EUROfusion

WPMST1-PR(18) 21169

T Vu et al.

Tokamak-agnostic actuator management for multitask integrated control with application to TCV and ITER

Preprint of Paper to be submitted for publication in Nuclear Fusion

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.

Tokamak-agnostic actuator management for multi-task integrated control with application to TCV and ITER

N.M.Trang VU*[1], T.C. BLANKEN[2], M. KONG[1], E. MALJAARS[2], F. FELICI[1], C. GALPERTI[1], O.SAUTER[1], the TCV team[3] and the EUROfusion MST1 team[4]

[1]École Polytechnique Fédérale de Lausanne (EPFL), Swiss Plasma Center (SPC), CH-1015 Lausanne, Switzerland

[2]Eindhoven University of Technology, Department of Mechanical Engineering, Control Systems Technology Group, P.O. Box 513, 5600 MB Eindhoven, The Netherlands

[3]See the author list of S. Coda et al 2017 Nucl. Fusion 57 102011

[4]See the author list of H. Meyer et al., 2017 Nucl. Fusion 57 102014

E-mail: `trang.vu@epfl.ch`

Abstract. The plasma control system (PCS) of a long-pulse tokamaks must be able to handle multiple control tasks simultaneously, and must be capable of robust event handling with a limited set of actuators. For ITER, this is particularly challenging given the large number of actuator-conflicting control requirements. In this work, a general PCS architecture is developed using a so-called task-based approach to deal with these issues. In this approach, a plasma supervisor and actuator manager make high-level decisions on how to handle the considered control tasks, using generic actuator resources and controllers. This simplifies the interface for operators and physicists since the generic control tasks (instead of controllers) can be directly defined from the general physics goals. The task-based approach also allows one to decompose the PCS into an *interface layer* (tokamak-dependent layer) and a *task layer* (tokamak-agnostic layer). The developed scheme is first implemented and tested on TCV for simultaneous β control, neoclassical tearing mode (NTM) control, central co-current drive, and H-mode control tasks. It is then applied to an ITER test scenario to prove its flexibility and applicability to systematically handle a large number of tasks and actuators.

Keywords: plasma control system, actuator management, plasma integrated control

1. Introduction

In order to fulfil the global physics goals in complex experiments on long-pulse tokamaks, a plasma control system (PCS) should aim to simultaneously reach multiple control objectives (control tasks) using a limited set of actuators [1]. It must solve the possibly conflicting requests of multiple control tasks and actuator sharing issues during normal operation. At the same time, undesired plasma events or hardware failures can suddenly occur and deteriorate the plasma as well as possibly damage the plasma facing components; for example a neoclassical tearing mode (NTM) that evolves into a locked mode can lead to a disruption, thus the PCS has to predict/detect these events and adapt its control solution to the new situation.

A combination of a supervisory controller and actuator manager have been proposed as a solution in the PCS design to manage various controllers and actuator sharing based on plasma and plant events [2, 3]. The plasma monitor and supervisory controller takes high-level control decisions on how to continue the discharge based on the observed state of the plasma. Then, the actuator manager determines the resource allocation for each controller and translates controller outputs to commands for the actuator control system. Some designs of PCS architecture have been proposed for AUG [4], DIII-D and KSTAR [5], WEST [6] and ITER [7].

