minimizing the cost function (2.10) subjected to the constraints (2.8) and (2.9). 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 (2.7) and CVP (2.12) as an online Quadratic Programming (QP) problem at each time instant $t_k$:

$$\begin{align*}
\text{minimize} & \quad \frac{1}{2} \xi_k^T H_k \xi_k + \tilde{x}_k^T F_k \xi_k, \\
\text{subject to} & \quad A_{\text{ineq},k} \xi_k \leq b_{\text{ineq},k}, \\
& \quad A_{\text{eq}} \xi_k = b_{\text{eq},k},
\end{align*}$$

(2.13)
Chapter 2. Control of the tokamak safety factor profile with time-varying constraints using MPC

Figure 2.6. Illustration of reducing the computational cost using concepts of input parametrization and coincidence points as used in the simulation results. The 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 (redo). The settings are explained in Section 2.3.5.

where $\xi^T_k = [\tilde{p}^T_{k,N}, \varepsilon]$. The matrices $H_k$, $F_k$, $A_{ineq,k}$ and $A_{eq}$ are computed offline and the vectors $b_{ineq,k}$ and $b_{eq,k}$ are externally provided by the user or set in real-time by a supervisory controller. All are defined in 2.A.4.

QP is computationally cheap and has a unique constrained solution. Many algorithms exist to solve the QP problem (2.13) online, that can handle a large number of free parameters and abundant linear constraints, e.g. [Ferreau et al., 2008; Wills, 2007]. Using QPC [Wills, 2007], results in a computational time per time instant of less than 8 ms in the simulation cases of Section 2.4 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.

2.2.10 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: \((2.4-2.5)\)
- Nominal state and controlled variable (reference) trajectories: \((x_k^o, y_k^o)\)

Choose prediction horizon \(N\)

Choose controller weights in \((2.10)\): \(Q, R_{\Delta U}\) and \(W_\varepsilon\)

Choose a future input sequence parametrization in \((2.12)\): \(P_{\text{map}}\)

Compute the prediction matrices at each time instant in \((2.7)\):
\[
\Gamma_{A_{k,N}}, \Gamma_{B_{k,N}}, \Gamma_{C_{k,N}} \text{ and } \Gamma_{D_{k,N}}
\]

Compute the pre-computable part of the matrices and vectors in \((2.13)\)

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.

### 2.3 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 [van Dongen et al., 2014]. 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 (Section 2.4).

#### 2.3.1 Physics model and settings in RAPTOR

A CRONOS simulation of an ITER L-mode ramp-up in [Hogewej et al., 2012] is used to supply the initial profiles of \(T_e\) and \(\psi\), the prescribed profile evolutions of \(n_e, n_i\) and \(Z_{\text{eff}}\) 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 chapter 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
Chapter 2. Control of the tokamak safety factor profile with time-varying constraints using MPC

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 $\tilde{x}_k$
    Receive from supervisory controller (or user): constraint vectors for actuators and states
    Receive currently active input $\tilde{u}_k$ to fix in optimization
    Compute $b_{ineq,k}$ and $b_{eq,k}$ using (2.31) and (2.32)
    Solve QP (2.13) for $\xi_T^T = [\tilde{p}_{k,N}^T \ \varepsilon]$.
    Compute $\tilde{U}_{k,N}$ using (2.12)
    Compute $U_{k,N}$, $X_{k,N}$ and $Z_{k,N}$ using (2.6) and (2.7)
    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

predicted L-H transition power threshold for this density evolution as reported in [Hogeweij et al., 2012].

In this chapter, 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:
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 $\sum_{i=1}^{4} P_{EC,i} = 20$ 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 [van Dongen et al., 2014] 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.

### 2.3.2 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:

- $0 \leq I_p \leq 10$ MA, during flat-top $6 \leq I_p \leq 8$ MA
- $-0.1$ MA/s $\leq \frac{dI_p}{dt} \leq 0.1$ MA/s
- $P_{EC,i} \geq 0$ MW, $\forall i \in [1, 2, 3, 4]$
- $\Sigma P_{EC} \leq 25$ 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 $I_p(t_{\text{end}}) \geq 7$ MA.

Large and steep variations of the plasma current $I_p$ during the flat-top phase are known 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 $l_i^{(3)}(t)$. 

<table>
<thead>
<tr>
<th>Beam</th>
<th>Location $\rho$</th>
<th>Type EC</th>
<th>Gaussian full width in $\rho$ [-]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>co</td>
<td>0.2</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>counter</td>
<td>0.2</td>
</tr>
<tr>
<td>3</td>
<td>0.2</td>
<td>co</td>
<td>0.1</td>
</tr>
<tr>
<td>4</td>
<td>0.4</td>
<td>co</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Chapter 2. Control of the tokamak safety factor profile with time-varying constraints using MPC

2.3.3 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 [Felici and Sauter, 2012a].

The nominal trajectory optimization aims at achieving any \( \imath \)-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 2.3.2. The flat-top phase starts after 120 s. The loop voltage profile \( U_{pl}(\rho) \) must be as flat as possible after 120 s, meaning that the ohmic current redistribution has almost stopped and therefore the \( \imath \)-profile is stationary. The optimization cost function contains therefore solely a penalty on the gradient of the final loop voltage profile \( U_{pl}(\rho) \).

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 2.7(a) and 2.7(e-h) respectively. The prescribed NBI-power and the total EC-power are shown in 2.7(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 overshots have been reported to be beneficial for achieving a stationary flat \( \imath \)-profile [Felici and Sauter, 2012a; Hobirk et al., 2012a; Hogeweij et al., 2012].

The optimization uses the flexibility in the EC-power allocation. At most times, the counter-ECCD power in the center (Figure 2.7(f)) significantly exceeds the co-ECCD power (Figure 2.7(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 \( \imath \leq 0.95 \).

The EC-beam at \( \rho = 0.2 \) (Figure 2.7(g)) provides significant heating and current drive during the ramp-up phase. Its peak-power occurs just before the NBI-power (Figure 2.7(b)) is ramped up between 60 s and 70 s. The EC-power deposition at \( \rho = 0.4 \) (Figure 2.7(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 2.7(b). The prescribed ramp-up of central electron and ion densities is given in Figure 2.7(d). The resulting central ion and electron temperatures \( T_{i,0} \) \( T_{e,0} \) are shown in 2.7(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 \( \imath \), \( T_e \), \( T_i \), \( n_e \), \( U_{pl} \) and the components of
Figure 2.7. Optimized nominal trajectories for achieving stationary $\iota$-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 $\iota, T_e, T_i, n_e, U_{pl}$ and components of $j_\parallel$ at 60, 70, 100 and 120 s. Note $I_p(120\text{ s}) = 7\text{ MA}$, that is the maximum achievable plasma current in our simulations for achieving a stationary $\iota$-profile within the given constraints in L-mode.
the parallel current density $j_{\parallel}$ are shown in the bottom panels of Figure 2.7. The $\iota$-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 $U_{\text{pl}}$-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 $U_{\text{pl}}$-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.

2.3.4 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 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 $u_{k}^{\text{ref}}$ and fed to RAPTOR. Comparison of the outputs of RAPTOR for the case with perturbation and without perturbation yields $\Delta T_{e, \text{nonlin}}$ and $\Delta \iota_{\text{nonlin}}$.

- Multiple linearized models: the perturbation input vector $\tilde{u}_k$ is fed into the models locally linearized at each moment in time which yields $\Delta T_{e, \text{multilin}}$ and $\Delta \iota_{\text{multilin}}$.

- 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 $\Delta T_{e, \text{singlelin}}$ and $\Delta \iota_{\text{singlelin}}$.

The results are given in Figure 2.8. Random sums of sinusoids with a limited frequency band (often used in system identification [Ljung, 1998] are used for the verification and are shown in column 1 (red- -), (blue- -) 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 (red- -) case and the cases with multiple linearizations.
2.3 Simulation setup

(blue-·-) and a single linear model (·-·). The effect of the perturbation inputs can be appreciated in column 2 (for $\Delta \iota$) and column 3 (for $\Delta T_e$) at several locations in the plasma.

The evolution of $\Delta T_e$ 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 $\iota$-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 $\iota$-profile of less than 0.01 in magnitude.

The response of the single linear model shows that the $\Delta T_e$ 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) cause 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.

2.3.5 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
Figure 2.8. 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 (red - -) 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 $\Delta T_e$-profile. Note the far more accurate description by multiple linearized models of both the $T_e$-profile and $\iota$-profile with respect to a single linear model.
2.3 Simulation setup

are dominant in the input-controlled variable behavior (analyzed via the so called Hankel singular values [Skogestad and Postlethwaite, 2005], 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.

Weights \( Q, R_{\Delta U} \) and \( W_\varepsilon \). In this chapter we obtained the settings of the weights \( Q, R_{\Delta U} \) and \( W_\varepsilon \) 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 \( W_\varepsilon \) 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 \( P_{\text{map}} \), including control horizon \( N_c \). This is chosen as shown in Figure 2.6 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 2.6. 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.
2.4 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.

2.4.1 Tracking the reference in flat-top phase

In the first simulation case, tracking of the reference \( \iota \)-profile during the flat-top phase will be considered. The reference \( \iota \)-profile at the beginning of the flat-top phase is not yet fully in stationary state, since the \( U_{pl} \)-profile in Figure 2.7 was not completely flat. Therefore the \( \iota \)-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 \( \iota \)-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 \( \iota \)-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 \( \iota \)-profile which is weighted in the controller cost function.

The results for tracking the reference \( \iota \)-profile in the flat-top phase are presented in Figure 2.9. In the feedforward only case (---), the actuator commands (Figure 2.9 (a,e-h)) are kept constant after the end of the ramp-up phase. This yields also a constant total EC-power (Figure 2.9(b)). The resulting error norm (Figure 2.9(c)) increases as the \( \iota \)-profile drifts away from the reference \( \iota \)-profile and violates also the soft constraint to a large extent (Figure 2.9(d)). The resulting \( \iota \)-profile is given in the bottom panels (Figure 2.9(i-l)) for the time steps also indicated by the gray dotted lines in Figure 2.9(c): 120, 150, 200 and 399 s. In Figure 2.9(l) can be observed clearly that the \( \iota \)-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 \( \iota \)-profile during the flat-top phase. The controller reduces the tracking error (Figure 2.9(c)) significantly and avoids the soft constraint violation (Figure 2.9(d)) by lowering the plasma current \( I_p \) (Figure 2.9(a)) and adjusting the powers to the co-EC-beam at \( \rho = 0 \) (Figure 2.9(e)) and the co-EC-beam at \( \rho = 0.4 \) (Figure 2.9(h)). The reduction of the tracking error is also clearly shown in Figure 2.9(l), where the \( \iota \)-profile is almost on top of its reference.

It can be concluded that the MPC-controller can significantly reduce the error during tracking a fixed reference \( \iota \)-profile in the flat-top phase, despite the
2.4 Simulation results

![Graphs and plots]

Figure 2.9. Tracking of reference $\iota$-profile in flat-top phase. Case with feed-forward 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 $\iota$-profiles at time steps also indicated by gray dotted lines in top panels (120, 150, 200 and 399 s). Reference $\iota$-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).

Fact that the $\iota$-profile was not yet fully in stationary state at the start of the flat-top.

2.4.2 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 $\iota$-profile during the flat-top phase. However, after 200 s we limit the maximum available EC-power to 14 MW instead of 25 MW. In Figure 2.10 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.
Chapter 2. Control of the tokamak safety factor profile with time-varying constraints using MPC

Figure 2.10. Saturation and handling of time-varying available EC-power constraint. The unconstrained feedback case from Figure 2.9 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 \( \iota \)-profile (i-l).

The unconstrained EC-power case from Figure 2.9 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 2.10(c)) and a soft constraint violation exceeding early \( \iota(\rho) \leq 1 \) (Figure 2.10(d)). In Figure 2.10(j-l), it can be 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 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
2.4 Simulation results

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 $\iota$-profile can be observed in Figure 2.10(i-l).

A detailed view of Figure 2.10 (d) in the time interval between 200 s and 400 s is given in Figure 2.11. Immediately 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 $\iota$-profile and also modifying the reference $\iota$-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.

2.4.3 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

Figure 2.11. 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.
Chapter 2. Control of the tokamak safety factor profile with time-varying constraints using MPC

Figure 2.12. 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 $\iota$-profile (c,i-l), while $\iota$-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 $\iota$-profile slightly violates (unpredicted) the soft state constraint (d).

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.

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 2.12 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 \( \iota \)-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.

**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 2.13 reveals the results with increased thermal transport. Using only feedforward, results in a large error in the \( \iota \)-profile (c,i-l), while the \( \iota \)-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. 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.

**2.5 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
Chapter 2. Control of the tokamak safety factor profile with time-varying constraints using MPC

Figure 2.13. 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 \( \iota \)-profile (c,i-l), while \( \iota \)-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.

Improve or extend the work presented in this chapter. 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, another 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 predictions. Disturbance estimations can be provided online, by using e.g. RAPTOR as observer [Felici et al., 2012b].

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 particularly 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 nonlinearily coupled evolution of shape and profiles.

2.6 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 chapter 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.
Appendixes

2.A Definitions and derivations

In this appendix we provide the definitions and derivations as used in Section 2.2.

2.A.1 Prediction matrices

For the use in the prediction equations (2.7), the following notation is introduced:

\[
\prod_{l=m}^{n} A_{k+l} \equiv \begin{cases} 
A_{k+n} A_{k+n-1} \ldots A_{k+m} & \text{if } m \leq n \\
I & \text{if } m > n 
\end{cases} \quad (2.14)
\]

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:

\[
\Gamma_{A_{k,N}} = \begin{bmatrix} 
\prod_{l=1}^{0} A_{k+l} \\
\prod_{l=1}^{1} A_{k+l} \\
\vdots \\
\prod_{l=1}^{N-1} A_{k+l} 
\end{bmatrix} A_{k}, \quad (2.15)
\]

\[
\Gamma_{B_{k,N}} = \begin{bmatrix} 
\prod_{l=1}^{0} A_{k+l} & B_{k} & 0 & \cdots & 0 & 0 \\
\prod_{l=1}^{1} A_{k+l} & B_{k} & \prod_{l=2}^{1} A_{k+l} & B_{k+1} & \cdots & \cdots & \vdots \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
\prod_{l=1}^{N-1} A_{k+l} & B_{k} & \prod_{l=2}^{N-1} A_{k+l} & B_{k+1} & \cdots & \prod_{l=N}^{N-1} A_{k+l} & B_{k+N-1} & 0 
\end{bmatrix}, \quad (2.16)
\]

\[
\Gamma_{C_{k,N}} = \begin{bmatrix} 
C_{k+1} & \prod_{l=1}^{0} A_{k+l} \\
C_{k+2} & \prod_{l=1}^{1} A_{k+l} \\
\vdots & \vdots \\
C_{k+N} & \prod_{l=1}^{N-1} A_{k+l} 
\end{bmatrix} A_{k}, \quad (2.17)
\]

and
\( \Gamma_{D_{k,N}} = \)
\[
\begin{bmatrix}
C_{k+1} \left[ \prod_{l=1}^{0} A_{k+l} \right] B_k & D_{k+1} & \cdots & 0 & 0 \\
C_{k+2} \left[ \prod_{l=1}^{1} A_{k+l} \right] B_k & C_{k+2} \left[ \prod_{l=2}^{1} A_{k+l} \right] B_{k+1} & \cdots & \cdots & \cdots \\
\vdots & \vdots & \ddots & \cdots & \cdots \\
C_{k+N} \left[ \prod_{l=1}^{N-1} A_{k+l} \right] B_k & C_{k+N} \left[ \prod_{l=2}^{N-1} A_{k+l} \right] B_{k+1} & \cdots & C_{k+N} \left[ \prod_{l=1}^{N} A_{k+l} \right] B_{k+N-1} & D_{k+N}
\end{bmatrix}.
\]

(2.18)

### 2.A.2 Constraints

Here we derive the linear inequality constraints (2.8) and (2.9) representing the actuator and state constraints.

**Actuator constraints**

The actuator amplitude constraints are given as:

\[
U_{\min,k} \leq U_{k,N} \leq U_{\max,k}.
\]

(2.19)

Using (2.6) these constraints can be rewritten as:

\[
\begin{bmatrix}
I \\
-I
\end{bmatrix} \tilde{U}_{k,N} \leq \begin{bmatrix}
U_{\max,k} - U^0_{k,N} \\
-U_{\min,k} + U^0_{k,N}
\end{bmatrix}.
\]

(2.20)

Following (2.11), actuator ramp-rate constraints are given as:

\[
\Delta U_{\min,k} \leq \Delta U_{k,N} \leq \Delta U_{\max,k}.
\]

(2.21)

Using (2.6) and (2.11) these constraints can be rewritten as:

\[
\begin{bmatrix}
\Gamma_{\Delta} \\
-\Gamma_{\Delta}
\end{bmatrix} \tilde{U}_{k,N} \leq \begin{bmatrix}
\Delta U_{\max,k} - \Gamma_{\Delta} U^0_{k,N} \\
-\Delta U_{\min,k} + \Gamma_{\Delta} U^0_{k,N}
\end{bmatrix}.
\]

(2.22)

The mixed actuator constraints are given as:

\[
A_{U_{\text{mix}}} U_{k,N} \leq b_{U_{\text{mix}}}.
\]

(2.23)

Using (2.6) these constraints can be rewritten as:

\[
A_{U_{\text{mix}}} \tilde{U}_{k,N} \leq b_{U_{\text{mix}}} - A_{U_{\text{mix}}} U^0_{k,N}.
\]

(2.24)
Combining now the equations (2.20), (2.22) and (2.24) we obtain:

\[
\begin{bmatrix}
I & -I \\
\Gamma_\Delta & -\Gamma_\Delta \\
A_{U,\text{mix}} & A_{\text{inp},k}
\end{bmatrix}
\begin{bmatrix}
\hat{U}_{k,N}
\end{bmatrix}
\leq
\begin{bmatrix}
U_{\text{max},k} - U_{0,k,N} \\
-U_{\text{min},k} + U_{0,k,N} \\
\Delta U_{\text{max},k} - \Delta U_{0,k,N} \\
-\Delta U_{\text{min},k} + \Delta U_{0,k,N} \\
b_{U,\text{mix}} - A_{U,\text{mix}}U_{0,k,N}
\end{bmatrix},
\]

(2.25)

which defines the matrices as given in (2.8).

**State constraints**

State constraints are defined as:

\[
A_{X,\text{mix}}X_{k+1,N} \leq b_{X,\text{mix}} + \varepsilon I.
\]

(2.26)

Rewriting (2.26) using (2.6) and (2.7) results in:

\[
\begin{bmatrix}
A_{X,\text{mix}} \Gamma B_{k,N} & -I \\
A,\text{state},k & \varepsilon
\end{bmatrix}
\begin{bmatrix}
\hat{U}_{k,N} \\
\varepsilon
\end{bmatrix}
\leq
\begin{bmatrix}
b_{X,\text{mix}} - A_{X,\text{mix}}[X_{k+1,N} + \Gamma A_{k,N}\tilde{x}_k] \\
b_{\text{state},k}
\end{bmatrix},
\]

(2.27)

which defines the matrices as given in (2.9).

**2.A.3 Cost function**

The cost function $J_k$ as given in (2.10) can be rewritten by using (2.7) and (2.11):

\[
J_k = \tilde{x}_k^T \Gamma_{C,k,N}^T Q \Gamma_{C,k,N} \tilde{x}_k + \tilde{x}_k^T \left[ 2 \Gamma_{C,k,N}^T Q \Gamma_{D,k,N} \right] \hat{U}_{k,N} + 
\]

\[
+ \tilde{U}_{k,N}^T \Gamma_{D,k,N}^T Q \Gamma_{D,k,N} + R_{\Delta U} \right] \hat{U}_{k,N} + w_\varepsilon \varepsilon^2.
\]

(2.28)

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

\[
\hat{J}_k = \tilde{x}_k^T \left[ 2 \Gamma_{C,k,N}^T Q \Gamma_{D,k,N} \right] \tilde{U}_{k,N} + 
\]

\[
+ \tilde{U}_{k,N}^T \Gamma_{D,k,N}^T Q \Gamma_{D,k,N} + R_{\Delta U} \right] \tilde{U}_{k,N} + w_\varepsilon \varepsilon^2.
\]

(2.29)
Chapter 2. Control of the tokamak safety factor profile with time-varying constraints using MPC

2.4 Quadratic programming matrices and vectors

By implementing the input vector parameterization in (2.12) and using \( \xi_k^T = [p_k^{T,N} \ \varepsilon] \), then the cost function (2.29), actuator constraints (2.25) and state constraints (2.27) can be rewritten into the standard form of the QP problem (2.13) at each time instant \( t_k \):

\[
\dot{J}_k = \tilde{x}_k^T \left[ \begin{array}{cc} \Gamma_{J1,k} P_{\text{map}} & 0 \\ \Gamma_{J2,k} P_{\text{map}} & 0 \\ w_{\varepsilon} \end{array} \right] \xi_k + \xi_k^T \left[ \begin{array}{cc} F_k & 0 \\ 0 & 2H_k \end{array} \right] \xi_k, \tag{2.30}
\]

\[
\begin{bmatrix} A_{\text{inp}} & 0 \\ A_{\text{state}} \end{bmatrix} \begin{bmatrix} P_{\text{map}} & 0 \\ 0 & 1 \end{bmatrix} \xi_k \leq \begin{bmatrix} b_{\text{inp},k} \\ b_{\text{state},k} \end{bmatrix}, \tag{2.31}
\]

\[
\begin{bmatrix} I_n \\ A_{\text{eq}} \end{bmatrix} \begin{bmatrix} P_{\text{map}} & 0 \\ 0 \end{bmatrix} \xi_k = \begin{bmatrix} \tilde{u}_k \\ b_{\text{eq},k} \end{bmatrix}, \tag{2.32}
\]

which defines the matrices \( F_k, H_k, A_{\text{ineq},k} \) and \( A_{\text{eq}} \) and the vectors \( b_{\text{ineq},k} \) and \( b_{\text{eq},k} \) in (2.13).
Chapter 3

Model predictive control of the current density distribution and stored energy in tokamaks using trajectory linearizations

The work presented in this chapter is published as refereed conference publication: Maljaars, E., et al. (2015), Model predictive control of the current density distribution and stored energy in tokamaks using trajectory linearizations, IFAC-PapersOnLine 48(23), 314321. Presented at the 5th IFAC Conference on Nonlinear Model Predictive Control NMPC 2015 Seville, Spain. The text has been revised after feedback of the thesis committee.
3.1 Introduction

The tokamak is the furthest developed nuclear fusion device in which a plasma is confined in a torus-shaped device using magnetic fields and heated to achieve fusion reactions. Due to the specific organization of the magnetic field, the plasma transport process can be seen as a 1 dimensional distributed parameter system in the direction of the minor radius, where the plasma quantities as function of the radial coordinate are called profiles (as illustrated in Figure 3.1). The plasma transport process can be described by a number of nonlinear coupled partial differential equations (fluid transport equations together with the Maxwell equations). A control-oriented introduction to tokamaks and their control is given in [Pironti and Walker, 2005, 2006].

![Figure 3.1. Illustration of plasma profiles in a tokamak. Radial coordinate $\rho$ (left) and typical profiles of electron temperature $T_e(\rho)$ and poloidal magnetic flux $\psi(\rho)$.](image)

Controlling profiles or scalar profile quantities of the nonlinear plasma transport process in (future) tokamaks is a challenging task for a number of reasons. First, the dynamics are described by nonlinear coupled transport equations that are only recently available in the form of control-oriented models [Felici et al., 2011b; Ou et al., 2007; Witrant et al., 2007]. Second, measurements in a hot plasma are technologically complicated, limited in quantity and are not routinely available in real-time on many devices, such that feedback control is often limited to a subset of the desired controlled parameters. Third, the plasma profile controller is part of a larger control problem with multiple controllers that may share common actuators that should be governed by a supervisory controller. Depending on the occurrence of actuator events (e.g. actuator failure) or plasma events (e.g. performance and stability degrading plasma instabilities), such a supervisory controller may reallocate actuators to the different control tasks and change control objectives and references in real-time [Winter et al., 2014]. Fourth, actuators are spatially distributed over the radial coordinate and are subjected to strict constraints in amplitude and ramp-rate that thus may change in real-time. Fifth, exceeding certain plasma parameter quantities is
3.1 Introduction

known to trigger performance and stability degrading plasma instabilities and hence it is desired to keep these plasma parameters within their limits. The energy stored in the plasma increases in larger devices such as ITER (under construction) and DEMO (planned demonstration reactor). Exceeding operational limits with high stored energy may result in unallowable damage to the machine at high cost and loss of expensive operational time. Sixth, the dominant process dynamics exhibit distinct time scales. On currently operational medium-sized tokamaks (e.g. the tokamaks ASDEX-Upgrade in Germany and TCV in Switzerland), the fastest time scale is in the order of a few milliseconds and this imposes a strong limit on the available computational time of 2 milliseconds. The slowest time scale is however in the order of seconds, indicating the wide range of time scales present.

Recently, a number of model-based feedback controllers have been proposed for the control of plasma profiles and scalar quantities in tokamaks, where we refer here only to a few of the most recent publications [Barton et al., 2012, 2014a; Boyer et al., 2013, 2014; Gaye et al., 2013; Moreau et al., 2013]. Only a few of these controllers have so far been used in dedicated experiments. These contributions impose fixed actuator constraints a posteriori by the use of e.g. an anti-windup loop. Handling of actuator constraints in the controller design itself is done in [Bribiesca Argomedo et al., 2013; Ou et al., 2011; Ouarit et al., 2011], where actuator constraints are fixed.

Model predictive control [Camacho and Bordons, 2004; Mayne, 2014; Rossiter, 2013] appears, due to its constraint handling capabilities and straightforward handling of MIMO systems, as a promising control method to increase the performance and stability of tokamak plasmas, while ensuring the satisfaction of important machine and physical limits. So far MPC has not been researched for profile control, except for the initial work in [Ou et al., 2011; Ouarit et al., 2011]. The controllers in these works handle only fixed actuator constraints, contain a model with a single PDE and yield nonlinear optimization problems that cannot be solved in real-time on existing devices.

In our recent work [Maljaars et al., 2015c] we proposed an MPC approach for the control of one particular magnetic profile important for the stability and performance of a plasma. We used trajectory linearizations to formulate an MPC controller that can track a predefined reference trajectory in the presence of time-varying actuator and state constraints. Simulation results of the future ITER tokamak showed the performance of the controller in reducing tracking error and handling time-varying constraints with computational times that enable implementation on currently operational tokamaks.

This chapter extends our previous work in various ways. In this chapter we not only control a magnetic profile, but also a kinetic quantity being the plasma stored energy. In addition, the controller can handle real-time-varying references that differ from the nominal trajectory used for the trajectory linearizations. Constraints are added that limit a weighted combination of inputs
and outputs. To improve tracking and constraint handling during the stationary phase of a tokamak plasma, state disturbance estimation is incorporated. The performance of the proposed MPC controller is shown in simulation using the nonlinear plasma transport simulator RAPTOR [Felici and Sauter, 2012a] with parameters that are chosen to approximately represent high performance plasma experiments in the ASDEX-Upgrade tokamak [Hobirk et al., 2012b].

The remainder of this chapter is organized as follows. In Section 3.2 we introduce the process model, the phases in a tokamak experiment and formulate the controller requirements. Based on these controller requirements, in Section 3.3 we present the MPC controller design. The performance of the designed controller is analyzed in simulations in Section 3.4. Finally, conclusions and an outlook are given in Section 3.5.

3.2 Process modeling and controller requirements

In this section we describe briefly the process model and the phases of a typical tokamak experiment after which we define the controller requirements.

3.2.1 Process model in nonlinear state space format

In this chapter we are interested in the spatial and temporal evolution of the current density distribution and the temporal evolution of the stored energy in the plasma in response to the various actuators. The underlying dynamics of both quantities are mainly the nonlinear coupled evolution of the poloidal magnetic flux $\psi(\rho,t)$ and the electron temperature $T_e(\rho,t)$ as a function of a radial coordinate $\rho$, represented by partial differential equations (PDEs) [Hinton and Hazeltine, 1976] that can be written as:

$$k_1 \frac{\partial \psi}{\partial t} = \frac{\partial}{\partial \rho} \left( k_2 \frac{\partial \psi}{\partial \rho} \right) + k_3 + k_u u_{int}, \quad (3.1)$$

$$c_1 \frac{\partial T_e}{\partial t} = \frac{\partial}{\partial \rho} \left( c_2 \frac{\partial T_e}{\partial \rho} \right) + c_3 + c_u u_{int}. \quad (3.2)$$

These are two non-linear coupled diffusion equations, where the quantities $k_1$, $k_2$, $k_3$, $k_u$, $c_1$, $c_2$, $c_3$ and $c_u$ are functions of the radial coordinate $\rho$ and nonlinearly dependent on the profiles $\psi$ and $T_e$ and other prescribed profiles including the geometry of the plasma. The radial coordinate $\rho \in \mathbb{R}$ equals 0 in the plasma core and 1 at the plasma edge.

The actuators act either on the boundary ($\rho=1$) or the interior of the plasma ($\rho \leq 1$). The plasma current $I_p$[MA] acts on the boundary of $\psi$. The interior actuators are given in the vector $u_{int} \in \mathbb{R}^{n_{int}}$ and the quantities $k_u(\rho)$ and $c_u(\rho)$ describe their spatial localization. We choose as interior actuators the auxiliary spatially distributed actuators that heat the plasma and can drive
current in the plasma. These auxiliary heating and current drives are comprised of a power request to two electron cyclotron beams \( P_{ec,1}[MW] \) and \( P_{ec,2}[MW] \) at different locations in \( \rho \) and a power request to the Neutral Beam Injection (NBI) system \( P_{nbi}[MW] \). We assume that low level controllers ensure that the requested power is delivered.

The current density distribution and the stored energy are a function of \( \psi \) and \( T_e \) respectively. The so-called inverse safety factor profile is a direct measure of the current density distribution in the plasma and an often used measure for stability and performance of the plasma. Therefore we choose as model outputs \( z \in \mathbb{R}^{n_z} \) the inverse safety factor profile \( \iota(\rho) \sim c_\iota(\rho) \frac{\partial \psi}{\partial \rho} \) at 6 different radial positions and the plasma stored thermal energy \( W_{th}[MJ] \sim \int_0^1 c_{W_{th}}(\rho,t)T_e(\rho)\,d\rho \), where \( c_\iota(\rho) \) and \( c_{W_{th}}(\rho,t) \) are known profiles.

In this chapter we choose to solve (3.1) and (3.2) using the control-oriented plasma transport simulator RAPTOR [Felici and Sauter, 2012a], in which (3.1) and (3.2) are written as a nonlinear state space model with the following state update equation and its corresponding output equation a time instant \( t_k \):

\[
\begin{align*}
  f(x_{k+1}, x_k, u_k) &= 0, \\
  z_k &= h(x_k).
\end{align*}
\]

In RAPTOR, (3.1) and (3.2) are discretized in space using \( n_b \) finite element basis functions for \( T_e \) as well as \( \psi \), such that the state \( x_k \in \mathbb{R}^{n_x} \) contains the corresponding basis function coefficients describing the profiles of \( T_e \) and \( \psi \). In our simulations we choose \( n_x = 2n_b = 28 \). Temporal discretization is done using a forward Euler implicit scheme with a fixed sample time.

In vector form the inputs, states and outputs read:

\[
\begin{align*}
  u_k &= \begin{bmatrix} I_p(t_k) \\
  P_{ec,1}(t_k) \\
  P_{ec,2}(t_k) \\
  P_{nbi}(t_k) \end{bmatrix}, \\
  x_k &= \begin{bmatrix} \hat{T}_{e1}(t_k) \\
  \vdots \\
  \hat{T}_{en_b}(t_k) \\
  \frac{\psi_1(t_k)}{\psi_1(t_k)} \\
  \vdots \\
  \frac{\psi_{n_b}(t_k)}{\psi_{n_b}(t_k)} \end{bmatrix}, \\
  z_k &= \begin{bmatrix} \iota_{\rho=0}(t_k) \\
  \iota_{\rho=0.1}(t_k) \\
  \vdots \\
  \iota_{\rho=0.5}(t_k) \\
  W_{th}(t_k) \end{bmatrix},
\end{align*}
\]

where the boundary actuator \( I_p \) with the interior actuator inputs \( u_{int} \) are put into the input vector \( u_k \in \mathbb{R}^{n_u} \) and by using the appropriate scale (e.g. [MA] and [MW]), all inputs and outputs are in the order of one.

