Actuator allocation for integrated control in tokamaks

Fusion Engineering and Design 122 (2017) 94–112

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes

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

E. Maljaars, F. Felici*

Eindhoven University of Technology, Mechanical Engineering, Control Systems Technology, PO Box 513, 5600MB Eindhoven, The Netherlands

ARTICLE INFO

Keywords:
Integrated control
Actuator allocation
Architectural design
Mixed-integer programming
ITER

ABSTRACT

Plasma control systems (PCS) in tokamaks need to fulfill a number of control tasks to achieve the desired physics goals. In present-day devices, actuators are usually assigned to a single control task. However, in future tokamaks, only a limited set of actuators is available for multiple control tasks at the same time. The priority to perform specific control tasks may change in real-time due to unforeseen plasma events and actuator availability may change due to failure. This requires the real-time allocation of available actuators to realize the requests by the control tasks, also known as actuator management.

In this paper, we analyze possible architectures to interface the control tasks with the allocation of actuators inside the PCS. Additionally, we present an efficient actuator allocation algorithm for Heating and Current Drive (H&CD) actuators. The actuator allocation problem is formulated as a Mixed-Integer Quadratic Programming optimization problem, allowing to quickly search for the best allocation option without the need to compute all allocation options. The algorithms performance is demonstrated in examples involving the full proposed ITER H&CD system, where the desired allocation behavior is successfully achieved. This work contributes to establishing integrated control routines with shared actuators on existing and future tokamaks.

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.

A control task typically compares present plasma quantities with their references and calculates commands to actuators such that references values are achieved. In present-day devices, actuators are usually assigned to a single control task for an entire experiment, e.g. to density control, plasma beta control or the control of Neoclassical Tearing Modes (NTMs). Performing multiple control tasks at the same time is sometimes done in tokamaks and is known as integrated control [1,2]. 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 [3–8]. Also, the priority to perform a specific control task may vary in real-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 importance and complexity of this integrated control problem is best illustrated for the ITER Electron Cyclotron (EC) Heating and Current Drive (H&CD) system. This system with its 24 gyrotrons and 11 steerable mirrors needs to be used by at least 4 control tasks with time-varying priorities in the flat-top alone: impurity control, sawtooth control, NTM control and profile control [9].

The PCS architecture defines the role of its components (such as a number of control tasks) and the interfaces between these components. PCS architectural designs are recently presented in literature for the tokamaks ASDEX Upgrade [10,11], WEST [12,13] and ITER [6,7]. Although different in details, these papers represent a coherent approach to the PCS architecture.

Recently an actuator allocation algorithm was developed and successfully implemented for the ECRH system at ASDEX Upgrade [14,15]. 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, as noted in [14,15], 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 work therefore provides an algorithm that is inspired by [14,15], but which can be

* Corresponding author.
E-mail addresses: e.maljaars@tue.nl (E. Maljaars), f.felici@tue.nl (F. Felici).

https://doi.org/10.1016/j.fusengdes.2017.09.004
Received 2 May 2017; Received in revised form 4 August 2017; Accepted 2 September 2017
Available online 03 October 2017

executed sufficiently rapidly for real-time implementation on e.g. ITER.

In this work we first evaluate possible architectures to interface the 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 work is organized as follows. In Section 2 possible PCS architectures for integrated control are evaluated. The MIQP-formulation of the actuator allocation problem is introduced in Section 3. Section 4 presents the performance of the actuator allocation algorithm in examples. Finally, conclusions and outlook are given in Section 5.

2. Overview and brief evaluation of architectures for integrated control

2.1. Introduction to PCS schemes

Tokamak plasmas need to be actively monitored and controlled by a plasma control system (PCS) to ensure that the desired plasma performance is achieved while operational and machine limits are satisfied. The architectural design of a PCS defines its components and the interfaces between these components. Recently, a number of PCS architectural designs were presented in literature [6,7,10–13]. These PCS architectural designs can be summarized as given in Fig. 1. Herein we can identify, next to the tokamak itself with its actuators and diagnostics, the following components:

Diagnostic signal processing and plasma state reconstruction. Here the signals of multiple diagnostics are processed and integrated (possibly by including model-based predictions) into an estimate of the present plasma state. In future tokamaks the PCS is expected to have a clear separation between plasma state reconstruction and control and supervision tasks [3].

Supervisory layer. In this supervisory layer (green) the important central decisions are taken to meet pre-defined plasma scenario objectives and handle events based on provided information about the plant and plasma state [6,10–13]. The supervisor activates control tasks and sets their parameters and references, possibly by switching between pre-defined segments, where each segment is applicable for a range of plasma and hardware states and contains a set of active control tasks with corresponding parameters and references. Also priorities of control tasks can be set by the supervisor in response to detected events [3]. More information on event handling strategies in the PCS can be found in [16].

Actuator control systems. These low-level 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.

Control tasks. A control task computes what actions are necessary such that the control task objective will be achieved. The control task objective maybe specified as minimizing the difference between a reference and present estimated value of a controlled variable. Depending on the interface between controllers and actuators (discussed in more detail in Section 2.4), the action may be specified directly as a command to an actuator, or as a more generic request for actuation.

Actuator allocation. The purpose of actuator allocation is to assign

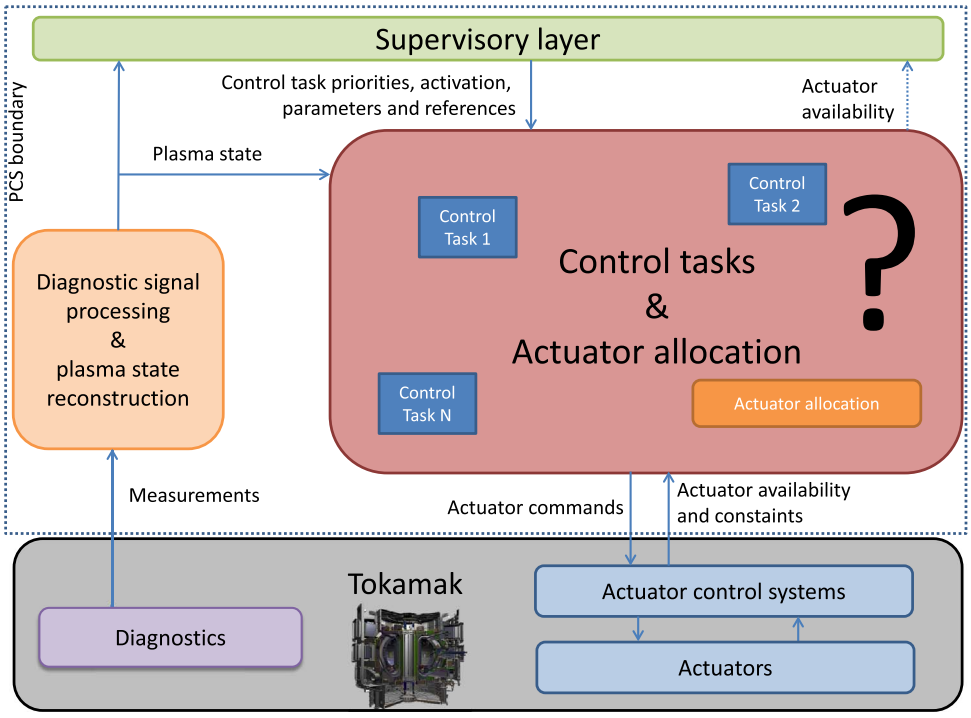

Fig. 1. General scheme of a plasma control system (PCS) with multiple control tasks.

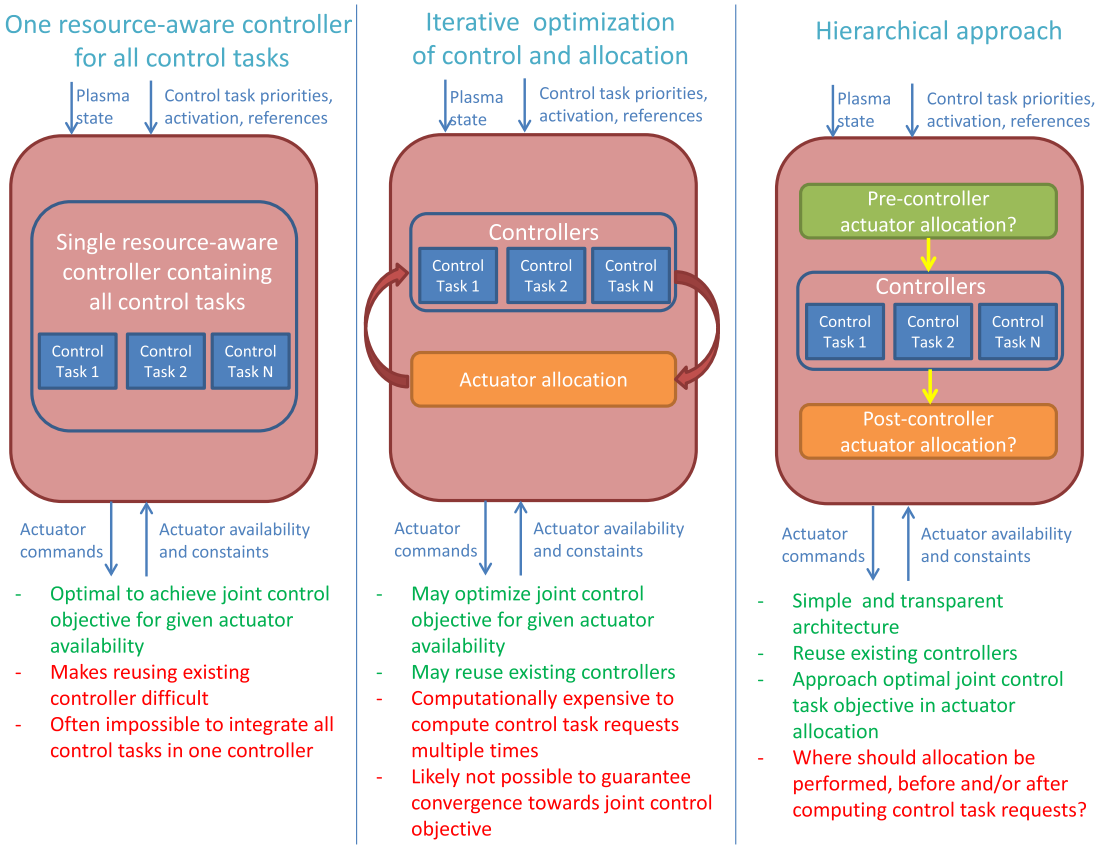

Fig. 2. Comparison of three possible architectures to interface control tasks and actuator allocation: all control tasks embedded in a single controller (left), an iterative scheme (left) and a hierarchical scheme (right). In the hierarchical scheme, the actuator allocation can be done before (pre-controller actuator allocation) and/or after (post-controller actuator allocation) executing the control tasks.

actuators such that the (prioritized) requests of the control tasks are realized with the available actuators.

Inside the PCS, the prioritized control tasks and actuator allocation (given in the red box in Fig. 1) need to be interfaced with each other. However, multiple architectures are possible here and presently no general 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.