The work in [2, 8] provides first examples of actuator manager for ASDEX-Upgrade, using Electron Cyclotron Resonance Heating (ECRH) system for NTM control. While this shows an excellent first demonstration of actuator management (AM) for this application, most of the decision for activating and prioritizing various NTM tasks is made by the NTM controller. When multiple control tasks and not only NTM control are to be considered, or multiple different actuators are to be managed, it would be beneficial to have strictly separate controllers, plasma monitoring pervision, and actuator management. On the other hand, [5] focuses on a supervisory logic capable of real-time off-normal event handling on DIII-D and KSTAR. In these examples, actuator management is done by direct selection of actuators via prioritization. A more optimal solution would require a more complex actuator assignment scheme as is proposed in this paper. In [3], a synthesis of possible architectures for integrated control and actuator sharing is presented, as well as a design of an efficient algorithm. It is argued that the choice of an appropriate PCS architecture depends on the scale and complexity of the tokamak actuator and diagnostics subsystems; for instance a pre-controller actuator manager (assigning resources before determining the controller requests) is recommended for tokamaks with a small number of actuators, and the post-controller actuator manager (assigning resources with knowledge of controller requests) for the tokamaks with complex actuator systems. The present work provides an extension and complete implementation of many of the ideas presented in [3], in particular demonstrating the first complete implementation of supervisory controller and actuator manager as well as experimental results on TCV.

In this work, we propose a generic PCS architecture which is capable of handling multiple control tasks in the same discharge within the limit of available actuators and of being easily adapted to every tokamak regardless of the complexity of the tokamak subsystems. The first contribution of this work is to define a *task-based approach* where a supervisory controller and an actuator manager handle control tasks using generic controllers and actuator resources. In a controller-based approach, the controllers can make high-level decisions on which control tasks to execute and which actuators to use. On the contrary, in the task-based approach, such high-level decisions are made by the supervisory controller and the actuator manager in order to resolve the controller conflict and actuator sharing issues, as well as to robustly handle abnormal plasma events. The task-based approach has several advantages:

- It provides an abstraction layer for operators or the physicists as they only need to specify the control tasks from physics goals (or pulse schedule), which are generic and similar among different tokamaks, regardless of the details of the relevant controllers and actuators. This greatly facilitates the interaction between operation and plasma control system software.
- It allows a separation of the components of the PCS into two layers: a tokamak-agnostic layer and an interface layer. The tokamak-agnostic layer can be easily adapted to other tokamaks irrespective of the complexity of the tokamak subsystems (diagnostics/actuators), while only the interface layer should be adapted for each tokamak.
- The fact that high-level decisions that depend on the plasma and actuator states are taken away from controllers makes it possible to design controllers in a more generic way, independent of the tokamak or the scenario specific properties.
- The task-based approach provides a natural way to prioritize the task execution order, which leads to clearly distinguished *parallel* and *sequential* tasks. Parallel tasks can be performed without being aware of the resource requests of the other

tasks, while sequential tasks depend on the resource requests of the other tasks.

The second contribution of this work is to propose a standardization of interfaces between the components in the tokamak-agnostic layer, including the supervisory controller, actuator manager and controllers. More precisely, the actuator manager and controller interfaces will be highlighted in the following while the supervisory controller is studied in detail in [9]. The standardized interfaces help to reduce implementation errors and improve maintainability, while granting the flexibility to add or remove controllers. Also, these standardized interfaces allow independent development and testing of each component of the tokamak-agnostic layer.

In Sec. 2, we first introduce the proposed architecture of the PCS for integrated control with the tokamak-agnostic layer and the interface layer. Then the task-based approach and each component in the tokamak-agnostic layer, including the supervisory controller, actuator manager and controllers are discussed in Sec. 2.1 and 2.2. The standardized interfaces between them are also presented in Sec. 2.2, and more details can be found in Appendix A. To clarify the idea of a task-based integrated control scheme, we analyze a concrete example in Sec. 2.3. The proposed solution is validated in simulation and implemented on TCV [10]. First experimental results are shown in Sec. 3, where integrated control of β, NTM and central heating and co-current drive are demonstrated. An application to a demonstration ITER scenario, studied in Sec. 4, confirms its applicability and flexibility to systematically handle a large number of tasks and actuators.