### 3.2.2 Phases during a typical tokamak experiment

A tokamak experiment follows a sequence of three phases:
Chapter 3. Model predictive control of the current density distribution and stored energy in tokamaks using trajectory linearizations

Phase I $t \leq t_{\text{ftstart}}$: current ramp-up phase.
This phase aims at obtaining the desired plasma profiles at its end by raising the plasma current assisted by the other actuators. Temperature and stored energy will rise and current density profile will evolve towards an equilibrium. During this phase nonlinearities play important roles in establishing the plasma profiles.

Phase II $t_{\text{ftstart}} < t < t_{\text{ftend}}$: current flat-top phase.
A more or less stationary phase in which the desired plasma conditions should be constant and hence feedforward inputs are expected to be constant to sustain the plasma conditions. This phase is usually the largest part of the experiment and should in future devices be the part where energy is produced.

Phase III $t \geq t_{\text{ftend}}$: current ramp-down phase.
Here, the current is ramped-down and the plasma should be terminated safely within known limits.

In this chapter we deal only with phase I and II. Specifically for the high performance plasmas we simulate in this chapter, the plasma undergoes during Phase I a so-called confinement mode transition in which the electron heat transport at the plasma edge drops, resulting in higher temperatures and a significant self-generated current in the plasma.

3.2.3 Controller requirements

Given the described process, the typical evolution of a tokamak experiment and the context of the controller as given in the introduction, we formulate that the to-be-designed controller should be able to simultaneously:

1. Minimize the tracking error for several sets of controlled variables, where during phase II the steady state error should be minimal in the presence of constant disturbances and model mismatches.

2. Handle hardware and physical constraints on:
   - Individual inputs and outputs
   - Combinations of inputs and outputs

3. Deal with real-time-varying:
   - References
   - Constraints

4. Include the nonlinear dynamics during Phase I

5. Deal with the long current diffusion time scale

6. Have a computational time of less than 5 milliseconds
3.3 Controller design

Note on physical constraints.

Depending on the scenario that will be simulated or the experiment that will be performed, physical limits will be present. For the high performance plasma scenario as simulated in this chapter, we require two types of additional physical constraints that will avoid the occurrence of a plasma instability that is undesired for this scenario. First the output constraint $\eta(\rho) \leq 1$ is required in the center of the plasma $\rho \leq 0.5$. Secondly the normalized plasma pressure $\beta_N$ is limited as $\beta_N \leq \beta_N^{\text{max}}$, where

$$
\beta_N = c_{\beta_N} W_{th}[\text{MJ}] / I_p[\text{MA}]
$$

(3.6)

and $c_{\beta_N}$ is a known constant. Under the condition that $I_p[\text{MA}] > 0$, this constraint can be rewritten as $c_{\beta_N} W_{th}[\text{MJ}] \leq \beta_N^{\text{max}} I_p[\text{MA}]$, thereby yielding a mixed input-output constraint.

3.3 Controller design

Given the controller requirements in the previous section, the MPC controller design is presented here. Using linearized models, a prediction model will be constructed. Then the control input parameterization, controller cost function and constraints will be given that allows us to formulate compact Quadratic Programming problems. We finish this section with discussing obtaining state and disturbance estimates.

3.3.1 Linearizations around nominal trajectory

Provided the available computational time of 5 milliseconds and the fact that each nonlinear simulation step takes already milliseconds, it is not presently feasible to apply fully nonlinear MPC. Fortunately we can benefit from the fact that commonly tokamak experiments rely strongly on feedforward input trajectories that are defined prior to the experiment. When applied to our model in (3.3-3.4), these feedforward inputs will, in the absence of disturbances and model mismatches, result in a particular nominal state evolution. By precalculating these input, state and output trajectories ($u_k^o, x_k^o, z_k^o, \forall k$), the nonlinear dynamics can then be linearized around them at each time instant $t_k$ resulting in a linear time-varying process model. This allows us to use linear MPC techniques that are computationally tractable. Examples of application of MPC to systems described by LTV models can be readily found in literature (e.g. [Falcone et al., 2008]).

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 (3.3):

$$
0 = \frac{\partial f}{\partial x_{k+1}} \tilde{x}_{k+1} + \frac{\partial f}{\partial x_k} \tilde{x}_k + \frac{\partial f}{\partial u_k} \tilde{u}_k.
$$

(3.7)
By solving (3.7) for \( \tilde{x}_{k+1} \) and linearizing the output equation (3.4) we obtain a LTV state space model given as

\[
\tilde{x}_{k+1} = A_k \tilde{x}_k + B_k \tilde{u}_k + \tilde{d}_k, \tag{3.8}
\]
\[
\tilde{z}_k = C_k \tilde{x}_k + D_k \tilde{u}_k, \tag{3.9}
\]

where \( \tilde{d}_k \) represent added state disturbances and the state space matrices are defined as:

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

For practical values of the underlying profiles (e.g. temperature), the Jacobian \( \partial f / \partial x_{k+1} \) will always have full rank and invertible.

### 3.3.2 Construction of prediction model

A prediction model is constructed to predict the outputs on the prediction horizon of \( N \) time instances. We define the stacked vectors

\[
U_k^T = [\tilde{u}_k^T \quad \tilde{u}_{k+1}^T \quad \cdots \quad \tilde{u}_{k+N-1}^T]
\]
\[
\tilde{Z}_k^T = [\tilde{z}_{k+1}^T \quad \tilde{z}_{k+2}^T \quad \cdots \quad \tilde{z}_{k+N}^T]
\]

and write

\[
U_k = U_k^0 + \tilde{U}_k \quad \text{and} \quad Z_k = Z_k^0 + \tilde{Z}_k.
\]

By using the obtained linearized state space models at each moment in time and adding to-be-estimated state disturbances that are assumed constant over the prediction horizon (see Section 3.3.7), we can write the future output deviations \( \tilde{Z}_k \) as:

\[
\tilde{Z}_k = \Gamma_{C_k} \tilde{x}_k + \Gamma_{D_k} \tilde{U}_k + \Gamma_{E_k} \tilde{d}_k. \tag{3.10}
\]

The matrices \( \Gamma_{C_k}, \Gamma_{D_k} \) and \( \Gamma_{E_k} \) at each time instant depend on \( N \) different linearized models. The matrices \( \Gamma_{C_k}, \Gamma_{D_k} \) are given in [Maljaars et al., 2015c] and \( \Gamma_{E_k} \) can be derived analogically. We use a single prediction model in Phase II, as the dynamics are in this phase well approximated by a linear model accompanied with state disturbance estimates.

### 3.3.3 Input parameterization and coincidence points

To reduce the number of input parameters to be computed, we parameterize \( \tilde{U}_k \) by a relatively small number of parameters \( \tilde{p}_k \). In fact, the parameters \( \tilde{p}_k \) are the inputs at some specified time instants (nodes), where we linearly interpolate between those nodes for the remaining time instants. While allowing more freedom in the early part in order to be able to react to fast dynamics, we keep all inputs constant after a so called control horizon \( N_c \), known as move blocking in MPC. We define:

\[
\tilde{U}_k = P_{\text{map}} \tilde{p}_k, \tag{3.11}
\]
where the parameterization mapping matrix $P_{\text{map}} \in \mathbb{R}^{(N+1) \cdot n_u \times n_p}$ is fixed and chosen offline. This is illustrated in Figure 3.1(a) for the settings discussed in Section 3.4.2.

**Remark.** An important ingredient in MPC stability proofs is recursive feasibility, meaning that the the tail of the previous input sequence cannot be fully contained in the next input sequence. Our choice of $P_{\text{map}}$ does not completely allow for recursive feasibility, but approaches that sufficiently. In the context of time-varying constraints, costs and references, one may also doubt the value of having the recursive feasibility.

To reduce the number of predicted outputs and constraints, we use the concept of coincidence points (see e.g. [Richalet et al., 2009]) such that the reference and predicted outputs are desired to coincide only at a limited number of time instants in the prediction horizon. Additionally, constraints depending on outputs are only imposed at these time instants. The concept of coincidence points is illustrated in Figure 3.1 (b).

**Remark.** We verified that choosing the time between coincidence points not larger than the time between the nodes in the input parameterization, using linear interpolation and considering the diffusive nature of the system, results in no significant excursions of the outputs in between those coincidence points.

### 3.3.4 Cost function

The control objective is to minimize the tracking error for $n_Q$ sets of controlled variables (in our case $\iota(\rho)$ and $W_{th}$), while also minimizing aggressive control
actions and soft constraint violations. This is reflected in the cost function:

\[
J_k = \sum_{i_Q=1}^{n_Q} w_{i_Q} (R_k - Z_k)^T S_{i_Q}^T Q_{i_Q} S_{i_Q} (R_k - Z_k)
\]

\[
+ w_{\Delta U} \Delta \tilde{U}_k^T R_{\Delta U} \Delta \tilde{U}_k + w_{\tilde{U}} \tilde{U}_k^T R_{\tilde{U}} \tilde{U}_k + \varepsilon^T W_{\varepsilon} \varepsilon.
\]

Each index \(i_Q\) corresponds to a set of controlled variables, and has a corresponding diagonal performance weight \(Q_{i_Q}\), scaled by a weight factor \(w_{i_Q}\). The matrix \(S_{i_Q}\) selects the relevant outputs for cost index \(i_Q\): \(Z_{i_Q}^k = S_{i_Q} Z_k\). The weights \(R_{\Delta U}\) and \(R_{\tilde{U}}\) are diagonal input difference and input weights respectively. The constraint violation weight \(W_{\varepsilon}\) sets the ‘softness’ of the (mixed input-) output constraints. The weight factors \(w_{i_Q}\), \(w_{\Delta U}\), \(w_{\tilde{U}}\) and \(W_{\varepsilon}\) facilitate closed-loop controller tuning without recomputing offline computed controller matrices.

We can rewrite the cost function into a more concise form. Given that \(R_k - Z_k = (R_k - Z_0^k) - \tilde{Z}_k\), we define \(\tilde{R}_k = R_k - Z_0^k\), which represents the reference for \(\tilde{Z}_k\). Using (3.10) and (3.11) and preserving only terms dependent on \(\tilde{p}_k\), we can write the cost function as:

\[
J_{\tilde{p},k} = \tilde{p}_k^T \Gamma_{J,\text{quad},k} \tilde{p}_k + \Gamma_{J,\text{lin},k} \tilde{p}_k + \varepsilon_k^T W_{\varepsilon} \varepsilon_k
\]

where the quadratic and linear part in \(\tilde{p}_k\) is respectively:

\[
\Gamma_{J,\text{quad},k} = w_{\Delta U} \Gamma_{J,\Delta U}^{\text{quad}} + w_{\tilde{U}} \Gamma_{J,\tilde{U}}^{\text{quad}} + \sum_{i_Q=1}^{n_Q} w_{i_Q} \Gamma_{J,i_Q}^{\text{quad},k},
\]

\[
\Gamma_{J,\text{lin},k} = \sum_{i_Q=1}^{n_Q} w_{i_Q} \left[ \tilde{R}_k^T S_{i_Q}^T \Gamma_{J,\tilde{R},i_Q}^{\text{lin,k}} \tilde{x}_k + \tilde{x}_k^T \Gamma_{J,\tilde{x},i_Q}^{\text{lin,k}} + \tilde{d}_k \Gamma_{J,\tilde{d},i_Q}^{\text{lin,k}} \right]
\]

The matrices \(\Gamma_{J,i_Q}^{\text{quad},k}, \Gamma_{J,\Delta U}^{\text{quad}}, \Gamma_{J,\tilde{U}}^{\text{quad}}, \Gamma_{J,\tilde{R},i_Q}^{\text{quad}}, \Gamma_{J,\tilde{x},i_Q}^{\text{lin,k}}\) and \(\Gamma_{J,\tilde{d},i_Q}^{\text{lin,k}}\) are computed offline for each time instant to reduce online computational cost and memory use.

### 3.3.5 Constraints

We show here how to impose input, output and mixed input-output constraints and construct the constraint matrices and vectors.

**Input constraints.**

Input magnitude and ramp-rate constraints are imposed as hard limits:

\[
U_{\text{min},k} \leq U_k \leq U_{\text{max},k}, \quad \Delta U_{\text{min},k} \leq \Delta U_k \leq \Delta U_{\text{max},k},
\]

\[
(3.15) \quad (3.16)
\]
respectively, where $\Gamma_\Delta$ is a difference matrix operator such that $\Delta U_k = [(u_{k+1} - u_k)^T \cdots (u_{k+N-1} - u_{k+N-2})^T] = \Gamma_\Delta U_k$. The mixed actuator constraints are put as hard limits:

$$A_{U,\text{mix}} U_k \leq b_{U,\text{mix}}. \quad (3.17)$$

Using (3.11), we can rewrite all input constraints as:

$$A_{\text{inp},\tilde{U}} P_{\text{map}} \tilde{p}_k \leq -A_{\text{inp},\tilde{U}} U_k^0 + b_{\text{set},\text{inp},k}, \quad (3.18)$$

where $A_{\text{inp},\tilde{U}} = [I \gamma_{\Delta}^T - \Gamma_{\Delta} U_{\text{mix}}^T]$. Matrix $A_{\text{inp},\tilde{U}}$ and vector $b_{\text{set},\text{inp},k}$ are calculated offline.

**Output constraints.**

For a number of types of output constraints $n_{Zc}$, we index each output constraint type with index $i_{Zc}$. As common in MPC (see e.g. [Maciejowski, 2002]) we soften the output constraints per type by a constraint relaxation parameter $\varepsilon_{i_{Zc}}$. We can write the output constraints for each type together as:

$$[A_{Z,1} \cdots A_{Z,n_{Zc}}] Z_k - [I_{Z,1} \cdots I_{Z,n_{Zc}}] \leq [b_{Z,1,k} \cdots b_{Z,n_{Zc},k}] \quad (3.19)$$

Using (3.10) and (3.11) we can rewrite this to:

$$[\Upsilon A_{Z,\tilde{x}_c,k} - I_{Z,\text{all}} \varepsilon_{UZ,k}] \leq b_{Z,\text{all},k} - b_{Z,\text{all},k} - \Upsilon A_{Z,\tilde{d}_c,k} \tilde{d}_k - \Upsilon A_{Z,\tilde{d}_c,k} \tilde{d}_k \quad (3.20)$$

The matrices $\Upsilon A_{Z,\tilde{x}_c,k} I_{Z,\text{all}} \Upsilon A_{Z,\tilde{d}_c,k} \Upsilon A_{Z,\tilde{d}_c,k}$ and the vector $b_{Z,\text{all},k}$ are calculated offline.

**Mixed input-output constraints.**

We impose mixed input-output constraints that constrain a weighted combination of inputs and outputs and are softened with constraint relaxation parameter $\varepsilon_{UZ,k}$. We write the mixed input-output constraints as:

$$\theta_{UZ,k} A_{UZ}^U U_k + A_{UZ}^Z Z_k - I_{UZ} \varepsilon_{UZ,k} \leq b_{UZ,k}, \quad (3.21)$$
where the scalar weight $\theta_{UZ,k} \in \mathbb{R}$ may also be set in realtime. Using (3.10) and (3.11) this can be rewritten as:

$$
\left[ \begin{array}{c}
\theta_{UZ,k} \Upsilon_{UZ, \tilde{p}_{k,k}} + \Upsilon_{A_{UZ, \tilde{x}_{k,k}}, \tilde{p}_{k,k}} \\
0 \\
- I_{UZ, \text{all}} 
\end{array} \right] \left[ \begin{array}{c}
\tilde{p}_{k} \\
\varepsilon_{Z,k} \\
\varepsilon_{UZ,k} 
\end{array} \right]
\leq
\left[ \begin{array}{c}
b_{UZ,k} - \theta_{UZ,k} b_{UZ,0} - b_{UZ,k} \\
-b_{UZ,0} \\
- b_{UZ,k} 
\end{array} \right] - \Upsilon_{A_{UZ, \tilde{x}_{k,k}}}. \quad (3.22)
$$

The matrices $\Upsilon_{UZ, \tilde{p}_{k,k}}$, $\Upsilon_{A_{UZ, \tilde{x}_{k,k}}, \tilde{p}_{k,k}}$, $I_{UZ, \text{all}}$, $\Upsilon_{A_{UZ, \tilde{x}_{k,k}}}$ and the vectors $b_{UZ,0}$ and $b_{UZ,k}$ are calculated offline.

### 3.3.6 Quadratic Programming problem formulation

The cost function and constraints as defined in the above equations allow to formulate a QP problem. By using $\xi_{k}^{T} = [\tilde{p}_{k}^{T} \varepsilon_{Z,k}^{T} \varepsilon_{UZ,k}^{T}]$ and $\varepsilon_{k}^{T} = [\varepsilon_{Z,k}^{T} \varepsilon_{UZ,k}^{T}]$, we can write the cost function (3.13) with the constraints (3.18), (3.20) and (3.22) into the standard format of a QP-problem:

$$
\begin{align*}
\text{minimize} & \quad J_{\tilde{p},k} = \xi_{k}^{T} \left[ \begin{array}{cc}
\Gamma_{\text{quad}} & 0 \\
0 & W_{k} 
\end{array} \right] \xi_{k} + \left[ \begin{array}{c}
\Gamma_{\text{lin}}^{\text{inp}} \\
0 
\end{array} \right] f_{k}^{T} \\
\text{subject to} & \quad \left[ \begin{array}{cc}
A_{\text{inp, } \tilde{p}} & 0 \\
A_{Z, \text{ineq,k}} & \Upsilon_{A_{UZ, \tilde{x}_{k,k}}, \tilde{p}_{k,k}} \\
A_{UZ, \text{ineq,k}} & \Upsilon_{A_{UZ, \tilde{x}_{k,k}}}
\end{array} \right] \xi_{k} \leq \left[ \begin{array}{c}
b_{\text{set, } \text{inp, } k}^{0} + b_{\text{set, } \text{inp, } k} \varepsilon_{Z,k}^{T} \\
b_{Z, \text{ineq,k}} \\
b_{UZ, \text{ineq,k}}
\end{array} \right] \\
\text{and} & \quad \left[ \begin{array}{cc}
I_{n_{a}} & 0 \\
P_{\text{map}} & 0 
\end{array} \right] A_{\text{eq}} \xi_{k} = \left[ \begin{array}{c}
\tilde{u}_{k} \\
b_{\text{eq}}
\end{array} \right],
\end{align*}
$$

where the equality constraints fix the currently active input on the system.

This QP-problem is in this chapter solved using the fast dual active set solver [Wills, 2007]. The simulations in this chapter yield a QP-problem with 50 variables, 564 inequality constraints and 4 equality constraints that is solved in less than 3 milliseconds on a standard off-the-shelf laptop equipped with an Intel i7 processor. Only a fraction of this time is required to build the QP-problem from the real-time information. Even considering a margin for worse cases with more active constraints, this is still fast enough for implementation on currently operational tokamaks with the existing dedicated hardware.

### 3.3.7 State and disturbance estimates

The proposed controller requires knowledge of the state $x_{k}$ and in addition we want to estimate low-frequency state disturbances $\tilde{d}_{k}$. Commonly, states and
state disturbances are written in augmented state space form (e.g. [Rossiter, 2013] and estimated together in e.g. a Kalman filter, in which case the states converge on the same time scale as the state disturbances. In our application other algorithms (real-time tokamak equilibrium reconstruction code (e.g. [Moret et al., 2015] or dynamic state observer (e.g. [Felici et al., 2014a]) deal with the estimation of plasma profiles, taking into account all available noisy measurements. As we can derive the state directly from these estimated profiles, we estimate only the low-frequent state disturbances, avoiding the additional lag in the convergence of the state. In this chapter we focus on the controller performance and choose to have direct access to the state of the plasma simulator, which has the same state representation as the controller.

We choose to estimate the constant state disturbances only in Phase II using a Kalman filter, based on the LTI state space model at the end of Phase I. The disturbance $\tilde{d}_k$ is modeled as an integrator driven by zero-mean white-noise $w_d$ having covariance matrix $Q_d$:

$$\tilde{d}_{k+1} = \tilde{d}_k + w_d. \quad (3.24)$$

The state disturbance ‘measurement’ $\tilde{d}_k\text{meas}$ is given by the discrepancy between the provided estimated state and the predicted state based on the model (3.3) without disturbance:

$$\tilde{d}_k\text{meas} = (x^\text{meas}_k - x^o_k) - A_k(x^\text{meas}_{k-1} - x^o_{k-1}) - B_k\tilde{u}_{k-1}. \quad (3.25)$$

The state disturbance ‘measurement’ $\tilde{d}_k\text{meas}$ has covariance matrix $R$ being the identity matrix. The covariance matrix $Q_d$ is chosen to enforce high correlation between neighboring spatial points (see [Felici et al., 2014a] for more details), as only spatially smooth disturbances are expected to be physically correct. We choose $Q_d$ as:

$$Q_d = Q_{\text{gain}} \begin{bmatrix} \Lambda^+_\psi Q_{\psi}(\Lambda^+_\psi)^T & 0 \\ 0 & \Lambda^+_T Q_T(\Lambda^+_T)^T \end{bmatrix}, \quad (3.26)$$

where $(\cdot)^+$ is the Moore-Penrose pseudo inverse and $\Lambda_\psi$ and $\Lambda_T$ contain the basis functions for $\psi$ and $T$: i.e. $[\psi^T_k T^T_{e,k}]^T = \text{diag}(\Lambda_\psi, \Lambda_T)x_k$. The matrix $Q_{\psi} = Q_T$ is a symmetric Toeplitz matrix with $\exp(-\hat{\rho}^2/w_s^2)$ being its first row and column and $w_s = 0.2$. We choose the scalar gain $Q_{\text{gain}} = 0.1$, resulting in a settling time of the disturbance estimates of about 5-10 sample times. The covariance matrix $Q_d$ is shown in Figure 3.2. The disturbance observer is given by:

$$\tilde{d}_{k+1} = \tilde{d}_k + M(\tilde{d}_k\text{meas} - \tilde{d}_k). \quad (3.27)$$

The discrete time Kalman gain $M = P(P + R)^{-1}$ is found by solving the discrete Riccati equation $P(P + R)^{-1}P = Q_d$. 

Chapter 3. Model predictive control of the current density distribution and stored energy in tokamaks using trajectory linearizations

\[ Q \psi = Q T e \]

\[ \rho_0 \]

\[ \rho_1 \]

\[ Q, \psi = Q T, \rho \]

Figure 3.2. Covariance matrix \( Q_\psi = Q_{Te} \), chosen to enforce high correlation between neighboring spatial points.

3.4 Performance analysis in simulations

In this section we analyze the performance of the proposed MPC-controller in simulations using the nonlinear plasma transport simulator RAPTOR with plasma parameters that approach high performance plasma experiments in the ASDEX-Upgrade tokamak as described in [Hobirk et al., 2012b]. First the plasma scenario with its feedforward inputs and corresponding nominal output evolution are introduced. Then we discuss the controller settings and conclude with presenting and analyzing the results.

3.4.1 Plasma scenario

We choose feedforward trajectories for the inputs as follows. The plasma current \( I_p \) is ramped-up from its initial value of 400 kA to its flat-top value of 1MA that is achieved at 1s. From the EC-beams we request after 0.5 seconds instantaneously: \( P_{ec,1} = 1\) MW and \( P_{ec,2} = 0.5\) MW. The power request to the NBI system \( (P_{nbi}) \) is ramped from 0 to 10 MW between 0.7 and 0.9 seconds. The feedforward input trajectories are given in Figure 3.1(a) as thin solid lines.

These feedforward trajectories provide a nominal profile evolution resulting in a plasma stored energy of about 1MJ during the flat-top phase (Figure 3.1(b) as thin solid line). The inverse safety factor profile \( \iota \) achieved at 1 second is still transient and violates briefly the \( \iota(\rho) \leq 1 \) limit (Figure 3.1(e)). The normalized plasma pressure \( \beta_N \) achieves a nominal value just below 2.8 in the flat-top phase (Figure 3.1(d) as thin solid line).
Figure 3.1. Simulation results indicating performance of designed MPC controller. Simulation is divided in six sections of 1 second, the labels on top indicate the events during that section. Panels from top to bottom: input trajectories (a), stored energy (b), relative error norm on $i(\rho)$ (c), normalized plasma pressure (d) and maximum $i(\rho)$ (e). Thin solid lines are the feedforward evolution and dashed thick lines are the feedback controlled case using the MPC controller.
Chapter 3. Model predictive control of the current density distribution and stored energy in tokamaks using trajectory linearizations

3.4.2 Controller settings

Here we describe the controller settings chosen for the closed-loop simulation. Using the nominal simulation with a sample time of 10 milliseconds we obtain the trajectory linearizations and build the prediction model (3.10) after which we can calculate the controller matrices and vectors using the settings below.

Input parameterization and coincidence points.

Relevant time scales were found by analyzing the linearizations in (3.8). The slowest time scale is 2.6 seconds and second slowest 0.8 seconds. We choose a control horizon of 36 steps and a prediction horizon of 120 steps, such that we include the second slowest time scale between control and prediction horizon, but not the slowest with the largest length scale. The choice of input parameterization is given in Figure 3.1(a) and choice of coincidence points in (b-c).

Controller weights.

The controller weights $Q_i$ are chosen equal to identity, except for the tail of the predicted sequence where the diagonal entries are equal to the number of time instances the coincidence points represent. The weights $R_{\Delta U}$ and $R_{\hat{U}}$ are also chosen as identity. The scalar weights are tuned in representative simulations based on rules of thumb and intuition. By choosing $w_{\hat{U}} = 1$, all other costs are relative to $w_{\hat{U}} = 1$ and chosen as $w_{iQ=1} = 10^4$, $w_{iQ=2} = 10^4$, $w_{\Delta U} = 2 \cdot 10^2$, and $W_\varepsilon = \text{diag}([1 1]^T) \cdot 10^8$.

Constraints.

The input constraints are chosen as:

- $I_p$: $0 \leq I_p \leq 1.5$ and $-1 \leq \frac{dI_p}{dt} \leq 1$
- $P_{ec}$: $0 \leq P_{ec,1} \leq 4$, $0 \leq P_{ec,2} \leq 4$ and $P_{ec,1} + P_{ec,2} \leq 4$
- $P_{nbi}$: $0 \leq P_{nbi} \leq 12.5$

with units in [MA], [MA/s] and [MW]. As output constraints we impose the constraint $\iota(\rho) \leq 1 \forall \rho \leq 0.5$. As mixed input-output constraint we choose $\beta_N \leq 2.8$, which is a conservative value [Wesson, 2011].

References.

During Phase I the reference is chosen as the nominal controlled variables evolution. In Phase II a fixed reference is chosen equal to the nominal controlled variables at the end of the nominal simulation ($t = 6$), which is a reachable target in steady state in the absence of model mismatch and disturbances. Note
that the prediction horizon at the first time instant $t = 0$ already includes the first 20 steps of Phase II including its steady state target.

### 3.4.3 Performance analysis of controller

The designed controller is analyzed in the simulations given in Figure 3.1 (time evolution) and Figure 3.2 (snapshots $\nu$-profile), where the controller was connected with the nonlinear simulator RAPTOR. The simulator was modified to have a model mismatch being a reduction of the electron heat transport by 30\% ($c_2$ in (3.2)). Without feedback, this results in significantly higher temperatures and stored energy and in a different evolution of $\nu(\rho)$. Hereafter we discuss the events during each period of 1 second and the controller performance in response to that.

**1: Ramp-up phase.** The model mismatch causes the controller to change its inputs significantly with respect to the feedforward case Figure 3.1(a). The error on $\nu(\rho)$ is increasing (c) whereas the error on the stored energy (b) is rather small. Although the $\nu(\rho)$-limit is handled correctly (e), the $\beta_N$-limit is violated (d). The $\nu$-profile at 1 second is not on its steady state target, as can also be noticed in Figure 3.2

**2: Set flat-top reference, disturbance estimation off.** The controller reduces the steady state error on the $\nu$-profile (c) by changing the heating powers (a). As disturbance estimation is still off, both an error in $W_{th}$ remains (b) and the $\beta_N$-limit is violated (d).

**3: Disturbance estimation on.** With disturbance estimation switched on, the error in $W_{th}$ is reduced (b) and the $\beta_N$-limit is no longer violated (d), which requires only minor adjustments of the inputs (a). The error in the $\nu$-profile approaches zero (c), as is also clear in Figure 3.2.

**4: Change reference to $W_{th,ref} = 1.2$.** This change in reference is tracked by the controller up to some error (b), whereas the error in $\nu(\rho)$ starts to rise (c), indicating a tradeoff between $W_{th}$ and $\nu(\rho)$. More heating power from NBI and EC is requested (a) to provide the additional stored energy (b). The plasma current is increased to prevent violation of the $\beta_N$-limit (e).

**5: Change limit to $P_{NBI} \leq 7.5$MW.** Mimicking a hardware failure of part of the NBI-system, the available NBI power is reduced (a). The controller request the EC-system to replace the missing NBI contribution (a), however the error in $\nu(\rho)$ is still rising (c) as the controller is not able to lower the plasma current $I_p$ due to the $\beta_N$-limit (d) given in (3.6).

**6: Increase limit $\beta_N \leq 3.2$.** The restriction of the $\beta_N$-limit is reduced by moving it to a less conservative value (d). The controller has now room to
reduce the plasma current $I_p$ (a), necessary to keep $\iota(\rho)$ at its reference (c) in the presence of the limited available NBI-power and the higher reference for $W_{th}$ (b). The error on $\iota(\rho)$ now indeed decreases (c), as well as the error on $W_{th}$ (e).

Evaluating the controller’s performance based on these results, we conclude that the controller designed in Section 3.3 is able to meet the requirements as given in Section 3.2.3. The tracking error is minimized for both the $\iota$-profile and the stored energy $W_{th}$ and during phase II the stationary error is minimized by exploiting the disturbance estimation. The controller can effectively handle hardware and physical constraints on inputs, outputs and combinations of these.

The control performance is limited by the combination of active constraints and the reachability of references. This is inherent to the control problem and the relevant physical and actuator constraints and not related to the MPC controller design. It is instead clear that the MPC controller allows the system to operate at the limits of its achievable performance.

3.5 Conclusions and outlook

In this chapter we applied model predictive control to the control of the current density distribution and stored energy in tokamaks. We introduced the controller requirements for profile controllers in the context of future tokamaks where a supervisory controller may allocate in real-time resources, objectives and references to multiple controllers. An MPC controller is designed so as to fulfill these requirements. Using a linear time-varying system description around a trajectory enables the application of linear MPC techniques that are computationally tractable for implementation on existing tokamaks. The controller performance has been analyzed in a closed-loop simulation that is representative of high performance plasma experiments in the ASDEX-Upgrade tokamak.
The performance analysis revealed that the controller can indeed minimize the tracking error in the presence of model mismatches and disturbances by incorporating state disturbance estimates. Changes in references, actuator constraints and mixed input-output constraints are effectively handled.

Future work includes further validation in closed-loop simulations and ultimately on existing tokamaks. The controllers ability to function in the context of multiple controllers governed by a supervisory controller will also be explored in the near future. In addition, responding to changing control objective requests in real-time by modifying scalar cost weights within reasonable bounds may be investigated. Future larger tokamaks exhibit longer timescales that might enable the use of fully nonlinear MPC techniques that can work in cases where the validity regions of the linearized models is exceeded. A particularly relevant application can be to terminate the plasma as fast as possible in case of unexpected events, without violating important physical and machine constraints.

This chapter has shown the potential of a model predictive controller in controlling the current density distribution and stored energy in plasma transport simulations and encourages further exploration and use in experiments.
Chapter 4

Profile control simulations and experiments in TCV: a controller test environment and results using model predictive controller

The work presented in this chapter will be submitted to Nuclear Fusion.
4.1 Introduction

Control over the safety factor profile and plasma pressure or stored energy is important for high performance tokamak operation, especially in hybrid and advanced scenarios [Gormezano et al., 2007]. Specific safety factor profiles can lead to improved confinement by reducing turbulent transport or to steady state operation by maintaining a significant amount of self-driven bootstrap current at zero loop voltage. Therefore, it is important to establish reliable profile control routines in currently operational tokamaks that can be transferred to future tokamaks.