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 performed simultaneously 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.

2.2. Remarks on cross-coupling between control tasks.

From the previous part it may seem that any combination of control tasks can be performed at the same time. However, individual control tasks may have a strong effect on each other as the underlying physical processes are coupled. This is known as cross-coupling between control tasks and potentially could lead to control stability loss, i.e. to undesired responses of the system. In case that control tasks exhibit cross-coupling, 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 [17,18].

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

In the remainder of this work we assume that these actions have been taken in case of cross-coupling between control tasks, also implying that actuator allocation itself does not need to deal with possible cross-coupling as this is solved at another level.

2.3. Interfacing control tasks and actuator allocation

We will now compare three architectures that interface control tasks and allocation of resources to achieve integrated control with shared actuators. We evaluate the architectures on the following aspects:

Joint control objective. Can a joint objective of the control tasks be optimized making optimal use of the available actuators? A joint objective may be seen as a weighted combination of the respective control task objectives, e.g. weighted according to their assigned

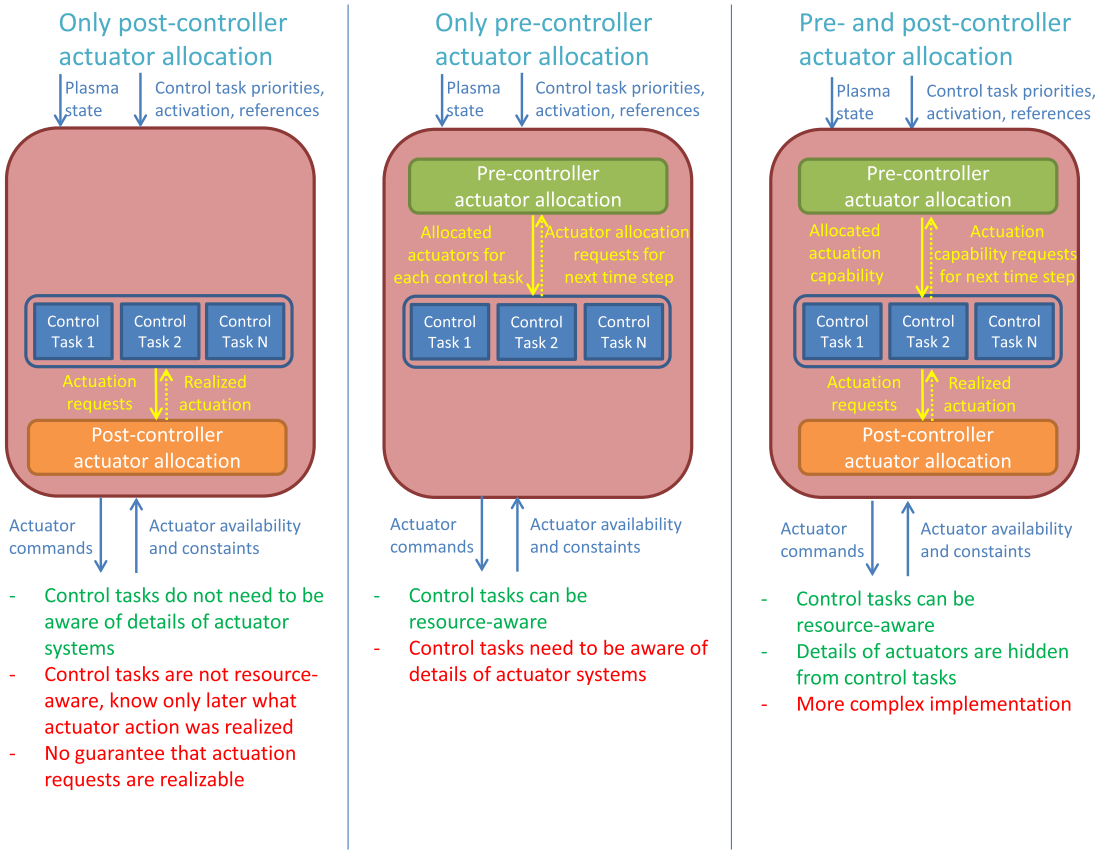

Fig. 3. Three options for places of actuator allocation in a hierarchical scheme: post-controller allocation (left), pre-controller allocation (middle) or both pre- and post-controller allocation (right).

priorities. This is important, since with scarce available actuators, inherently a tradeoff needs to be made in assigning the available actuators to the various prioritized control tasks.

Resource-awareness. Do control tasks know which (actuator) resources are at their disposal when calculating the requests (i.e. are they resource-aware)? A resource-aware controller can make optimal use of its assigned resources.

Controller reusability. Is it possible to reuse existing control task algorithms, i.e. controllers that have been developed and are implemented on presently operational tokamaks and typically involve much expert knowledge?

Implementation complexity. Is the architecture easy and transparent to implement?

Three architectures will be compared: all control tasks embedded in a single controller, an iterative scheme and a hierarchical scheme. The comparison of the three architectures is summarized in Fig. 2. We will now discuss these architectures in more detail:

1. One resource-aware controller for all control tasks (left). Embedding control tasks and allocation in a single resource-aware controller allows to optimize the joint control objective for a given actuator availability. Embedding control tasks in a single controller is preferred for a set of control tasks for which their associated plasma dynamics can be effectively integrated in a control-oriented model (see the example of stored energy and current profile control discussed in Section 2.2). However, for other control tasks integration in a single controller is not straightforward due to strongly nonlinear physics relations (e.g. combination of plasma profile control and NTM control) or the combination of continuous plasma dynamics and discrete events (e.g. sawteeth reconnection and NTM triggering). Reusing existing control algorithms is difficult or even impossible in this scheme.

2. Iterative optimization of control and allocation (middle). Here the joint control objective is optimized by iteratively computing the control task requests for a set of assigned actuators, while updating the assigned actuators at each iteration. Hence control tasks requests need to be calculated multiple times at each time step, which increases the computational time. Reusing existing controllers may be possible in this scheme. However, it is likely impossible to provide any guarantees that this iterative scheme will converge towards an optimum in terms of the joint control objective and therefore this scheme is practically not feasible.

3. Hierarchical approach (right). Here the actuator allocation is done before (pre-controller actuator allocation) and/or after (post-controller actuator allocation) executing the control tasks at each time step. 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. The optimization of a joint objective of the control tasks can be approached by taking the control task priorities into account in the actuator allocation.

In the literature all schemes [6,7,10–13] are hierarchical. From the comparison of the various schemes in this section, we conclude also that a hierarchical scheme is favorable if control tasks cannot be integrated in a single controller. This is mainly thanks to its transparency and ease of real-time implementation of a hierarchical scheme. In the next subsection we will look in more detail on the possible places of actuator allocation in hierarchical schemes.

2.4. The place(s) of actuator allocation in hierarchical schemes

In a hierarchical scheme, the actuator allocation can be performed either before or after executing the control tasks, or both. In Fig. 3 we compare three options for the place of actuator allocation.

1. **Only post-controller actuator allocation (left).** Here control tasks generate generic *actuation requests*, and the post-controller actuator allocation decides how to realize these requests with the available actuator systems. Requests sent by the control tasks need not specify which physical actuators should be used to realize them, thus providing a layer of abstraction between the control task and the actuator systems. For example, an actuation request might be to deposit a given amount of power at a specified radial location. This simplifies the design of controllers in case of complex actuator systems. However, the control tasks are not resource-aware and will only know at a next time step to what extent their control request has been realized.
2. **Only pre-controller actuator allocation (middle).** Here, actuator systems are directly allocated to specific control tasks based on requests from control tasks at the previous time step. To achieve this, control tasks must be aware of details the actuator systems. For example, a control task may request to control a given power source and injection system to achieve its control objective. The advantage is that control tasks can be made resource-aware, since they know (before execution), what resources they have at their disposal. However, since each control task need to be aware of details of the actuator systems, it may be practical to implement only if the number of actuators is relatively small. In [14] this scheme is employed based on pre-defined actuation requests for each control task. This scheme was also used in [19], where actuation requests from the previous time step were used to distribute the available power over the prioritized control tasks for integrated control simulations.
3. **Pre- and post-controller actuator allocation (right).** This scheme combines the previous two solutions. The pre-controller actuator allocation assigns actuators to the control tasks based on requests from the previous time step. However, contrarily to (2), the control tasks do not receive detailed information about allocated actuators, but only general information of the actuation capability provided by these actuators. The control tasks send, as in (1), actuation requests to the post-controller allocation block, which translates the requests to actuator system commands. With this architecture, a layer of abstraction is present between the control tasks and the actuators, and at the same time control tasks can be made resource-aware. However, out of the three options, this is the most complicated to implement in practice.

Summarizing: option 1 and 3 are best for tokamaks with many and complex actuator systems, where option 3 is preferred but more cumbersome to implement. Option 2 is best for tokamaks with a small number of relatively simple actuator systems.

2.5. 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 pre- and post-controller actuator allocation to provide a layer of abstraction between the control tasks and details of the actuator systems.

3. H&CD actuator allocation problem formulation

In the previous section we evaluated PCS architectures for integrated control. In the remainder of this work we will focus on Heating and Current Drive (H&CD) actuators and develop a real-time 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 [9,5,20].

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. Then 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.

3.1. Actuator allocation problem definition and interfaces

Consider the post-actuator allocation block in cases (1) and (3), as well as the pre-allocation block in case (2) of Fig. 3. The task of these blocks is to allocate the actuators to realize the (prioritized) requests coming from the control tasks, subject to constraints in the actuator availability and capability. In this case, the interfaces can be written as in Fig. 4.

The actuator allocation block receives the following information:

- Prioritized actuation requests. For each target:
  - Power requested at target
  - Current to be driven at target
  - Deposition location at target (e.g. normalized toroidal magnetic flux ρ of desired deposition)
  - 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 real-time actuator allocation algorithm unless strictly necessary, e.g. in case of sudden failures of actuators)

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

- Actuator control systems:
  - Actuator power commands
  - Actuator deposition location commands
  - Actuator configuration commands (e.g. desired settings of transmission line switches)
- Control tasks
  - Allocated power, etc. per target
- Supervisory controller
  - Allocated resources and total available resources at this time step

Note that the same interface can also be used for the pre-controller actuator allocation block in cases (2) and (3) of Fig. 4, where instead of

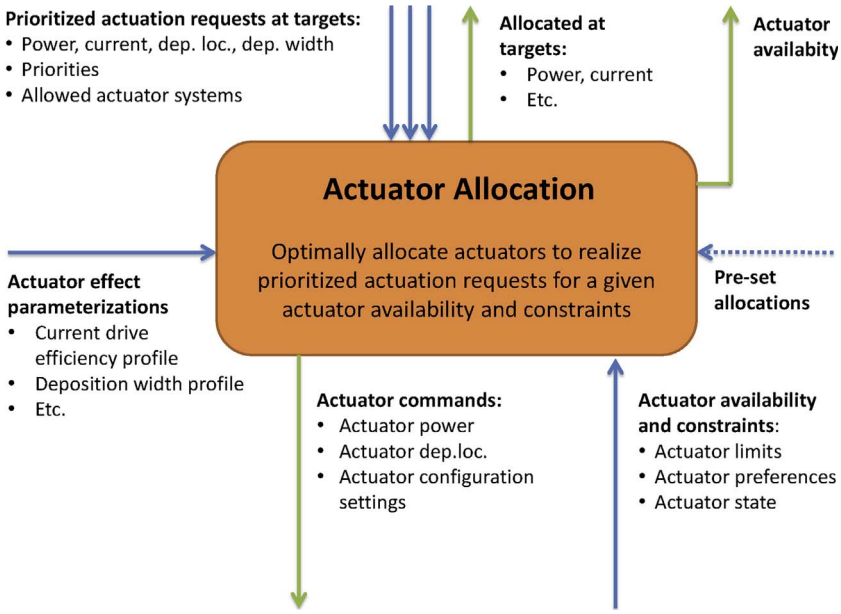

Fig. 4. Interfaces of actuator allocation block with input information (blue) and available information after performing actuator allocation (green). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sending actuator commands to the actuator control systems, the block sends actuator allocations (case 2) or actuation capabilities (case 3) to the control tasks.