2. Generalized plasma control system structure

In this section, the proposed architecture of our generic PCS is presented. Fig. 1 shows the overview of this architecture comprising the tokamak system, the PCS and the user interface. The user interface allows the tokamak operator to configure the parameters of the PCS from the pulse schedule. The PCS is split into two layers: the tokamak-agnostic layer and the interface layer, using a task-based approach (2.1). The interface layer has to be designed for each tokamak but the structure may remain the same as proposed in this work (Fig. 1). It is composed of an actuator interface which converts the controller commands into the input signals for local actuator controller and generic plasma and actuator state reconstruction which provides real-time information on the plasma state, actuator states and limits, by merging information from multiple plasma diagnostics and local actuator control systems. This state reconstruction may include sophisticated algorithms suchas equilibrium reconstruction, profile estimation, ray tracing, etc. The output of the state reconstruction is a representation, in a generic form, of the overall state of the plasma and the key tokamak subsystems. This representation should not depend on the details of the diagnostics used to estimate the plasma state. One example of such a generic state reconstruction code is the RAPTOR-observer [11, 12] which is presently implemented on AUG and TCV.

The tokamak-agnostic layer contains a plasma state monitor, a supervisory controller, an actuator manager and controllers (detail in 2.2). It is worth noting that these components are tokamak independent in the sense that their layouts and the communication between them (their connection signals) are defined in a general way independent of the tokamak specific subsystems. However, their configuration parameters may depend on the specifics of the tokamak, diagnostics and actuators, as well as the control goals of a given discharge. These parameters are specified by the user via the user interface (Fig. 1). For example, a feedforward controller is defined with a general input-output interface like the other controllers (see Appendix A.4), the feedforward power is however determined by the user and is indeed different for each tokamak and plasma scenario.

2.1. Task-based vs. controller-based approach

A standard approach to develop an actuator manager is to directly link controllers and actuators together. We refer to this as a controller-based approach, and this is the paradigm followed e.g. in [2, 3]. In this approach, the user has to select an appropriate set of controllers to be used in order to fulfil the general physics goals (the pulse schedule). In general, the controllers are responsible for making high-level decisions

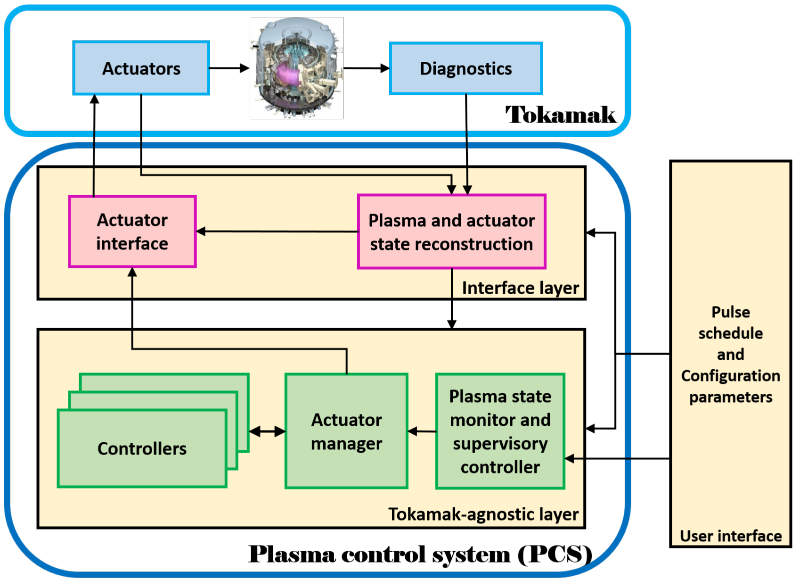

Figure 1: Overview of a generic tokamak control system architecture: the tokamak system and the plasma control system (PCS). The components in the PCS are split into two layers: the interface layer, and the tokamak-agnostic layer. The latter is tokamak independent and can be applied to different tokamaks.

on what they should do according to the plasma and actuator states. Therefore, they often include detailed knowledge about the specific subsystems of a tokamak. The actuator manager also needs to be aware of details of a particular controller in order to assign relevant actuator resources w.r.t the controller requests. Thus, the actuator manager and controllers built this way will be machine dependent.