Recently, many model-based profile controllers have been developed using a wide variety of controller models and control methods. The applied methods include adaptive control [Kim and Lister, 2012], backstepping control [Boyer et al., 2014], passivity-based control [Vu et al., 2014], Lyapunov control [Bribiesca Argomedo et al., 2013], linear-quadratic-integral control [Boyer et al., 2013; Moreau et al., 2013], model predictive control [Maljaars et al., 2015c; Wehner et al., 2016] and robust control [Barton et al., 2014a]. Some of these have been implemented in experiments, others only in simulations.

The highest plasma performance in tokamaks is often achieved close to actuator limits (e.g. maximum available heating source power) and in addition close to areas of the plasma parameter space that are prone to disruptions or deleterious MHD behavior (e.g. Neoclassic Tearing Modes (NTMs)). Even if these actuator and plasma physics limits may not be restrictive at the target operating point, these are still limiting during transient phases (e.g. ramp-up, ramp-down or transitions between operating points). Therefore, control methods are required that can effectively deal with these actuator limits and ensure operation within the safe plasma parameter space. Most of the control methods mentioned above impose only limits on the actuator signal after it has been computed by the controller. This approach has the important limitation that the controller is not aware of the limits in actuators and plasma in the actuator input calculation and cannot anticipate for these limits.

Model predictive control (MPC) is a well established control method that can take these time-varying actuator and process parameter limits into account in the optimal actuator input calculation itself [Camacho and Bordons, 2004; Maciejowski, 2002; Mayne, 2014; Rossiter, 2013]. MPC was first applied to profile control in simulations with simple models and only fixed actuator constraints [Ou et al., 2011; Ouarit et al., 2011]. We presented in [Maljaars et al., 2015c] a model predictive controller that uses multiple linearized models to control the safety factor profile while effectively dealing with constraints on actuators and physics limits in ITER simulations. In [Maljaars et al., 2015a] we extended this work by including other controlled variables, time-varying references and nonlinear constraints, and the estimation of state disturbances. The controller performance was demonstrated in ASDEX-Upgrade H-mode simulations. Even more recently,
input-constrained MPC is also successfully applied in profile control experiments at the DIII-D tokamak [Wehner et al., 2016].

In this chapter we present a model predictive profile controller and its performance in simulations and experiments in TCV L-mode discharges. Successful tracking of the plasma pressure and inverse safety factor profile in the presence of disturbances is achieved as well as effective handling of time-varying input-constraints. It is important to note that, since measurements of the core current density profile on TCV are presently missing, the $q$-profile that is used as a basis for feedback control in this chapter is only a model-based estimate. As such, we present these results as a demonstration of implementation and operation of a profile controller rather than a demonstration of having achieved a given $q$-profile in TCV plasmas. The results in this chapter encourage further exploration and use of MPC for profile control in future experiments.

Implementation of controllers for e.g. the density, temperature and current distribution requires knowledge of these quantities in real-time. These quantities can be reconstructed using real-time equilibrium reconstruction [Ferron et al., 1998; Moret et al., 2015] or dynamic state observers [Felici and Sauter, 2012a] that integrate the available (noisy) diagnostic signals into an estimate of the plasma state.

It is beneficial to prepare controllers well before being tested in experiments in order to minimize testing and commissioning time on the experiment. This requires a controller development and test environment involving interfaces to plasma state reconstruction, a fast simulator and possibly experimental data. One example of such a system has been used on DIII-D [Ferron et al., 1995] and is now also used at EAST, KSTAR, and NSTX-U [Walker et al., 2015b]. A similar software environment is being prepared for ITER, where it is known as PCSSP [Walker et al., 2015a]. These environments allow to simulate controllers in closed-loop (also on the control system hardware).

In this chapter we present also a profile controller development and implementation software environment that is employed for pre-experiment simulations of various profile controllers as well as testing their performance in TCV experiments. This software environment allows to develop profile controllers in representative simulations (both on a local computer and in the TCV control system) and afterwards test them in experiments without changing the controller code itself.

The remainder of this chapter is organized as follows. In Section 4.2 we summarize the plasma transport modelling using RAPTOR, that is used to design the MPC controller as well as for closed-loop testing. The experimental physics scenario is presented in Section 4.3 where also the control problem is defined. The controller development and testing environment is presented in Section 4.4. The MPC controller design is summarized in Section 4.5. The performance of the designed controller is analyzed in simulations (Section 4.6) and experiments on TCV (Section 4.7). A qualitative comparison of the MPC
controller’s performance to the performance of two other profile controllers is provided in Section 4.8. Finally, conclusions and suggestions for further research are given in Section 4.9.

4.2 Modeling plasma transport using RAPTOR

A rapid plasma transport simulator is used in this chapter for closed-loop controller testing and plasma state reconstruction (Section 4.4), as well as for deriving the MPC controller (Section 4.5). We employ RAPTOR [Felici and Sauter, 2012a; Felici et al., 2011b] in this chapter, a control-oriented, physics-based 1D transport code. RAPTOR solves the simplified non-linear coupled transport of the electron temperature $T_e$ and the poloidal magnetic flux $\psi$ as a function of the normalized square-root toroidal flux $\rho$, represented by partial differential equations (PDEs) [Hinton and Hazeltine, 1976]:

\[
\sigma_\parallel \left( \frac{\partial \psi}{\partial t} - \frac{\rho \dot{\Phi}_b}{2\Phi_b} \frac{\partial \psi}{\partial \rho} \right) = \frac{F^2}{16\pi^2\mu_0\Phi_b^2\rho} \frac{\partial}{\partial \rho} \left[ \frac{g_2 g_3}{\rho} \frac{\partial \psi}{\partial \rho} \right] - \frac{B_0}{2\Phi_b \rho} V'_\rho (j_{aux}(u_k) + j_{bs}),
\]

(4.1)

\[
\frac{3}{2} (V'_\rho)^{-5/3} \left( \frac{\partial}{\partial t} - \frac{\dot{B}_0}{2B_0} \frac{\partial}{\partial \rho} \rho \right) \left[ (V'_\rho)^{5/3} n_e T_e \right] + \frac{1}{V'_\rho} \frac{\partial}{\partial \rho} \left( -\frac{g_1}{V'_\rho} n_e \chi_e \frac{\partial T_e}{\partial \rho} \right) = P_e(u_k)
\]

(4.2)

Details of the terms in these equations in RAPTOR depend on the plasma scenario or application. We choose here:

- Bootstrap current profile $j_{bs}$ and neoclassical conductivity $\sigma_\parallel$ are calculated using the Sauter-equations [Sauter et al., 1999, 2002a].
- Multiple sources and sinks of thermal energy are modeled ($P_e$) including:
  - Ohmic heating.
  - Power deposition by Electron Cyclotron Heating (ECH). The EC actuator power is given in the actuator inputs $u_k$.
  - Losses from Bremsstrahlung, line radiation, and electron-ion heat exchange. In these calculations, the $T_e(\rho)/T_i(\rho)$ ratio is prescribed.
- The auxiliary current drive $j_{aux}(u_k)$ by Electron Cyclotron Current Drive (ECCD) is calculated using a scaling law including a $T_e/n_e$ scaling [Felici and Sauter, 2012a].
- An ad-hoc transport model in [Felici and Sauter, 2012a] is used to calculate the electron thermal diffusivity $\chi_e$. 
4.3 Plasma scenario and control problem definition

- Geometry profiles quantities $V' = \frac{\partial V}{\partial \rho}, F = RB\phi, g_1 = \langle (\nabla V)^2 \rangle, g_2 = \langle (\nabla V)^2 / R^2 \rangle, g_3 = \langle 1/R^2 \rangle$ as well as the scalar $\Phi_b$ (toroidal flux enclosed by plasma) can be time-varying and provided by a real-time equilibrium reconstruction codes such as LIUQE [Moret et al., 2015].

- The electron density profile $n_e$ is prescribed, but can be time-varying and provided by a real-time density profile reconstruction code such as [Blanken et al., 2015].

- A boundary condition for (4.1) is obtained by prescribing the total plasma current, which is seen as an actuator input in $u_k$. For (4.2), the edge electron temperature is prescribed.

Inside RAPTOR, (4.1) and (4.2) are written as a nonlinear state space model with the following state update equation and its corresponding output equation at time instant $t_k$:

\begin{align*}
f(x_{k+1}, x_k, u_k) &= 0, \quad (4.3) \\
y_k &= h(x_k). \quad (4.4)
\end{align*}

The state vector $x_k$ contains the coefficients that are used to parameterize the $T_e$ and $\psi$ profiles and the vector $u_k$ contains the actuator inputs, in this case the total plasma current and EC actuator powers. The state update equation gives the state at the next time instant $x_{k+1}$ based on the present state $x_k$ and actuator inputs $u_k$. The output vector $y_k$ can contain many quantities that are a function of the state. Further details on the numerical methods applied to evolve (4.1) and (4.2) in RAPTOR are given in [Felici and Sauter, 2012a]. The extension of RAPTOR to include time-varying geometry and density profiles is detailed in [Teplukhina et al., 2017]. The used nonlinear state space format allows RAPTOR to provide linearizations of these state and output equations that can be employed in an observer and controller.

4.3 Plasma scenario and control problem definition

In this section we will define the plasma scenario for the profile control experiments in TCV presented in this chapter. From this plasma scenario we will also define a control problem in this section.

4.3.1 Plasma scenario

The designed plasma scenario for the TCV experiments described in this chapter will be introduced here in terms of chosen hardware and plasma configuration.
The physics goal for these experiments is to routinely achieve and maintain a broad range of targets for the inverse safety-factor profile $\mathcal{I}(\rho)$ and the plasma pressure $\beta$ by using the available actuators. We may distinguish three cases where we wish to achieve and maintain targets for:

- Only the plasma pressure $\beta$.
- Only the inverse safety-factor profile $\mathcal{I}(\rho)$.
- Simultaneously the plasma pressure $\beta$ and the inverse safety-factor profile $\mathcal{I}(\rho)$.

TCV has the feature that many different stationary safety factor profiles can actually be achieved in a single experiment. This is due to the small current redistribution time w.r.t. the total shot time (about 150ms versus 2.5 seconds in this plasma scenario). Hence, it is possible to investigate control of the $q$-profile during the flat-top phase. The plasma pressure $\beta$ evolves on the energy confinement time (here about 5ms), indicating the wide range of time scales of the quantities of interest.

To achieve a broad range of safety-factor profiles as well as plasma pressures in these plasmas, central ECH and central co- and counter-ECCD is employed at low density in L-mode discharges. In addition, the plasma current $I_p$ is feedback controlled via the inductive voltage from external coils, and can be adjusted in real-time to broaden the range of achievable profiles. The plasma heating and current drive setup is illustrated in Figure 4.1. The plasma boundary shape (last closed flux surface) shows a plasma in limiter configuration.

Gyrotrons in TCV are grouped in so-called clusters, which share a single power supply. So-called cluster A uses one gyrotron, connected to launcher 1 (L1) to drive counter-current in these experiments. Cluster B has two gyrotrons that are used to drive co-current via launchers L4 and L6. The magnetic axis is located at the resonant magnetic surface so that the EC system effectively provides central co/counter ECCD. The position of the magnetic axis is feedback controlled, whereas only feedforward shape control is used.

The actuator deposition profiles and the achievable plasma profiles are illustrated in Figure 4.2 for shot #54423 during the flat-top phase. The normalized volume integrated power density profile retrieved from TORAY-GA (a) shows a central and narrow deposition profile. The EC deposition width used in RAPTOR is larger than that retrieved from TORAY-GA, this allows the reduce the required spatial resolution so as to achieve real-time capable calculations. A minor offset in the surface integrated driven current can be noticed, mainly caused by using different electron temperature and electron density profiles in these calculations (Thomson measurements for TORAY-GA and real-time profile reconstructions for RAPTOR). The normalized surface integrated current density profile (b) reveals the opposite current drive direction of the EC-clusters.
4.3 Plasma scenario and control problem definition

Figure 4.1. Experimental plasma scenario for TCV plasma profile control experiments using two clusters of gyrotrons/launchers and plasma current $I_p$ as actuators. Plasma is in limiter configuration. Ray-tracing using the TORAY-GA code [Matsuda, 1989] is used to visualize path EC beams and absorption near to the resonant magnetic surface. Cluster A drives counter-current via launcher L1, cluster B drives co-current via launchers L4 and L6.

The reconstructed profiles at this time for $T_e$ and $\iota = 1/q$ are given in panels (c) and (d) respectively. The temperature profile shows that the scenario is in L-mode (no pedestal). The range of achieved profiles in multiple experiments presented in this chapter is also provided (grey, shaded). Note that some of the reconstructed $\iota$-profiles are nonphysical, significantly higher than 1, which would correspond to $q < 1$. In the plasma, the sawtooth instability would prevent such a $q$-profile, but this effect was not included in the reconstructions shown here (see Section 4.3.4).

The density (c) is purposely chosen low ($n_e(\rho = 0) \approx 2 \cdot 10^{19} m^{-3}$) in order to maximize the driven current by ECCD, which scales with $T_e/n_e$. The volume-averaged density was controlled to constant value in this plasma scenario.

4.3.2 Control problem

The defined physics goal in the previous section cannot be achieved using only pre-programmed (feedforward) actuator requests. In that case the $\iota$-profile and plasma pressure $\beta$ will likely vary due to shot-to-shot varying experimental conditions and the presence of disturbances and MHD-events. Hence active feedback control is required to ensure that targets are achieved and maintained. The
Chapter 4. Profile control simulations and experiments in TCV: a controller test environment and results using model predictive controller

Figure 4.2. Illustration of actuator deposition profiles and achievable plasma profiles. Normalized volume integrated power and surface integrated current density deposition profiles for EC clusters A and B (a)-(b) obtained from TORAY and RAPTOR; RT reconstructed temperature $T_e$ and density $n_e$ profiles (c), and $\iota$-profile (d). Data from shot #54423 at 0.862s. Note that the EC deposition width used in RAPTOR is larger than the experimental value; this is done to reduce the required spatial resolution to allow the calculations to be real-time capable. The achieved range of $T_e$ and $\iota$-profiles in a number of experiments is also given in (grey, shaded).

The physics goal is therefore translated into a control problem with requirements for the profile controller.

The profile controller should be able to:

- Minimize the tracking error for time-varying $\iota$- and/or $\beta$-references as given in a performance index (see Section 4.3.3).
- Be robust against model mismatches and disturbances in the specific class of L-mode discharges as defined in the experiment overview during the current flat-top phase.
- Handle time-varying actuator input constraints (amplitude and ramp-rates).
- Stay within the available 0.7ms computational time per 1ms time step.

The controller is only activated during the current flat-top phase. Before controller activation, the plasma shape, density profiles and EC power can evolve
towards their nominal values. The controller cycle time $T_s = 1\text{ms}$ is chosen roughly 5 times smaller than the energy confinement time ($\tau_E \approx 5\text{ms}$).

To achieve the control goal, the controller can act on the plasma current $I_p$ and the power requests to the EC-clusters $P_A$ and $P_B$ such that the actuator input vector $u_k$ at time $t_k$ can be defined as:

$$u_k = \begin{bmatrix} I_p(t_k) [A] \\ P_A(t_k) [W] \\ P_B(t_k) [W] \end{bmatrix}.$$  \hfill (4.5)

The actuator inputs are subject to (time-varying) amplitude and ramp-rate constraints that can be written as:

$$u_{\text{min},k} \leq u_k \leq u_{\text{max},k},$$  \hfill (4.6)

$$\Delta u_{\text{min},k} \leq \frac{u_{k+1} - u_k}{T_s} \leq \Delta u_{\text{max},k}.$$  \hfill (4.6)

Time-varying power limits are especially important for cluster A. For technical reasons, powers on cluster A in the range between 100kW and 550kW cannot be delivered longer than 470ms. After this time, the requested power must be in the range 550kW-750kW. Cluster B with its two connected gyrotrons does not have this technical limit and can deliver powers between 360kW and 900kW (2x180kW and 2x450kW). No ramp-rate limits are imposed on the power requests to cluster A and B. Amplitude and ramp-rate limits for the plasma current vary between simulations and experiments and are given in the results in Sections 4.6 and 4.7.

### 4.3.3 Controller performance criterion

The performance of a profile controller is evaluated based on control error indicators for $\beta$ and the $\iota$-profile. To obtain a scalar error indicator from a $\iota$-profile error, we introduce a weighted norm of the difference between the reference and achieved $\iota$-profile. A mixed norm can also be defined, representing a weighted average of the two norm functions:

$$J_\beta(t_k) = \left(\frac{\beta(t_k) - \beta_{\text{ref}}(t_k)}{\beta_{\text{ref}}(t_k)}\right)^2,$$

$$J_{\iota}(t_k) = \sum_{i=1}^{n_{\iota}} W(\rho_i) \left(\frac{\iota(\rho_i, t_k) - \iota_{\text{ref}}(\rho_i, t_k)}{\iota_{\text{ref}}(\rho_i, t_k)}\right)^2,$$

$$J_{\text{tot}}(t_k) = \nu_\beta J_\beta(t_k) + \nu_{\iota} J_{\iota}(t_k).$$  \hfill (4.7)

The weighting profile $W(\rho)$ allows to set the importance of tracking the $\iota$-profile in a certain region. The chosen weighting profile $W(\rho)$ for this chapter
is given in Figure 4.3, indicating that we choose to track the $\iota$-profile in the plasma core, leaving freedom at the edge to use $I_p$ as actuator (since $I_p$ strongly influences $\iota(\rho = 1)$). The norm $J_\iota(t_k)$ is computed by evaluating the $\iota$-profile on $n_\rho = 11$ equidistant $\rho$-grid points.

The weights $\nu(\cdot)$ allow to set the relative importance of $\beta$ or $\iota$-control and can be used for the three profile control purposes defined in the physics problem:

- $\beta$-only control: $\nu_\beta = 1, \nu_\iota = 0$
- $\iota$-only control: $\nu_\beta = 0, \nu_\iota = 1$
- $\beta$ and $\iota$ control: $\nu_\beta = \nu_\iota = \frac{1}{2}$

### 4.3.4 Note on current density profile estimates

As will be discussed in Section 4.4.2, the $\iota$-profile and plasma pressure (and other plasma profiles and parameters) are computed by a model-based state observer algorithm, which merges model-based predictions with diagnostic measurements. For the temperature profile, a central temperature measurement provides an effective real-time constraint. However, as TCV presently lacks measurements of the internal magnetic field in the plasma region, it is important to note that the $q$-(or $\iota$-) profile estimates are calculations based exclusively on a poloidal flux diffusion model. As a consequence, the $q$-profile drops well below 1 in many cases since the effect of sawteeth was not considered, while this is physically unrealistic. In this chapter we focus, nevertheless, on the performance of the controller in this imperfect situation. While we expect the true $q$-profile to globally follow the trends reported here, we do not claim that the core $q$-profile estimate accurately represents the situation in the plasma. A comparison of the real-time reconstructed $q$-profiles used for control with another reconstruction is given in Section 4.7.2.
4.4 Controller development and implementation environment

As mentioned in the introduction, it is important to prepare controllers thoroughly before testing them in experiments. In this section we present the set of controller development and implementation tools that facilitates the efficient preparation of profile controllers in simulations and implementation in experiments. These tools are all prepared in the MATLAB Simulink block-programming language, which greatly facilitates re-using components in different stages of the development and implementation with minimal changes.

First a closed-loop simulation tool is prepared, separate from the TCV control system software environment that allows to largely prepare the profile controller on a local computer. Next we present the controller development and implementation tools inside the TCV control system environment that allows to develop the controller in more comprehensive closed-loop simulations and finally test it in experiments. It is important to note that it is straightforward to transfer these tools to another control testing environment, e.g. PCSSP [Walker et al., 2015a], for re-use in simulations for ITER and other tokamaks.

4.4.1 Controller development and validation in simulations

A simulation environment has been developed to interface the profile controllers with the RAPTOR simulator for closed-loop simulations. The aim of this simulation environment is to test the profile controller in stand-alone in an ideal situation, without needing to consider interfaces to e.g. state reconstruction codes. This simulation environment is visualized in Figure 4.1.

The RAPTOR simulator uses the actuator inputs and prescribed density and geometry information to evolve the plasma profile state and provides the plasma profiles and parameters at the next time step to the profile controller. The profile controller calculates the actuator inputs at each time step based on the provided signals including the RT actuator limits. Details on the provided signals to the controller are given in Appendix 4.A.

To test controllers under circumstances that could be expected during experiments, test cases with different sets of plasma parameters were defined with various targets for the $\imath$-profile and $\beta$. These test cases range from different parameters for current drive efficiency and electron heat transport, to simulated events such as the appearance of an Neoclassical Tearing Mode (NTM) resulting in a confinement drop. Results of closed-loop control simulations using this setup are shown in Section 4.6.
Chapter 4. Profile control simulations and experiments in TCV: a controller test environment and results using model predictive controller

4.4.2 Controller development and implementation in TCV control system software environment

Once the controller has been tested and verified in the environment described in Section 4.4.1, it is ready to be tested in the more complete TCV control system software environment that includes other real-time algorithms. The digital TCV control system is distributed over several computational nodes linked via shared reflective memory and accommodates hundreds of diagnostic inputs and actuator outputs [Le et al., 2014]. Details on the recent upgrades to and applications of the TCV control system can be found in [Anand et al., 2017]. The control system is entirely programmable by MATLAB Simulink, allowing rapid algorithm development and implementation using code generation [Felici et al., 2014b]. The developed profile controller development and implementation environment within the TCV control system allows three crucial applications:

1. Closed-loop testing in simulation using a plant simulator, on a host computer.

2. Hardware-in-the-Loop simulation on the actual real-time hardware, after automated generation of C-code.

3. Experimental testing on tokamak.

Figure 4.1. Controller test environment where profile controller is interfaced with the RAPTOR simulator. Profile controller computes actuator inputs using the activation, reference and feedforward signals, the actuator limits and the plasma profiles and parameters provided by RAPTOR simulator. The RAPTOR simulator takes the actuator inputs to compute the plasma profiles for the next time step.
More details on these three applications will be given next after which we discuss the profile controller implementation and state reconstruction algorithms.

**Closed-loop control simulations using a plant simulator**

The profile controller can be further developed and tuned under experimental conditions in closed-loop simulations e.g. by using experimental geometry and density profiles of a specific shot. In addition most dynamics and delays of the closed-loop are included that may affect the controller performance: e.g. signal routing delays and reconstruction delays. The effect of controller gains can be easily analysed in these closed-loop simulations. In addition, the design of new shot references is eased by checking e.g. the expected proximity of actuator limits.

**Hardware-in-the-Loop simulations**

In this case, the simulator runs on a dedicated computational node of the control system, using the same interface to the true experimental signals. This Hardware-in-the-Loop test can be seen as the ultimate verification step before deploying the entire suite of reconstruction and control systems in experiments. It is particularly useful for detecting exceeding of computational time limits, or issues related to the integrated use of all control system components. The executable code is automatically generated using the Simulink Embedded Coder.

**Experimental testing**

The final stage is testing the profile controller in closed-loop experiments. A profile controller can be tested in experiments if it has been shown to work properly in closed-loop simulations inside the controller development environment, it passes code compilation tests and satisfies the computational time constraint in hardware-in-the-loop simulations.

The codes can also be run in open-loop in background during experiments performed for other purposes (the actuators do not respond to the profile controller, but to e.g. pre-programmed actuator requests). These are known as piggyback experiments and allow to test and improve the codes without the need for dedicated experiments. This has been especially useful in developing and interfacing the state reconstruction algorithms.

**Overview of implementation and involved algorithms**

An overview of the profile controller implementation is given in Figure 4.2.

The profile controllers can be tested in experiments in closed-loop on the TCV plant or in closed-loop simulations. The controllers receive the plasma state from a set of reconstruction algorithms that use either the diagnostic measurements
Chapter 4. Profile control simulations and experiments in TCV: a controller test environment and results using model predictive controller

Figure 4.2. Implementation of profile controller in TCV RT control system allowing to test the controller in experiments on the TCV plant and also in closed-loop simulations. Profile controller and density controller receive plasma state from a set of RT plasma state reconstruction algorithms (see Figure 4.3 for more details).

or the simulated diagnostics in combination with stored diagnostic signals from a previously performed shot. The set of RT state reconstruction algorithms is explained next. The interface to the profile controller itself is identical to Figure 4.1.

Density control in the experiments is performed using a recently developed robust density controller [Blanken et al., 2015], where more details will be provided in a separate publication. This controller computes the request to a single gas valve to achieve and maintain a specified volume averaged electron density value, which actual value is supplied by the RT state reconstruction algorithms explained next.

Plasma state reconstruction

Reconstruction of the plasma state from the scarce available measurements is essential for closed-loop control of plasma profiles and parameters. This plasma state involves here knowledge of the 2D plasma equilibrium, density profile and the electron temperature and poloidal magnetic flux profiles. The algo-
4.4 Controller development and implementation environment

Figure 4.3. Real-time state reconstruction algorithms and there interfaces for estimation of electron temperature and poloidal magnetic flux, density and equilibrium.

The following real-time reconstruction algorithms provide every 1ms a new plasma state estimate:

**RAPTOR-observer** The electron temperature and poloidal magnetic flux is reconstructed in an Extended Kalman Filter (EKF) scheme [Felici et al., 2014a]. This algorithm uses the density profile from RAPDENS-observer, the geometry and plasma current estimate from LIUQE, a core electron temperature measurement from soft X-ray (XTe) and the EC power requests from the profile controller to merge diagnostics with RAPTOR model predictions. The estimated central temperature $T_e(\rho = 0)$ will slowly track the XTe value, as the EKF estimates a disturbance heating source/sink that compensates for discrepancies between measurements and the undisturbed model predictions. Due to the lack of RT current density distribution measurements in TCV, the $\iota$-profile is fully determined by the RAPTOR model predictions, constrained only by the plasma geometry and experimental total plasma current measurement. The RAPTOR-observer gives many scalar and profile outputs, the most important of which are, for the present purposes:

- Electron temperature profile $T_e(\rho)$
- Poloidal magnetic flux profile $\psi(\rho)$
- Inverse safety factor profile $\iota(\rho)$
- Plasma pressure $\beta$

More details on the implementation of RAPTOR-observer in TCV can be found in [Felici et al., 2016].
RAPDENS-observer Using the same EKF approach as in RAPTOR-observer, RAPDENS-observer reconstructs the density profile from far-infrared (FIR) measurements in combination with model-based predictions from a density evolution model [Blanken et al., 2015]. Automatic detection and correction for diagnostic artefacts like fringe jumps in FIR channels is included.

LIUQE This algorithm reconstructs the 2D plasma equilibrium based on magnetic signals using the real-time implementation of the LIUQE-code [Moret et al., 2015]. It supplies equilibrium geometry information to the RAPTOR-observer, the RAPDENS-observer and the profile controller. At present the RAPTOR $q$-profile estimate is not fed back to LIUQE.

More details on recent advances of these state reconstructions algorithms will be provided in separate publications.

4.5 Model Predictive Controller design

We will now design a profile controller using Model Predictive Control (MPC) following the controller requirements defined in Section 4.3.2. First we discuss MPC after which we summarize the controller design. We conclude with the main controller settings and their effect on the controller performance.

4.5.1 Brief introduction to MPC

MPC is a well-established advanced control method that has been used for decades to control multiple-input-multiple-output (MIMO) processes in industry that are subjected to input and state constraints [Camacho and Bordons, 2004; Maciejowski, 2002; Mayne, 2014; Rossiter, 2013]. Its principle is visualized in Figure 4.1.

The compute the optimal actuator requests, an MPC controller relies on a prediction model of the involved process to calculate the future evolution of the controlled variables as a function of the present state and future actuator input sequence up to a horizon. This prediction model is used in an optimization problem to find the future actuator input sequence that minimizes a cost function. If the cost function is set-up to minimize the difference between controlled variables and references up to the horizon, the MPC controller will yield actuator commands to track these references. Only the first step of the computed actuator sequence is sent to the actuators and at the next step a new optimization problem is solved based on the estimated present state and possibly updated references. In this optimization problem constraints on actuator inputs and other variables can be taken into account, which may be time-varying due to changing conditions in actuators and plant.
Figure 4.1. Illustration of MPC principle. MPC uses a prediction model to predict the evolution of controlled variables up to a prediction horizon (a) based on the present state and future actuator inputs (b). The MPC controller computes future actuator input trajectories that will minimize the tracking error between the controlled variable and its reference, even in the presence of constraints on actuators or controlled variables.

4.5.2 Outline of controller design procedure

Here the controller design is summarized, which is largely based on previous work in [Maljaars et al., 2015a,c], with minor changes to simplify the implementation and meet the set (computational) requirements. We refer to Appendix 4.B for further details on the controller design.

Selected state and controlled variable representation

The controller requires a definition of the process state and of the controlled variables. We will employ a linearized model from RAPTOR in this chapter (will be explained next), therefore we choose to use the same state vector $x_k$ as in RAPTOR (see Section 4.2), which contains coefficients $\hat{\psi}$ and $\hat{T}_e$ to parameterize the $T_e$ and $\psi$ profiles ($n_b = 12$ coefficients for each profile). As controlled variables $z_k$ we choose both the $\iota \propto \frac{1}{\rho} \frac{\partial \psi}{\partial \rho}$ profile (at the locations of the provided 11 point equidistant $\rho$-grid where $W(\rho) \neq 0$, see Figure 4.3), and the the plasma pressure $\beta \propto \int_0^1 V' n_e T_e d\rho$. Both can be calculated as combinations of the state
Chapter 4. Profile control simulations and experiments in TCV: a controller test environment and results using model predictive controller

vector for a given plasma geometry and density profile. In vector form these read as:

\[
x_k = \begin{bmatrix}
\hat{T}_{e1}(t_k) \\
\vdots \\
\hat{T}_{e_{\text{ph}}}(t_k) \\
\hat{\psi}_1(t_k) \\
\vdots \\
\hat{\psi}_{\text{nb}}(t_k)
\end{bmatrix}
\quad \text{and} 
\quad z_k = \begin{bmatrix}
t_{\rho=0}(t_k) \\
\vdots \\
t_{\rho=0.1}(t_k) \\
\vdots \\
\beta(t_k)
\end{bmatrix}.
\]  

(4.8)

Prediction model

The prediction model used to predict the future controlled variable evolution is constructed from a single linearized model provided by RAPTOR (see Section 4.2). As part of the control design procedure, a simulation is run at plasma current and EC-power levels expected in the experiment, until stationary profiles are achieved and then the linearized model is selected at the linearization point \((u^0, x^0, z^0)\). Noting that as the current redistribution time in TCV is approximately 150ms for these plasmas, we choose as prediction horizon 100 time steps of 1ms to capture the main part of the slow evolution of the \(\iota\)-profile within the predictions.

The controller is expected to be able to compensate for the inevitable mismatches between the linearized model and the actual plasma transport dynamics. The plasma transport dynamics are expected to be only weakly nonlinear in these experiments due to starting control in the current flat-top phase, keeping (almost) the same actuator and plasma configuration, and the absence of confinement mode transitions. A comparison of the observed nonlinearity of the plasma transport dynamics in the conducted experiments (presented in Section 4.7) to the linearized controller model is given in Appendix 4.C.1.

Controller performance objective

The controller performance objective is to minimize the tracking error on the controlled variables and to avoid aggressive control actions, which can be expressed in the cost function \(J_k\) at time \(t_k\) as follows:

\[
J_k = \nu_\iota \left( (R_k - Z_k)^T Q_\iota (R_k - Z_k) + \nu_\beta \left( (R_k - Z_k)^T Q_\beta (R_k - Z_k) + w_{\Delta \tilde{U}} \Delta \tilde{U}_k^T R_{\Delta \tilde{U}} \Delta \tilde{U}_k \right) \right) \\
+ \nu_\beta \left( (R_k - Z_k)^T Q_\beta (R_k - Z_k) + w_{\Delta \tilde{U}} \Delta \tilde{U}_k^T R_{\Delta \tilde{U}} \Delta \tilde{U}_k \right)
\]

(4.9)

The first two terms penalize deviations between the future reference sequence \(R_k\) and future controlled variable sequence \(Z_k\) from time \(t_k\) to the horizon \(t_{k+N}\).
The matrices $Q_\iota$ and $Q_\beta$ select the appropriate controlled variables for $\beta$ or $\iota$ respectively and includes the weighting profile $W(\rho)$. The parameter $w_{\Delta U}$ can be used to set the importance of penalizing aggressively changing inputs in the future input sequence, where $\Delta U$ is the sequence of input changes between time steps. More freedom to set this penalty for individual actuators is available in $R_{\Delta U}$. The chosen cost function ensures that if the controller cost $J_k \rightarrow 0$, also the provided error norm $J_{\text{tot}} \rightarrow 0$ in (4.7).