In Section 3.6 we define in more detail the input and output signals of the actuator allocation block.

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 Fig. 5.

Unless each source is connected to an independent power supply,

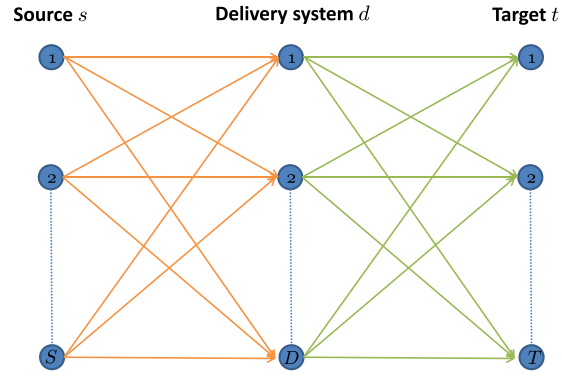

Fig. 5. Connectivity network between sources, delivery systems and targets.

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 3.7.

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^{S2D} 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:

$$\#_{(S,D,T)} = (T+1)^D \prod_{s=1}^{S} n_s^{\mathrm{S2D}} \tag{1}$$

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 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 (1) can be reduced to:

$$\#_{(S,T)} = (T+1)^S \tag{2}$$

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 [14].

In the actuator allocation algorithm in [14], the best allocation option was found by computing for all allocation options the

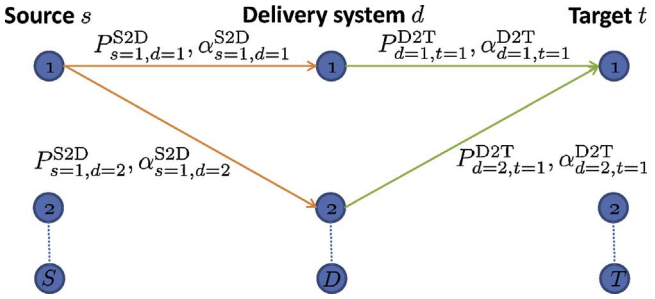

Fig. 6. Definition of optimization variables: continuous variables for power transferred along a connection $P^{S2D}_{s,d}$, $P^{D2T}_{d,t}$ and binary variables $\alpha^{S2D}_{s,d}$, $\alpha^{D2T}_{d,t}$ to indicate if a connection is active.

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 0.3 ms 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 4.2.5), even if code-optimization, parallelization and future improved hardware is considered.

3.4. Formulation as a generic optimization problem

The discussed actuator allocation problem formulation in [14] 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 [21,22]. 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:

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & J(z) = z^{\top} \mathrm{H}\, z + f^{\top} z && \text{(cost-function)} \\ \text{subject to} \quad & A_{\text{ineq}} z \le b_{\text{ineq}} && \text{(linear inequality constraints)} \\ & z_{\min} \le z \le z_{\max} && \text{(bounds)} \\ & z_i \in \mathbb{N} && \text{(integer variables)} \end{aligned} \tag{3}$$

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.

3.5. Optimization variables choice

The first step in writing the optimization problem is the definition of the optimization variables z. As discussed before, this vector z in MIQP problems can contain both continuous and binary variables. We choose to use for each connection between a source and a delivery system (and between a delivery system and a target) both a continuous variable P for the power transferred along a connection and a binary variable α to indicate if a connection is active (see Fig. 6). The continuous variables allow to optimize over the individual powers supplied by sources and delivery systems and the binary variables are essential to model discrete choices: e.g. a source can only be connected to one delivery system. The continuous and binary variables have the following relation:

- $\alpha^{S2D}_{s,d} = 1$ if $P^{S2D}_{s,d} > 0$, otherwise $\alpha^{S2D}_{s,d} = 0$
- $\alpha^{D2T}_{d,t} = 1$ if $P^{D2T}_{d,t} > 0$, otherwise $\alpha^{D2T}_{d,t} = 0$

We can now define the optimization variable vector z in (3) as:

$$z = \begin{bmatrix} \overline{P}^{\mathrm{S2D}} \\ \overline{P}^{\mathrm{D2T}} \\ \overline{\alpha}^{\mathrm{S2D}} \\ \overline{\alpha}^{\mathrm{D2T}} \end{bmatrix} \tag{4}$$

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

$$\overline{P}^{\mathrm{S2D}} = [[P^{\mathrm{S2D}}_{s=1,d=1} \cdots P^{\mathrm{S2D}}_{s=S,d=1}]^{\top} \quad [P^{\mathrm{S2D}}_{s=1,d=D} \cdots P^{\mathrm{S2D}}_{s=S,d=D}]^{\top}]^{\top}.$$

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 C.5.

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 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 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 1.

The interfaces of the actuator allocation algorithm were briefly introduced in words in Section 3.1 (see also Fig. 4). Here we need to

[1] Alternatively a linear cost function could be chosen (see Appendix C.5).

Table 1
System configuration definition.

Description	Variable	Possible values
Resource type		
Source type	S_s^{type}	Integer: (1) EC, (2) IC and (3) NB
Delivery system type	D_d^{type}	Integer: (1) EC, (2) IC and (3) NB
Connection topology maps between		
Sources and delivery systems	$M_{s,d}^{\text{S2D}}$	{0 (no connection possible), 1 (connection possible)}
Power supplies and sources	$M_{h,s}^{\text{H2S}}$	{0 (no connection possible), 1 (connection possible)}
Actuator efficiency		
Power transfer efficiency between source and delivery system	$\eta_{s,d}^{\text{S2D}}$	$0 \leq \eta_{s,d}^{\text{S2D}} \leq 1$

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 actuation requests, source and delivery system information, the present allocation, actuator effect parameterizations and feedforward (pre-set) allocations. This input information is specified in Table 2.

The normalized toroidal magnetic flux is used for the deposition location ρ as well as the full Gaussian deposition width w in this work, 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).

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 3, where also the relation to the optimization variables is given.

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 MIQP-problem (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 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.

3.7.1. Penalize difference between requested and allocated values at targets

The main objective of the actuator allocation is to achieve the prioritized actuation 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 actuation request importance specifications (that include the control task priorities) and normalized.[2] These cost penalties can be expressed as follows:

Power.

$$J_P = \nu_P \frac{1}{T} \sum_{t=1}^{T} W_t^{P,\text{req}} \frac{1}{(P_t^{\text{req,norm}})^2} (P_t^{\text{T,alloc}} - P_t^{\text{req}})^2 \tag{5}$$

[2] An appropriate choice of the normalization for required power $P_t^{\text{req,norm}}$ (5), required current $I_t^{\text{req,norm}}$ (6) and present power $P_s^{\text{S,pres,norm}}$ (8) is discussed in Appendix E.

Table 2
Definition of real-time information available to the actuator allocation algorithm.

Description	Variable	Unit	Possible values
Actuation requests per target			
Volume integrated power	P_t^{req}	[MW]	
Surface integrated driven current	I_t^{req}	[MA]	
Deposition location	ρ_t^{req}	–	
Desired full gaussian deposition width	w_t^{req}	–	
Importance of power matching	$W_t^{P,\text{req}}$	–	$0 \leq W_t^{P,\text{req}} \leq 1$
Importance of current matching	$W_t^{I,\text{req}}$	–	$0 \leq W_t^{I,\text{req}} \leq 1$
Importance of deposition width matching	$W_t^{w,\text{req}}$	–	$0 \leq W_t^{w,\text{req}} \leq 1$
Delivery system type is allowed at target	$D_t^{\text{allow}}(i_{\text{D,type}})$	–	{0(false), 1(true)} ∀ $i_{\text{D,type}} \in \{1, 2, 3\}$
Source information			
Power minimum	$P_s^{\text{S,min}}$	[MW]	
Power maximum	$P_s^{\text{S,max}}$	[MW]	
Source avoidance penalty	$W_s^{\text{S,avoid}}$	–	$0 < W_s^{\text{S,avoid}} \leq 1$
Source is connected to its power supply	ξ_s^{H2S}	–	{0(false), 1(true)}
Source is being switched to other del. sys.	$\sigma_{s,d}^{\text{S2D}}$	–	{0(false), 1(true)}
Delivery system information			
Power maximum	$P_d^{\text{D,max}}$	[MW]	
Delivery system avoidance penalty:	$W_d^{\text{D,avoid}}$	–	$0 < W_d^{\text{D,avoid}} \leq 1$
Present allocation per source and delivery system			
Source power request	$P_s^{\text{S,pres}}$	[MW]	
Delivery system deposition location	$\rho_d^{\text{D,pres}}$	–	
Source is active on delivery system	$\alpha_{s,d}^{\text{S2D,pres}}$	–	{0(false), 1(true)}
Delivery system is active on target	$\alpha_{d,t}^{\text{D2T,pres}}$	–	{0(false), 1(true)}
Actuator effect parameterization maps			
Power transfer efficiency del. sys. to target	$\eta_{d,t}^{\text{D2T}} = f_\eta^{\text{D2T}}(d, \rho_t^{\text{req}})$	–	
Deposition width del. sys. at target	$w_{d,t}^{\text{D2T}} = f_w(d, \rho_t^{\text{req}})$	–	
Current drive efficiency del. sys. at target	$\eta_{d,t}^{\text{cd}} = f_\eta^{\text{cd}}(d, \rho_t^{\text{req}})$	–	
Minimum deposition location del. sys.	$\rho_d^{\min}$	–	
Maximum deposition location del. sys.	$\rho_d^{\max}$	–	
Feedforward (pre-set) allocations per source and delivery system:			
Source is active on delivery system	$\alpha_{s,d}^{\text{S2D,ff}}$	–	{0(false), 1(true)}
Delivery system is active on target	$\alpha_{d,t}^{\text{D2T,ff}}$	–	{0(false), 1(true)}

Driven current.

$$J_I = \nu_I \frac{1}{T} \sum_{t=1}^{T} W_t^{I,\text{req}} \frac{1}{(I_t^{\text{req,norm}})^2} (I_t^{\text{T,alloc}} - I_t^{\text{req}})^2 \tag{6}$$

Deposition width.

$$J_w = \nu_w \frac{1}{T} \sum_{d=1}^{D} \sum_{t=1}^{T} W_t^{w,\text{req}} (w_{d,t}^{\text{D2T,del}} - w_t^{\text{req}})^2 \alpha_{d,t}^{\text{D2T}} \tag{7}$$

Table 3
Definition of the algorithm output and relation to optimization variables.