The task-based approach we present here is based on the idea that, from the physics goals of each discharge, one can directly distill a set of high-level *control tasks* which are tokamak independent, such as β control, NTM control, H-mode entry control, plasma shape control, etc. In the task-based approach, high-level decisions on control tasks to be carried out are taken away from controllers and are made in a separate supervisory layer, making them tokamak independent.

Thus, we are able to define a tokamak-agnostic layer or *task layer* in the PCS (Fig. 1) which includes a plasma state monitor and supervisory controller, actuator manager and controllers. In this work, we only focus on the development of the *task layer*. Fig. 2 is a zoom of this layer with more details on the components as well

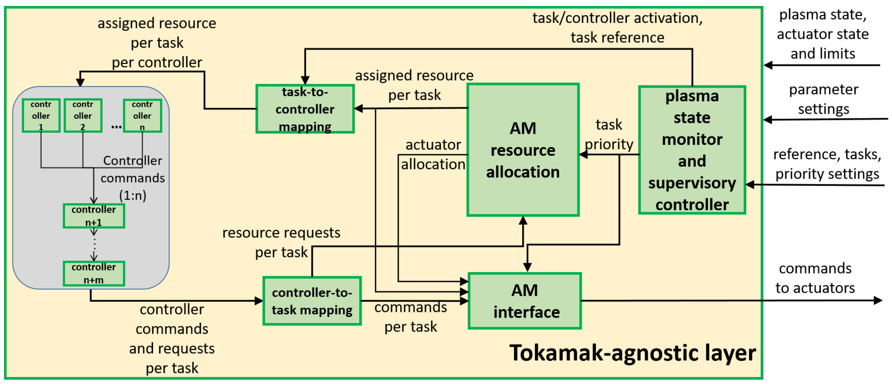

Figure 2: Schematic block diagram of the tokamak-agnostic *task layer* of the PCS, including supervisory controller, actuator manager and controllers

as on the signals propagating among them. These components are elaborated in the following subsections, while their interfaces are studied in detail in Appendix A.

2.2. Details of the task layer (tokamak-agnostic layer)

The tokamak-agnostic layer or *task layer*, Fig. 2, includes the plasma state monitor and supervisory controller; the actuator manager (AM) resource allocation and AM interface; the task-to-controller mapping and controller-to-task mapping; and the controllers. We call this the *task layer* since it specifically deals with the execution of control tasks, without requiring information on features of controllers, diagnostics and actuators (kind, functionality, etc.).

As inputs, the *task layer* receives the plasma state, the actuator state and limits from the plasma and actuator state reconstruction; the parameter settings for all components in the layer and the physics goals represented by the references, tasks, and priority settings via the user interface. These signals serve to make high-level decisions in the supervisory controller and to perform the AM resource allocation. As outputs, the commands to the actuators are sent to the actuator interface in order to convert these signals into the physical actuator commands.

Communication between the various components of this layer is performed using only task-based information, which requires an abstract representation of a task. Specifically, a task is considered as an action that needs to be taken by some actuator acting on the plasma. For example, a plasma current profile control task may require a specific amount of power deposited at a given deposition location and with a given direction of current drive from Heating and Current Drive (H&CD) actuators. A density control task may ask for the opening of a gas valve with a specific flow rate, etc. Nevertheless, we can generalize these requests for each task into the triplet (*amplitude*, *position*, and *type*). In this way, amplitude can represent either a power amplitude, gas flux amplitude, or pellet particle flux (injected particles per second), etc. Position is the location of actuation in the plasma for heating and current drive purpose, or the relative position of actuators (gas valves), etc. Type may be the direction for current drive, or other information which allows the AM resource allocation to classify the actuators that can fulfil the needs of each task. As a result, each actuator's state and capabilities also have to be parametrized in this generalized form. The actuators are enumerated (i), $i = 1, 2, 3, ...$ and their capacities are represented as ranges between the minimum and maximum values of (*amplitude*, *position*, *type*) and an activation signal for actuator state (*on* when it is capable of acting on the plasma, and *off* when it is not). Note that all signals in the *task layer*, except for activation and task parameter signals, are composed of such a triplet (*amplitude*, *position*, *type*). This abstract representation can be extended for more components when we expand the type of control tasks and actuators in the future work.