**Optimization problem**

The optimal future actuator inputs are computed by minimizing the quadratic control performance objective (4.9) subjected to the actuator amplitude and ramp-rate limits (4.6). This can be formulated as a standard Quadratic Programming optimization problem that is solved at each time step using a fast open-source QP-solver (Quadprog++) that was embedded in Simulink. The actuator requests for the next time step are taken from the QP-problem solution and sent to the actuators.

It should be noted that although constraints on plasma parameters (state constraints) can be readily imposed in the optimization problem, only actuator input constraints are applied in this chapter. Given the available computational time of 0.7ms per time step, imposing also state constraints was expected to be infeasible at this timescale. See also the discussion on the used computational time in Appendix 4.C.3.

**State and disturbance estimation**

The controller requires the present plasma profile state $x_k$ as defined in (4.8). This state can be retrieved from the $T_e$ and $\psi$ profiles and their gradients that are provided by RAPTOR-observer.

In the absence of integral action in a profile controller, both static disturbances acting on the system and model mismatches between the controller model and the actual plasma profile evolution may result in steady state tracking errors. One way to overcome this in an MPC controller is to estimate in a disturbance observer the discrepancies between the controller-model-based expected states and controlled variables and those estimated from measurements. In this chapter we use such a disturbance observer to estimate the slowly varying component of these discrepancies. The MPC controller uses these disturbance estimates to improve the model-based prediction and ensure that targets are achieved with minimal steady state error. This effectively introduces integral action in the MPC controller [Maciejowski, 2002; Rossiter, 2013].
4.5.3 Main controller settings

The controller behavior can be tuned by changing a relatively small number of controller parameters with a clear impact on the control behaviour:

- The penalty on input changes $w_{\Delta \tilde{U}}$ given in (4.9). Increasing $w_{\Delta \tilde{U}}$ results in smoother actuator trajectories but a slower convergence of controlled variables to the references.

- The relative penalty on changes in $I_p$ vs. EC powers. This parameter $R_{\Delta \tilde{U}I_p/P_{EC}}$ is part of the matrix $R_{\Delta \tilde{U}}$ defined in (4.9). Increasing this value results in more aggressive use of EC rather than $I_p$.

- The time required to let the disturbance estimates converge towards the observed disturbances, defined in (4.23). This sets how fast the controller anticipates on changes in the disturbances, high values yield a slow response to changes in disturbances, which translates to a slow convergence towards the references in stationary conditions.

The chosen controller settings in the simulation and experiments are given in Appendix 4.B in Table 4.B.1.

4.6 Simulation results

Offline simulations have been performed using the MPC-controller interfaced with the RAPTOR-simulator as described in Section 4.4.1. These closed-loop simulations are used to tune the controller parameters and to test the controller’s robustness in various test cases. The parameters of the transport model and current drive efficiency in the RAPTOR-simulator are chosen representatively of the physics setup described in Section 4.3.

We present here a test case involving both control of the $\iota$-profile and plasma pressure $\beta$ to demonstrate the performance and main properties of the controller. We introduce a significant model mismatch between the physics parameters in the controller and the simulator. This model mismatch is a 30% higher current drive efficiency for both clusters in the simulator parameters, resulting in different required powers to reach the references than those powers used to generate the references. We use the controller settings in Table 4.B.1 and use both clusters A (counter-ECCD) and B (co-ECCD) as well as the plasma-current. Actuator amplitude limits are shown next and the ramp-rate limits for $I_p$ are set to $\| \frac{dI_p}{dt} \| \leq 2$MA/s.

Figure 4.1 presents the performance of the MPC controller in the closed-loop simulation, where three references for $\beta$ and $\iota$ are successfully tracked. The actuators (panel (a)) are all used to track the $\iota$-profile (c) and $\beta$ (d), which can also be noticed in the performance measure (b) that reaches small values despite the model mismatch. Actuator limits are given in dashed lines (a).
4.6 Simulation results

Figure 4.1. Performance of MPC profile controller in closed-loop simulation using plasma current and gyrotron clusters A (counter-ECCD) and B (co-ECCD) as actuators (a), where various $\iota$ (c) and $\beta$ (d) targets are successfully tracked. This good tracking is also visible in the performance measures (b).

As can be seen in (a) at $t=0.1$s, the controller uses the overshoot in the plasma current $I_p$ together with the maximum co-current (cluster B) and minimum counter-current (cluster A) to steer the $\iota$-profile as quickly as possible towards the first target, at the expense of some overshoot in $\beta$ (b). The same behavior can be noticed during transition to the third target and opposite behavior can be observed at the start of the second target. When the $\iota$-target is almost reached, both cluster requests are changed so as to minimize the error on $\beta$ and prevent an overshoot on $\iota$. The transition to the second target is slowed down by the active actuator limits on both clusters. A very small steady state error is achieved for all controlled variables (b) at all targets, thanks to the disturbance estimation that captures the steady state effect of the model mismatch in the disturbance estimates.

This simulation result shows the successful tracking of both $\iota$ and $\beta$ references and effective handling of the actuator limits so as to realize fast transitions between the targets. We conclude from this (and other) simulations that the controller is ready for testing in the TCV control system software environment.
4.7 Experimental results on TCV

In this section we present the results of using the controller in closed-loop on TCV in four discharges with different control objectives, references and actuators, using the implementation described in Section 4.4.2. The results are presented in the order of increasing complexity, starting with $\beta$-only control using only EC-clusters in feedback, followed by combined $\beta$ and $\iota$ control using both EC-clusters and finally adding $I_p$ as an actuator. The plasma scenario was discussed in Section 4.3 and the controller settings used are listed in Table 4.B.1.

4.7.1 Plasma pressure $\beta$ control using both EC clusters

In the first presented shot (#54385), the controller needs to track a $\beta$-reference with multiple steps, where the controller has the freedom to use both cluster A and B. The $\beta$ value at the moment of controller activation is taken as the initial reference value and a staircase reference is added to this value. In addition, a disturbance is purposely applied to the system by adding a ramp-up and ramp-down in the plasma current $I_p$ after 1.4s. In the controller settings a high penalty on aggressive input changes ($w_{\Delta U}$) was chosen, such that smooth actuator input trajectories are expected.

Figure 4.1 presents the performance of the controller in this $\beta$-control shot. We first briefly introduce the various traces shown in the figures here, since these will come back in all experimental results. The actuator inputs are given in (a)-(c), where the shaded area indicates the active feedback control phase for the actuators used by the profile controller. The density control is visualized in (d) where the density reference and estimate as well as the gas command are given. The $\beta$ and $\iota$ estimates and references (dashed, for those controlled) are given in (e) and (f) respectively, and the shaded area indicates the active control phase for the controlled variables. The performance measures (error norms) are given in the panel (g) for the controlled variables. Other information about the temperature (h) and density evolution (i), as well as the estimated NTM normalized likelihood for several mode numbers (j) is also provided. The normalized NTM likelihood is provided by the recent implementation of [Galperti et al., 2014]. If the NTM likelihood marker is close to 1, then this mode is strongly present in the magnetic signals.

The steps in the $\beta$-reference are successfully tracked (e) by adjusting the powers of the EC-clusters (b)-(c). As a high penalty on input changes was chosen, the actuator input traces are smooth but the controller cannot react quickly to changes in the density (i), resulting in some overshoot in $\beta$.

After about 1.2s, cluster A has spent too much time below 550kW and the lower limit is raised to 550kW, avoiding EC power supply shut-off. The power requested on cluster B drops to the minimum power and the reference cannot be reached.
**Figure 4.1.** Performance of $\beta$-control in shot #54385 using both cluster A and B with a prescribed perturbation from the nominal $I_p$ (a). Calculated actuator inputs for cluster A and B (b)-(c) to successfully track the $\beta$-reference with effective handling of actuator limit change in cluster A (b). During the applied $I_p$ disturbance (a) the density increases (d,i), resulting in a high $\beta$. The heating power cannot be reduced as both clusters are on their respective minimum powers (b) and (c). At the moment the plasma current $I_p$ drops below 100kA (a), a NTM is triggered (j) and the plasma disrupts just before 2s.

When the $I_p$ disturbance (a) starts to increase, the density increases even though the gas valve is closed (i). The EC-powers cannot be decreased (b)-(c) due to the minimum power limits, resulting in a significant error in $\beta$ ((e) and (g)). When the $I_p$ disturbance decreases again (a), the density decreases too (i) as well as $\beta$ (e), such that the controller quickly raises the EC requests (b)-(c). The plasma disrupts when the plasma current $I_p$ drops below 100kA (a), possibly due to the onset of a strong 2/1 NTM (j).

This experiment shows that the controller can successfully control the plasma pressure $\beta$ as far as it is reachable within the actuator limits. Also the change in the minimum power for cluster A is effectively taken into account. The small overshoot in $\beta$ could be reduced by choosing a lower penalty on actuator input changes, such that the controller can react rapidly to the changes in the density. This is done in the next presented experiment.
4.7.2 Combined $\beta$ and $\imath$-control using both EC-clusters

In the next two experiments both EC-clusters are used to control simultaneously $\beta$ and $\imath$, where we start in the first experiment with multiple steps in $\beta$ while keeping the $\imath$-profile constant. Subsequently an experiment is performed where the controller needs to track a sequence of four targets for both $\beta$ and $\imath$.

**Steps in $\beta$ reference at constant $\imath$**

In this experiment (#54423) the controller needs to track a $\beta$-reference with multiple steps whereas the $\imath$-target is kept constant.

![Figure 4.2](image)

**Figure 4.2.** Successful tracking of steps in $\beta$ (d) at constant $\imath$ (f) in shot #54423 using both EC-clusters (b)-(c).

The results for shot #54676 are presented in Figure 4.2, where the controller successfully uses both clusters A and B (b)-(c) to keep the $\imath$-profile constant (f), while at the same time quickly reaching the $\beta$-targets with hardly any overshoot (e). The clusters A and B move in the same direction during the $\beta$-steps to effectively not change the driven current. Immediately after controller activation, the EC-clusters make tiny opposite movements so as to compensate the ohmic current density relaxation while keeping the $\beta$ constant. Although the volume
averaged density is constant (d), the density profile distribution peaks slightly during the phase with the highest $\beta$-target.

We conclude for this experiment that the controller can tightly control the plasma pressure $\beta$ while at the same time the $\iota$-profile can be kept constant. Smooth and quick transitions are achieved without overshoot in $\beta$ or the $\iota$-profile, indicating that the controller effectively takes relevant couplings into account.

**Multiple targets for $\beta$ and $\iota$**

In the next shot #54402 the controller needs to track four different targets for both $\beta$ and $\iota$ where it can use both clusters.

![Figure 4.3](image)

*Figure 4.3.* Successful tracking of several $\beta$ (d) and $\iota$ (f) targets in shot #54402 using both EC-clusters (b)-(c). Four different targets are requested, where the first and last are not fully reachable with the used actuator settings and density profile. Fast transitions between targets are obtained by exploiting knowledge of actuator limits in the control input computation.

The controller performance is presented in Figure 4.3, where the second (between $t = 700\text{ms}$ and $t = 1100\text{ms}$) and third target (between $t = 1100\text{ms}$ and $t = 1500\text{ms}$) are achieved with a very small remaining error (e)-(g), while the other two targets have some remaining error for various reasons. The transition
from one target to another (e.g. from second to third at $t = 1100\text{ms}$) is very fast without overshoot, by pushing first both clusters against opposite actuator limits (b)-(c) and then adapting so as to prevent overshoot on $\iota$ (f) while keeping $\beta$ constant (e). A similar result was obtained in simulations, see Figure 4.1. This can only be achieved by accounting for the actuator limits over the prediction horizon in the control input computation. It demonstrates how a predictive controller can outperform more traditional feedback control methods in terms of constraint handling (see also Section 4.8).

During the short feedforward phase between 0.5 and 0.7s, a transient electron internal transport barrier (eITB) was created by the high counter current drive from cluster A (b), resulting in high temperature $T_e$ (h) and high $\beta$ (e). The high temperature is estimated from the XTe measurement, while the transport model parameters do not use such scenarios with local confinement improvement at reversed shear profiles (the related model parameters are set to zero). The eITB is lost at the start of the feedback control phase when the cluster A power is lowered to reach the $\beta$ target and MHD activity is detected for a short time (j). Post-shot analysis of soft X-ray data and electron temperature and density profile evolution revealed a minor disruption that leads to the loss of the eITB, which happens when internal MHD stability is lost [Turri et al., 2008]. It was shown in [Martynov, 2005] that ideal internal modes can trigger minor disruptions due to the high pressure gradient in a low magnetic shear region (infernal modes), even at low normalized plasma pressure.

The first target cannot be reached due to the tight actuator limits. Cluster B delivers less power, which is required to lower the $\iota$-profile in the center by reducing the co-current drive. At the same time the $\beta$-reference cannot be reached due to the low density (i) compared to the $n_{e,0} = 2 \cdot 10^{19} m^{-3}$ used to generate the references. The last $\beta$-target cannot be reached due to the presence of NTMs (j), while the last $\iota$-target cannot be reached as the last target was generated at a higher plasma current $I_p = 130kA$.

The reference and achieved $\iota$-profiles at 100ms after the target switch are given in Figure 4.4. Target 2 and target 3 are best achieved as discussed before. Note that the controller uses the weighting profile $W(\rho)$ such that e.g. for target 3 (at $t = 1600s$) the profile error is minimal at the peak of the weighting profile $W(\rho)$.

The real-time reconstructed $q$-profiles from the RAPTOR-observer at the time steps given in Figure 4.4 are compared in Figure 4.5 to real-time reconstructions from LIUQE [Moret et al., 2015]. Note the agreement from mid axis to plasma edge. The discrepancy in the center is due to the fact that while RAPTOR-observer includes the simulated effect of the EC current drive on the core $q$-profile, the real-time implementation of LIUQE employs a parameterization of the $q$-profile with only a few degrees of freedom to fit the (magnetics-only) measurements.

The plasma pressure evolution as reconstructed in the RAPTOR-observer
4.7 Experimental results on TCV

Figure 4.4. Reference and achieved $\iota$-profiles for the 4 targets in shot #54402 at 100ms after a target switch. Targets are best achieved close to the peak of the weighting profile $W(\rho)$.

Figure 4.5. Comparison of reconstructed $q$-profiles at four time steps from the RAPTOR-observer to real-time reconstructions of LIUQE.
is also compared to offline reconstructions from LIUQE in Figure 4.6. Note the agreement between both reconstructions, indicating that the actual plasma pressure $\beta$ behaves as reconstructed in the RAPTOR-observer.

This experimental result shows the controller’s capability to simultaneously control $\beta$ and the $\iota$-profile, ensuring fast transitions between targets without overshoot. The controller takes advantage of knowing the actuator limits in the control input calculation to realize these fast transitions, such that targets are closely achieved within 100ms, even faster than the current redistribution time scale of about 150ms.

4.7.3 Combined $\beta$ and $\iota$ control including plasma current

In the last presented experiment #54414 we repeated the previous experiment #54402 with the four targets for $\beta$ and $\iota$ while adding the plasma current $I_p$ as actuator. This should provide more control freedom, especially enabling to reach the last target that was generated with $I_p = 130\text{kA}$. In this shot the plasma current ramp-rate was restricted to $-0.5\text{MA/s} \leq \frac{dI_p}{dt} \leq 0.5\text{MA/s}$, changes in $I_p$ where significantly more penalized than changes in cluster powers, and a slow convergence of the disturbance estimates to measured disturbances was set (see Table 4.B.1).

This last experimental result is presented in Figure 4.7, where in general the performance of the controller seems worse than without including $I_p$. The control-loop involving the plasma current $I_p$ is clearly oscillating, while being bounded by the ramp-rate limits (a).

Changes in $I_p$ immediately result in changes in the density (i) due to a changing particle confinement, for which the density controller tries to compensate (d). The fluctuations in $I_p$ lead to plasma volume variations and in alternating be-
4.7 Experimental results on TCV

Figure 4.7. Performance of controller in shot #54414 where the same four targets for $\beta$ and $\iota$ as in #54402 need to be tracked, but with the plasma current $I_p$ as third actuator. Although the $\iota$ tracking improves for the first and last target (f)-(g) by lowering or increasing $I_p$ (a), the control-loop involving $I_p$ is clearly unstable (but bounded by the ramp-rate limits). Changes in $I_p$ correspond to changes in density (d,i) and lead to the presence of NTM (j), resulting in a bad $\beta$-tracking (e).

Remarkably, comparing the error norm for $\iota$ (g) to the one in Figure 4.3(g) shows improved $\iota$-tracking in this experiment for the first and last target. To reach the first target, $I_p$ is lowered to its minimum so as to make the $\iota$-profile less peaked. The plasma current $I_p$ is correctly raised in the last target, oscillating around the 135kA, close to the value used to generate the reference. Contrarily, the error in $\beta$ is clearly larger, mainly due to the induced density oscillations and the inability of the controller to compensate quickly for these using the clusters as the disturbance estimates are converging too slow to the measured disturbances.

The main expected origins of the closed-loop instability in the loop involving $I_p$ are listed here in order of their likely contribution:
• Unmodelled closed-loop delays and actuator dynamics for $I_p$ resulting in a large delay of 7-15 ms before the $I_p$-values requested by controller can be seen in the measured $I_p$ fed to the RAPTOR-observer (see Appendix 4.C.2). This delay exceeds by far the energy confinement time (5ms).

• Unmodelled effects of changing the plasma current $I_p$ on the density profile, whereas it is known that particle confinement and recycling changes with the plasma current.

• No active plasma shape feedback control is used. The plasma volume changes significantly in shot #54414 and also changes frequently from limiter to divertor shape.

We expect that correctly modeling the delays and actuator dynamics will already remove the closed-loop instability. The last two expected origins of the closed-loop instability can be solved by adding an equation for the density in RAPTOR and including shape control in the experiments.

This experiment shows the potential and limitations of using the plasma current $I_p$ in feedback in the present software configuration. It is shown that using the plasma current as feedback actuator allows to achieve a broader range of $q$-profiles. Applying the described remedies to remove the closed-loop instability should enable the controller to efficiently use the additional control freedom provided by $I_p$.

4.8 Qualitative comparison of MPC controller performance to other control methods.

It would be insightful to compare the performance of the MPC controller, as demonstrated in the simulations and experiments in Section 4.6 and 4.7, to the performance of profile controllers using different control methods as presented in literature. An analysis of various control methods in terms of overshoot, convergence rate of the controlled variables to the references and actuator limit handling depends significantly on the plasma conditions and tokamak involved. Results of testing other profile controllers in simulations and experiments in a comparable plasma scenario in TCV, as reported here for the MPC controller, are only available for two other profile controllers. We will restrict ourselves in this qualitative performance comparison to these two control methods.

4.8.1 Comparison to robust control

The first control method that is also used in simulations and experiments for TCV is robust control. A robust controller is typically designed using a description of the expected model uncertainty and disturbances such that the controller
4.8 Qualitative comparison of MPC controller performance to other control methods.

can ensure a specified performance despite these uncertainties. Simulations results indicating the performance of a robust controller applied in a similar plasma scenario on TCV are presented in [Barton et al., 2014b] and [Barton et al., 2015b]. Robust controllers employing the same controller model and controller design have also been tested successfully in ITER simulations [Barton et al., 2015a] and on the DIII-D tokamak in experiments [Barton et al., 2012, 2014a].

In [Barton et al., 2014b] control of only the $q$-profile is considered, where three different controllers are designed that put emphasis on different regions of the $q$-profile. The $q$-profile targets and controller gains have been chosen in these simulations such that actuator limits are not restrictive (see Figure 6 in [Barton et al., 2014b]), therefore the constraint handling of this robust controller cannot be evaluated. If a higher bandwidth would have been chosen, more aggressively changing inputs would result in actuator requests that hit these actuator limits.

In [Barton et al., 2015b], a robust controller is designed to control both the electron temperature and safety factor profile using four EC actuators. Figure 7 in [Barton et al., 2015b] presents the performance of the controller for tracking a constant $q$-profile but time-varying $T_e$ profile target. All targets are achieved in the end ((a)-(f)), thanks to integral action included in the controller. The controller is tuned such that actuator limits are not restrictive in these results ((g)-(i)). A convergence time for $T_e$ of about 100ms can be noticed, while the typical timescale of $T_e$ (energy confinement time) is about 5-10ms in that scenario and hence a faster convergence rate should be possible. This simulation is comparable to our experiment #54423, where the $q$-profile is kept constant while fast transitions in the plasma pressure (i.e. different temperatures) are realized without overshoot.

A potential limitation of robust control is that actuator limits are not taken into account in the control input itself, but are imposed using anti-windup schemes after the actuator input has been calculated. As such, the controller will not compensate for e.g. saturation (or sudden unavailability) of actuator 1 by changing the request for actuator 2 at the same time instant, but only after some time the request for actuator 2 may convergence to the appropriate value to compensate for the missing power in actuator 1. A typical example of this can be found in profile control simulations with ITER parameters using robust control in [Barton et al., 2015a] in Figure 14 and Figure 15. There, at about 180s, a positive step in thermal stored energy is requested (Figure 14(m)) after which the Ion Cyclotron power request quickly saturates at full power (Figure 15(f)) and the Neutral Beam Injection power only afterwards slowly converges to the power level required to achieve the thermal stored energy target (Figure 15(d)).

An MPC controller, taking the actuator limits into account in the actuator input calculation, is expected to typically use both actuators immediately after the reference step change, so as to more quickly realize the transition between the targets for thermal stored energy.

The inherent robustness against model uncertainties and disturbances, that
is a major advantage of a robust controller, can also be achieved in MPC by using a specific type of MPC, called Robust MPC. An introduction to Robust MPC and discussion of recent advances can be found in [Mayne, 2014].

4.8.2 Comparison to IDA-PBC

The second type of controller that is used in simulations and experiments for TCV, is IDA-PBC (interconnection and damping assignment - passivity based control). This control method uses a Port-Hamiltonian formulation of the system to design a controller that provides the desired closed-loop performance description in terms of stability and convergence rate. Simulation results as well as preliminary experimental results are reported in [Vu et al., 2016, 2017].

In [Vu et al., 2016], controllers using two different controller models have been tested in simulations with the RAPTOR code. One controller uses only a poloidal magnetic flux diffusion model, the other a coupled model of both poloidal magnetic flux and electron temperature evolution. When integral action is included in the controller, the targets are achieved with minimal steady state error (see Figure 9 and 13 in [Vu et al., 2016]). The convergence time to achieve the targets is about 800ms, while the relevant current diffusion timescale is about 100-150ms. It is also shown that the controller compensates for artificial perturbations in model parameters. In this paper, targets and controller gains are chosen such that actuator limits are not active and hence actuator limit handling cannot be analyzed. Although it is discussed that actuator limits are included in the calculation of feedforward actuator requests, it is not clear from this paper how these are imposed on the sum of feedforward and feedback actuator requests. This paper includes also a preliminary result of testing the controller in closed-loop on the TCV tokamak, however, due to actuator failure in that experiment, the controller performance cannot be analyzed.

Recently, in [Vu et al., 2017] new results of the IDA-PBC controller in simulations and experiments on the TCV tokamak are reported, using the same simulator settings and experimental setup as used in this work. In Figure 5.1 (top), good tracking of the $q$-profile reference is demonstrated. This result benefits from a careful chosen feedforward heating power and the references are chosen such that only feedforward inputs provide already zero tracking error. In Figure 5.1 (bottom) tracking of both the $q$-profile reference and plasma pressure reference is shown. In Figure 5.1 at $t = 0.6s$, both EC actuators (one driving co-current, one counter-current) saturate at full power, which results in an overshoot in the plasma pressure, while a lower co-current would speed up the convergence of the $q$-profile as well as reduce the overshoot in plasma pressure (as is demonstrated in this chapter in Figure 4.1). Preliminary results from experimental testing the IDA-PBC controller in TCV experiments show successful tracking of a constant $q$-profile reference at a single radial location using the difference between both EC actuator powers (driving only co-current, no heating)
as a single actuator input, while keeping the effective heating power constant.
Although the single target is tracked successfully, a drift of the $q$-profile at other
radial locations can be noticed, which could probably be reduced by using both
actuators in the controller. As no transitions between targets are requested
in this experiment, the controller performance cannot be analyzed in terms of
convergence rate, overshoot etc.

4.8.3 Summary

The considered controllers show a slower convergence time for the controlled
variables towards the reference than we found with the MPC-controller and
than can be expected on the basis of the system time scales. Both convergence
rate and overshoot strongly depend on controller tuning, such that it can not be
excluded that the other controllers could give an improved performance, but this
has not been shown. The considered controllers have often been tuned to avoid
that actuator limits are restrictive in simulations and experiments, although
actuator limits are often restrictive in high performance tokamak operation.
One example revealed the drawback of not using the actuator limits in the
actuator input calculation itself. Until proven otherwise, the MPC-controller has
shown the best performance of the examined controllers in terms of overshoot,
convergence rate of the controlled variables to the references and actuator limit
handling.

A more complete comparison (preferable quantitative) of the available profile
control methods would be useful to direct further research for establishing profile
control for reliable high performance operation of future tokamaks like ITER in
advanced scenarios.

4.9 Conclusions and outlook

This chapter has demonstrated the successful performance of a model predictive
profile controller in experiments in the TCV tokamak by employing a profile
controller test environment.

We designed a linear MPC controller including disturbance estimation and
demonstrated its performance in both simulations and experiments in a TCV
L-mode plasma scenario. The results show successful tracking of the inverse
safety factor profile as well as the plasma pressure using two clusters of gyrotrons/launchers in the presence of uncertain plasma conditions and distur-
bances. The controller exploits the knowledge of the time-varying actuator lim-
its in the actuator input calculation such that fast transitions between targets
are achieved without overshoot, demonstrating how a predictive controller can
outperform other control methods in terms of effective input-constraint handling.

Secondly, a profile controller development and implementation environment
is presented that is used to prepare and test profile controllers both in simula-
Chapter 4. Profile control simulations and experiments in TCV: a controller test environment and results using model predictive controller

...tions and experiments on TCV. It allowed us to first prepare profile controllers interfaced to a simulator on a local computer. Next the controllers were tested more comprehensively inside the TCV real-time control system including plasma state reconstruction codes and experimental data. The employed software was used to automatically generate code for the control system including profile controllers to test the real-time performance in hardware-in-the-loop simulations and finally in experiments.

The work in this chapter can be extended in several ways to further utilize the advantages of model predictive profile controllers. The controller performance can be further improved by accurately modeling the actuator dynamics and delays. The controller would be able to use the plasma current more successfully as actuator if the plasma current actuator dynamics and delays as well as its coupling with the density profile dynamics would be included in the linearized controller model and shape feedback control would be employed in experiments. Finally, adding internal current profile diagnostics as input for the RAPTOR-observer would yield more accurate $q$-profile estimates, ensuring that the tracked $q$-profile is indeed the true $q$-profile of the plasma rather than a purely model-based estimate as is presently the case. We noticed that the reconstructed $q$-profiles are sensitive to the transport model parameters. We expect that using improved transport models [Citrin et al., 2015; Kim et al., 2016] will alleviate this effect.

Exploiting model predictive profile control throughout multiple plasma regimes (e.g. H-mode, internal transport barriers) would be eased if the controller was independent of a predefined linear model. This can be realised by using real-time linearizations provided by the RAPTOR-observer. Fully nonlinear MPC, involving a nonlinear prediction model, is not computationally feasible on currently operational devices, but can be considered for e.g. ITER with its slower characteristic time scales and hence increased available computational time.

Demonstrating handling of plasma parameter limits (e.g. normalized plasma pressure) in experiments on currently operational tokamaks is important for developing the control expertise to ensure reliable high-performance operation close to these limits in (future) large tokamaks. In this chapter we imposed only input constraints, whereas MPC can also handle state constraints (as shown in [Maljaars et al., 2015a,c]). Sufficient computational time is available to add such a limit to the controller in future profile control experiments at TCV.

In (future) plasma control systems, a profile controller will be interfaced with a supervisory controller that may set controller activation, references and parameters as well as available actuators in real-time based on the plasma state and detected events [Humphreys et al., 2015]. The MPC controller’s ability to deal with time-varying references, activations and actuator limits encourages to use it in the further development of integrated control strategies involving the management of actuators shared between several control tasks.
4.A Available information to controllers

Acknowledgements

We acknowledge Hugo van den Brand (DIFFER) for the real-time implementation of the Quadratic Programming solver. The work in this chapter has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. The work in this chapter is also supported by the Netherlands Organisation for Scientific Research (NWO) via the Innovational Research Incentives Scheme.

Appendixes

4.A Available information to controllers

The controller is supplied in real-time with information that it can use to compute the new actuator inputs. The origin of the signals is shown in Figure 4.1 (outside TCV control system) and Figure 4.2 (inside TCV control system). The information supplied to the controllers is given in Table 4.A.1.

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>External pre-defined signals:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>References</td>
<td>$\beta_{\text{ref}}, \iota_{\text{ref}}$</td>
<td></td>
</tr>
<tr>
<td>Control activation signals</td>
<td>$\beta_{\text{act}}, \iota_{\text{act}}$</td>
<td>Boolean. If true, $\beta$ and/or $\iota$ control is activated</td>
</tr>
<tr>
<td>Feedforward actuator inputs</td>
<td>$u_{\text{ff}}$</td>
<td></td>
</tr>
</tbody>
</table>

| Provided by real-time state reconstruction or simulator: | | |
| Plasma profiles on $p$-grid | $\psi, T_e, T_i, n_e, z_{\text{eff}}, \frac{\partial \Omega}{\partial \rho}, \frac{\partial^2 \Omega}{\partial \rho^2}, t, U_{\|}, \chi_{\|}, j_{\|,\text{bs}}, j_{\|}$ | |
| Plasma parameters | $\beta, \beta_{\text{N}}(\|)$ | |
| Geometry profiles | $\frac{\partial \Lambda}{\partial \rho}, \frac{\partial^2 \Lambda}{\partial \rho^2}, F, g_1, g_2, g_3, \frac{\partial q}{\partial \rho}$ | |
| Geometry parameters | $\Phi_{\|}$ | |
| ECH & ECCD dep. profiles | $p_{\text{ec}}, j_{\text{ec}}, \frac{\partial p_{\text{ec}}}{\partial \rho}, \frac{\partial^2 p_{\text{ec}}}{\partial \rho^2}, \frac{\partial q}{\partial \rho}, \frac{\partial q}{\partial \rho}$ | |
| State validity flag | | Boolean. If true, provided profiles and parameters are valid |

| Real-time computed actuator limits: | | |
| Minimum amplitude | $u_{\text{min},k}$ | |
| Maximum amplitude | $u_{\text{max},k}$ | |
| Minimum ramp-rate | $\Delta u_{\text{min},k}$ | |
| Maximum ramp-rate | $\Delta u_{\text{max},k}$ | |

| Other: | | |
| Previous actuator inputs | $u_{k-1}$ | E.g. for anti-windup controller |
4.B Details MPC controller design

Here we provide more details on the design of the MPC controller as summarized in Section 4.5.2.

Obtaining linearized model

The single linearized model is obtained by linearizing the nonlinear state and output equations (4.3)-(4.4) in RAPTOR at the linearization point \((u^0, x^0, z^0)\). The linearization point satisfies equations (4.3)-(4.4). We define variations of the inputs, states and outputs as \(\tilde{u}_k = u_k - u^0, \tilde{x}_k = x_k - x^0\) and \(\tilde{z}_k = z_k - z^0\). The obtained linear time-invariant state space model reads as:

\[
\begin{align*}
\tilde{x}_{k+1} &= A\tilde{x}_k + B\tilde{u}_k + \tilde{d}_x^k, \quad (4.10) \\
\tilde{z}_k &= C\tilde{x}_k + \tilde{d}_z^k, \quad (4.11)
\end{align*}
\]

where \(\tilde{d}_x^k\) and \(\tilde{d}_z^k\) represent additional state and output disturbances.