Description	Variable	Units	Relation to optimization vars
Information to actuator control systems about allocated			
Source power	$P_s^{\mathrm{S,alloc}}$	[MW]	$\sum_{d=1}^{D}\sum_{t=1}^{T} P_{s,d}^{\mathrm{S2D}}\alpha_{d,t}^{\mathrm{D2T}}$
Del. sys. power	$P_d^{\mathrm{D,alloc}}$	[MW]	$\sum_{s=1}^{S}\sum_{t=1}^{T} \eta_{s,d}^{\mathrm{S2D}} P_{s,d}^{\mathrm{S2D}}\alpha_{d,t}^{\mathrm{D2T}}$
Del. sys. per source	$D_s^{\mathrm{S,alloc}}$	–	$\sum_{d=1}^{D}\alpha_{s,d}^{\mathrm{S2D}} d$
Del. sys. dep. loc.	$\rho_d^{\mathrm{D,alloc}}$	–	$\sum_{t=1}^{T}\alpha_{d,t}^{\mathrm{D2T}}\rho_t^{\mathrm{req}}$
Information to control tasks about allocated			
Power at target	$P_t^{\mathrm{T,alloc}}$	[MW]	$\sum_{d=1}^{D}\eta_{d,t}^{\mathrm{D2T}} P_{d,t}^{\mathrm{D2T}}$
Driven current at target	$I_t^{\mathrm{T,alloc}}$	[MA]	$\sum_{d=1}^{D}\eta_{d,t}^{\mathrm{D2T}}\eta_{d,t}^{\mathrm{cd}}(d,\rho_t^{\mathrm{req}}) P_{d,t}^{\mathrm{D2T}}$
Information to supervisory level about available			
Power per act. sys. type	$P^{\mathrm{max,S,type}}(i_{\mathrm{S,type}})$	[MW]	$\sum_{s=1}^{S} P_s^{\mathrm{S,max}}$ if $S_s^{\mathrm{type}} = = i_{\mathrm{S,type}}$, $i_{\mathrm{S,type}} \in \{1, 2, 3\}$

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

3.7.2. Penalize changes with respect to 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 with respect to the present allocation (present abbreviated as 'pres' in equations) for:

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

$$J_{\Delta P} = \nu_{\Delta P}\frac{1}{S}\sum_{s=1}^{S}\frac{1}{(P_s^{\mathrm{S,pres,norm}})^2}(P_s^{\mathrm{S,alloc}} - P_s^{\mathrm{S,pres}})^2 \tag{8}$$

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

$$J_{\Delta\rho} = \nu_{\Delta\rho}\frac{1}{D}\sum_{d=1}^{D}\sum_{t=1}^{T}(\rho_t^{\mathrm{req}} - \rho_d^{\mathrm{D,pres}})^2\alpha_{d,t}^{\mathrm{D2T}} \tag{9}$$

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.

$$J_{\mathrm{ac,pres}}^{\mathrm{S2D}} = \nu_{\mathrm{ac,pres}}^{\mathrm{S2D}}\frac{1}{S}\sum_{s=1}^{S}\sum_{d=1}^{D}(1 - \alpha_{s,d}^{\mathrm{S2D,pres}})\alpha_{s,d}^{\mathrm{S2D}} \tag{10}$$

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 (B.1). In addition, it can be desired to keep using the same delivery systems at their already allocated targets if possible, which is expressed in (B.2).

3.7.3. 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.

$$J_{S,\mathrm{avoid}} = \nu_{S,\mathrm{avoid}}\frac{1}{S}\sum_{s=1}^{S}\sum_{d=1}^{D} W_s^{\mathrm{S,avoid}}\alpha_{s,d}^{\mathrm{S2D}} \tag{11}$$

Setting $W_{s_i}^{S,\mathrm{avoid}} < W_{s_j}^{S,\mathrm{avoid}}$ gives preference to use source s_i with respect to source s_j.

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

$$J_{D,\mathrm{avoid}} = \nu_{D,\mathrm{avoid}}\frac{1}{D}\sum_{d=1}^{D}\sum_{t=1}^{T} W_d^{\mathrm{D,avoid}}\alpha_{d,t}^{\mathrm{D2T}} \tag{12}$$

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.

$$J_{S2D,\mathrm{avoid}} = \nu_{\mathrm{avoid}}^{\mathrm{S2D}}\frac{1}{S}\sum_{s=1}^{S}\sum_{d=1}^{D} W_{s,d}^{\mathrm{S2D,avoid}}\alpha_{s,d}^{\mathrm{S2D}} \tag{13}$$

3.7.4. Penalize changes with respect to pre-defined (feedforward) allocation

In some situations, it may be desirable to keep the allocation close to a predefined 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 with respect to pre-set allocations for using sources on delivery systems (B.3), and delivery systems on targets (B.4).

3.7.5. 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 [9]). This can be avoided using the cost penalty (B.5). 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 (B.6).

3.7.6. Remarks on tuning allocation behavior

In present day practice, an expert operator needs to select the appropriate settings for individual actuators such as gyrotron power waveforms for a specific (control) experiment. This often requires extensive manual fine-tuning over several discharges. Using an actuator allocation algorithm as presented here, the individual actuator settings follow from the global specification of the desired actuator allocation behavior in terms of the tuning parameters $\nu_{(\cdot)}$, as well as from the specification of the system configuration, actuator effect parameterizations and actuator constraints. The advantage of specifying the actuator behavior at this higher level is that the setting is more general and is able to cope, for example, with unexpected faults in actuators. Also, since the behavior is defined in more generic manner, the same high-level settings can be re-used for different plasma targets, without needing to change the individual actuator settings to match the target for each different case.

To choose the tuning parameters $\nu_{(\cdot)}$, a user of the algorithm would typically define a broad set of representative cases expected during plasma discharges for which the allocation will be used, with the

corresponding desired allocation behavior. The tuning procedure may start by setting the tuning parameter corresponding to the most important cost penalty equal to 1 and checking the allocation behavior in simulations of all representative cases. Subsequently, the tuning parameters $\nu_{(\cdot)}$ corresponding to less important cost penalties can be set one by one, again checking their effect on the overall allocation behavior.

Normalization of power and current request is applied in the algorithm to ensure that a single set of tuning parameters works for different power and current request levels as well as different numbers of sources, delivery systems and targets. Still, it is not possible to formally guarantee that a single set of tuning parameters will ensure that the exact desired allocation behavior is obtained in any event and operation point of a tokamak. Therefore it remains essential to validate the tuning parameters in a broad set of simulation cases that are representative of intended experiments.

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 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: (C.1)–(C.3).
- Constraints to ensure that a source can only connect to a single delivery system (C.4), and similarly a delivery system can only be allocated to a single target (C.5).
- Technical constraints to ensure that active sources sharing the same power supply have equal source power (C.6).
- 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 (C.7), the target is out of reach of the delivery system (C.8) or the delivery system type is not allowed at the target (C.9).

These constraints can be imposed on the optimization problem using linear inequality constraints and bounds that fit in the MIQP formulation (3).

3.9. Constructing and solving MIQP-problem

We can now proceed to formulate the MIQP-problem (3) by using the definitions of the desired allocation behavior in cost penalties (Section 3.7) and the allocation feasibility and actuator availability in constraints (Section 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 (4). The Hessian H and gradient vector f can be derived from (5)–(13) and (B.1)–(B.6), and the inequality constraint matrix A_{ineq} and vector b_{ineq} from (C.1)–(C.6). The bounds $z_{\min}$ and $z_{\max}$ can be derived using (C.7)–(C.9) and the variables $P_{s,d}^{\text{S,max}}$ and $P_{d,t}^{\text{D2T,max}}$.

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 (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 [23] and Gurobi [24] (both with free acedemic license). CPLEX and Gurobi are among the fastest available solvers, see [25] for a frequently updated benchmark of MIP-solvers. In this work we use the solver CPLEX, called from Matlab [26].

MIQP-problems and the underlying decision problem are known to be non-deterministic polynomial-time hard (NP-hard) [21], 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.

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.

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.[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.

Fig. 7 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: $W_t^I = W_t^P$).

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

4.1.1. Matching power request at target

We start with only a penalty on the difference between required and allocated power: $\nu_P = 1$ (see (5)). It is convenient to set the main penalty equal to one, allowing to easily interpret the relative importance of other penalties. The result is given in Fig. 8, where the allocated values are indicated with a circle and the corresponding change with an arrow.

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).

4.1.2. 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)), indicating that we consider this penalty relatively less important than the power mismatch penalty. The

[3] System details are only given for the full size example in Section 4.2.

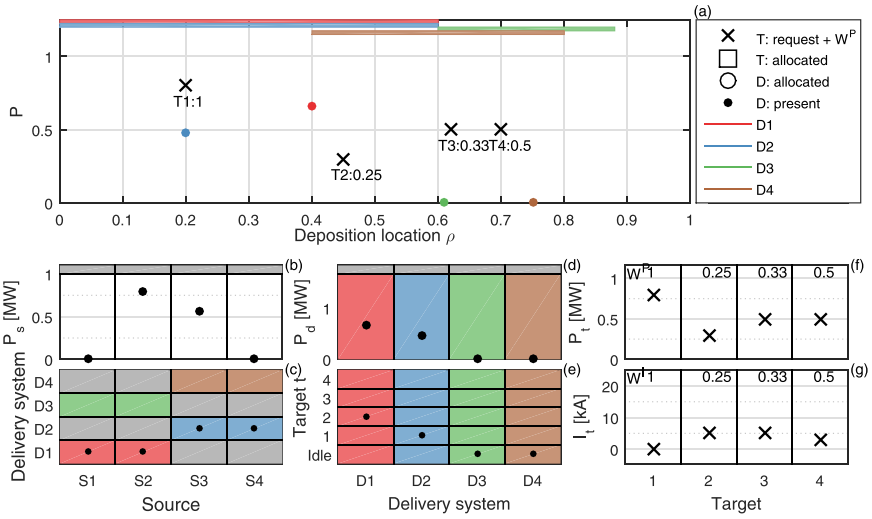

Fig. 7. Targets and present allocations for small system size illustration. Top (a): present allocation (dot) for power and deposition locations of delivery systems. Targets are indicated with (x) and their target importance below these. Deposition location ranges for delivery systems are given by horizontal 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.