Inside the layer, the hierarchical architecture [3, 13, Sec. 2] is employed for the component connection. Highlighted by the advantages of transparency and ease of implementation, this architecture has been elaborated in various applications [7, 4, 6]. Here, we include the supervisory controller together with the actuator manager and controllers as shown in Fig. 2. The detailed function of each component in the *task layer* is described next.

- The *plasma state monitor and supervisory controller* computes the task priority and task activation based on the plasma state, the actuator state and limits.

Depending on the plasma evolution w.r.t. physics goals, the supervisory controller decides how to continue the discharge by activating relevant tasks and giving them appropriate priorities. It can change the references and task parameters according to the pulse schedule settings. The cross-coupling between control tasks is also taken into account by this supervisory level, e.g. the case when some control tasks cannot be performed simultaneously. Details of the supervisory controller within the scope of the generalized PCS architecture described in this paper are discussed in [9].

- The *AM resource allocation* assigns resources (e.g. range of amplitude, position, and type) to each activated task, by solving an optimization problem based on resource limits, task priority and resource requests per task. Details are described in Sec. Appendix A.1).
- The *AM interface* maps the resource commands from each control task to the corresponding actuators while ensuring these commands are within the assigned resources per task given by the AM resource allocation and within the actual actuator limits.
- The *task-to-controller mapping* and *controller-to-task mapping* provide the interface between the AM components and the controllers using the task-controller mapping. The task-to-controller mapping distributes tasks and the corresponding signals (activation, assigned resources and task parameters) to the different controllers. Conversely, the controller-to-task mapping combines all controller commands into commands per task for the AM interface, and all controller requests into resource requests per task for the AM resource allocation. Since multiple controllers can potentially fulfil the same task, the distribution of tasks to controllers (by the supervisory controller) has to be such that one task is assigned to one controller, in order to have a unique mapping between tasks and controllers.
- The *controllers* receive their assigned resources (e.g. ranges of amplitude, position, type) as well as plasma state information, and execute the (feedback)

control laws for their tasks. They send out the controller commands per task (commands of power amplitude, position, and type for the actuators) and the controller requests per task (a range of power amplitude, position, and type for the actuator manager). It is important to distinguish between these two outputs of the controllers: the controller commands (after being limited by the assigned resources) will be distributed to the actuators; while the resource requests, without limits, will be communicated to the AM to be considered for the actuator allocation in the following time step.

The controllers can be executed either in parallel or in sequence. The controllers in the parallel group (Fig. 2, controllers $1 \rightarrow n$) handle parallel tasks, which are mutually decoupled, meaning that these controllers can execute their control laws independently from the other controllers. On the contrary, the controllers in the sequential group (Fig. 2, controllers $n+1 \rightarrow m$) handle sequential tasks, which need to be aware of the commands of the other tasks. Hence, these controllers need to receive the commands of the other controllers so that they can compute their own outputs. An example thereof is the LH-mode controller, which is discussed in detail in Sec. 4.1. Note that the controllers interact only with the actuator manager via the task-to-controller mapping and the controller-to-task mapping; and there is no direct connection between controllers and the actuator systems.

This proposed interface makes it easy to add or remove tasks and controllers for each discharge without changing the generic interface. The interfaces of various components in this *task layer* are generally shown in Fig. 2 with main inputs and outputs. For more details of these interfaces, see Appendix A.

To clarify the task and controller concepts that we proposed in this work, a concrete example is discussed in Sec. 2.3 below.