Constructing prediction model with simplifications

A prediction model is constructed to predict the controlled variables on the prediction horizon of \(N\) time instants. We define the column stacked vectors \(\tilde{U}_k^T = [\tilde{u}_k^T \tilde{u}_{k+1}^T \cdots \tilde{u}_{k+N-1}^T]^T\) and \(\tilde{Z}_k^T = [\tilde{z}_{k+1}^T \tilde{z}_{k+2}^T \cdots \tilde{z}_{k+N}^T]\) and write \(\tilde{U}_k = U^0 + \tilde{U}_k\) and \(\tilde{Z}_k = Z^0 + \tilde{Z}_k\), where \(U^0\) and \(Z^0\) are column stacks of the inputs \(u^0\) and controlled variables \(z^0\) at the linearization point. By using the single linear-time-invariant (LTI) state space model and adding the to-be-estimated state disturbances that are assumed constant over the prediction horizon, we can write the future controlled variable deviations \(\tilde{Z}_k\) as:

\[
\tilde{Z}_k = \Gamma_C\tilde{x}_k + \Gamma_D\tilde{U}_k + \Gamma_E\tilde{d}_k, \quad (4.12)
\]

The matrices \(\Gamma_C\), \(\Gamma_D\) and \(\Gamma_E\) can be constructed using the LTI state space matrices \(A\), \(B\) and \(C\) (analogously to [Maljaars et al., 2015c]).

In order to construct a compact prediction model and optimization problem, we make the following simplifications:

- The future inputs \(\tilde{U}_k\) are parameterized by a number of input parameters \(\tilde{p}_k\) (see Figure 4.B.1(a)) using a fixed input parameterization matrix \(P_{\text{map}}\) (see [Maljaars et al., 2015a] for a detailed definition):

\[
\tilde{U}_k = P_{\text{map}}\tilde{p}_k, \quad (4.13)
\]

where the input parameterization map \(P_{\text{map}}\) includes an input scaling such that the input parameters \(\tilde{p}_k\) are of order 1.
4.B Details MPC controller design

Figure 4.B.1. Illustration of the chosen input parameterization and coincidence points. Input parameters $\tilde{p}_k$ (nodes) for $P_A$ and corresponding inputs $\tilde{U}_k$ (a), yielding more control freedom in the beginning and constant future inputs in the tail. Corresponding predicted evolution of output $\beta$ (b) with coincidence points.

- As the controlled variables are required to track their reference signals at a subset of time instants within the receding horizon, the future controlled variables are predicted only at these coincidence time instants (see Figure 4.B.1(b)).

Rewriting cost function

The cost function (4.9) can be rewritten into a compact form (for details see [Maljaars et al., 2015a]). Defining $\tilde{R}_k = R_k - Z_k^0$ and using the prediction model in (4.12), input parameterization in (4.13) and preserving only terms dependent on $\tilde{p}_k$, the compact cost function can be written as:

$$J_{\text{red},k} = \frac{1}{2} \tilde{p}_k^T H_k \tilde{p}_k + f_k^T \tilde{p}_k,$$

(4.14)

where $H_k$ is the quadratic cost matrix (Hessian) and $f_k = [\tilde{R}_k^T \tilde{x}_k^T \tilde{d}_k^T] F_k$ the gradient vector of this quadratic cost function.

Rewriting input constraints

The input magnitude and ramp-rate constraints defined in (4.6) are rewritten as follows:

$$U_{\text{min},k} \leq U_k \leq U_{\text{max},k},$$

(4.15)

$$\Delta U_{\text{min},k} \leq \Gamma \Delta U_k \leq \Delta U_{\text{max},k},$$

(4.16)
where $\Gamma_\Delta$ is a difference matrix operator such that $\Delta U_k = [(u_{k+1} - u_k)^\top \cdots (u_{k+N-1} - u_{k+N-2})^\top]^\top / T_s = \Gamma_\Delta U_k$. These constraints can be written as a linear inequality constraint on $\tilde{p}_k$ using the input parameterization in (4.13) (for more details see [Maljaars et al., 2015a]):

$$A_{\text{ineq}} \tilde{p}_k \leq b_{\text{ineq}, k}$$

(4.17)

As in this chapter $U^0$ is constant over the prediction horizon for each actuator input, it is sufficient to impose amplitude constraints only at the parameterization knots and ramp-rate constraints at one point in between these knots. In addition, ramp-rate constraints are here not important for EC-powers and therefore only imposed on $I_p$.

**Formulation and conditioning of QP-problem**

The quadratic programming (QP) problem is composed of the cost function (4.14) and input constraints (4.17). The final standard QP-problem reads as:

$$\text{minimize} \quad \frac{1}{2} \tilde{p}_k^\top (\alpha H_k + \epsilon I) \tilde{p}_k + (\alpha f_k^\top) \tilde{p}_k,$$

subject to

$$\epsilon A_{\text{ineq}} \tilde{p}_k \leq \epsilon b_{\text{ineq}, k},$$

$$\epsilon A_{\text{eq}} \tilde{p}_k = \epsilon b_{\text{eq}, k},$$

(4.18)

where the equality constraints are used to fix the currently active input on the system and $f_k = [\tilde{R}_k^\top \tilde{x}_k^\top \tilde{d}_k^\top] F_k$.

The QP-problems have been conditioned to avoid numerical issues in the solver. Although input scaling is used to ensure $\tilde{p}_k$ is order 1, the matrices $H_k$, $A_{\text{ineq}}$ and $A_{\text{eq}}$ contain elements in the order of the inputs ($10^6$). This may cause numerical problems in the matrix calculations inside the solver if no scaling is applied. Choosing the scaling factor $\alpha$ such that the largest eigenvalue of $\alpha H_k$ equals 1, and choosing the scalar $\epsilon = 10^{-6}$ results in a positive definite Hessian matrix $\alpha H_k$ with condition number $\approx 10^6$, regardless of the chosen controller parameters. The (in)equality constraints are also scaled at both sides with $\epsilon$.

**Solving QP-problem**

The QP-problem at each time step is solved using the open-source library Quadprog++ [Di Gaspero, 2016], which is an efficient C++ implementation of the Goldfarb-Idnani dual active-set method [Goldfarb and Idnani, 1983]. An active-set solver iteratively adds or removes constraints from a so-called working-set of constraints until the working-set includes all constraints that are active at the optimal solution. The solver code is interfaced via a S-function wrapper, enabling it to be included in Simulink for simulations and compiling it correctly with all other codes for experiments. All QP-problems have been solved correctly during experiments and the available computational time was never exceeded (see Appendix 4.C).
Post-processing QP-solution

Once the QP-solution yielding the future input parameters has been obtained, the actuator input for the next step can be computed using (4.13). Feedforward inputs are assigned for those actuators that are not in feedback.

Disturbance estimation details

The MPC controller requires disturbance estimates to improve the model-based prediction and ensure that targets are achieved without steady state error. Disturbances are estimated in a disturbance observer where predicted disturbances are corrected by measured disturbances. The disturbance model assumes that the disturbances $\tilde{d}_k$ are constant over time:

$$\tilde{d}_{k+1} = \tilde{d}_k.$$ (4.19)

The state disturbance ‘measurement’ $\tilde{d}^\text{meas}_k$ is given by the discrepancy between the provided estimated state and the predicted state based on the model (4.10) without disturbance:

$$\tilde{d}^\text{meas}_k = (x_k - x^o) - A(x^\text{meas}_{k-1} - x^o) - B\tilde{u}_{k-1} - N_{\text{obs} - \text{delay}},$$ (4.20)

$$\tilde{d}^\text{meas}_k = (z_k - z^o) - C(x^\text{meas}_{k-1} - x^o),$$ (4.21)

where $x^\text{meas}_{k-1}$ is the retrieved state from the RAPTOR-observer profiles. The variable $N_{\text{obs} - \text{delay}}$ is the additional number of delay time steps between when the actuator input is calculated and the response is seen in the RAPTOR-observer output.

The disturbance observer is finally modeled as:

$$\tilde{d}_{k+1} = \tilde{d}_k + M(\tilde{d}^\text{meas}_k - \tilde{d}_k),$$ (4.22)

where $\tilde{d}^\text{meas}_k = [\tilde{d}^\text{meas}_k, \tilde{d}^\text{meas}_k, \tilde{d}^\text{meas}_k]^\top$ and the observer gain $M$ is computed as

$$M = 3T_s \begin{bmatrix} \frac{I}{\tau_x} & 0 \\ 0 & \frac{I}{\tau_z} \end{bmatrix}$$ (4.23)

yielding a simple discrete time disturbance observer with a low-pass filter characteristic. The parameters $\tau_x$ and $\tau_z$ set approximately the 95% convergence time of the disturbance estimates in response to a step change in disturbances on states and controlled variables respectively.

It is also possible to compensate for the time delay between when the actuator input is calculated and its response is seen in the provided signals by RAPTOR-observer. This is done by predicting the state $\tilde{x}_k$ by $N_{\text{obs} - \text{delay}}$ time steps ahead using the first updated disturbance estimates.
Table 4.B.1. Controller settings in simulation and experiments. Listed in order as presented.

<table>
<thead>
<tr>
<th>Sim/Exp</th>
<th>$w_{\Delta \tilde{U}}$</th>
<th>$R_{\Delta \tilde{U}I_p/P_{EC}}$</th>
<th>$\tau_x = \tau_z$</th>
<th>Observer delay compensation</th>
<th>$N_{\text{obs-delay}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sim 1</td>
<td>5</td>
<td>10</td>
<td>10ms</td>
<td>true</td>
<td>2</td>
</tr>
<tr>
<td>54385</td>
<td>40</td>
<td>20</td>
<td>10ms</td>
<td>true</td>
<td>2</td>
</tr>
<tr>
<td>54423</td>
<td>30</td>
<td>20</td>
<td>10ms</td>
<td>false</td>
<td>2</td>
</tr>
<tr>
<td>54402</td>
<td>20</td>
<td>20</td>
<td>10ms</td>
<td>true</td>
<td>2</td>
</tr>
<tr>
<td>54414</td>
<td>20</td>
<td>80</td>
<td>60ms</td>
<td>false</td>
<td>2</td>
</tr>
</tbody>
</table>

Handling invalid provided plasma state information

It may occur that the RAPTOR-observer cannot provide the controller with valid plasma state information in which case the controller should go to a fail-safe strategy. The validity of the reconstructed state is available to the controller in the state validity flag (see Table 4.A.1). We choose that in case of invalid states, the controller keeps the actuator request $u_k$ at the previous value $u_{k-1}$ until a valid state is received in a subsequent cycle. In addition the state estimate $\tilde{x}_k$ and disturbance estimates $\tilde{d}_k$ are not updated with this invalid information. However, this did not occur during the feedback control phase in the experiments reported in this chapter.

Chosen controller settings

An overview of the chosen controller settings in the simulation and the experiments are given in Table 4.B.1.

4.C Analysis aspects experimental results

Here several aspects of the experimental results are analyzed in more detail.

4.C.1 Analysis nonlinearity of plasma transport dynamics

We are interested in the extent of the nonlinearity of the plasma transport dynamics as observed in these L-mode discharges. This is analyzed by considering the nonlinear terms in the PDEs (4.1) and (4.2) as evolved in the RAPTOR-observer. In addition, we would like to analyze how these nonlinear terms compare to those at the linearization point of the controller model.

Figure 4.C.1 shows the nonlinear terms in the PDEs (4.1) and (4.2) at the controller model’s linearization point $(z^0, x^0, u^0)$, together with the range of values as have been observed in the following experiments:

- #54402, #54423 and #54767: full control window.
• #54385: start of control window till 1.85s (before disruption).
The shot #54414 is excluded as the loop involving $I_p$ was unstable.

![Graphs](image)

**Figure 4.C.1.** Nonlinear terms in PDEs (4.1) and (4.2) at the linearization point ($z^0, x^0, u^0$) (red). In addition the range of observed values in experiments is given (shaded grey area). Note that the linearized model is for most terms close to the mean of the range.

The plasma transport in these L-mode discharges is nonlinear as expected and can be seen in the wide range of observed values in these experiments. Note that the chosen linearization point is for many terms the average of the observed values in experiments. The given terms are now discussed individually:

- The electron temperature $T_e$ (a) in Figure 4.C.1 (a) is not a nonlinear term in the PDEs, but shown as background information as it influences $\sigma_\parallel$, $j_{bs}$ and $j_{ec}/P_{\text{cluster}}$.

- The parallel conductivity $\sigma_\parallel$ (a) is proportional to $T_e^{3/2}$ and changes the resistive diffusion time scale of the poloidal magnetic flux and hence the $\nu$-profile.

- The bootstrap current $j_{bs}$ changes significantly with the temperature and density profile, but has only a small contribution in these L-mode discharges.
Chapter 4. Profile control simulations and experiments in TCV: a controller test environment and results using model predictive controller

- The density profile $n_e(d)$ covers a wide range and is especially important for $j_{\text{ec}}/P_{\text{cluster}}$ and the plasma pressure $\beta$.
- The $j_{\text{ec}}/P_{\text{cluster}}$ profile (e) changes significantly, due to the scaling with temperature and density.
- The geometry profiles (f)-(j) do change, but not dramatically, although no shape control was used in these shots.

The controller worked properly in these experiments using its single linearized model, despite the observed nonlinearity. Significant changes in actuator deposition locations or confinement mode transitions would be more difficult for a controller using a single linearized model.

4.C.2 Analysis plasma current actuator dynamics and delays

Here we briefly investigate the plasma current actuator dynamics and delays between the requested plasma current and the plasma current as reconstructed from measurements, where the latter is used in RAPTOR-observer.

The actuator delays and dynamics of the plasma current $I_p$ are analyzed by using data from shot #54414, during the long current ramp. The controller-request as well as the input to RAPTOR-observer are given in Figure 4.C.2.

![Controller request and input RAPTOR-Observer](image)

**Figure 4.C.2.** Visualization of delay between the requested current and the current reconstructed from measurements, where the reconstructed current is used as an input to the RAPTOR-observer.

Observe that the RAPTOR-observer input is clearly 7-15 ms behind the controller request. As the RAPTOR-observer uses the reconstructed $I_p$ from LIUQE as input, the delay originates from the full closed-loop. The delay includes signal routing delays, coil power supply delays, filtering of magnetic signals and
reconstruction delays. In addition, the bandwidth of the low-level $I_p$-controller is relatively low, resulting in a relatively slow convergence of the observed $I_p$ to the requested $I_p$. Further analysis is required to make a detailed delay budget and accurately model the actuator dynamics. In contrast to here, in experiments, only 1ms delay was present between the controller request and the RAPTOR simulator input in the simulation involving $I_p$ as shown in Section 4.6.

These observed plasma current actuator dynamics and delay can dramatically alter the stability and performance of the controller as we encountered in experiment #54414. The 7-15ms may be reduced by changing signal routing or increasing the bandwidth of the $I_p$ controller. But far more important is to include the actual present dynamics and delays in the controller model and settings, to avoid stability loss and increase the controller performance.

### 4.C.3 Analysis of the computational time of the MPC controller

In this subsection, the computational time used by the MPC controller per cycle in the experiments is analyzed to investigate if it satisfies the computational time constraints (<0.7ms).

The computational time per cycle is shown in Figure 4.C.3(a) for the shots #54402 (both EC clusters in feedback) and #54414 (also $I_p$ in feedback). The computations were performed on an Intel® Core™ i7-5930K CPU @ 3.50GHz with 6 active real cores running at 3.5GHz, where one core was dedicated to the MPC controller. The number of working-set iterations performed by the QP-solver is also given (b), see Appendix 4.B for a short discussion of the solver’s iterative strategy.

The maximum measured computational time is 300μs, well within the available 700μs, as can be noticed in Figure 4.C.3(a). Only at the first time instant, when the controller is not yet activated, the computational time goes up to 800μs during initialization and memory allocation (outside plotted range). The changes in computational time follow the number of QP-solver iterations shown in (b), which is related to the number and type (amplitude or ramp-rate limits) of active constraints at each time instant.

The relevant dimensions of the QP-problems in these shots are:

<table>
<thead>
<tr>
<th>Number of:</th>
<th>#54402</th>
<th>#54414</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimization variables</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>Equality constraints</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Inequality constraints</td>
<td>36</td>
<td>72</td>
</tr>
</tbody>
</table>

We conclude that the computational time restriction was always satisfied and 400μs computational time is left per cycle, which is available for the increase of computational complexity such as the addition of actuators and state constraints.
Chapter 4. Profile control simulations and experiments in TCV: a controller test environment and results using model predictive controller

**Figure 4.C.3.** Computational time of MPC controller per cycle in 2 experiments (a), with the corresponding QP-solver iterations (b). The QP-solver iterations determine most of the computational time.
Part II

Actuator management for integrated control
Simultaneous control of plasma profiles and neoclassical tearing modes with actuator management in tokamaks

The work presented in this chapter is published as non-refereed conference paper: Maljaars, E., et al. (2015), Simultaneous control of plasma profiles and neoclassical tearing modes with actuator management in tokamaks, 42nd EPS Conference on Plasma Physics, Lisbon, Portugal, 2015, European Physical Society. The text has been revised after feedback of the thesis committee.
Chapter 5. Simultaneous control of plasma profiles and neoclassical tearing modes with actuator management in tokamaks

5.1 Introduction

Tokamaks require a plasma control system (PCS) to help ensure that the physics goals are achieved while remaining within the operational and machine limits. The PCS can use multiple actuators to affect the plasma state in real-time. In the PCS multiple control tasks can be performed that calculate requests to the actuators in order to achieve the various control task goals. In the chapters 2 to 4 we discussed profile control, which is only one of the control tasks that may need to be executed.

In present day devices, actuators are usually assigned to a specific control task. Especially in future long pulse devices (e.g. ITER), several control tasks need to be executed simultaneously that share a limited set of actuators, such that actuators can no longer be assigned to the same control task for an entire experiment. The priority of the control tasks and the availability of the actuators may also change suddenly due to events in the plasma (e.g. the occurrence of magnetic instabilities) or hardware (such as actuator failure) [Humphreys et al., 2015]. This imposes new challenges for integrating multiple control tasks sharing a set of actuators.

The architectural design of the PCS has recently gained much attention in the literature in view of future long pulse tokamaks (e.g. ITER [Treutterer et al., 2014b] and WEST [Ravenel et al., 2014]) and currently operational tokamaks (e.g. ASDEX Upgrade [Treutterer et al., 2014a]). These papers present a coherent approach to the architectural design of the PCS with a prominent role for a supervisory control layer that activates control tasks and provides their references.

In the PCS design, actuator management is needed to resolve the conflicting requests of multiple control tasks that need to share a limited set of actuators [Humphreys et al., 2015]. A first demonstration of actuator management is presented in [Rapson et al., 2015] for the Electron Cyclotron (EC) beams at ASDEX-Upgrade, where gyrotrons are optimally allocated to low level controllers for gyrotron power and launcher steering, based on higher level power requests for control tasks with corresponding importance and effectiveness. In Chapter 6 of this thesis we will introduce the reader in more detail to integrated control and we will provide an overview and analysis of PCS architectures for integrated control.

In this chapter we demonstrate integrated control of two control tasks employing a PCS architecture including actuator management. We show simultaneous control of the safety factor profile \((q\)-profile) and suppression of neoclassical tearing modes (NTMs) by using a shared amount of available EC-power that is often used for each of these individual control tasks. NTMs are magnetic islands that may develop at rational \(q\)-surfaces resulting in increased heat transport and hence confinement degradation. These can be controlled or suppressed by depositing EC current drive within the magnetic island. The proposed PCS archi-
tecture contains a high level actuator management layer that allocates resources to the profile and NTM-controller before execution of these control tasks. As such, a profile controller based on model predictive control (MPC) (as developed in Chapter 3) can be used to ensure satisfaction of (time-varying) operational limits, which would be impossible if calculated actuator commands would be modified significantly afterwards.

## 5.2 Control architecture

Based on the existing PCS architecture designs in [Ravenel et al., 2014; Treutterer et al., 2014a; Treutterer et al., 2014b], we propose a control architecture to facilitate the simultaneous control of profiles and NTMs, that could be extended to include more tasks. The control architecture is presented in Figure 5.1.

In this PCS architecture, a central decision layer sets control task priorities based on the state of plasma and status of hardware (top). These priorities are used by the high level actuator management layer to allocate a limited set of resources to each control task, such that these controllers are aware of their resources.

The NTM controller provides the actuator request for the next step to the low-level actuator management layer and in addition the desired allocated power for the next step is given back to the high level actuator management. The MPC controller computes at every time instant the future optimal control inputs for 120 time steps (of 1s) ahead, based on profile evolution models such that a $q$-profile reference is tracked and actuator and operational limits are satisfied [Maljaars et al., 2015a,c]. This MPC controller could also provide profile predictions and warnings for expected constraint violations back to the central decision layer, but this is not done in this work.

The actuator inputs as given by the NTM-controller and MPC profile controller are in this chapter simply joined with feedforward actuator inputs in the low level actuator management block. In practice, this low level actuator management layer needs to deal with the actuator systems and take its availability into account. The low level actuator management could be extended based on the work in [Rapson et al., 2015].

Finally the actuator inputs are given to the plasma transport simulator RAPTOR [Felici and Sauter, 2012a; Felici et al., 2011b], that simulates the non-linear coupled evolution of the electron temperature and poloidal magnetic flux, where the latter defines the $q$-profile.

## 5.3 Simulation results

A simulation environment is built in the block programming language Simulink [The MathWorks, 2015a] containing the designed PCS, connected to the plasma
transport simulator RAPTOR. RAPTOR has been extended to include the effect of NTMs on the plasma profiles. To simulate the effect of NTMs, the local heat transport is enhanced proportional to the island width, while the island width evolution is modeled by the Modified Rutherford Equation [Sauter et al., 2002b]. This includes also the effect of EC-actuators on the island width evolution such that stabilization or suppression of the NTM using localized EC current drive can be simulated.

A plasma scenario simulation in RAPTOR with parameters representative of the ITER tokamak as described in [van Dongen et al., 2014] has been adopted.
5.3 Simulation results

and modified for the purposes in this chapter. The nominal actuator inputs in the current flat-top phase are chosen as follows: plasma current $I_p = 12$ MA, 33 MW NBI power, and 25.7 MW EC power. The actuator trajectories are optimized to achieve stationarity of profiles at the beginning of the current flat-top (100 s) using the method described in [Felici and Sauter, 2012a; van Dongen et al., 2014].

The MPC-controller can request EC-power deposition at three fixed $\rho$-locations: $\rho = [0.05, 0.2, 0.3]$, where $\rho$ is the normalized toroidal magnetic flux coordinate. The NTM controller can request EC-power deposition at the location of the NTMs if present, assuming perfect alignment and no power modulation. We assume that in practice a low level control system will ensure that gyrotrons and steering mirrors deliver the requested EC-power deposition. The total available EC-power is set 30MW, which exceeds the power presently planned for ITER, but note that no Ion Cyclotron Resonance Heating power is used in these simulations, which is included in the proposed ITER heating and current drive mix [Snipes et al., 2012]. Priorities are set such that in case of an NTM, the NTM controller will request 12MW per NTM (at $q=2/1$ or $q=3/2$). With the remaining available power, the MPC-control objective is to track the nominal $q$-profile in the region $\rho \leq 0.5$ while satisfying the operation limit $q \geq 1$ (to avoid saw-teeth in Hybrid scenario), where it can use the power to the three EC-beams, NBI power, and plasma current $I_p$ as actuators.

The simulation results are presented in Figure 5.1. The various panels show: the actuator inputs as calculated by the MPC and NTM controllers ((a) and (b)), the NTM width evolution (c), the EC power deposition contours (d), the relative tracking error on the $q$-profile (e) and contour lines of the electron temperature evolution (f). The $q$-profile (g) and its inverse ($\iota$-profile) (h) are also given at two time steps.

In this simulation, seed islands of 3 cm are created after 200 s for the 2/1 NTM and after 400 s for the 3/2 NTM, respectively (c). The 2/1 NTM is suppressed in 16 s and the 3/2 NTM in 6 s using the localized EC current drive in the center of the NTM (d) as requested by the NTM-controller (b). At the times of maximum island widths, an off-axis peak can be noticed clearly in the deposited EC-power $V'P_{ec}$, this is the EC deposition requested by the NTM-controller. At the same time a drop in $T_e$ is visible (f), which is mainly due to the loss of central EC heating power (d). By modifying the actuator inputs ((a)-(b)), the MPC controller can limit the growth of the relative error in the $q$-profile during the presence of the NTMs (e), while keeping $q \geq 1$. The MPC controller quickly recovers the target $q$-profile after the NTMs have been suppressed (e) and full power is given back to the MPC controller (b). The maximum error in the $q$-profile is small, as can be noticed in the profiles shown in (g) and (h). An important side effect of the minimal changes in the $q$-profile is that the $\rho$-positions of the NTMs do not change significantly.
Chapter 5. Simultaneous control of plasma profiles and neoclassical tearing modes with actuator management in tokamaks

Figure 5.1. Simulation results of the effective simultaneous control of NTMs and the $q$-profile. Seed islands are created for the 2/1 NTM at 200s and the 3/2 NTM at 400s (c), resulting first in a significant drop in $T_e$ (f). The NTMs are then fully suppressed using EC deposition inside the NTM (d), while MPC profile controller uses the plasma current $I_p$ (a) and the remaining EC-power (b) to reduce the $q$-profile tracking error (e,g,h) and keep $q \geq 1$. 
5.4 Conclusion and outlook

We have shown simulations of the simultaneous operation of a profile controller and an NTM controller within a PCS architecture where these controllers are aware of their resources, allocated by a high level actuator management layer. Closed-loop simulations for the ITER tokamak show that the proposed design can effectively respond to the occurrence of an NTM by suppressing it, while at the same time the MPC profile controller maintains the safety factor profile close to its reference and within the operational limits.

The preliminary work in this chapter can be continued in multiple ways. The simultaneous control of profiles and NTMs can be tested on the newly developed Plasma Control System Simulation Platform (PCSSP) [Walker et al., 2015a]. Other control tasks sharing the same actuators could also be added. The provided expected profile predictions and constraint violations can be used by the central decision layer to improve decision making. Extending the MPC controller to track also plasma parameters like $\beta_{pol}$ may enable temporarily lowering $\beta_{pol}$ for faster suppression of NTMs. Also more advanced NTM-controllers could be implemented.

In this chapter we investigated integrated control using actuator management in a practical example. In the next chapter we will analyze in more detail the possible PCS architectures to achieve integrated control with shared actuators. From this chapter follows also that we need a more sophisticated low level actuator management. Therefore, in the next chapter we will also present an actuator allocation algorithm that can handle the complexity of actuator systems such as proposed for ITER.
Chapter 6

Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

The work presented in this chapter will be submitted to Fusion Engineering and Design.
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

6.1 Introduction

Tokamaks require a plasma control system (PCS) to control plasma quantities of interest, in order to ensure that physics goals are met while remaining within operational and machine limits. For this purpose, the PCS can use multiple actuators to affect the plasma state in real-time.

In present-day devices, actuators are usually assigned to a single control task for an entire experiment, e.g. density control, beta control or NTM control. Executing multiple control tasks at the same time is sometimes performed in tokamaks and is known as integrated control [Humphreys et al., 2005; Joffrin et al., 2007]. This is still an area of research and integrated control of all relevant phenomena is not performed routinely today.

However, in future tokamaks it will become increasingly important to use a limited set of actuators for multiple purposes during a plasma discharge [Humphreys et al., 2015; Snipes et al., 2010, 2012; Treutterer et al., 2014b; Winter et al., 2014]. Also, the priority to execute a control task may vary in time due to unforeseen plasma events and the availability of the actuators may change due to failure. Hence, real-time management of actuators is required to achieve integrated control using these shared actuators.

The PCS architecture defines the role of PCS components and the interfaces between these components. PCS architectural designs are recently presented in literature for the tokamaks ASDEX-Upgrade [Treutterer et al., 2013, 2014a], WEST [Nouailletas et al., 2013; Ravenel et al., 2014] and ITER [Treutterer et al., 2014b]. Although different in details, these papers represent a coherent approach to the PCS architecture with a prominent role for a supervisory control layer that activates control tasks and provides their references. Such a supervisor may also set control task priorities [Humphreys et al., 2015] that allow to allocate actuators to the control tasks according to these priorities. However, missing is an overview and evaluation of possible architectures to interface prioritized control tasks and the allocation of actuators that have to realize these control tasks requests.

Recently an actuator allocation algorithm was developed and successfully implemented for the ECRH system at ASDEX-Upgrade [Rapson et al., 2015, 2016]. This algorithm computes in real-time for all possible allocation options the benefits (are control task requests achieved) minus the costs (required movements of launchers etc), while taking actuator availability into account. This is an excellent first demonstration of real-time actuator management. However,
computing all allocation options for a large and complex actuator system like the one foreseen in ITER may not be feasible in real-time. This chapter therefore provides an algorithm that is inspired by [Rapson et al., 2015, 2016], but which can be executed sufficiently rapidly for real-time implementation on e.g. ITER.

In this chapter we first evaluate possible architectures to interface the prioritized control tasks with the allocation of actuators inside the PCS. We confirm that hierarchical schemes are favorable and recommendations are given to choose a specific hierarchical architecture dependent on the scale and complexity of the actuator systems involved.

Secondly we provide a generic actuator allocation algorithm for the H&CD systems using a Mixed-Integer Quadratic Programming (MIQP) optimization problem formulation, allowing to quickly search for the best allocation option without the need to compute all allocation options. The desired allocation behavior can be clearly defined in a cost function, whereas actuator availability and infeasible allocation options can be described in constraints. Simple examples are used to illustrate that the chosen desired allocation behavior is effectively achieved. Examples involving the full planned ITER H&CD system size, including Neutral Beam Injection (NBI), Ion Cyclotron (IC) and EC H&CD systems, demonstrate the algorithm's capability to perform the actuator allocation in real-time in correspondence to the desired allocation behavior. Simulations of a 100s ITER shot show effective handling of actuator failures by selecting redundant actuators according to a defined actuator preference.

The remainder of this chapter is organized as follows. In Section 6.2 possible PCS architectures for integrated control are evaluated. The MIQP-formulation of the actuator allocation problem is introduced in Section 6.3. Section 6.4 presents the performance of the actuator allocation algorithm in examples. Finally, conclusions and outlook are given in Section 6.5.

### 6.2 Overview and brief evaluation of architectures for integrated control

#### 6.2.1 Introduction to PCS schemes

Tokamak plasmas need to be actively monitored and controlled by a plasma control system to ensure that the desired plasma performance is achieved while operational and machine limits are satisfied. In future tokamaks the PCS is expected to have a clear separation between plasma state reconstruction (estimation of the plasma state from multiple diagnostics) and control and supervision tasks [Humphreys et al., 2015]. A brief general scheme of a PCS is given below in Figure 6.1.

A supervisory controller (green) is required that takes the important central decisions to handle events and that sets activation, parameters and references for
control tasks in real-time [Nouailletas et al., 2013; Ravenel et al., 2014; Treutterer et al., 2013, 2014a; Treutterer et al., 2014b]. Also priorities of control tasks can be set by a supervisor (green) in response to detected events [Humphreys et al., 2015].

Actuator control systems deal with the control of the actuator hardware to ensure that the PCS actuator commands are realized, e.g. to regulate the operating settings of a gyrotron such that it will deliver the requested power. At the same time, the actuator control systems will provide information to the PCS with information on the actuator status, parameters and constraints.

The purpose of actuator allocation is to assign actuators in real-time such that the prioritized control task requests are realized with the available actuators. For this purpose the prioritized control tasks and actuator allocation (given in the red box in Figure 6.1) need to be interfaced with each other. However, multiple architectures are possible here and presently no guidelines are available to choose an architecture for a specific tokamak given the scale and complexity of the actuator systems and the number of control tasks involved.

Figure 6.1. General scheme of a plasma control system (PCS) with multiple control tasks.
Therefore, in the remainder of this section, we will evaluate the advantages and disadvantages of possible architectures to interface the control tasks and actuator allocation (red box), where multiple control tasks need to be executed that share a set of actuators. After some remarks on cross-coupling between control tasks, we will compare three different options of computing the actuator control systems demands based on the plasma and hardware state, such that the numerous prioritized control task objectives are achieved. We will conclude with recommendations to choose a specific architecture dependent on the scale and complexity of the actuator systems involved.