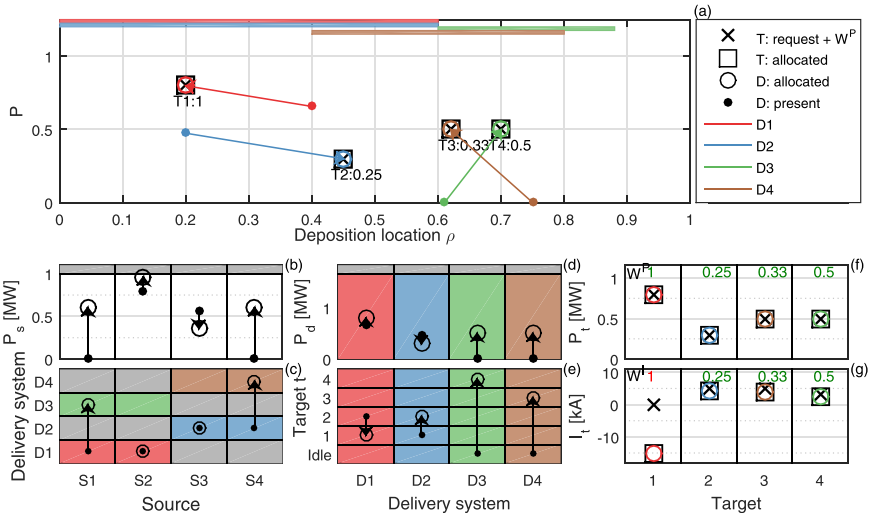

Fig. 8. Allocation result with only a penalty on the power mismatch at the targets. Allocated values (circles) and required changes (arrows) are added to the present situation as given in Fig. 7. All target powers are achieved as allocated values (o) coincide with targets (x) in (a), see also (f). This 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).

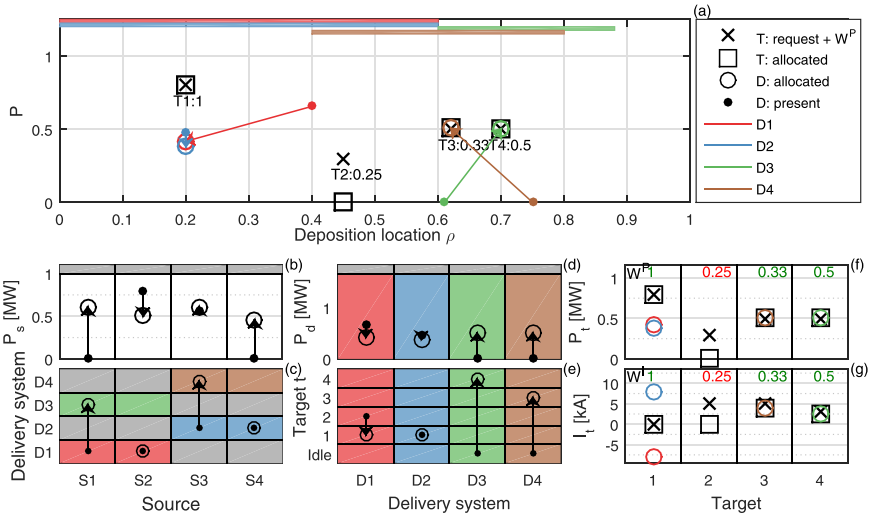

Fig. 9. Illustration of allocation with penalties on both the power and current mismatch at target. Note that the current at target T1 is matched by using both a counter-current (D2) and co-current delivery system (D1), at the expense of losing target T2 with lowest priority.

resulting allocation is given in Fig. 9.

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.

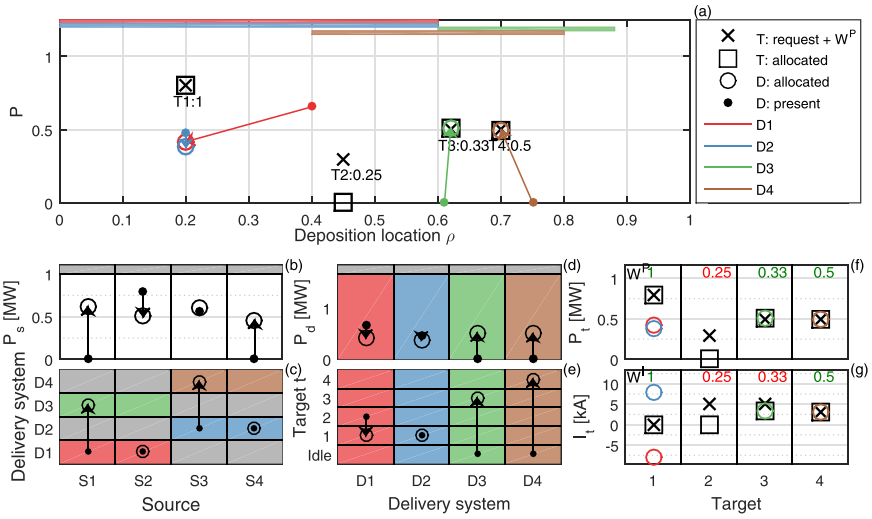

Fig. 10. Illustration of allocation after adding movement costs for the delivery systems. Note that the movement of D3 and D4 is minimized by allocating them to their most nearby target, at the expense of a slightly worse current matching for target T3 compared to Fig. 9.

4.1.3. Including a movement cost

We will now try to minimize the delivery system movement costs by setting $\nu_{\Delta\rho} = 0.1$ (see (9)) such that the most nearby delivery systems should be selected.

Indeed Fig. 10 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). Suppose we would have chosen a very high movement penalty, only power would have been delivered to target T1 by D2 which is then the best allocation option that requires no delivery system movement. Contrarily, if a very small movement penalty was chosen, we would obtained the allocation in Fig. 9.

These three results reveal the clear impact of each of these cost penalties on the allocation behavior and how the allocation algorithm chooses the best allocation option corresponding to the user-defined preferences.

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.

4.2.1. 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:

1 **Electron Cyclotron system** [9,20]**.** 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[4] (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.

2 **Ion Cyclotron system** [5,20]**.** 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 [20]. In this article 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 [5]. Each IC source can only connect to a single IC delivery system and has its own power supply.

3 **Neutral Beam Injection system** [5,20]**.** 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 $\eta_{d,t}^{\mathrm{D2T}}$ is fixed at 1 (no losses, e.g. full power absorbtion).
- The deposition width parameterization $w_{d,t}^{\mathrm{D2T}} = f_w(d, \rho_t^{\mathrm{req}})$ is not used in these examples and not defined here.
- The current drive efficiency parameterization $\eta_{d,t}^{\mathrm{cd}} = f_\eta^{\mathrm{cd}}(d, \rho_t^{\mathrm{req}})$ is given in Appendix 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 $\eta_{s,d}^{\mathrm{S2D}}$ is fixed at $\frac{20}{24}$ 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[5] are shown directly in the figures.

In the next examples we choose the cost penalties as given in Table 4. Following the tuning remarks given in Section 3.7.6, we have tuned these cost penalties on a number of example problems such that the desired allocation behavior is obtained. The values in Table 4 indicate that we consider target power matching most important, followed by target current matching and avoiding delivery system movements. Furthermore, we use the default settings and tolerances of the CPLEX MIP-solver [23] in these examples.

4.2.2. 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 ρ = 0.6. The resulting allocation is given in Fig. 11.

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

[5] Details of the proposed EC source to delivery system connections were given in a personal communication [27].

Table 4
Cost penalties in ITER examples.

Cost penalty on	Coefficient	Value	Equation
Power mismatch	ν_P	1	(5)
Current mismatch	ν_I	10^{-2}	(6)
Source power changes	$\nu_{\Delta P}$	10^{-5}	(8)
Delivery system movements	$\nu_{\Delta\rho}$	10^{-2}	(9)
Allocation of source to other delivery system	$\nu_{\mathrm{ac,pres}}^{\mathrm{S2D}}$	10^{-4}	(10)
Source use (equal source preference)	$\nu_{S,\mathrm{avoid}}$	10^{-4}	(11)
Delivery system use (equal preference)	$\nu_{D,\mathrm{avoid}}$	10^{-5}	(12)
Allocation of active sources sharing a power supply to different delivery systems	$\nu_{H2S,\mathrm{sameD}}$	10^{-4}	(B.6)

is suddenly unavailable. Both EC and IC systems can be used to replace the missing NB-system. The results are given in Fig. 12.

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.

4.2.4. Example: Simulating EC management for 100s actuation request sequence

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 ITER example involving multiple trips in EC sources (gyrotrons) during a 100s actuation request sequence. It should be stressed that we do not perform a closed-loop simulation including all PCS components and a plant simulator (e.g. simulating the entire loop shown in Fig. 1, but we only evaluate the performance of the actuator allocation algorithm examining the actuator allocation in response to a pre-defined sequence of actuation requests.

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: $P_s^{\mathrm{S,min}} = P_s^{\mathrm{S,max}} = 1\mathrm{MW}$. 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 $W_s^{\mathrm{S,avoid}}$ increases linearly from 0.1 for GY1 to 1 for GY24, giving