6.2.2 Remarks on cross-coupling between control tasks.

Control tasks may have a strong effect on each other (cross-coupling) that potentially could lead to control stability loss. In that case it is necessary to take appropriate actions, where we may identify the following cases:

**Control tasks can be integrated.** A Multiple Input Multiple Output (MIMO) controller could be designed that combines control tasks as much as possible. For example stored energy and current profile control can be integrated into a MIMO controller, as the physics of their evolution is strongly coupled, and both are strongly affected by heating and current drive schemes [Barton et al., 2015a; Maljaars et al., 2015a].

**Control tasks can be decoupled.** This separation can be achieved either in the physics (different actuators affect different physical quantities), in space (different actuators affect different regions of the plasma), or in time (different tasks require different time scales). In this case multiple independent controllers can be designed e.g. by exploiting time scale separations between processes.

**Control tasks have only a one sided coupling.** Here one control task affects another control task, but not vice versa. In this case it may help to pass the requests from one control task to the other control task so that the second task is aware of the effects of the first. (see Section 6.2.5).

**Control tasks cannot be executed simultaneously.** If control tasks are mutually exclusive, the supervisor should only execute the control task with highest priority.

6.2.3 Interfacing control tasks and actuator allocation

We compare three architectures to interface control tasks and allocation of resources as summarized in Figure 6.2.
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

Figure 6.2. Comparison of three possible architectures to interface control tasks and actuator allocation.

1. **One resource aware controller for all control tasks** (left). Executing control tasks and allocation in one step allows to find the optimal solution. Although it might be possible to realize such a resource-aware controller that can also deal with discrete events (e.g. using hybrid model predictive control techniques [Borrelli et al., 2017]), it will be a very complex controller. Reusing existing controllers (involving much expert knowledge) is difficult or even impossible.

2. **Iterative optimization of control and allocation** (middle). In this case optimal control requests and optimal allocation may be achieved, as well as reusing existing controllers. However, providing guarantees on convergence of such a scheme is likely impossible.

3. **Hierarchical approach** (right). Here the actuator allocation is done before and / or after executing the control tasks. The main advantage of this scheme is its transparency and ease of implementation, while enabling reusing existing controllers. However, the main question here is where the actuator allocation is performed in this scheme and to what extent the optimal tradeoff between control tasks and allocation can be achieved.
6.2 Overview and brief evaluation of architectures for integrated control

### Control task priorities, activation, references
- Plasma state
- Actuator requests
- Actuator availability and constraints

### Not resource-aware controllers
- Post-controller actuator allocation

### Fully resource-aware controllers
- Pre-controller actuator allocation
- Allocated actuators for each controller

### Partially resource-aware controllers
- Post-controller actuator allocation

---

**Figure 6.3.** Three options for places of actuator allocation in a hierarchical scheme: post-controller allocation resulting in not resource-aware controllers (left), pre-controller allocation making controllers resource-aware (middle) or both pre- and post-controller allocation giving partially resource aware controllers. (right).

In the literature all schemes [Nouailletas et al., 2013; Ravenel et al., 2014; Treutterer et al., 2013, 2014a; Treutterer et al., 2014b] are hierarchical. From these comparison of the various schemes in this section, we conclude also that a hierarchical scheme is favorable, mainly thanks to its transparency and ease of real-time implementation. In the next subsection we will look in more detail on the place of actuator allocation in hierarchical schemes.

### 6.2.4 The place(s) of actuator allocation in hierarchical schemes

The main choice in a hierarchical scheme is when to perform the actuator allocation: before and / or after executing the control tasks, where this choice determines if the controller knows which actuators it can use to perform its task (if it is resource-aware or not). In Figure 6.3 we compare three options for the place of actuator allocation.
1. **Not resource-aware controllers (left).** Here only post-controller allocation is performed based on the prioritized control task requests. The post-controller actuator allocation can deal with large or complex actuator systems. This prevents that control tasks have to deal with the details of numerous and complex actuators. However, the controllers do not know their available resources when executed, they may only know at a next time step what has been allocated from information supplied by the actuator allocation.

2. **Fully resource-aware controllers (middle).** Actuators are here allocated prior to executing the control tasks, such that these can make optimal use of their assigned resources and the computed control task requests are always realizable. The pre-controller actuator allocation needs to rely on pre-defined knowledge or knowledge from a previous time step, since the control tasks request for the current time step are not yet computed. This scheme is mainly advantageous for tokamaks with simple actuator systems, where control tasks only need to know the details of a few actuators. In [Rapson et al., 2015] this scheme is employed based on pre-defined control task requests. This scheme has also been implemented in integrated control simulations in [Maljaars et al., 2015b], where control task requests from the previous time step were used to distribute the available power over the prioritized control tasks.

3. **Partially resource-aware controllers (right).** In this combined pre- and post-controller allocation, the post-controller actuator allocation block has the same role as left and in addition the pre-controller actuator allocation block can globally distribute the available actuator power over the prioritized control tasks using some ad-hoc rules, based on the control task priorities and information of previous timesteps (e.g. control task’s desired power for next time-step). Hence the control tasks approximately know their available resources and can make optimal use of these, which reduces the main drawback of option 1. However, no guarantee can be given that the pre-allocated available power per control task can be realized, which may require introducing conservatism in the global distribution of available actuator power.

   To summarize: option 1 and 3 are best for tokamaks with many and complex actuator systems, where option 3 is preferred as control tasks are globally aware of their available actuators. Option 2 is best for tokamaks with relatively simple actuator systems.

### 6.2.5 Hierarchy of controllers and allocations

We discussed in Section 6.2.3 that it is important to consider couplings between control tasks if present. Specifically, controllers could use information about the
actions of other controllers that may influence their performance. By introducing some hierarchy in executing the control tasks and allocations, coupling can be taken into account to some extent. In Figure 6.4 we compare different hierarchies in execution of controllers (and allocations).

**No hierarchy in controllers.** This is the simplest architecture, but the control tasks do not know each other’s request, which is sufficient in the absence of strong coupling between control tasks.

**Hierarchical execution of controllers.** Here each control task knows the requests of the previous executed control tasks, at the expense of a more complicated architecture. Coupling can be taken into account or the control tasks down the hierarchy can be informed about remaining actuator power. However, since the requests of the previous executed control tasks are not yet allocated, we may rely on requests that are afterwards not assigned in the allocation.

**Hierarchical execution of controllers and allocation.** This over-
comes the above mentioned problem by first allocating the request of the previous control task, before executing the next control task. However, this may lead to suboptimal actuator allocation, since for example an actuator that is essential for control task 2 may already be assigned to control task 1 (which had a higher priority), while this actuator was not essential for control task 1 and sufficient redundant actuators were available.

To summarize: to keep the simplest and most transparent architecture, further hierarchy should be avoided, unless it is required to overcome coupling between control tasks. In that case option 2 can be chosen taking into account that the requests may not be actually allocated, whereas option 3 should be avoided for its suboptimal allocation.

6.2.6 Summary and recommendations

In this section we compared several architectures to interface multiple control tasks with the allocation of actuators. We conclude that hierarchical schemes are favorable due to their transparency and ease of implementation. We recommend for tokamaks with a small number of actuator systems to use pre-controller allocation (actuators are assigned prior to executing the control tasks). For tokamaks with many and complex actuator systems we recommend to use post-controller allocation (actuators are allocated after the control tasks are executed) or a combination of these. Further hierarchy in the execution order of controllers should be avoided, as this introduces further complexity.

6.3 H&CD actuator allocation problem formulation

In the previous section we evaluated PCS architectures for integrated control and recommended to use post-controller allocation for tokamaks with complex actuator systems. In the remainder of this chapter we will focus on Heating and Current Drive (H&CD) actuators and develop a real-time post-controller actuator allocation algorithm capable of handling large and complex H&CD actuator systems. The presented algorithm is generic and can be extended to any tokamak actuator system, but we will use, as an example, the ITER H&CD systems composed of EC, NBI and IC actuator system as defined in [Henderson et al., 2015; Singh, 2016; Snipes et al., 2012].

In this section we will formulate the actuator allocation problem as a specific optimization problem. We start with specifying the considered actuator allocation problem in more detail and modelling it as a resource allocation problem. Afterwards we formulate it as an Mixed-Integer Programming problem where a
cost function defines the desired allocation behavior and constraints ensure that only physically realizable allocations are performed.

### 6.3.1 Considered actuator allocation problem and interfaces

We consider here the post-controller allocation problem in the hierarchical PCS scheme as given in Figure 6.1. This scheme allows to use also some form of pre-controller allocation (see Section 6.2.4), but this is not considered here.

![Diagram of Considered post-controller actuator allocation problem in PCS scheme](image)

**Figure 6.1.** Considered post-controller actuator allocation problem in PCS scheme. This scheme allows also some form of pre-controller allocation.

The task of the post-controller allocation block is to allocate the actuators to realize the prioritized controller requests for given actuator effect parameterizations, actuator availability and actuator constraints. The proposed interfaces to the allocation block are given in Figure 6.2.

The actuator allocation block receives the following information:

- Prioritized requests for each target:
Prioritized controller requests at targets:
- Power, current, dep. loc., dep. width
- Priorities
- Allowed actuator systems

Allocated at controller targets:
- Power, current
- Etc.

Actuator availability and constraints:
- Actuator limits
- Actuator preferences
- Actuator state

Figure 6.2. Interfaces of actuator allocation block with input information (blue) and available information after performing actuator allocation (green).

- Power requested at target.
- Current to be driven at target.
- Deposition location at target (e.g. in 1D description using normalized flux label).
- Allowed actuator systems for target.

- Parametrization of actuator effect per actuator as a function of deposition location, required to compute the potential effect of an actuator at a target location in the plasma. These actuator parameterizations can be calculated for a given plasma equilibrium and kinetic profiles, e.g. by ray-tracing:
  - Power absorption efficiency profile.
  - Current drive efficiency profile.
  - Maximum/minimum radial deposition location.

- Actuator availability and constraints:
– Actuator limits (e.g. maximum/minimum power).
– Actuator preferences (e.g. avoid using sources that have lower reliability).
– Actuator state (e.g. present state of launcher mirror angles).

• Pre-set allocations (allocations which are pre-determined and may not be changed by the RT actuator allocation algorithm unless strictly necessary, e.g. in case of suddenly unavailable actuators).

After the allocation has been performed, the following information is available to:

• Actuator control systems:
  – Actuator power request.
  – Actuator deposition location command.
  – Actuator configuration settings (e.g. desired settings of transmission line switches).

• Control tasks (available to controllers at the next time step e.g. for anti-windup):
  – Allocated power etc. per target.

• Supervisory controller:
  – Allocated resources and total available resources at this time step.

In Section 6.3.6 we define in more detail the input and output signals of the actuator allocation block.

### 6.3.2 System modeling as a resource allocation problem

The actuator allocation problem considered here can be seen as a general resource allocation problem where resources need to be assigned to a task. To model the resource allocation problem, we introduce first the following definitions:

• Power supply \( h \in \{1, \ldots, H\} \): provides electrical power to (multiple) sources.

• Source \( s \in \{1, \ldots, S\} \): converts electrical energy to energy form to be delivered in the plasma (e.g. gyrotron, NB source, RF generator).

• Delivery system \( d \in \{1, \ldots, D\} \): delivers source power to target (e.g. launcher or antenna).
• Target \( t \in \{1, \ldots, T\} \): request by control task for power/current deposition at a specific location in the plasma.

These sources can connect to (some) delivery system and a delivery system can connect to a target. This connectivity network for sources, delivery systems and targets is given below in Figure 6.3.

![Connectivity network between sources, delivery systems and targets](image)

**Figure 6.3.** Connectivity network between sources, delivery systems and targets.

Unless each source is connected to an independent power supply, the connections between power supplies and sources may impose additional requirements on the source allocations. However, this is not important in the modeling at this point, we will return on this point in Section 6.3.7.

### 6.3.3 Allocation options scaling with system size

The number of allocation options is dependent on the system size (i.e. \( S, D \) and \( T \)) and the available connection options between each of these. Obviously, a source cannot be simultaneously connected to multiple delivery systems and a delivery system not to multiple targets. In case a source \( s \) can connect to \( n_s^{SD} \) delivery systems and assuming here that all delivery systems can connect to all targets, we can write the number of allocation options as a function of the system dimensions \( S, D \) and \( T \):

\[
\#(S,D,T) = (T + 1)^D \prod_{s=1}^{S} n_s^{SD}.
\]  

(6.1)
Note that a delivery system does not need to be connected to a controller target but can be idle, which can be seen as being connected to an additional idle target (i.e. $T + 1$ instead of $T$ in the scaling).

As example, the ITER EC system has 24 gyrotrons and 11 launchers, where in the present design 16 gyrotrons can connect to 2 launchers and 8 gyrotrons can connect each to 3 launchers (see for more details the summary on the ITER H&CD system in Section 6.4.2). Using a modest number of 5 targets we can compute the number of allocation options for the ITER EC system as $\#(S,D,T) = (5 + 1)^{11} \cdot 2^{16} \cdot 3^8 \approx 1.6 \cdot 10^{17}$. Note that we did not take into account in this simple scaling the fact that not all launchers necessarily can connect to all targets (e.g. target outside their deposition range). This means that the actually feasible allocation options will be lower.

In case sources have a fixed connection to a delivery system the scaling (6.1) can be reduced to:

$$\#(S,T) = (T + 1)^S.$$  \hfill (6.2)

The EC system at ASDEX-Upgrade tokamak has such fixed connections between gyrotrons and launchers. Using 4 sources (gyrotrons) for 5 targets (central heating, 3/2 and 2/1 NTM control, $q = 1.5$ and $q = 2.0$ surface tracking to pre-empt NTMs) resulted in $(5+1)^4 = 1296$ allocation options [Rapson et al., 2015].

In the actuator allocation algorithm in [Rapson et al., 2015], the best allocation option was found by computing for all allocation options the effectiveness (how well power requests are achieved) minus its costs (movement of launchers, allocation switch and non-idle gyrotrons), where infinite costs were assigned to infeasible allocation options. Solving this allocation problem for a single time step could be performed in less than 0.3ms for the above mentioned 1296 allocation options, which is sufficiently fast for the ASDEX-Upgrade tokamak.

Computation of all allocation options for an ITER-like actuator system with the above calculated more than $10^{17}$ allocation options may not be computationally feasible within a reasonable time (e.g. 100 ms, see remarks in Section 6.4.2), even if code-optimization, parallelization and future improved hardware is considered.

### 6.3.4 Formulation as a generic optimization problem

The discussed actuator allocation problem formulation in [Rapson et al., 2015] may be considered as a special case of an optimization problem formulation. There a cost function is evaluated using brute-force computing to select the allocation option with minimum cost. Alternatively, the actuator allocation problem can be formulated as a different class of optimization problems that
allows to use techniques that can quickly find satisfactory allocation options without the need to evaluate all allocation options.

Resource allocation problems have often been formulated as Mixed Integer Programming (MIP) problems [Appa et al., 2006; Nemhauser and Wolsey, 1988]. A MIP-problem consists typically of a cost function to be minimized over optimization variables and a set of constraints that have to be satisfied. In a resource allocation problem the cost function typically promotes desired / penalizes undesired allocations whereas constraints can be used to ensure that only physically realizable allocations are selected.

MIP-problems involve a mixture of continuous and integer optimization variables. In our actuator allocation problem a source will be either connected to or disconnected from a certain delivery system (a discrete choice, that can be modelled using an integer variable). At the same time, the power of a given source may be a continuous variable and could also be optimized.

We choose a quadratic cost function, resulting in a Mixed-Integer Quadratic Programming (MIQP) problem\(^1\). The MIQP-problem can be written as follows:

\[
\begin{align*}
\text{minimize} & \quad J(z) = z^\top Hz + f^\top z & \text{ (cost-function)} \\
\text{subject to} & \quad A_{\text{ineq}} z \leq b_{\text{ineq}} & \text{ (linear inequality constraints)} \\
& \quad z_{\text{min}} \leq z \leq z_{\text{max}} & \text{ (bounds)} \\
& \quad z_i \in \mathbb{N} & \text{ (integer variables)}
\end{align*}
\]

The vector \( z \) contains the optimization variables, where some of these are restricted to be integer. The cost function \( J(z) \) contains both a quadratic and linear component. Linear inequality constraints and bounds can be used to impose further restrictions on the optimization variables. The transpose of a vector \( v \) is denoted as \( v^\top \).

The formulation as a MIQP optimization problem has two major strengths:

- Components of the cost function and constraints can be easily added or removed.
- Well-established methods can be used to search in a computationally efficient manner for the best allocation option without the need to evaluate all allocation options.

In the following section, we shall see how the actuator allocation problem can be structured in this MIQP form.

### 6.3.5 Optimization variables choice

The first step in writing the optimization problem is the definition of the optimization variables \( z \). We choose to use for each connection between a source and

\(^1\)Alternatively a linear cost function could be chosen (see Appendix 6.D)
a delivery system (and between a delivery system and a target) both a continuous variable $P$ for the power ‘flow’ along a connection and a binary variable $\alpha$ to indicate if a connection is active (see Figure 6.4), with the following relation:

- $\alpha_{s,d} = 1$ if $P_{s,d} > 0$, otherwise $\alpha_{s,d} = 0$
- $\alpha_{d,t} = 0$ if $P_{d,t} > 0$, otherwise $\alpha_{d,t} = 0$

We can now define the optimization variable vector $z$ in (6.3) as:

$$z = \begin{bmatrix} \bar{P}_{S2D} \\ \bar{P}_{D2T} \\ \bar{\alpha}_{S2D} \\ \bar{\alpha}_{D2T} \end{bmatrix}$$

where the variables $\bar{P}_{S2D}$, $\bar{P}_{D2T}$, $\bar{\alpha}_{S2D}$ and $\bar{\alpha}_{D2T}$ are also vectors, e.g.

$$\bar{P}_{S2D} = \left[ \left[ P_{s=1,d=1} \ldots P_{s=1,d=D} \right]^{\top} \left[ P_{s=1,d=1} \ldots P_{s=1,d=D} \right]^{\top} \ldots \left[ P_{s=1,d=1} \ldots P_{s=1,d=D} \right]^{\top} \right]^{\top}.$$

This choice of optimization variables leads to $(SD + DT)$ continuous and $(SD + DT)$ binary variables, but providing linear constraints and a quadratic cost function that has a structure that can be exploited in a MIP-solver. In case sources have a fixed connection to a delivery system, a more compact optimization problem could be formulated. This is discussed in Appendix 6.D.

### 6.3.6 System configuration and algorithm input/output definition

The remainder of this section goes on to explain in detail the interfaces as well as the various cost function terms and constraints that enter into the MIQP
optimization problem. Given the multitude of technical constraints and possible optimization quantities of interest, this section may be skipped upon first reading, and readers interested in illustrative examples can proceed directly to Section 6.4.

A description of the H&CD actuator system in terms of the actuator allocation algorithm description is required. This system configuration is assumed to be fixed and known before a simulation or experiment starts. Besides the dimensions of the actuator system involved (number of sources $S$ and delivery systems $D$ as defined in Section 6.3.2), we also need to define the various actuator types, the connection topologies between actuator components and the power transfer efficiency between a source and delivery system. The system configuration definition is given in Table 6.1.

### Table 6.1. System configuration definition.

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Resource type:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source type</td>
<td>$S_{\text{type}}^s$</td>
<td>Integer: 1) EC, 2) IC and 3) NB</td>
</tr>
<tr>
<td>Delivery system type</td>
<td>$D_{\text{type}}^d$</td>
<td>Integer: 1) EC, 2) IC and 3) NB</td>
</tr>
<tr>
<td><strong>Connection topology maps between:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sources and delivery systems</td>
<td>$M_{s,d}^{S2D}$</td>
<td>${0$ (no connection possible), $1$ (connection possible)$}$</td>
</tr>
<tr>
<td>Power supplies and sources</td>
<td>$M_{h,s}^{H2S}$</td>
<td>${0$ (no connection possible), $1$ (connection possible)$}$</td>
</tr>
<tr>
<td><strong>Actuator efficiency:</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power transfer efficiency from source to delivery system</td>
<td>$\eta_{s,d}^{S2D}$</td>
<td>$0 \leq \eta_{s,d}^{S2D} \leq 1$</td>
</tr>
</tbody>
</table>

The interfaces of the actuator allocation algorithm where briefly introduced in words in Section 6.3.1 (see also Figure 6.2). Here we need to define in more detail the input and output information of the algorithm by specifying the variables that contain this information with their corresponding units.

The actuator allocation algorithm requires real-time knowledge of the control task requests, source and delivery system information, the present allocation, actuator effect parameterizations and feedforward (pre-set) allocations. This input information is specified in Table 6.2.

The normalized toroidal magnetic flux is used for the deposition location $\rho$ as well as the full Gaussian deposition width $w$ in this chapter, which assumes that back and forth transformation in physical actuator variables (such as launcher angles) is done outside the algorithm using e.g. real-time equilibrium reconstruction, ray tracing and machine geometry data. For a deposited power we always use the volume integrated power and for driven current always the surface integrated driven current (by delivery system on a target).
Table 6.2. Definition of real-time information available to the actuator allocation algorithm.

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable</th>
<th>Unit</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller requests per targets:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Volume integrated power</td>
<td>$P^\text{req}_t$</td>
<td>[MW]</td>
<td></td>
</tr>
<tr>
<td>Surface integrated driven current</td>
<td>$I^\text{req}_t$</td>
<td>[MA]</td>
<td></td>
</tr>
<tr>
<td>Deposition location</td>
<td>$\rho^\text{req}_t$</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Desired full gaussian deposition width</td>
<td>$w^\text{req}_d$</td>
<td>-</td>
<td>$0 \leq w^\text{req}_d \leq 1$</td>
</tr>
<tr>
<td>Importance of power matching</td>
<td>$W^\text{P,req}_t$</td>
<td>-</td>
<td>$0 \leq W^\text{P,req}_t \leq 1$</td>
</tr>
<tr>
<td>Importance of current matching</td>
<td>$W^\text{I,req}_t$</td>
<td>-</td>
<td>$0 \leq W^\text{I,req}_t \leq 1$</td>
</tr>
<tr>
<td>Importance of deposition width matching</td>
<td>$W^\text{w,req}_t$</td>
<td>-</td>
<td>$0 \leq W^\text{w,req}_t \leq 1$</td>
</tr>
<tr>
<td>Delivery system type is allowed at target</td>
<td>$D^\text{allow}(i_d,\text{type})$</td>
<td>-</td>
<td>$(0(\text{false}),1(\text{true}))\ \ \ \ \forall i_d,\text{type} \in {1,2,3}$</td>
</tr>
<tr>
<td>Source information:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power minimum</td>
<td>$P^\text{S,min}_s$</td>
<td>[MW]</td>
<td></td>
</tr>
<tr>
<td>Power maximum</td>
<td>$P^\text{S,max}_s$</td>
<td>[MW]</td>
<td></td>
</tr>
<tr>
<td>Source avoidance penalty</td>
<td>$W^\text{S,avoid}_s$</td>
<td>-</td>
<td>$0 &lt; W^\text{S,avoid}_s \leq 1$</td>
</tr>
<tr>
<td>Source is connected to its power supply</td>
<td>$\xi^{\text{H2S}}_s$</td>
<td>-</td>
<td>$(0(\text{false}),1(\text{true}))$</td>
</tr>
<tr>
<td>Source is being switched to other del. sys.</td>
<td>$\alpha^{\text{S2D}}_{s,d}$</td>
<td>-</td>
<td>$(0(\text{false}),1(\text{true}))$</td>
</tr>
<tr>
<td>Delivery system information:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power maximum</td>
<td>$P^\text{D,max}_d$</td>
<td>[MW]</td>
<td></td>
</tr>
<tr>
<td>Delivery system avoidance penalty</td>
<td>$W^\text{D,avoid}_d$</td>
<td>-</td>
<td>$0 &lt; W^\text{D,avoid}_d \leq 1$</td>
</tr>
<tr>
<td>Present allocation per source and delivery system:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source power request</td>
<td>$P^\text{S,press}_s$</td>
<td>[MW]</td>
<td></td>
</tr>
<tr>
<td>Delivery system deposition location</td>
<td>$\rho^\text{D,press}_d$</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Source is active on delivery system</td>
<td>$\alpha^{\text{S2D,press}}_{s,d}$</td>
<td>-</td>
<td>$(0(\text{false}),1(\text{true}))$</td>
</tr>
<tr>
<td>Delivery system is active on target</td>
<td>$\alpha^{\text{D2T,press}}_{d,t}$</td>
<td>-</td>
<td>$(0(\text{false}),1(\text{true}))$</td>
</tr>
<tr>
<td>Actuator effect parameterization maps:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power transfer efficiency del. sys. to target</td>
<td>$\eta^{\text{D2T}}_{d,t} = \eta^{\text{D2T}}_d(d,\rho^\text{req}_t)$</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Deposition width del. sys. at target</td>
<td>$w^{\text{D2T}}_{d,t} = w(d,\rho^\text{req}_t)$</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Current drive efficiency del. sys. at target</td>
<td>$\eta^{\text{cd}}_{d,t} = \eta^{\text{cd}}_d(d,\rho^\text{req}_t)$</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Minimum deposition location del. sys.</td>
<td>$\rho^\text{min}_d$</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Maximum deposition location del. sys.</td>
<td>$\rho^\text{max}_d$</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Feedforward (pre-set) allocations per source and delivery system:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source is active on delivery system</td>
<td>$\alpha^{\text{S2D,ff}}_{s,d}$</td>
<td>-</td>
<td>$(0(\text{false}),1(\text{true}))$</td>
</tr>
<tr>
<td>Delivery system is active on target</td>
<td>$\alpha^{\text{D2T,ff}}_{d,t}$</td>
<td>-</td>
<td>$(0(\text{false}),1(\text{true}))$</td>
</tr>
</tbody>
</table>

Once the actuator allocation has been performed, information can be sent to the actuator control system, control tasks and supervisory controller. This output information is specified in Table 6.3, where also the relation to the optimization variables is given.

### 6.3.7 Cost function penalties

In order to be able to define the desired allocation behavior of the algorithm, we include penalties on undesired allocation behavior in the cost function of the
Table 6.3. Definition of the algorithm output and relation to optimization variables.

<table>
<thead>
<tr>
<th>Description</th>
<th>Variable</th>
<th>Units</th>
<th>Relation to optimization variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information to actuator control systems about allocated:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source power</td>
<td>$P_{S,alloc}^d$</td>
<td>[MW]</td>
<td>$\sum_{d=1}^{D} \sum_{t=1}^{T} P_{S,d}^d T_{d,t}^2 \alpha_{d,d}$</td>
</tr>
<tr>
<td>Del. sys. power</td>
<td>$P_{D,alloc}^d$</td>
<td>[MW]</td>
<td>$\sum_{d=1}^{D} \sum_{t=1}^{T} \eta_{S,d}^d P_{S,d}^d \alpha_{d,t}$</td>
</tr>
<tr>
<td>Del. sys. per source</td>
<td>$D_{S,alloc}^d$</td>
<td>-</td>
<td>$\sum_{d=1}^{D} \alpha_{d,d}$</td>
</tr>
<tr>
<td>Del. sys. dep. loc.</td>
<td>$P_{d}^t$</td>
<td>-</td>
<td>$\sum_{t=1}^{T} \alpha_{d,t} \rho_{t}$</td>
</tr>
</tbody>
</table>

| Information to control tasks about allocated: | | | |
| Power at target | $P_{T,alloc}^t$ | [MW] | $\sum_{d=1}^{D} \eta_{d,T}^t P_{D,d}^t T_{d,t}$ |
| Driven current at target | $I_{T,alloc}^t$ | [MA] | $\sum_{d=1}^{D} \eta_{d,T}^t \eta_{d,T}^d (d, \rho_{req}) P_{D,d}^t T_{d,t}$ |

| Information to supervisory level about available: | | | |
| Power per act. sys. type | $P_{max,S,type}^{i_S,typ} (i_S,typ)$ | [MW] | $\sum_{s=1}^{S} P_{max,S}^{s}$ if $S_{type} = i_S,typ, i_S,typ \in \{1, 2, 3\}$ |

MIQP-problem (6.3). Various cost penalties can be used to describe different aspects of the allocation behavior. A number of these cost penalties will be described here in detail, whereas for others we will refer to Appendix 6.B. These cost penalties are examples of what is possible, and depending on the details of the actuator systems other cost penalties may be added as needed.

Penalize difference between requested and allocated values at targets

The main objective of the actuator allocation is to achieve the prioritized control tasks requests at the targets. Therefore we penalize the difference between the allocated and requested power, driven current and deposition width at the targets. Quadratic penalties are used to penalize large deviations from the requests significantly more than small deviations. In addition, the requests are weighted according to the control task priorities and normalized. These cost penalties can be expressed as follows:

Power.

\[ J_P = \nu_P \frac{1}{T} \sum_{t=1}^{T} W_{P,req}^t \frac{1}{(P_{P,req, norm})^2} (P_{T, alloc}^t - P_{T, req}^t)^2 \]  \hspace{1cm} (6.5)

Driven current.

\[ J_I = \nu_I \frac{1}{T} \sum_{t=1}^{T} W_{I,req}^t \frac{1}{(I_{I, req, norm})^2} (I_{T, alloc}^t - I_{T, req}^t)^2 \]  \hspace{1cm} (6.6)

\[ \text{An appropriate choice of the normalization for required power } P_{T,req, norm}^t \text{ (6.5), required current } I_{I,req, norm}^t \text{ (6.6) and present power } P_{S, pres, norm}^s \text{ (6.8) is discussed in Appendix 6.E.} \]
Deposition width.

\[ J_w = \nu_w \frac{1}{T} \sum_{d=1}^{D} \sum_{t=1}^{T} W_{t,\text{req}}^w (w_{d,t}^{D2T,\text{del}} - w_{t}^{\text{req}})^2 \alpha_{d,t}^{D2T} \]  

(6.7)

The tuning parameters \( \nu(\cdot) \) can be used to set user-defined preferences of the allocation behavior.

Penalize changes w.r.t. present allocations

To prevent unnecessary changes, it may be desirable to keep the system as close as possible to the present allocation. Therefore we penalize changes w.r.t. the present allocation for:

**Source powers.** To promote solutions where the required change in power allocation is small.

\[ J_{\Delta P} = \nu_{\Delta P} \frac{1}{S} \sum_{s=1}^{S} \left( \frac{1}{(P_s^{\text{alloc}} - P_s^{\text{pres}})^2} \right)^{\frac{1}{2}} \]  

(6.8)

**Delivery system deposition location.** To promote selecting delivery systems that are already close to the target.

\[ J_{\Delta \rho} = \nu_{\Delta \rho} \frac{1}{D} \sum_{d=1}^{D} \sum_{t=1}^{T} (\rho_{t,\text{req}} - \rho_{d,\text{pres}})^2 \alpha_{d,t}^{D2T} \]  

(6.9)

**Connected delivery system to source.** To avoid unnecessary switching the connections between sources and delivery systems since this may lead to temporarily unavailable power and fatigue.

\[ J_{S2D,\text{ac, pres}} = \nu_{S2D,\text{ac, pres}} \frac{1}{S} \sum_{s=1}^{S} \sum_{d=1}^{D} (1 - \alpha_{s,d,\text{pres}}^{S2D}) \alpha_{s,d}^{S2D} \]  

(6.10)

As connecting a source to a different delivery system may take some time, it could be undesired to reallocate the source to the previously connected delivery system while this switch is performed. Such a situation can be avoided by adding a cost penalty (6.15). In addition, it can be desired to keep using the same delivery systems at their already allocated targets if possible, which is expressed in (6.16).
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

Penalize use of specific resources

We may prefer to avoid using certain sources, delivery systems or their connections. This is reflected in the penalization of, for example:

**Sources.** This enables to set a preference for specific actuators to e.g. avoid using sources that have proven to be less reliable.

\[
J_{S,\text{avoid}} = \nu_{S,\text{avoid}} \frac{1}{S} \sum_{s=1}^{S} \sum_{d=1}^{D} W_{s}^{S,\text{avoid}} \alpha_{s,d}^{S2D} \tag{6.11}
\]

Setting \( W_{s_i}^{S,\text{avoid}} < W_{s_j}^{S,\text{avoid}} \) gives preference to use source \( s_i \) w.r.t. source \( s_j \).