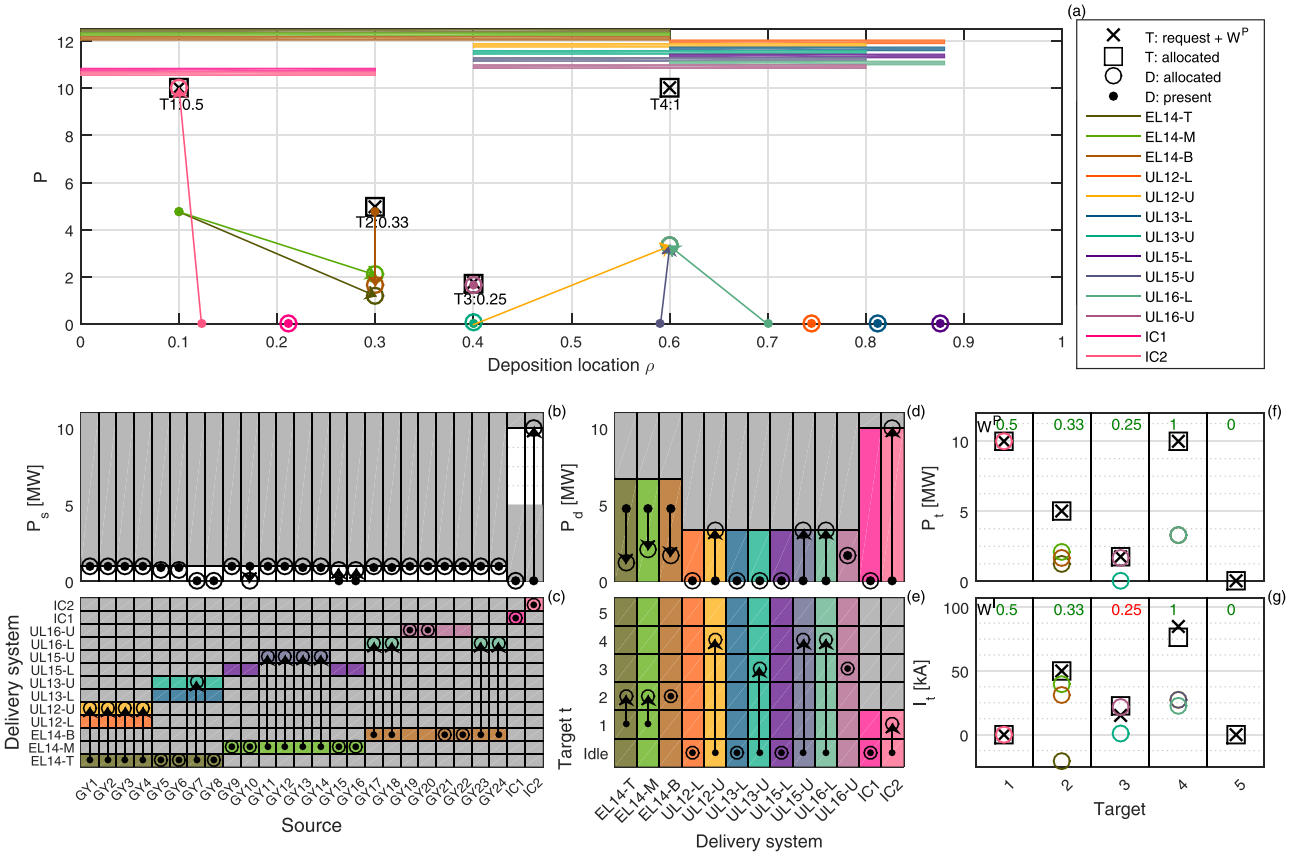

Fig. 11. Results for example with sudden NTM control request while performing central heating / profile control. The central heating target T1 is taken over by the nearby IC2, providing freedom in the EC-system to redistribute the power, mainly to achieve the 10MW NTM control target T4 with highest priority ((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).

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).

4.2.3. 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 ρ = 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

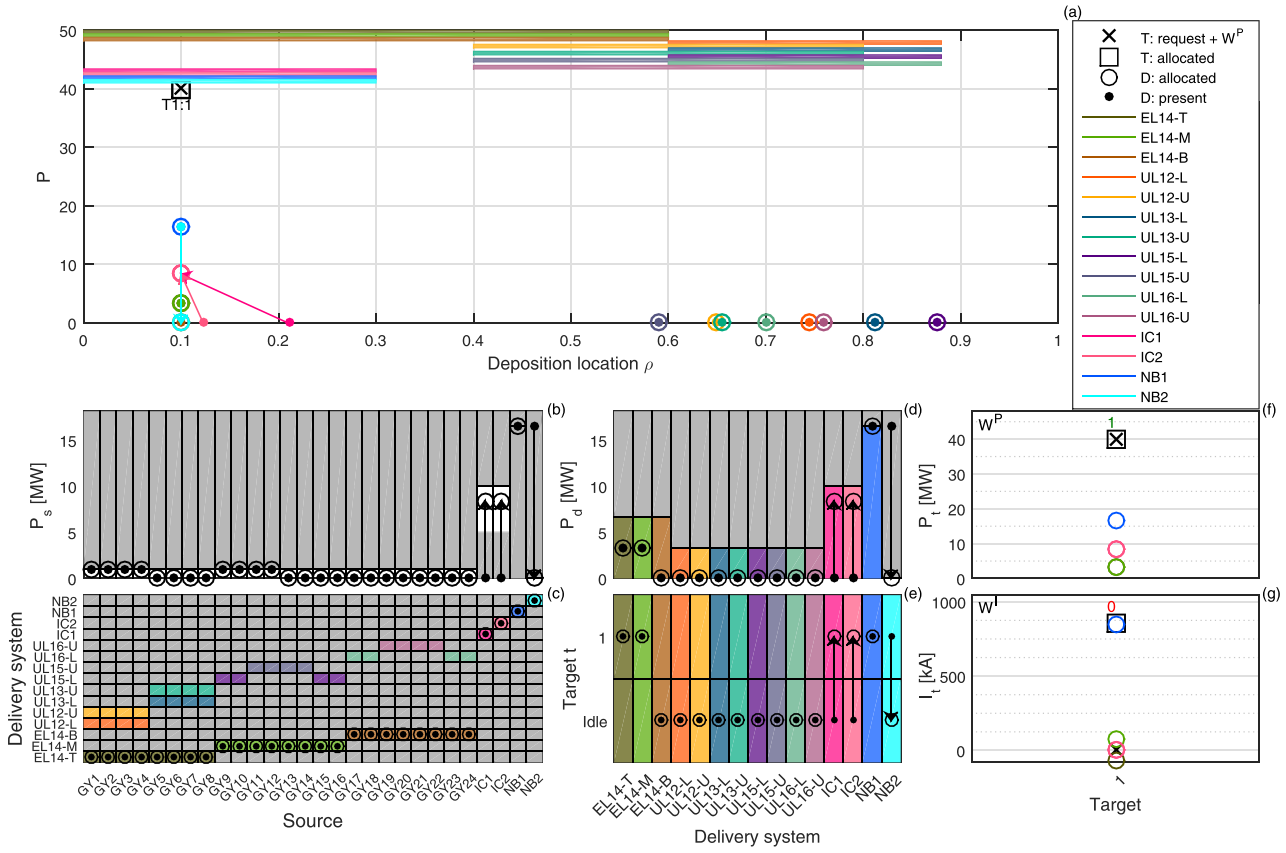

Fig. 12. Results example for replacing suddenly unavailable NB power with IC power. A single target requires 40MW power (panels (a) and (f), where the current importance (g) is zero. Nearby located IC-systems take over the missing NB power (panels (a), (b) and (e)). EC allocations remain unchanged, so as to minimize source changes and the number of used sources.

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. The plasma state (electron temperature and density profiles) determining the current drive efficiency parameterization map is assumed to be fixed in time (see Appendix A).

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 Fig. 13. 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.

4.2.5. 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 a single thread assigned to the solver. Solving the same problems using the Gurobi-solver [24] gave similar computational times.

In some cases the solver was not able to guarantee that the solution satisfies the MIPgap-tolerance (see Section 3.9), within the set 10 seconds time limit of the solver. These cases are characterized by high cost penalties on integer optimization variables such that integer decisions dominate the quality of solution. Analyzing the solutions in these cases indicated that the solutions obtained after 1 second were already corresponding to the set desired allocation behavior e.g. the targets were correctly achieved although minor improvements were possible in some cases.

The maximum available computational time to perform the actuator allocation at ITER may be derived from the maximum allowed time to respond to events that require the H&CD systems. We take here 10% of the maximum allowed latency between seeding a 2/1 NTM and the required start of EC-power deposition within the NTM that is assumed to be about 1-3s [28,29]. Therefore the actuator allocation algorithm should be run at least every 100ms (and the sources and delivery systems should already be allocated to track the relevant q-surface). Taking also into account the potential of dedicated hardware and exploiting MIP-solvers parallelization capabilities, the allocation algorithm can be used to readily solve an ITER-size allocation problem in real-time.

The small 4x4x4 examples of Section 4.1 were solved using CPLEX in less than 25ms. This indicates that the required computational time does not scale with the number of possible allocation options. These small problems were also solved with Gurobi in less than 10 ms, indicating that other solvers than CPLEX can be significantly faster for small problems.

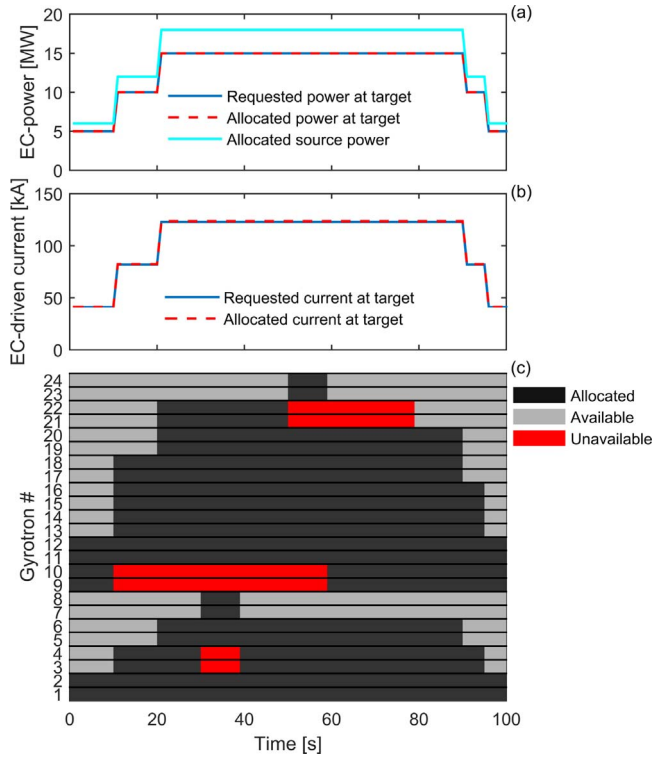

Fig. 13. Illustration of effective EC management in 100s ITER actuation request sequence involving multiple trips in gyrotrons. Requested power (a) and current (b) is achieved by adding multiple pairs of gyrotrons (c), where GY1-GY8 are connected to a launcher mirror driving counter-current and GY9-GY24 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 are replaced by others, when becoming available again, these are allocated again due to their higher preference.

5. Conclusions and outlook

This work 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 a combination of pre-and post-controller allocation.

The second part of this paper presented an efficient algorithm for allocating H&CD actuators in real-time based on prioritized requests by the control tasks, 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 an Intel® i7-2670QM CPU running at 2.20GHz.

The developed algorithm can be readily 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.

Acknowledgements

The authors acknowledge fruitful discussions with Chris Rapson (IPP Garching), Marco de Baar and Hugo van den Brand (DIFFER), Thomas Blanken, Erjen Lefeber and Jasper Verhoeven (TU/e), Cristian Galperti, Tim Goodman and Olivier Sauter (SPC), Mark Henderson, Doohyun Kim and Joe Snipes (ITER). This work 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. This work is also supported by the Netherlands Organisation for Scientific Research (NWO) via the Innovational Research Incentives Scheme.

Appendix 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 $\eta_{d,t}^{\mathrm{cd}}(d, \rho_t^{\mathrm{req}})$ as:

$$\eta_{d,t}^{\mathrm{cd}}(d, \rho_t^{\mathrm{req}}) = \eta_d^{\mathrm{cd},0}(d, \rho_t^{\mathrm{req}}) \frac{T_e(\rho_t^{\mathrm{req}})}{n_e(\rho_t^{\mathrm{req}})}, \tag{A.1}$$

where $\eta_d^{\mathrm{cd},0}$ can be negative, zero or positive value to distinguish e.g. counter-current drive, pure heating and co-current drive respectively. The values of $\eta_d^{\mathrm{cd},0}$ given in Table A.5 are approximated for EC from [30,31], while for IC and NB we choose zero.

We used in our examples a fixed electron temperature profile $T_e(\rho)$ and electron density profile $n_e(\rho)$ from ITER H-mode simulations in [32] to compute the current drive efficiency map based on the plasma state.