**Delivery systems.** Similarly to sources, e.g. to avoid the use of specific delivery systems for technical reasons.

\[
J_{D,\text{avoid}} = \nu_{D,\text{avoid}} \frac{1}{D} \sum_{d=1}^{D} \sum_{t=1}^{T} W_{d}^{D,\text{avoid}} \alpha_{d,t}^{D2T} \tag{6.12}
\]

**Connections between sources and delivery systems.** This can be used to avoid e.g. the use of specific connections for technical reasons, e.g. higher losses.

\[
J_{S2D,\text{avoid}} = \nu_{\text{avoid}}^{S2D} \frac{1}{S} \sum_{s=1}^{S} \sum_{d=1}^{D} W_{s,d}^{S2D,\text{avoid}} \alpha_{s,d}^{S2D} \tag{6.13}
\]

Penalize changes w.r.t. pre-defined (feedforward) allocation

In some situations, it may be desirable to keep the allocation close to a pre-defined allocation (e.g. to use a manually pre-set allocation). Imposing this as a strict constraint would exclude the ability to react to actuator failure, changing priorities etc. Therefore penalties are introduced in the cost function for changes w.r.t. pre-set allocations for using sources on delivery systems (6.17), and delivery systems on targets (6.18).

Penalize specific allocations of sources sharing a power supply

In case two sources are sharing the same power supply, there are specific situations that are undesired and could be avoided using the algorithm. Connecting or disconnecting one of these sources to its power supply might be undesired, since this may take a significant amount of time (up to 3s for ITER [Henderson et al., 2015]). This can be avoided using the cost penalty (6.19). In addition, connecting these sources to different delivery systems may be undesired, as...
sources connected to the same power supply should have equal power and a future change for power at one delivery system will require also a power change in the other delivery system. This situation can be avoided using the cost penalty (6.20).

Remarks on tuning allocation behavior

To choose the tuning parameters $\nu(\cdot)$, a user of the algorithm would typically define a broad set of representative cases expected during system operation, with the corresponding desired allocation behavior. One can then seek values of $\nu(\cdot)$ to obtain the desired behavior of the algorithm. Tuning is eased by the normalization of all cost terms such that these are all in the same order.

6.3.8 Constraints defining allocation feasibility and actuator availability

The actuator allocation algorithm should only perform allocations that are technically realizable by the available sources and delivery systems. This requires a description of the allocation feasibility and actuator availability, which can be formulated as constraints. We describe here briefly the constraints and refer to Appendix 6.C for the details. Constraints have been formulated for the following reasons:

- Technical constraints to relate the active flags (binary variables) and power flows (continuous variables), and actuator availability: (6.21)-(6.23).

- Constraints to ensure that a source can only connect to a single delivery system (6.24), and similarly a delivery system can only be allocated to a single target (6.25).

- Technical constraints to ensure that active sources sharing the same power supply have equal source power (6.26).

- Constraints to exclude allocation options that are physically not realizable due the fact that there is no connection present between a source and delivery system (6.27), the target is out of reach of the delivery system (6.28) or the delivery system type is not allowed at the target (6.29).

These constraints can be imposed on the optimization problem using linear inequality constraints and bounds that fit in the MIQP formulation (6.3).

6.3.9 Constructing and solving MIQP-problem

We can now proceed to formulate the MIQP-problem (6.3) by using the definitions of the desired allocation behavior in cost penalties (Section 6.3.7) and
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

the allocation feasibility and actuator availability in constraints (Section 6.3.8). For this purpose, the cost function and constraints are written in matrix/vector format in terms of the optimization variables vector $z$ given in (6.4). The Hessian $H$ and gradient vector $f$ can be derived from (6.5)-(6.13) and (6.15)-(6.20), and the inequality constraint matrix $A_{ineq}$ and vector $b_{ineq}$ from (6.21)-(6.26). The bounds $z_{min}$ and $z_{max}$ can be derived using (6.27)-(6.29) and the variables $P_{s,d}^{S,\max}$ and $P_{d,t}^{D2T,\max}$.

Once the matrices and vectors of the MIQP-problem are constructed, it can be readily solved using existing solvers. This yields the optimal choice for the optimization variable vector $z$ (6.4). The outputs of the actuator allocation block are then computed from this MIQP-solution.

Many MIP-solvers are available, including the state-of-the-art commercial solvers such as CPLEX [IBM, 2016] and Gurobi [Gurobi Optimization, 2016] (both with free academic license). CPLEX and Gurobi are among the fastest available solvers, see [Mittelmann, 2016] for a frequently updated benchmark of MIP-solvers. In this chapter we use the solver CPLEX, called from Matlab [The MathWorks, 2015a].

MIQP-problems and the underlying decision problem are known to be non-deterministic polynomial-time hard (NP-hard) [Nemhauser and Wolsey, 1988], implying that a-priori no guarantee can be given that not all decision options have to be evaluated to choose the best. Fortunately, in practice only a subset of possible decision options needs to be evaluated, such that MIQP-problems can be solved in a reasonable time. MIP-solvers like CPLEX can quickly provide good solutions, while ensuring that there is no better solution available, up to a set tolerance (MIPgap tolerance), than the best feasible solution found sofar during solving. This eliminates the need to evaluate many (almost) identical solutions, which is important for allocation problems with many similar allocation options. The solver can also yield good feasible solutions even if it is stopped by a time limit before the currently best feasible solution is within the MIPgap tolerance. This is important for real-time application of the actuator allocation algorithm.

6.4 Performance of H&CD actuator allocation algorithm

We illustrate here the principle and performance of the allocation algorithm in examples. We begin with an example with a limited system size and then present examples with the full planned ITER H&CD system size.

6.4.1 Illustrating principle for small system size example

We start with illustrating the working principle in a simple example where we selected 4 sources (called S1 to S4) and 4 delivery systems (called D1 to D4)
from the ITER EC system with corresponding system parameters\textsuperscript{3}. Delivery system D1 will drive a negative (counter) current, whereas the others will drive a positive (co) current. Also 4 target are chosen with requests for power, current and deposition location.

Figure 6.1 shows the targets and present allocations. In (a) the targets T1 to T4 are given for power versus deposition location (x) with each priority weight, indicating that target T1 has highest priority. The present allocation of the delivery systems is indicated by dots, meaning that D3 and D4 are presently idle. The source allocations and powers are given in (b) and (c) respectively, indicating that sources S2 and S3 are presently the only used sources. In (d) and (e) the present delivery system allocations are given, showing that delivery system D1 was active at target T2 and D2 at target T1. Panels (f) and (g) present the target requests for power and current respectively, with their corresponding priority (we choose the priorities equal for power and current: $W^t_I = W^t_P$).

\textbf{Figure 6.1.} Targets and present allocations for small system size illustration. Top (a): present allocation (dot) for power and deposition locations of delivery systems. Targets are indicated with (x) and their target importance below these. Deposition location ranges for delivery systems are given by horizontal bars. Bottom left: present source powers (b) and allocations c). The maximum source power is given by the grey bar. The feasible connections for a source to a delivery system are colored. Bottom middle: present delivery system allocations (d) and powers (e). Bottom right: required and allocated power (f) and current (g) at the indicated priorities $W^t_P$ and $W^t_I$.

\textsuperscript{3}System details are only given for the full size example in Section 6.4.2.
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

We will now add cost function components step by step to visualize the corresponding allocation behavior.

Matching power request at target

We start with only a penalty on the difference between required and allocated power: $\nu_P = 1$ (see (6.5)). The result is given in Figure 6.2, where the allocated values are indicated with a circle and the corresponding change with an arrow.

![Figure 6.2. Allocation result with only a penalty on the power mismatch at the targets. Allocated values (circles) and required changes (arrows) are added to the present situation as given in Figure 6.1. All target powers are achieved as allocated values (o) coincide with targets (x) in (a), see also (f). This is achieved by changing the deposition location and power of all delivery systems and requires two connection switches (c). All delivery systems move to a different target than the presently allocated target (e).](image)

As expected, the power requests are achieved (see (a) and (f)), requiring the use of all 4 delivery systems. Note that these make large movements, which seems not necessary. Two connection switches between sources and delivery systems are required for sources S1 and S3 (c). The requested current for target T1 is not achieved (red priority weight), while for the other targets (green priority weight) this is achieved, but by coincidence (as no cost term corresponding to current matching was set).
6.4 Performance of H&CD actuator allocation algorithm

Including current matching at targets

Now we add a penalty on the current mismatch at the targets by choosing $\nu_I = 0.1$ (see (6.6)). The resulting allocation is given in Figure 6.3.

![Figure 6.3](image)

Figure 6.3. Illustration of allocation with penalties on both the power and current mismatch a target. Note that the current at target T1 is matched by using both a counter-current (blue) and co-current delivery system (red), at the expense of losing target T2 with lowest priority.

The requested zero current at target T1 (highest priority) is now achieved (g) by allocating both the counter delivery system D1 and co-current delivery system (D2) to this target. As these delivery systems do not have equal current drive efficiency, their power is slightly different (f) but sums up to the requested power for target T1. No power is allocated to the lowest priority target T2 as it is the least important and no delivery system and source is left to allocate to this target.

Including a movement cost

We will now try to minimize the delivery system movement costs by setting $\nu_\rho = 0.1$ (see (6.6)) such that the most nearby delivery systems should be selected.

Indeed Figure 6.4 shows that the movement of D3 and D4 is minimized by allocating them to the most nearby targets 3 and 4 respectively ((a) and e)), at the small expense that the allocated current at target T3 is slightly off (g).
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

6.4.2 Performance in typical ITER examples

We will now demonstrate the performance of the allocation algorithm in two examples with the full dimensions of the ITER H&CD system including EC, IC and NBI.

System configuration ITER H&CD and algorithm settings

First we need to represent the system configuration of the ITER H&CD actuator system in terms of our allocation algorithm:

Electron Cyclotron system [Henderson et al., 2015; Singh, 2016]. The ITER EC actuator system has 11 steerable mirrors (delivery systems): 3 at the Equatorial Launcher (EL) and 2 in each of the 4 Upper Launchers (UL). Its 24 gyrotrons (sources) can connect to up to 3 delivery systems.
Two gyrotrons share a power supply, where the first power supply feeds the first and second gyrotron, the second the next two gyrotrons etc.

**Ion Cyclotron system** [Singh, 2016; Snipes et al., 2012]. The IC actuator system designed for ITER involves 8 3MW RF sources that can each connect to one of the two delivery systems (antennas) and deliver 20MW into the plasma, where a spare RF source can replace one of the 8 sources in case of failure [Singh, 2016]. In this chapter we assumed that we can model the IC system as 2 IC sources / delivery systems that are assumed here to be able to modulate power between half and full power (10MW), following the brief description in [Snipes et al., 2012]. Each IC source can only connect to a single IC delivery system and has its own power supply.

**Neutral Beam Injection system** [Singh, 2016; Snipes et al., 2012]. Two NB sources / delivery systems of 16.5 MW can only be on or off and although the deposition location cannot be changed, these NB delivery systems are assumed here to be able to satisfy power requests in a deposition range due to their broad deposition profile. Also each NB source can only connect to its own NB delivery system and has its own power supply.

The actuator parameterizations are chosen as follows:

- The power transfer efficiency of delivery systems to targets $\eta_{D2T}^{d,t}$ is fixed at 1 (no losses, e.g. full power absorption).
- The deposition width parameterization $w_{D2T}^{d,t} = f_w(d, \rho_{req}^{t})$ is not used in these examples and not defined here.
- The current drive efficiency parameterization $\eta_{\text{cd}}^{d,t} = f_{\eta}(d, \rho_{req}^{t})$ is given in Appendix 6.A with corresponding parameters.
- The deposition ranges of the delivery systems are given in the figures presenting the results.

The power transfer efficiency of sources to delivery systems $\eta_{S2D}^{d,t}$ is fixed at $\frac{20}{27}$ for EC, such that 24 1MW gyrotrons can deliver 20 MW to the plasma, whereas no losses are assumed for IC and NB. Other parameters such as the minimum and maximum source and delivery system powers, the deposition ranges of the delivery systems, the feasible connections between sources and delivery systems are shown directly in the figures.

---

4We assume in these examples that the gyrotrons can achieve on average the continuous power requests using e.g. modulation. One could also set $P_{S, \text{min}} = P_{S, \text{max}}$ such that each source can either provide full power or no power.

5Details of the proposed EC source to delivery system connections were given in a personal communication [Henderson, 2016].
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

In the next examples we choose the cost penalties as given in Table 6.1. Furthermore we use the default settings and tolerances of the CPLEX MIP-solver [IBM, 2016].

Table 6.1. Cost penalties in ITER examples.

<table>
<thead>
<tr>
<th>Cost penalty on</th>
<th>Coefficient</th>
<th>Value</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power mismatch</td>
<td>$\nu_P$</td>
<td>1</td>
<td>(6.5)</td>
</tr>
<tr>
<td>Current mismatch</td>
<td>$\nu_I$</td>
<td>$10^{-2}$</td>
<td>(6.6)</td>
</tr>
<tr>
<td>Source power changes</td>
<td>$\nu_{\Delta P}$</td>
<td>$10^{-5}$</td>
<td>(6.8)</td>
</tr>
<tr>
<td>Delivery system movements</td>
<td>$\nu_\rho$</td>
<td>$10^{-2}$</td>
<td>(6.9)</td>
</tr>
<tr>
<td>Allocation of source to other delivery system</td>
<td>$\nu_{S2D,ac,pres}$</td>
<td>$10^{-4}$</td>
<td>(6.10)</td>
</tr>
<tr>
<td>Source use (equal source preference)</td>
<td>$\nu_{S,avoid}$</td>
<td>$10^{-4}$</td>
<td>(6.11)</td>
</tr>
<tr>
<td>Delivery system use (equal preference)</td>
<td>$\nu_{D,avoid}$</td>
<td>$10^{-5}$</td>
<td>(6.12)</td>
</tr>
<tr>
<td>Allocation of active sources sharing a power supply to different delivery systems</td>
<td>$\nu_{S2D,ac,pres}$</td>
<td>$10^{-4}$</td>
<td>(6.20)</td>
</tr>
</tbody>
</table>

Example: Sudden EC power request for NTM control requires additional central IC

We assume that the system starts in a situation where power is concentrated in the plasma core on 3 targets (central heating and profile control). Suddenly, 10MW of power is requested for NTM control at $\rho = 0.6$. The resulting allocation is given in Figure 6.5. The allocation algorithm uses the second IC system to take over the central heating target with zero current request, enabling the EC system to redistribute its power over the other targets (panel (a)). The NTM control target T4 gets the required power from three nearby UL-launchers (panels (a) and (f)). Redistributing the EC-power requires many switches between sources and delivery systems (c), however, these switches are the minimum required to satisfy all power targets. All power requests are achieved and only a small mismatch is left on the current (g).

Example: NBI failure requires replacement by other actuator systems

In this second example we show how the algorithm can effectively compensate for actuator failure. We have a single target request for 40MW power at $\rho = 0.1$ (e.g. for beta-control), that was in the present allocation achieved by a combination of two NB-systems and 8 gyrotrons from the EC-system. In this allocation step the second NB system is suddenly unavailable. Both EC and IC systems can be used to replace the missing NB-system. The results are given in Figure 6.6. The algorithm automatically selects the two IC sources to take over the unavailable NB system (panels (a), (b) and (e)). This allocation option requires minimum changes in source powers and the minimum amount of active sources.
6.4 Performance of H&CD actuator allocation algorithm

The central heating target T1 is taken over by the nearby IC2, providing freedom in the EC-system to redistribute the power, mainly to achieve the 10MW NTM control target T4 with highest priority (a) and (e)). Note that sources sharing the same power supply and both in use have the same power (b). This allocation result involves many allocation switches between source and delivery system, but minimum for satisfying simultaneously all power targets (c).

Figure 6.5. Results for example with sudden NTM control request while performing central heating/ profile control.
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

Figure 6.6. Results example for replacing suddenly unavailable NB power with IC power. A single target requires 40MW power (panels (a) and (f), where the current importance (g) is zero. Nearby located IC-systems take over the missing NB power (panels (a), (b) and (e)). EC allocations remain unchanged, so as to minimize source changes and the number of used sources.
Example: Simulating EC management for 100s ITER shot

The examples shown may only become important during a late phase of ITER operation. However, even during early ITER operation it is also important to be able to compensate for actuator failure by selecting redundant actuators according to a set actuator preference order. We will now illustrate the algorithm's capability to handle this in an example involving multiple trips in EC sources (gyrotrons) during a 100s ITER shot.

We specify a single target with a power request reference with staircases of each 5 MW increase or decrease that can be achieved by adding or removing 6 gyrotrons. Gyrotrons can only be at full power or fully off: $P_{s,\text{min}} = P_{s,\text{max}} = 1\text{MW}$. The current request is linearly scaled with the power request and reachable by having equal power at all three mirrors of the Equatorial Launcher. The source avoidance weight $W_{s,\text{avoid}}$ increases linearly from 0.1 for GY1 to 1 for GY24, giving preferences to use gyrotrons with lowest index. We used mostly the same cost penalties as in the previous two examples, we only disabled the power change penalty ($\nu_{\Delta P} = 0$) and increased the source avoidance penalty to $\nu_{s,\text{avoid}} = 10^{-3}$, so as to clearly visualize the source prioritization effect.

In addition, we simulate the following gyrotron failures:

- GY3 and GY4 are not available between 30 and 40s.
- GY9 and GY10 are not available between 10 and 60s.
- GY21 and GY22 are not available between 50 and 80s.

The resulting allocation behavior is given in Figure 6.7. The requested power (a) is achieved by adding 6 gyrotrons (c) for each stepwise increase in the request of 5 MW. To achieve also the requested current (a), 2 gyrotrons connected to the co-current driving mirror (GY1-GY8 connect to EL14-T) are added together with 4 gyrotrons connected to the co-current driving mirrors (GY9-GY24 connect to EL14-B and EL14-M). Gyrotrons are selected according to the set preference for gyrotrons with a low index. Gyrotrons that are temporarily unavailable due to gyrotron trip and corresponding shut-down of its power supply (red in (c)) are replaced by gyrotrons with the highest available preference. If gyrotrons are available again, these are allocated as having a higher preference than the replacing gyrotrons.

Remarks on computational times

The computational time is measured for the shown examples and in addition for a number of examples with different cost penalty settings and different target requests. In all cases the optimal solution (optimal within the set tolerances) or at least a good solution is found within 1 second. This was calculated on a laptop equipped with an Intel® i7-2670QM CPU running at 2.20GHz and
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

Figure 6.7. Illustration of effective EC management in 100s ITER shot involving multiple trips in gyrotrons. Requested power (a) and current (b) is achieved by adding multiple pairs of gyrotrons (c), where GY1-GY8 are connected to a launcher mirror driving counter-current and GY9-GY24 to launcher mirrors driving co-current. Gyrotrons are selected according to their set preference: the source avoidance increases linearly from GY1 to GY24 such that gyrotrons with low index are selected. Temporarily unavailable gyrotrons (red) are replaced by others, when becoming available again, these are allocated again due to their higher preference.

In some cases the solver was not able to guarantee that the solution satisfies a single thread assigned to the solver. Solving the same problems using the Gurobi-solver [Gurobi Optimization, 2016] gave similar computational times.
Conclusions and outlook

This chapter has given a twofold contribution to integrated control in (future) tokamaks. First, multiple architectural schemes of the plasma control system were evaluated, focused on integrating multiple control tasks sharing limited available actuators. It is argued that a variety of hierarchical schemes are most promising due to their transparency and ease of implementation. We recommend for tokamaks with a small number of actuators to use pre-controller allocation (actuators are assigned prior to executing the control tasks). For tokamaks with numerous and complex actuators, we recommend to use post-controller allocation (actuators are allocated after the control tasks are executed).

The second part of this chapter presented an efficient algorithm for allocating H&CD actuators in real-time based on prioritized control task requests, actuator parameterizations and actuator availability. The actuator allocation problem was formulated in the flexible format of a Mixed-Integer Quadratic Programming problem, where the cost function reflects the desired allocation behavior and the constraints ensure that only feasible allocations are performed. The algorithm can be easily adapted to specific tokamaks or users needs as many given elements
of the desired allocation behavior can be set or cost components and constraints can be added or removed easily.

The principle of the algorithm was visualized in an example with a small system size, where different settings of the desired allocation behavior were clearly achieved in the computed allocation. Next the algorithm performance was demonstrated in representative examples involving the full proposed ITER H&CD system, where the desired allocation behavior is achieved. Simulations of a 100s ITER shot illustrated the effective compensation for actuator failure by selecting redundant actuators according to a defined actuator preference, indicating that the algorithm can also be very useful in early ITER operation where integrated control is not yet involved. ITER-size allocation problems were solved using this algorithm in about 1 second on a single core of a Intel® i7-2670QM CPU running at 2.20GHz.

The developed algorithm can be exploited in establishing integrated control in (future) tokamak operation. It can be used in simulations of the entire PCS (including supervisory layer) with multiple control tasks to analyse the impact of hardware and PCS design choices on the integrated control closed-loop. The impact of delays in hardware, but also between the layers of the hierarchical architecture could be analyzed in closed-loop simulations. Real-time implementation on existing tokamaks should experimentally prove its performance and reliability, and would require a fast MIP-solver.

The algorithm could be generalized for other resource allocation problems in tokamaks. For example the fuelling actuator allocation problem, where multiple gas valves and/or pellet injection systems must be allocated for density and impurity control, whereas the availability of these actuators may change in real-time.

6.6 Acknowledgements

The authors acknowledge fruitful discussions with Chris Rapson (IPP Garching), Marco de Baar and Hugo van den Brand (DIFFER), Erjen Lefeber (TU/e), Cristian Galperti, Tim Goodman and Olivier Sauter (SPC), Mark Henderson, Doohyun Kim and Joe Snipes (ITER). The work in this chapter has been carried out within the framework of the EUROfusion Consortium and has received funding from the Euratom research and training programme 2014-2018 under grant agreement No 633053. The views and opinions expressed herein do not necessarily reflect those of the European Commission. The work in this chapter is also supported by the Netherlands Organisation for Scientific Research (NWO) via the Innovational Research Incentives Scheme.
Appendixes

6.A Current drive parametrization

The actuator allocation algorithm requires a description of the driven current per actuator at a target location in the plasma. We define here the current drive parametrization \( \eta_{cd}^{d,t}(d, \rho_{\text{req}}^t) \) as:

\[
\eta_{cd}^{d,t}(d, \rho_{\text{req}}^t) = \eta_{cd,0}^{d}(d, \rho_{\text{req}}^t) \frac{T_e(\rho_{\text{req}}^t)}{n_e(\rho_{\text{req}}^t)},
\]

(6.14)

where \( \eta_{d,0}^{cd} \) can be negative, zero or positive value to distinguish e.g. counter-current drive, pure heating and co-current drive respectively. The values of \( \eta_{d,0}^{cd} \) given in Table 6.A.1 are approximated for EC from [Farina et al., 2014; Figini et al., 2015], while for IC and NB we choose zero.

<table>
<thead>
<tr>
<th>Delivery system</th>
<th>( d )</th>
<th>( \eta_{d,0}^{cd} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>EL14-T</td>
<td>1</td>
<td>-0.971</td>
</tr>
<tr>
<td>EL14-M,EL14-B</td>
<td>2-3</td>
<td>1.068</td>
</tr>
<tr>
<td>UL12-L to UL16-L</td>
<td>4,6,8,10</td>
<td>0.75</td>
</tr>
<tr>
<td>UL12-U to UL16-U</td>
<td>5,7,9,11</td>
<td>0.9</td>
</tr>
<tr>
<td>IC1,IC2</td>
<td>12-13</td>
<td>0</td>
</tr>
<tr>
<td>NB1,NB2</td>
<td>14-15</td>
<td>0</td>
</tr>
</tbody>
</table>

We used in our examples a temperature and density profile of ITER H-mode simulations in [van Dongen et al., 2014] to compute the current drive efficiency map based on the plasma state.

6.B Additional cost penalties

In Section 6.3.7 a number of cost penalties were introduced to define the desired allocation behavior. Some cost penalties where only briefly described and are given here in more detail.

Other penalties on changes w.r.t. present allocations

To keep the system as close as possible to the present allocation, several cost penalties were introduced in the cost function. Next to the cost penalties defined in (6.5) to (6.7), we would also like to penalize changes w.r.t. the present allocation for:
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

Connected delivery system to source during the time a switch is being performed. To avoid reallocation of the source to the previously connected delivery system while a switch to a different delivery system is being performed.

\[ J_{ac,\text{pres},\sigma}^{S2D} = \nu_{ac,\text{pres},\sigma}^{S2D} \frac{1}{S} \sum_{s=1}^{S} \sum_{d=1}^{D} \sigma_{s,d}^{S2D} (1 - \alpha_{s,d}^{S2D,\text{pres}}) \alpha_{s,d}^{S2D} \]  \hspace{1cm} (6.15)

**Connected target to delivery system.** To promote using already allocated delivery systems for a given target.

\[ J_{ac,\text{pres}}^{D2T} = \nu_{ac,\text{pres}}^{D2T} \frac{1}{D} \sum_{d=1}^{D} \sum_{t=1}^{T} (1 - \alpha_{d,t}^{D2T,\text{pres}}) \alpha_{d,t}^{D2T} \]  \hspace{1cm} (6.16)

Penalize changes w.r.t. pre-defined allocation

In some situations, it may be desirable to keep the allocation close to a pre-defined (feedforward (ff)) allocation. Therefore cost penalties are introduced on changes in the following quantities:

**Connected delivery system to source.**

\[ J_{ac,\text{ff}}^{S2D} = \nu_{ac,\text{ff}}^{S2D} \frac{1}{S} \sum_{s=1}^{S} \sum_{d=1}^{D} (1 - \alpha_{s,d}^{S2D,\text{ff}}) \alpha_{s,d}^{S2D} \]  \hspace{1cm} (6.17)

**Connected target to delivery system.**

\[ J_{ac,\text{ff}}^{D2T} = \nu_{ac,\text{ff}}^{D2T} \frac{1}{D} \sum_{d=1}^{D} \sum_{t=1}^{T} (1 - \alpha_{d,t}^{D2T,\text{ff}}) \alpha_{d,t}^{D2T} \]  \hspace{1cm} (6.18)

Penalize specific allocations of sources sharing a power supply

In case two sources are sharing the same power supply, there are specific situations that are undesired and could be avoided using the algorithm. Therefore the option is added to penalize allocations for sources \( s_i \) and \( s_j \) that are sharing the same power supply \( h \) to avoid:

**Connecting or disconnecting one of these sources to its power supply.** This might be undesired, since disconnecting a source from a power supply may take a significant amount of time (up to 3s for ITER).
This penalty requires also the knowledge if a source is presently connected to its power supply or not, provided as $\xi_{s}^{H2S}$.

$$J_{H2S,\text{connect}} = \nu_{H2S,\text{connect}} \frac{1}{S} \sum_{h=1}^{H} \sum_{s_i=1}^{S} \sum_{s_j=1}^{S} \sum_{d_i=1}^{D} M_{h,s_i}^{H2S} M_{h,s_j}^{H2S}$$

\[
\left[ \xi_{s}^{H2S} \alpha_{s_i,d}^{S2D} + \xi_{s_i}^{H2S} \alpha_{s_j,d}^{S2D} - \xi_{s_j}^{H2S} \xi_{s_i}^{H2S} \alpha_{s_i,d}^{S2D} \alpha_{s_j,d}^{S2D} - \xi_{s_j}^{H2S} \xi_{s_i}^{H2S} \alpha_{s_j,d}^{S2D} \alpha_{s_i,d}^{S2D} \right]
\] (6.19)

Connecting these sources to different delivery systems. This may be undesired, because a future change for power at one delivery system will affect the other and vice versa.

$$J_{H2S,\text{sameD}} = \nu_{H2S,\text{sameD}} \frac{1}{S} \sum_{h=1}^{H} \sum_{s_i=1}^{S} \sum_{s_j=1}^{S} \sum_{d_i=1}^{D} \sum_{d_j=1}^{D} M_{h,s_i}^{H2S} M_{h,s_j}^{H2S}$$

\[
M_{s_1,d_i}^{S2D} M_{s_2,d_j}^{S2D} \alpha_{s_i,d_i}^{S2D} \alpha_{s_j,d_j}^{S2D} \] (6.20)

where $s_i \neq s_j$ and $d_i \neq d_j$.

6.C Details regarding constraints

In Section 6.3.8 we introduced briefly the various constraints that are required to ensure that only technically feasible allocations are performed and the actuator availability is taken into account. Here we provide more details on these constraints and formulate them mathematically.

Technical constraints relating to optimization variables and imposing actuator availability

The continuous and integer optimization variables are related to each other and this relation can be specified in constraints. Therefore inequality constraints are imposed for the following purposes:

Relate source active flags and powers and impose source power constraints. The on-off flags $\alpha_{s,d}^{S2D}$ and powers $P_{s,d}^{S2D}$ for each source at a delivery system should be related as follows:

- Source on at delivery system: $\alpha_{s,d}^{S2D} = 1 \leftrightarrow P_{s,d}^{S2D,\text{min}} \leq P_{s,d}^{S2D} \leq P_{s,d}^{S2D,\text{max}}$.
- Source off at delivery system: $\alpha_{s,d}^{S2D} = 0 \leftrightarrow P_{s,d}^{S2D} = 0$

This requirement can be written as follows:
Chapter 6. Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm.

\[ P_{s,d}^{S2D} - P_{s,max}^{S2D} \alpha_{s,d}^{S2D} \leq 0, \quad \forall s \in \{1, \ldots, S\}, d \in \{1, \ldots, D\}, \]
\[ -P_{s,d}^{S2D} + P_{s,min}^{S2D} \alpha_{s,d}^{S2D} \leq 0, \quad \forall s \in \{1, \ldots, S\}, d \in \{1, \ldots, D\}. \]  \hspace{1cm} (6.21)

Note that this is a linear inequality constraint in the optimization variables \( \alpha_{s,d}^{S2D} \) and \( P_{s,d}^{S2D} \). To ensure that \( \alpha_{s,d}^{S2D} = 0 \) in case \( P_{s,d}^{S2D} = 0 \) and \( P_{s,min}^{S2D} = 0 \), we add a (small) penalty on non-idle sources \( \alpha_{s,d}^{S2D} \) in the cost function. Choosing \( P_{s,min}^{S2D} = P_{s,max}^{S2D} \) allows a source to be only on at maximum power or off.

**Relate delivery system active flags and powers.** The on-off flags and powers for each delivery system at a target are related as follows:

- Delivery system on at target: \( \alpha_{d,t}^{D2T} = 1 \) \( \leftrightarrow \) \( P_{d,t}^{D2T} > 0 \)
- Delivery system off at target: \( \alpha_{d,t}^{D2T} = 0 \) \( \leftrightarrow \) \( P_{d,t}^{D2T} = 0 \)

These constraints are formulated as follows:

\[ P_{d,t}^{D2T} - P_{d,max}^{D2T} \alpha_{d,t}^{D2T} \leq 0, \quad \forall d \in \{1, \ldots, D\}, t \in \{1, \ldots, T\}, \]
\[ -P_{d,t}^{D2T} \leq 0, \quad \forall d \in \{1, \ldots, D\}, t \in \{1, \ldots, T\} \]  \hspace{1cm} (6.22)

where \( P_{d,max}^{D2T} = P_{d,max}^{D} \), an alternative is discussed in Appendix 6.E. Again we ensure that \( \alpha_{d,t}^{D2T} = 0 \) in case \( P_{d,t}^{D2T} = 0 \) by adding a (small) penalty on non-idle delivery systems \( \alpha_{d,t}^{D2T} \) in the cost function.

**Couple source powers to delivery system power.** If a delivery system \( d \) is active at target \( t \) (\( \alpha_{d,t}^{D2T} = 1 \)), then the delivered power at this target \( P_{d,t}^{D2T} \) should equal the power delivered by the sources to this delivery system \( P_{d}^{D,alloc} \):

- Delivery system on at target: \( \alpha_{d,t}^{D2T} = 1 \) \( \rightarrow \) \( P_{d,t}^{D2T} = P_{d}^{D,alloc} \).
- Delivery system off at target: \( \alpha_{d,t}^{D2T} = 0 \) \( \rightarrow \) \( P_{d,t}^{D2T} = 0 \)

The second bullet is imposed already by (6.22). The first bullet can be imposed by the following constraint:

\[ 0 \leq P_{d}^{D,alloc} - P_{d,t}^{D2T} \leq (\sum_{t=1}^{T} \alpha_{d,t}^{D2T}) - \alpha_{d,t}^{D2T} P_{d,max}^{D}\]
\[ \forall d \in \{1, \ldots, D\}, t \in \{1, \ldots, T\} \]  \hspace{1cm} (6.23)
Single allocation per source and per delivery system

Another important constraint follows from the fact that sources and delivery systems can connect to only one destination:

Each source can be active at one delivery system.

\[ \sum_{d=1}^{D} \alpha_{s,d}^{S2D} \leq 1, \quad \forall \ s \in \{1, \ldots, S\} \]  

(6.24)

Each delivery system can be active at one target.

\[ \sum_{t=1}^{T} \alpha_{d,t}^{D2T} \leq 1, \quad \forall \ d \in \{1, \ldots, D\} \]  

(6.25)

Constraints induced by power supplies

Additional constraints are imposed to ensure that active sources sharing the same power supply have equal source power. We show now the specific case for a maximum of 2 sources sharing a single power supply, which can be readily generalized to more sources. For sources \( s_i \) and \( s_j \) that are connected to the same power supply \( h \) (hence \( M_{h,s_i}^{H2S} = 1 \) and \( M_{h,s_j}^{H2S} = 1 \)) we require:

- Both sources active \((\sum_{d=1}^{D} [\alpha_{s_i,d}^{S2D} + \alpha_{s_j,d}^{S2D}] = 2)\): must have same power \((P_{S,alloc}^{s_i} = P_{S,alloc}^{s_j})\)

- Otherwise: this constraint does not apply.

This can be imposed using the following constraint for each power supply \( h \) with its two connected sources \( s_i \) and \( s_j \):

\[-(2 - \sum_{d=1}^{D} [\alpha_{s_i,d}^{S2D} + \alpha_{s_j,d}^{S2D}])P_{h}^{H,max} \leq P_{S,alloc}^{s_i} - P_{S,alloc}^{s_j} \leq (2 - \sum_{d=1}^{D} [\alpha_{s_i,d}^{S2D} + \alpha_{s_j,d}^{S2D}])P_{h}^{H,max} \]  

(6.26)

where \( P_{h}^{H,max} = \sum_{s=1}^{S} M_{h,s}^{H2S} P_{s,max} \) is the maximum power that can be delivered by a power supply \( h \).

Constraints to exclude other physically not realizable allocation options

Some allocation options are technically not realizable for the following reasons:
No connection is possible between source and delivery system.

\[
\alpha_{d,t}^{D2T} = \begin{cases} 
0, & \text{if } M_{s,d}^{S2D} = 1 \\
1, & \text{otherwise}
\end{cases} \quad (6.27)
\]

Target is out of reach of a delivery system.

\[
\alpha_{d,t}^{D2T} = \begin{cases} 
0, & \text{if } \rho_{d}^{\min} \leq \rho_{t}^{\text{req}} \leq \rho_{d}^{\max} \\
1, & \text{otherwise}
\end{cases} \quad (6.28)
\]

Delivery system type is not allowed at target. As the function \(D_t^{\text{allow}}(D_d^{\text{type}})\) is 1 if delivery system \(d\) is allowed to be used for target \(t\), and 0 otherwise, the constraint can be written as:

\[
\alpha_{d,t}^{D2T} = \begin{cases} 
0, & \text{if } D_t^{\text{allow}}(D_d^{\text{type}}) = 1 \\
1, & \text{otherwise}
\end{cases} \quad (6.29)
\]

6.D Remarks on alternative problem formulations

Formulation for fixed connections between sources and delivery systems

A more compact optimization problem can be formulated in case all sources have a fixed connection to a delivery system. This requires only modelling a connection between the source and target and using both continuous variables \(P_{s,t}^{S2T}\) and binary variables \(P_{s,t}^{S2T}\) as the optimization variables. All relevant costs and constraints can then be rewritten in terms of these optimization variables. This reduces the total number of variables from \(2(S^2 + ST)\) in the original description, to \(2ST\) in this description. As most solvers can quickly eliminate redundant variables during their pre-solve step, the potential reduction in computational time by writing a more compact optimization problem is limited.

MILP-problem formulation

MIP-problems are solved by solving many subproblems in which the originally integer variables are fixed or treated as continuous variables. As Linear Programming (LP) problems are generally computationally cheaper to solve than QP problems (for the same problem size), a MILP formulation could be advantageous for our allocation problem.

The allocation problem can be rewritten into a MILP by using the absolute norm instead of quadratic penalties in the cost terms related to the continuous variables: e.g. using \(\|(P_{t}^{T,\text{ALLOC}} - P_{t}^{\text{REQ}})\|\) instead of \((P_{t}^{T,\text{ALLOC}} - P_{t}^{\text{REQ}})^2\). Similarly, the product of binary variables can be linearized. However, both require the
addition of auxiliary constraints and auxiliary variables, which limits the potential decrease of computational complexity by solving subproblems involving LPs instead of QPs.

We investigated both formulations and found that the allocation results are very similar for both MILP and MIQP and we noticed only minor differences related to the absolute norm instead of quadratic penalty. In terms of computational time, either MIQP or MILP can be faster, depending on the cost penalties and constraints taken into account. If cost penalties on integer variables are included, the MILP-formulation requires often less computational time than the MIQP-formulation. Even though MILP could thus be faster, we have chosen in this chapter a MIQP-formulation, which is more flexible and transparent due the absence of the auxiliary variables and constraints.

6.E Practical issues

Here we discuss some practical and more technical issues.

Normalization factors

The normalization factors $P_t^{\text{req, norm}}$ in (6.5), $I_t^{\text{req, norm}}$ in (6.6) and $P_s^{S, \text{pres, norm}}$ (6.8) are chosen such that all entries are normalized individually, but also to avoid dividing by zero (or numbers close to zero):

$$
P_t^{\text{req, norm}} = \max(P_{\text{norm, min}}, P_t^{\text{req}}) \quad (6.30)
$$

$$
I_t^{\text{req, norm}} = \max(I_{\text{norm, min}}, I_t^{\text{req}}) \quad (6.31)
$$

$$
P_s^{S, \text{pres, norm}} = \max(P_{\text{norm, min}}, P_s^{\text{max}}) \quad (6.32)
$$

where $P_{\text{norm, min}}$ and $I_{\text{norm, min}}$ should take values corresponding to the considered actuator system, we choose in the ITER examples $P_{\text{norm, min}} = 0.1$ and $I_{\text{norm, min}} = 0.001$.

Tighter bounds on optimization variables

To find the solution of a MIQP-problem, MIP-solvers solve many subproblems in which some of the integer variables are treated as continuous variables. It is beneficial to construct the MIQP such that during solving the subproblems these integer variables will naturally appear to be either close to zero or close to one. This extensively helps the MIP-solver to quickly find the best integer feasible allocation option.

A tighter bound on $P_{D2T}^{D, t}$ in (6.22) can help to realize this for the active flags between delivery systems and targets ($\alpha_{D2T}^{D, t}$). A tighter bound can be formulated by including the target power request $P_t^{\text{req}}$ in computing the bound:

$$
P_{D2T, \text{max}, t} = \min((1 + f^{P, \text{add}})P_t^{\text{req}}, P_d^{\text{D, max}}). \quad (6.33)
$$
Choosing $f^{P,\text{add}} > 0$ allows to allocate more power than requested at a target, which may be helpful to get a better current matching. We choose $f^{P,\text{add}} = 0.1$ in the examples.
Chapter 7

Conclusions, discussion and recommendations

This thesis has addressed two important problems in the control and operations of tokamaks. In Part I of this thesis, three model predictive controllers have been designed to control key plasma parameters in tokamak plasmas that can deal effectively with actuator and operational limits. The successful performance of these controllers has been demonstrated in simulations and experiments. The second part of this thesis evaluated architectures and developed an algorithm to systematically integrate multiple plasma control tasks that share a set of actuators.

In this chapter we will first for each part conclude on the research conducted by summarizing the contributions and reflecting on the research objectives. Next we will give a discussion and recommendation for improving, extending and exploiting the developed methods in (future) tokamaks.

7.1 Conclusions

7.1.1 Model predictive profile control

In Chapter 2 an MPC controller was designed to track a predefined reference safety factor profile evolution using 5 actuators in the presence of model mismatches and disturbances, while handling actuator and operational limits. The used prediction model is based on multiple linearized models around the simulated nominal evolution of the plasma profiles during the current ramp-up phase to account for the nonlinearities in this phase. Techniques such as parameterization of the future input sequence were used to construct compact Quadratic
Programming problems that could be solved within 8ms. The potential of the MPC controller was demonstrated in closed-loop simulations with parameters representative of the ITER tokamak, where successful tracking of the $q$-profile was obtained in the presence of under- or overestimated heat transport while satisfying the operational limit $q(\rho) > 1$. It was also shown that the controller can provide warnings of expected constraint violations in the near future if constraints become too stringent.

Chapter 3 presented several extensions to the previous designed MPC controller to improve the steady state performance and extend its capabilities. The MPC controller can now track real-time varying references that differ from the nominal plasma profile evolution used to provide the linearized models and includes state disturbance estimation to improve steady state tracking and constraint handling. It controls not only the $q$-profile, but also the plasma stored energy, while satisfying an important limit on the normalized plasma pressure. Analysis of the controller performance in closed-loop simulations with parameters representative of high performance plasmas in the ASDEX Upgrade tokamak using 4 actuators demonstrated successful tracking. Changes in references, actuator constraints and state constraints are effectively handled, demonstrating how a predictive controller can outperform other control methods in terms of effective input-constraint handling.

In Chapter 4 a profile controller test environment is presented and the performance of an MPC controller in simulations and experiments in the TCV tokamak is demonstrated. The presented controller development and implementation environment facilitates thorough off-line preparation and experimental implementation of profile controllers in the TCV tokamak. The designed input-constrained MPC controller uses a single linearized model complemented with disturbance estimation to track $q$-profile and plasma pressure references. The performance of the controller is analyzed in both simulations and experiments using 3 actuators where successful tracking of the inverse safety factor profile as well as the plasma pressure is demonstrated under uncertain plasma conditions and disturbances. The controller exploits the knowledge of the time-varying actuator limits in the actuator input calculation such that fast transitions between targets are achieved without overshoot, demonstrating how a predictive controller can outperform other control methods in terms of effective input-constraint handling. The maximum required computational time of the MPC controller using 3 actuators was 300 $\mu$s, well within the available 700 $\mu$s.

Returning to the first research objective:

*Design a controller that ensures that desired plasma profile and parameters are achieved in the presence of disturbances while time-varying actuator and operational limits are satisfied.*

The contributions in Chapters 2-4 have shown the successful performance and distinguishing capabilities of MPC to track a variety of plasma profiles and
parameters while effectively dealing with time-varying actuator and operational limits. This encourages to further exploit the capabilities of a model predictive profile controller in experiments as we will discuss next and provide recommendations for further research.

7.1.2 Actuator management for integrated control

In Chapter 5 simultaneous control of the plasma $q$-profile and neoclassical tearing modes (NTMs) was demonstrated using a shared amount of EC-power. A PCS architecture is proposed that allocates the available EC-power to the profile and NTM-controller before execution of these control tasks. Closed-loop control simulations, with parameters that are representative of the ITER tokamak, demonstrate that the proposed PCS design enables effective response to the events of appearing NTMs by suppressing these using an NTM-controller, while an MPC profile controller (as designed in Chapter 3) simultaneously maintains the $q$-profile close to the reference and within the operational constraint $q > 1$.

In Chapter 6, first multiple architectures were evaluated to integrate multiple control tasks sharing limited available actuators. It is argued that a variety of hierarchical schemes are most promising due to their transparency and ease of implementation. Recommendations are given to choose a specific hierarchical scheme, dependent on the scale and complexity of the actuator system and the number of control tasks involved.

The second part of Chapter 6 proposed an efficient actuator allocation algorithm for allocating H&CD actuators in real-time based on prioritized control task requests and actuator availability. The actuator allocation problem is formulated as a Mixed-Integer Programming problem format, where the desired allocation behavior is defined in a cost function and constraints ensure that only feasible allocations are performed. The performance of the actuator allocation algorithm is demonstrated in representative examples involving the full proposed ITER H&CD system, where the desired allocation behavior is achieved. A 100s ITER shot simulation illustrated the effective compensation for actuator failure by selecting redundant actuators according to a defined actuator preference, indicating that the algorithm can also be very useful in early ITER operation where integrated control is not yet required. ITER-size allocation problems can be readily solved using this algorithm in about 1 second on a single core of an Intel® i7-2670QM CPU running at 2.20GHz. The algorithm can be easily adapted to the needs of a specific tokamak or desired actuator behaviour, as many given elements of the desired allocation behavior can be set or cost components and constraints can be added or removed easily.

The second research objective of this thesis is:

*Evaluate PCS architectures to integrate multiple control tasks requiring the same actuators.*
A PCS architecture to integrate two control tasks was evaluated in Chapter 5 in simulations. A broader evaluation of multiple PCS architectures to integrate control tasks using actuator allocation was conducted in Chapter 6. Both evaluations have resulted in recommendations to choose a specific architecture depending on a tokamak’s needs. These will provide guidance to explore integrated control in experiments on various existing tokamaks as well as in simulations for future tokamaks.

The third research objective states:

*Develop a real-time actuator allocation algorithm that can deal with large and complex H&CD systems such as proposed for ITER.*

The efficient algorithm developed in the second part of Chapter 6 can successfully allocate actuators in real-time according to a defined desired allocation behavior for large and complex H&CD systems such as proposed for ITER. The chosen problem formulation can be easily adapted to a tokamak’s specific needs, allowing exploitation of this actuator allocation algorithm in integrated control experiments with shared actuators on existing tokamaks as well as in simulation for future tokamaks.

### 7.2 Discussion and recommendations

#### 7.2.1 Model predictive profile control

The designed MPC controllers have demonstrated the expected performance in both tracking and constraint handling capabilities. The application range of the model predictive profile controller can be further extended in several ways.

Adapting the controller model to the present plasma state would make the controller independent of pre-defined linearizations. The designed controllers as presented in this thesis depend on linearized models that are chosen offline for a dedicated operating range. It is expected that these models will not give the desired performance when references are changed during experiments to operating points outside the validity range of the linearized models. In addition it might be difficult to obtain a sufficient controller performance in regions where nonlinear couplings between profiles are dominant, such as in realizing reversed shear $q$-profiles. Therefore it would be beneficial if the controller model would adapt to the present plasma state, which can be achieved by using real-time linearized models provided by the RAPTOR-observer as used in Chapter 4. Fully nonlinear MPC, involving a nonlinear prediction model [Grüne and Pannek, 2017], is probably not computationally feasible on currently operational devices, but can be considered for e.g. ITER with its slower time scales.

Including the actuator dynamics and all delays involved in the closed-loop is important to achieve a satisfactory controller performance. It was observed in one of the experiments presented in Chapter 4 that for the plasma current
actuator a large delay was involved and the actuator had a long response time. This was not included in the controller model and led to unstable controller behavior. This would not have occurred if these actuator dynamics and delays were included in the controller model.

Ensuring that the tracked $q$-profiles are indeed the true $q$-profile of the plasma is essential for appropriately using the MPC controller. This requires the addition of internal current profile diagnostics in the plasma profile state reconstruction that were not available in the experiments presented in this thesis. The Motional Stark Effect (MSE) diagnostic can provide such measurements and has been used in profile experiments in e.g. the DIII-D tokamak [Barton et al., 2014a; Boyer et al., 2014].

Demonstrating the handling of operational limits (e.g. normalized plasma pressure) in experiments on currently operational tokamaks is important for developing the control capability to ensure reliable high-performance operation close to these limits in (future) large tokamaks such as DEMO [Biel et al., 2015]. In this thesis handling of operational limits was only shown in simulations (Chapters 2 and 3) and not in experiments.

Extending the nonlinear physics models in RAPTOR, underlying the linearized controller models, would broaden the successful application range of the MPC controller. Other important physics couplings than those modelled presently can than be taken into account. One particular example is the coupling between the current profile and the electron density profile as observed in experiments in Chapter 4 that is presently not included in RAPTOR. Another example is including the ion temperature and density evolution such that the effect of actuators that also affect the plasma ions (e.g. NBI) can be more appropriately taken into account in the controller.

Generalizing the MPC controller such that any plasma quantity that is a function of the plasma state can be controlled would ease the use of the controller for different kinds of experiments. This would allow to not only control profiles, but also their gradients or other derived quantities. Also, other operational limits can then be taken into account and other actuators can be used.

As such, model predictive profile control could develop further into a control tool that facilitates physics studies in a controlled environment. For future tokamaks like ITER and the proposed demonstration power plant DEMO it can be used to ensure reliable high performance operation within a proven stable plasma parameter space.

7.2.2 Actuator management for integrated control

The recommendations given in Chapter 6 can be used to choose the best architecture for existing and future tokamaks to establish integrated control routines with shared actuators. Validation of the provided recommendations by evaluating these in simulations for various tokamaks would provide an even stronger
basis to choose a specific architecture for a tokamaks needs.

The algorithm proposed in Chapter 6 can be exploited in integrated control simulations with shared actuators for various purposes. It can be used in simulations of the entire PCS of future tokamaks with multiple control tasks, which would be eased if the algorithm is embedded in the recently developed Plasma Control System Simulation Platform (PCSSP) [Walker et al., 2015a]. This would allow thorough evaluation of the PCS design choices such as possible delays between the layers of the hierarchical architecture.

Including a representative model of the ITER EC actuator hardware dynamics and delays in the PCSSP may then also enable the comprehensive analysis of the impact of hardware choices on the integrated control closed-loop. An important example to analyse is the impact of delays induced by mechanical switches in the transmission lines in the ITER EC system on the required time to suppress quickly growing NTMs.

A thorough testing and tuning of the actuator allocation algorithm under various ITER operational conditions could be performed using PCSSP. One of the test cases could be the scenario of simultaneous control of plasma profiles and NTMs as simulated in Chapter 5.

Real-time implementation of the algorithm on existing tokamaks allows to experimentally prove its performance and reliability. The developed algorithm can be exploited in establishing integrated control routines that are required for future tokamaks. The algorithm can then be adapted to a tokamaks needs and a fast MIP-solver is required that can solve the actuator allocation problem in real-time.

The actuator allocation algorithm can deal with all H&CD actuators, although the present available cost penalties and used actuator parameterizations are best suited for EC actuators. Effective allocation of Ion Cyclotron and Neutral Beam Injection resources may require the addition of cost penalties and actuator parameterizations that take the characteristics of these actuator systems appropriately into account.

7.2.3 Extensions to other applications in controlled fusion

Model predictive control will be useful in any application that involves strict actuator constraints or operational constraints. Also in controlled nuclear fusion many other processes than plasma transport processes could benefit from an MPC approach.

Shape control may benefit from an MPC approach as the poloidal field coils used to regulate the plasma shape possess strict actuator limits. Also, the plasma should stay at a safe distance from the wall also during transitions to other shapes, which are operational limits. The dynamics have strong coupling between the inputs and controlled variables. MPC would be an ideal candidate with its straightforward handling of MIMO systems and actuator and operational
limits. Very recently MPC is investigated for shape control at ITER [Gerkšič et al., 2016] in simulations, indicating that it is at least computationally feasible to realize MPC-based shape control on large tokamaks with slow time scales. It may be computationally challenging to implement shape control on currently operational tokamaks (e.g. TCV) due to fast time scale of the vertical instability. Applying a command-governor strategy by first pre-stabilizing the shape dynamics and then modifying the references to satisfy actuator limits and process limits (as applied to shape control simulations in [Mattei et al., 2013]) may be a viable solution to relax the computational time constraints.

Power exhaust control may be another interesting application for MPC, where impurities are injected into the edge plasma to radiate power over a large region of the plasma facing wall to regulate the intense power exhaust that may otherwise damage the wall if concentrated in a small region. This may have as a side effect that the high core plasma confinement may be lost suddenly [Biel et al., 2015]. Power exhaust control has been demonstrated at multiple devices [Kallenbach et al., 2013; Kolemen et al., 2015], but may benefit from an MPC approach as satisfaction of operational limits is important here.

Also other allocation problems exist in tokamaks that may benefit from the Mixed-Integer Programming approach applied in this thesis to the H&CD actuator allocation problem. The developed algorithm could therefore be generalized for other resource allocation problems in tokamaks. For example the fuelling actuator allocation problem, where multiple gas valves and/or pellet injection systems must be allocated for density and impurity control, whereas the availability of these actuators may change in real-time.

This thesis has contributed to the reliable high performance operation of (future) tokamaks within operational limits, where multiple control tasks share a set of actuators, by developing control algorithms based on state-of-the-art control engineering practices.


BIBLIOGRAPHY


O. Sauter, C. Angioni, and Y. R. Lin-Liu. Neoclassical conductivity and bootstrap current


A. A. Teplukhina, O. Sauter, F. Felici, A. Merle, D. Kim, the TCV Team, the ASDEX-Upgrade Team, and the EUROfusion MST1 Team. Simulation of profile evolution from ramp-up to ramp-down and optimization of tokamak plasma termination with the raptor code. To be submitted to Plasma Physics and Controlled Fusion, 2017.


J. van Dongen, F. Felici, G. M. D. Hogeweij, P. Geelen, and E. Maljaars. Numerical optimiza-


<table>
<thead>
<tr>
<th>Glossary Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ASDEX-Upgrade</strong></td>
<td>Axially Symmetric Divertor EXperiment Upgrade, tokamak at Max-Planck-Institut für Plasmaphysik, Garching-bei-München, Germany</td>
</tr>
<tr>
<td><strong>CRONOS</strong></td>
<td>Suite of codes for integrated tokamak modelling</td>
</tr>
<tr>
<td><strong>CPU</strong></td>
<td>Central processing unit</td>
</tr>
<tr>
<td><strong>CV</strong></td>
<td>Controlled variable</td>
</tr>
<tr>
<td><strong>CVP</strong></td>
<td>Control vector parameterization</td>
</tr>
<tr>
<td><strong>DEMO</strong></td>
<td>DEMOnstration power station, tokamak envisioned to demonstrate fusion power production</td>
</tr>
<tr>
<td><strong>EC</strong></td>
<td>Electron cyclotron</td>
</tr>
<tr>
<td><strong>EC(R)H</strong></td>
<td>Electron cyclotron (resonance) heating</td>
</tr>
<tr>
<td><strong>ECCD</strong></td>
<td>Electron cyclotron current drive</td>
</tr>
<tr>
<td><strong>EKF</strong></td>
<td>Extended Kalman filter</td>
</tr>
<tr>
<td><strong>FIR</strong></td>
<td>Far-infrared</td>
</tr>
<tr>
<td><strong>GY</strong></td>
<td>Gyrotron</td>
</tr>
<tr>
<td><strong>H&amp;CD</strong></td>
<td>Heating and current drive</td>
</tr>
<tr>
<td><strong>IC</strong></td>
<td>Ion cyclotron</td>
</tr>
<tr>
<td><strong>ICRH</strong></td>
<td>Ion cyclotron resonance heating</td>
</tr>
<tr>
<td><strong>ITB</strong></td>
<td>Internal transport barrier</td>
</tr>
<tr>
<td><strong>ITER</strong></td>
<td>International Thermonuclear Experimental Reactor, tokamak presently under construction in St. Paul-lez-Durance, France</td>
</tr>
<tr>
<td><strong>JT-60SA</strong></td>
<td>Japan Torus-60 Super Advanced: tokamak at JAEA Naka Fusion Institute, Japan</td>
</tr>
<tr>
<td><strong>LCFS</strong></td>
<td>Last closed flux surface</td>
</tr>
<tr>
<td><strong>LIUQE</strong></td>
<td>Equilibrium reconstruction code used at TCV</td>
</tr>
<tr>
<td><strong>LP</strong></td>
<td>Linear programming</td>
</tr>
<tr>
<td><strong>LTI</strong></td>
<td>Linear time-invariant</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LTV</td>
<td>Linear time-varying</td>
</tr>
<tr>
<td>MHD</td>
<td>Magnetic-hydrodynamics</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multiple-input-multiple-output</td>
</tr>
<tr>
<td>MIP</td>
<td>Mixed-integer quadratic programming</td>
</tr>
<tr>
<td>MILP</td>
<td>Mixed-integer linear programming</td>
</tr>
<tr>
<td>MIQP</td>
<td>Mixed-integer quadratic programming</td>
</tr>
<tr>
<td>MPC</td>
<td>Model predictive control</td>
</tr>
<tr>
<td>MSE</td>
<td>Motional Stark effect</td>
</tr>
<tr>
<td>NBI</td>
<td>Neutral beam injection</td>
</tr>
<tr>
<td>NP</td>
<td>Non-deterministic polynomial-time</td>
</tr>
<tr>
<td>NTM</td>
<td>Neoclassical tearing modes</td>
</tr>
<tr>
<td>PCS</td>
<td>Plasma control system</td>
</tr>
<tr>
<td>PCSSP</td>
<td>Plasma control system simulation platform</td>
</tr>
<tr>
<td>PDE</td>
<td>Partial differential equation</td>
</tr>
<tr>
<td>QP</td>
<td>Quadratic programming</td>
</tr>
<tr>
<td>RAPDENS</td>
<td>RAPid DEnsity evolution model</td>
</tr>
<tr>
<td>RAPTOR</td>
<td>RAPid Plasma Transport simulatOR</td>
</tr>
<tr>
<td>RF</td>
<td>Radio-frequency</td>
</tr>
<tr>
<td>RT</td>
<td>Real-time</td>
</tr>
<tr>
<td>TORAY-GA</td>
<td>Ray-tracing code developed at General Atomics</td>
</tr>
<tr>
<td>TCV</td>
<td>Tokamak ´ a Configuration Variable: tokamak at the Swiss Plasma Center, Lausanne, Switzerland</td>
</tr>
<tr>
<td>WEST</td>
<td>Tungsten (chemical symbol 'W') Environment in Steady-state Tokamak: tokamak at CEA, Cadarache, France</td>
</tr>
<tr>
<td>XTe</td>
<td>X-ray electron temperature measurement</td>
</tr>
</tbody>
</table>
Dankwoord

Een promotietraject is net een bergbeklimming. Inspannende klimmetjes, verwondering over al het grote en kleine moois onderweg, bomen waardoor je soms het bos niet meer ziet, de kans om elk moment voor een nieuwe verrassing te staan, bij de pakken neer gaan zitten om vervolgens weer vol goede moed door te gaan, het reikhalzend uitzien naar de top. Het hoort er allemaal bij. Graag wil ik alle mensen bedanken die mij hebben gesteund tijdens mijn bergtocht en een bijdrage hebben geleverd aan mijn onderzoek.

Marco, bedankt voor de kans en het vertrouwen om een promotietraject te starten. Vooral aan het begin ben je een grote motivator en inspirator voor mij geweest. Ik waardeer je nuchtere kijk op dingen en het altijd weer zien van perspectief. Dankjewel ook voor het regelen van alle formele zaken rondom mijn promotie.

Maarten, dank voor de support die je mij al sinds het begin van mijn masteropleiding hebt gegeven. Ik waardeer je persoonlijke aandacht op de momenten dat ik dat nodig had, je was dan altijd beschikbaar. Dankjewel voor deze geweldige promotieplaats die je mij hebt gegeven binnen de Control Systems Technology groep.

Federico, wat heb ik het getroffen met jou als co-promotor. Ik heb onzettend veel geleerd van jouw indrukwekkende kennis en kunde op het gebied van fusie, control, problemen oplossen en teksten schrijven. Wat heb je veel tijd genomen om naar mij te luisteren, mij aan te moedigen en steeds weer de puntjes op de i te zetten. Bedankt voor je vertrouwen in mij en voor al je geduld, alleen al bij het eindeloos verbeteren van therefor in therefore. Mede door jou ben ik als persoon gegroeid en kan ik terugkijken op een onzettend leerzame en mooie periode.

I also would like to thank the other members of the committee: Siep Weiland, Massimiliano Mattei, Bart de Schutter and Joseph Snipes. Thanks for your valuable feedback on my thesis and for taking part in my defense committee.

I would like to thank the TCV team for allowing me to do the profile control
experiments on this fantastic machine. Thanks for the pleasant collaboration. I would like to thank Olivier Sauter who has greatly contributed to this thesis. Your outstanding knowledge of plasma physics, endless flow of suggestions and great enthusiasm have helped me to give this thesis a strong physics basis. Thank you also for introducing me to real Swiss food! Cristian Galperti, your great experience with real-time control systems has been essential to perform the profile control experiments and I enjoyed working together. Thanks also to Anna, Bojan, Doohyun, Francesco, Himank, Mengdi and Tran for the fruitful collaboration and the good time during my stays in Lausanne. Chris Rapson, I am very grateful for your helpful ideas in the many (e-mail) discussions we had on actuator management.

Verschillende studenten heb ik mogen begeleiden: Bas, Gijs, Stijn, Bas en Jasper. Dank jullie wel voor jullie inbreng. Jullie onderzoeken hebben mij geholpen om de juiste kennis te verzamelen en zijn op verschillende manieren in dit proefschrift terecht gekomen.

Dank ook aan alle collega’s op -1. Ik kijk terug op een gezellige tijd met jullie. Een speciaal dankjewel voor het control-voor-fusie groepje in Eindhoven door de jaren heen: Artur, Gilles, Hugo, Matthijs, Menno, Thomas en Timo. Bedankt voor de leuke tijd die we met elkaar hebben gehad en de goede en leerzame gesprekken. Thomas en Hugo, het was fijn om jullie als kamergenoten te mogen hebben. Dennis, Thijs and Jiquan, thank you for the good time we had when we shared our office. Sorry voor de rommel op mijn bureau, Dennis. Je foto heeft mij geholpen om mijn opruimvaardigheden te verbeteren. Petra en Geertje, dank jullie wel voor de gezelligheid en al jullie hulp bij de administratieve zaken.

Vrienden, heel erg bedankt voor al jullie meelevens en het luisteren naar mijn donutverhalen. Ik denk hierbij aan de het-was-weer-fantastisch-groep, de vriendengroep uit Aagtekerke, de EQ-oudjes en de Bijbelkring in Dordrecht. Hans, Janetta en mevrouw Roubos, bedankt voor de warme plek die ik in Eindhoven bij jullie vond tijdens de begintijd van mijn promotie.

Pa en ma, enorm bedankt voor alles wat jullie de afgelopen jaren voor mij hebben betekend, en al mijn hele leven. Papa en mama Bassa, jullie oprechte belangstelling in mijn onderzoek heb ik erg gewaardeerd, maar vooral ook de onderlinge band. Broers, schoonzussen en zwagers, dank jullie wel voor alle belangstelling en steun in de afgelopen jaren!

Lieve Hanneke, mijn trouwe levensmaatje, dank je wel voor al jouw liefde, wijdheid en geduld. Je was er altijd voor mij! Ik hoop dat we samen nog veel bergen zullen beklimmen! Lieve Joël, wat een kostbaar geschenk ben jij! Wat heb je papa al vaak blij gemaakt, juist ook op die momenten dat ik door de bomen het bos even niet meer zag. Door jou word ik er ook steeds weer aan herinnerd Wie de meeste dank toekomt: Soli Deo Gloria.
List of publications

Peer-reviewed journal articles


Articles in peer-reviewed conference proceedings

• E. Maljaars, F. Felici, M.R. de Baar and M. Steinbuch. Model predictive control of the current density distribution and stored energy in tokamak fusion experiments using trajectory linearizations. IFAC-PapersOnLine 48(23), 314321. 5th IFAC Conference on Nonlinear Model Predictive Control NMPC, Seville, Spain, 2015. Chapter 3 of this thesis.
List of publications

Articles in not peer-reviewed conference proceedings


Other presentations


Bert Maljaars was born on November 10th 1986 in Vlissingen, the Netherlands. He finished secondary school in 2005 at the Calvijn College in Goes. Afterwards he studied Mechanical Engineering at the Eindhoven University of Technology, the Netherlands, where he received his Master degree in 2012 (cum laude). As part of this program he performed an internship at Industrial Systems and Control Ltd. in Glasgow, United Kingdom, where he worked on car engine control. His master thesis was entitled ‘Model predictive control of safety factor profile in ITER hybrid mode’. He also completed the Certificate Program on Technology Management during his Master program.

In 2012 he started his PhD in the Control Systems Technology group with a project under the supervision of Marco de Baar, Maarten Steinbuch and Federico Felici. His project focussed on the model predictive control of key plasma parameters in tokamaks and the systematic integration of multiple control tasks using actuator management in these nuclear fusion devices. As part of his project, experiments were performed on the TCV tokamak at the Swiss Plasma Center in Lausanne, Switzerland. The main results of his PhD work are presented in this thesis.