Appendix B. Additional cost penalties

In Section 3.7 a number of cost penalties were introduced to define the desired allocation behavior. Some cost penalties were only briefly

Table A.5
Current drive efficiency delivery systems.

Delivery system	d	$\eta_d^{\mathrm{cd},0}$
EL14-T	1	−0.971
EL14-M,EL14-B	2–3	1.068
UL12-L to UL16-L	4, 6, 8, 10	0.75
UL12-U to UL16-U	5, 7, 9, 11	0.9
IC1,IC2	12–13	0
NB1,NB2	14–15	0

described and are given here in more detail.

B.1 Other penalties on changes with respect to 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 (5) to (7), we would also like to penalize changes with respect to the present allocation for:

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.

$$J_{\mathrm{ac,pres},\sigma}^{\mathrm{S2D}} = \nu_{\mathrm{ac,pres},\sigma}^{\mathrm{S2D}} \frac{1}{S} \sum_{s=1}^{S} \sum_{d=1}^{D} \sigma_{s,d}^{\mathrm{S2D}} (1 - \alpha_{s,d}^{\mathrm{S2D,pres}}) \alpha_{s,d}^{\mathrm{S2D}} \tag{B.1}$$

Connected target to delivery system. To promote using already allocated delivery systems for a given target.

$$J_{\mathrm{ac,pres}}^{\mathrm{D2T}} = \nu_{\mathrm{ac,pres}}^{\mathrm{D2T}} \frac{1}{D} \sum_{d=1}^{D} \sum_{t=1}^{T} (1 - \alpha_{d,t}^{\mathrm{D2T,pres}}) \alpha_{d,t}^{\mathrm{D2T}} \tag{B.2}$$

B.2 Penalize changes with respect to 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.

$$J_{\mathrm{ac,ff}}^{\mathrm{S2D}} = \nu_{\mathrm{ac,ff}}^{\mathrm{S2D}} \frac{1}{S} \sum_{s=1}^{S} \sum_{d=1}^{D} (1 - \alpha_{s,d}^{\mathrm{S2D,ff}}) \alpha_{s,d}^{\mathrm{S2D}} \tag{B.3}$$

Connected target to delivery system.

$$J_{\mathrm{ac,ff}}^{\mathrm{D2T}} = \nu_{\mathrm{ac,ff}}^{\mathrm{D2T}} \frac{1}{D} \sum_{d=1}^{D} \sum_{t=1}^{T} (1 - \alpha_{d,t}^{\mathrm{D2T,ff}}) \alpha_{d,t}^{\mathrm{D2T}} \tag{B.4}$$

B.3 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 ξ_s^{H2S}.

$$J_{\mathrm{H2S,connect}} = \nu_{\mathrm{H2S,connect}} \frac{1}{S} \sum_{h=1}^{H} \sum_{s_i=1}^{S} \sum_{s_j=1}^{S} \sum_{d=1}^{D} M_{h,s_i}^{\mathrm{H2S}} M_{h,s_j}^{\mathrm{H2S}} \ldots$$
$$[\xi_{s_j}^{\mathrm{H2S}} \alpha_{s_i,d}^{\mathrm{S2D}} + \xi_{s_i}^{\mathrm{H2S}} \alpha_{s_j,d}^{\mathrm{S2D}} - \xi_{s_i}^{\mathrm{H2S}} \xi_{s_j}^{\mathrm{H2S}} \alpha_{s_i,d}^{\mathrm{S2D}} \alpha_{s_j,d}^{\mathrm{S2D}} - \xi_{s_i}^{\mathrm{H2S}} \xi_{s_j}^{\mathrm{H2S}} \alpha_{s_j,d}^{\mathrm{S2D}} \alpha_{s_i,d}^{\mathrm{S2D}}] \tag{B.5}$$

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.

$$J_{H2S,\mathrm{sameD}} = \nu_{H2S,\mathrm{sameD}} \frac{1}{S} \sum_{h=1}^{H} \sum_{s_i=1}^{S} \sum_{s_j=1}^{S} \sum_{d_i=1}^{D} \sum_{d_j=1}^{D} M_{h,s_i}^{\mathrm{H2S}} M_{h,s_j}^{\mathrm{H2S}} \ldots$$
$$M_{s_i,d_i}^{\mathrm{S2D}} M_{s_j,d_j}^{\mathrm{S2D}} \alpha_{s_i,d_i}^{\mathrm{S2D}} \alpha_{s_j,d_j}^{\mathrm{S2D}}. \tag{B.6}$$

where $s_i \neq s_j$ and $d_i \neq d_j$.

Appendix C. Details regarding constraints

In Section 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.

C.1 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 $\alpha_{s,d}^{\mathrm{S2D}}$ and powers $P_{s,d}^{\mathrm{S2D}}$ for each source at a delivery system should be related as follows:

- Source on at delivery system: $\alpha_{s,d}^{\mathrm{S2D}} = 1 \quad \leftrightarrow \quad P_s^{\mathrm{S,min}} \leq P_{s,d}^{\mathrm{S2D}} \leq P_s^{\mathrm{S,max}}$.
- Source off at delivery system: $\alpha_{s,d}^{\mathrm{S2D}} = 0 \quad \leftrightarrow \quad P_{s,d}^{\mathrm{S2D}} = 0$

This requirement can be written as follows:

$$\begin{aligned} &P_{s,d}^{\mathrm{S2D}} - P_s^{\mathrm{S,max}}\alpha_{s,d}^{\mathrm{S2D}} \leq 0, \quad \forall \ s \in \{1,...,S\}, \ d \in \{1,...,D\}, \\ &-P_{s,d}^{\mathrm{S2D}} + P_s^{\mathrm{S,min}}\alpha_{s,d}^{\mathrm{S2D}} \leq 0, \quad \forall \ s \in \{1,...,S\}, \ d \in \{1,...,D\} \end{aligned} \tag{C.1}$$

Note that this is a linear inequality constraint in the optimization variables $\alpha_{s,d}^{\mathrm{S2D}}$ and $P_{s,d}^{\mathrm{S2D}}$. To ensure that $\alpha_{s,d}^{\mathrm{S2D}} = 0$ in case $P_{s,d}^{\mathrm{S2D}} = 0$ and $P_s^{\mathrm{S,min}} = 0$, we add a (small) penalty on non-idle sources $\alpha_{s,d}^{\mathrm{S2D}}$ in the cost function. Choosing $P_s^{\mathrm{S,min}} = P_s^{\mathrm{S,max}}$ 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: $\alpha_{d,t}^{\mathrm{D2T}} = 1 \quad \leftrightarrow \quad P_{d,t}^{\mathrm{D2T}} > 0$
- Delivery system off at target: $\alpha_{d,t}^{\mathrm{D2T}} = 0 \quad \leftrightarrow \quad P_{d,t}^{\mathrm{D2T}} = 0$

These constraints are formulated as follows:

$$\begin{aligned} &P_{d,t}^{\mathrm{D2T}} - P_{d,t}^{\mathrm{D2T,max}}\alpha_{s,d}^{\mathrm{D2T}} \leq 0, \quad \forall \ \mathrm{d} \in \{1,...,\mathrm{D}\}, \ \mathrm{t} \in \{1,...,\mathrm{T}\}, \\ &\qquad -P_{d,t}^{\mathrm{D2T}} \leq 0, \quad \forall \ \mathrm{d} \in \{1,...,\mathrm{D}\}, \ \mathrm{t} \in \{1,...,\mathrm{T}\} \end{aligned} \tag{C.2}$$

where $P_{d,t}^{\mathrm{D2T,max}} = P_d^{\mathrm{D,max}}$, an alternative is discussed in Appendix E. Again we ensure that $\alpha_{d,t}^{\mathrm{D2T}} = 0$ in case $P_{d,t}^{\mathrm{D2T}} = 0$ by adding a (small) penalty on non-idle delivery systems $\alpha_{d,t}^{\mathrm{D2T}}$ in the cost function.

Couple source powers to delivery system power. If a delivery system d is active at target t ($\alpha_{d,t}^{\mathrm{D2T}} = 1$), then the delivered power at this target $P_{d,t}^{\mathrm{D2T}}$ should equal the power delivered by the sources to this delivery system $P_d^{\mathrm{D,alloc}}$:

- Delivery system on at target: $\alpha_{d,t}^{\mathrm{D2T}} = 1 \quad \rightarrow \quad P_{d,t}^{\mathrm{D2T}} = P_d^{\mathrm{D,alloc}}$.
- Delivery system off at target: $\alpha_{d,t}^{\mathrm{D2T}} = 0 \quad \rightarrow \quad P_{d,t}^{\mathrm{D2T}} = 0$

The second bullet is imposed already by (C.2). The first bullet can be imposed by the following constraint:

$$0 \leq P_d^{\mathrm{D,alloc}} - P_{d,t}^{\mathrm{D2T}} \leq ((\sum_{t=1}^{T} \alpha_{d,t}^{\mathrm{D2T}}) - \alpha_{d,t}^{\mathrm{D2T}})P_d^{\mathrm{D,max}} \quad \forall \ d \in \{1, ...,D\}, \ t \in \{1, ...,T\} \tag{C.3}$$

C.2 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.

$$\sum_{d=1}^{D} \alpha_{s,d}^{\mathrm{S2D}} \leq 1, \quad \forall \ s \in \{1, ...,S\} \tag{C.4}$$

Each delivery system can be active at one target.

$$\sum_{t=1}^{T} \alpha_{d,t}^{\mathrm{D2T}} \leq 1, \quad \forall \ d \in \{1, ...,D\} \tag{C.5}$$

C.3 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 $M_{h,si}^{\mathrm{H2S}} = 1$ and $M_{h,sj}^{\mathrm{H2S}} = 1$) we require:

- Both sources active ($\sum_{d=1}^{D} [\alpha_{si,d}^{\mathrm{S2D}} + \alpha_{sj,d}^{\mathrm{S2D}}] = 2$): must have same power ($P_{si}^{\mathrm{S,alloc}} = P_{sj}^{\mathrm{S,alloc}}$)
- 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:

$$-(2 - \sum_{d=1}^{D} [\alpha_{si,d}^{\mathrm{S2D}} + \alpha_{sj,d}^{\mathrm{S2D}}])P_h^{\mathrm{H,max}} \leq P_{si}^{\mathrm{S,alloc}} - P_{sj}^{\mathrm{S,alloc}} \leq (2 - \sum_{d=1}^{D} [\alpha_{si,d}^{\mathrm{S2D}} + \alpha_{sj,d}^{\mathrm{S2D}}])P_h^{\mathrm{H,max}} \tag{C.6}$$

where $P_h^{\mathrm{H,max}} = \sum_{s=1}^{S} M_{h,s}^{\mathrm{H2S}} P_s^{\mathrm{S,max}}$ is the maximum power that can be delivered by a power supply h.

C.4 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.

$$\alpha_{d,t}^{\mathrm{D2T}} = \begin{cases} \{0,\ 1\}, & \text{if } M_{s,d}^{\mathrm{S2D}} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{C.7}$$

Target is out of reach of a delivery system.

$$\alpha_{d,t}^{\mathrm{D2T}} = \begin{cases} \{0,\ 1\}, & \text{if } \rho_d^{\min} \le \rho_t^{\mathrm{req}} \le \rho_d^{\max} \\ 0, & \text{otherwise} \end{cases} \tag{C.8}$$

Delivery system type is not allowed at target. As the function $D_t^{\mathrm{allow}}(D_d^{\mathrm{type}})$ is 1 if delivery system d is allowed to be used for target t, and 0 otherwise, the constraint can be written as:

$$\alpha_{d,t}^{\mathrm{D2T}} = \begin{cases} \{0,\ 1\}, & \text{if } D_t^{\mathrm{allow}}(D_d^{\mathrm{type}}) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{C.9}$$

Appendix D. Remarks on alternative problem formulations

D.1 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 $P_{s,t}^{\mathrm{S2T}}$ and binary variables $P_{s,t}^{\mathrm{S2T}}$ as the optimization variables. All relevant costs and constraints can than 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.

D.2 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 $\|(P_t^{\mathrm{T,alloc}} - P_t^{\mathrm{req}})\|$ instead of $(P_t^{\mathrm{T,alloc}} - P_t^{\mathrm{req}})^2$. 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 work a MIQP-formulation, which is more flexible and transparent due the absence of the auxiliary variables and constraints.

Appendix E. Practical issues

Here we discuss some practical and more technical issues.

E.1 Normalization factors

The normalization factors $P_t^{\mathrm{req,norm}}$ in (5), $I_t^{\mathrm{req,norm}}$ in (6) and $P_s^{\mathrm{S,pres,norm}}$ (8) are chosen such that all entries are normalized individually, but also to avoid dividing by zero (or numbers close to zero):

$$P_t^{\mathrm{req,norm}} = \max(P^{\mathrm{norm,min}}, P_t^{\mathrm{req}}) \tag{E.1}$$

$$I_t^{\mathrm{req,norm}} = \max(I^{\mathrm{norm,min}}, I_t^{\mathrm{req}}) \tag{E.2}$$

$$P_s^{\mathrm{S,pres,norm}} = \max(P^{\mathrm{norm,min}}, P_s^{\mathrm{S,max}}) \tag{E.3}$$

where $P^{\mathrm{norm,min}}$ and $I^{\mathrm{norm,min}}$ should take values corresponding to the considered actuator system, we choose in the ITER examples $P^{\mathrm{norm,min}} = 0.1$ and $I^{\mathrm{norm,min}} = 0.001$.

E.2 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 $P_{d,t}^{\mathrm{D2T}}$ in (C.2) can help to realize this for the active flags between delivery systems and targets ($\alpha_{d,t}^{\mathrm{D2T}}$). A tighter bound can be formulated by including the target power request P_t^{req} in computing the bound:

$$P_{d,t}^{\mathrm{D2T,max}} = \min((1 + f^{P,\mathrm{add}})P_t^{\mathrm{req}}, P_d^{\mathrm{D,max}}). \tag{E.4}$$

Choosing $f^{P,\mathrm{add}} > 0$ allows to allocate more power than requested at a target, which may be helpful to get a better current matching. We choose $f^{P,\mathrm{add}} = 0.1$ in the examples.

References

[1] D.A. Humphreys, R.D. Deranian, J.R. Ferron, A.W. Hyatt, R.D. Johnson, R.R. Khayrutdinov, R.J. La Haye, J.A. Leuer, B.G. Penaflor, J.T. Scoville, et al., Integrated plasma control in DIII-D, Fusion Sci. Technol. 48 (2) (2005) 1249–1263.

[2] E. Joffrin, O. Barana, D. Mazon, P. Moreau, F. Turco, J. Artaud, V. Basiuk, C. Bourdelle, S. Brémond, J. Bucalossi, et al., Integrated plasma controls for steady state scenarios, Nucl. Fusion 47 (12) (2007) 1664.

[3] D.A. Humphreys, G. Ambrosino, P. De Vries, F. Felici, S. Kim, G. Jackson, A. Kallenbach, E. Kolemen, J. Lister, D. Moreau, et al., Novel aspects of plasma control in ITER, Phys. Plasmas (1994-present) 22 (2) (2015) 021806.

[4] J.A. Snipes, Y. Gribov, A. Winter, Physics requirements for the ITER Plasma Control System, Fusion Eng. Des. 85 (3) (2010) 461–465.

[5] J.A. Snipes, D. Beltran, T. Casper, Y. Gribov, A. Isayama, J. Lister, S. Simrock, G. Vayakis, A. Winter, Y. Yang, et al., Actuator and diagnostic requirements of the ITER Plasma Control System, Fusion Eng. Des. 87 (12) (2012) 1900–1906.

[6] W. Treutterer, D.A. Humphreys, G. Raupp, E. Schuster, J.A. Snipes, G. De Tommasi, M.L. Walker, A. Winter, Architectural concept for the ITER Plasma Control System, Fusion Eng. Des. 89 (5) (2014) 512–517.

[7] W. Treutterer, C. Rapson, G. Raupp, J. Snipes, P. de Vries, A. Winter, D. Humphreys, M. Walker, G. de Tommasi, M. Cinque, et al., Towards a preliminary design of the iter plasma control system architecture, Fusion Eng. Des. 115 (2017) 33–38.

[8] A. Winter, P. Makijarvi, S. Simrock, J.A. Snipes, A. Wallander, L. Zabeo, Towards the conceptual design of the ITER real-time Plasma Control System, Fusion Eng. Des. 89 (3) (2014) 267–272.

[9] M.A. Henderson, G. Saibene, C. Darbos, D. Farina, L. Figini, M. Gagliardi, F. Gandini, T. Gassmann, G. Hanson, A. Loarte, et al., The targeted heating and current drive applications for the ITER electron cyclotron system, Phys. Plasmas (1994-present) 22 (2) (2015) 021808.

[10] W. Treutterer, R. Cole, K. Lüddecke, G. Neu, C.J. Rapson, G. Raupp, D. Zasche, T. Zehetbauer, ASDEX Upgrade Team, ASDEX Upgrade discharge control system – a real-time plasma control framework, Fusion Eng. Des. 89 (3) (2014) 146–154.

[11] W. Treutterer, G. Neu, C.J. Rapson, G. Raupp, D. Zasche, T. Zehetbauer, ASDEX Upgrade Team, Event detection and exception handling strategies in the ASDEX Upgrade discharge control system, Fusion Eng. Des. 88 (6) (2013) 1069–1073.

[12] R. Nouailletas, P. Moreau, S. Brémond, O. Barana, F. Saint-Laurent, J.-F. Artaud, J. Bucalossi, L. Colas, A. Ekedahl, O. Semlali, et al., Plasma discharge management for long-pulse tokamak operation, Fusion Sci. Technol. 64 (1) (2013) 13–28.

[13] N. Ravenel, R. Nouailletas, O. Barana, S. Brémond, P. Moreau, B. Guillerminet, S. Balme, L. Allegretti, S. Mannori, Architecture of west Plasma Control System, Fusion Eng. Des. 89 (5) (2014) 548–552.

[14] C.J. Rapson, M. Reich, J. Stober, W. Treutterer, et al., Actuator management for ecrh at ASDEX Upgrade, Fusion Eng. Des. 96 (2015) 694–697.

[15] C.J. Rapson, F. Felici, C. Galperti, P.T. Lang, M. Lennholm, E. Maljaars, et al., Experiments on actuator management and integrated control at ASDEX Upgrade, Proceedings of 29th Symposium on Fusion Technology (SOFT 2016), Prague, 2016.

[16] G. Raupp, G. Pautasso, C. Rapson, W. Treutterer, J. Snipes, P. de Vries, A. Winter, D. Humphreys, M. Walker, G. Ambrosino, et al., Preliminary exception handling analysis for the iter plasma control system, Fusion Eng. Des. (2017).

[17] J.E. Barton, K. Besseghir, J. Lister, E. Schuster, Physics-based control-oriented modeling and robust feedback control of the plasma safety factor profile and stored energy dynamics in iter, Plasma Phys. Control. Fusion 57 (11) (2015) 115003.

[18] E. Maljaars, F. Felici, M.R. de Baar, M. Steinbuch, Model predictive control of the current density distribution and stored energy in tokamak fusion experiments using trajectory linearizations, IFAC-PapersOnLine 48 (23) (2015) 314–321.

[19] E. Maljaars, H. van den Brand, F. Felici, C.J. Rapson, W. Treutterer, O. Sauter, M.R. de Baar, Simultaneous control of plasma profiles and neoclassical tearing modes with actuator management in tokamaks, 42nd EPS Conference on Plasma Physics, European Physical Society, 2015.

[20] M.J. Singh, Status of heating and current drive systems planned for iter, IEEE Trans. Plasma Sci. 44 (9) (2016) 1514–1524.

[21] G.L. Nemhauser, L.A. Wolsey, Integer and Combinatorial Optimization, Wiley-Interscience, New York, 1988.

[22] G.M. Appa, L.S. Pitsoulis, H.P. Williams, Handbook on Modelling for Discrete Optimization vol. 88, Springer Science & Business Media, 2006.

[23] IBM, IBM ILOG CPLEX Optimization Studio, Version 12.6.2, (2016) http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/.

[24] Inc. Gurobi Optimization, Gurobi Optimizer Reference Manual, (2016) http://www.gurobi.com.

[25] H. Mittelmann, MIQP Benchmark, (2016) Available from: http://plato.asu.edu/ftp/miqp.html [29.11.16].

[26] The MathWorks, Matlab Release 2015a, (2015).

[27] M.A. Henderson, Personal Communication, (2016).

[28] O. Sauter, M.A. Henderson, G. Ramponi, H. Zohm, C. Zucca, On the requirements to control neoclassical tearing modes in burning plasmas, Plasma Phys. Control. Fusion 52 (2) (2010) 025002.

[29] H. van den Brand, M.R. de Baar, N.J. Lopes Cardozo, E. Westerhof, Integrated modelling of island growth, stabilization and mode locking: consequences for ntm control on iter, Plasma Phys. Control. Fusion 54 (9) (2012) 094003.

[30] D. Farina, M.A. Henderson, L. Figini, G. Saibene, Optimization of the ITER electron cyclotron equatorial launcher for improved heating and current drive functional capabilities, Phys. Plasmas (1994-present) 21 (6) (2014) 061504.

[31] L. Figini, D. Farina, M.A. Henderson, A. Mariani, E. Poli, G. Saibene, Assessment of the ITER electron cyclotron upper launcher capabilities in view of an optimized design, Plasma Phys. Control. Fusion 57 (5) (2015) 054015.

[32] J. van Dongen, F. Felici, G.M.D. Hogeweij, P. Geelen, E. Maljaars, Numerical optimization of actuator trajectories for ITER hybrid scenario profile evolution, Plasma Phys. Control. Fusion 56 (12) (2014) 125008.