2.3. Example of pulse schedule: tasks and controllers in a discharge

To illustrate the task-based approach described in the previous section, we present an example of integrated control featuring several tasks and several controllers. The physics goal, defined by the pulse schedule, is to control the plasma performance (e.g. plasma kinetic pressure β control) while maintaining an H-mode regime. In other words, the physics goal is not only to drive plasma current and β to their references, but simultaneously to ensure the total heating power is higher than the critical threshold P_{LH} to keep the plasma in H-mode. This threshold power is typically determined via a scaling law [14]. Also, during the discharge, NTMs may appear, so NTM control [15] is required to stabilize (or suppress) NTMs. It has been shown that preventing an NTM by preemption is more efficient than stabilizing it [16, 17], so our NTM control strategy will use a combination of both. The 2/1 and 3/2 NTMs (NTMs at the q surfaces $q = 2/1$ and $q = 3/2$ respectively) are the most significant ones in this example. Moreover in the worst case when the plasma is strongly destabilized, leading to a disruption, disruption mitigation [18] must be considered.

According to the pulse schedule described above, several tasks can be defined. These are listed in Table 1. The controllers capable of executing these tasks are enumerated in Table 2, and mapping between tasks and controllers are defined in Table 3. In order to avoid possible conflicts between controllers and between tasks, it has to be ensured that each task is assigned to only one controller.

Task 1	β control
Task 2	2/1 NTM stabilization
Task 3	2/1 NTM preemption
Task 4	3/2 NTM stabilization
Task 5	3/2 NTM preemption
Task 6	H-mode
Task 7	Disruption mitigation

Table 1: Example of task list in a discharge

Controller 1	performance controller
Controller 2	NTM controller
Controller 3	LH-mode controller
Controller 4	disruption mitigation valve (DMV) controller

Table 2: Example of controller list

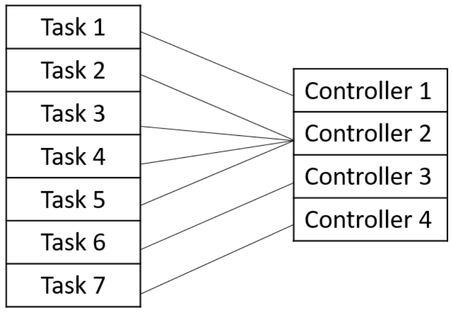

Table 3: Task-controller mapping, linking the tasks in Table 1 and the controllers in Table 2

At each PCS cycle time during the discharge, the supervisory controller will generate activation signals and priorities for relevant tasks (Table 1) according to priorities defined through the pulse schedule as well as the actual evolution of the plasma and actuator states. The AM resource allocation will assign the resources to the active tasks. The controllers will execute their control laws and send the commands (indirectly) to the actuators to achieve their tasks.

Note here the crucial difference between the task-based approach and a controller-based approach: in a controller-based approach, the NTM controller would have to decide if preemption or stabilization should be performed, while with the task-based approach, such a high-level decision is carried out by the supervisory controller and actuator manager instead.

Note as well that despite this being a relatively simple example of β control in H-mode, seven tasks and four controllers are needed, not to mention the actuators, but the generic nature of the interfaces helps to maintain this manageable even with more complicated cases.

3. First results on TCV

The proposed integrated control scheme has been tested and implemented on TCV. For this purpose, it was implemented in Matlab/Simulink [19], from which C code was generated for use on the TCV digital real-time control system [20], [21].

Two test scenarios are described in this section. In the first test, the physics goal is to stabilize a 2/1 NTM by Electron Cyclotron (EC) co-CD (current drive in the same direction as the plasma current). In TCV, creation of the mode also requires co-CD, but aimed at the center of the plasma [15, 17]. In the second test, β control is demonstrated using EC power, while switching to the control of a 2/1 NTM when the mode appears during the discharge. For both scenarios, only the power and injection angle of the EC H&CD system are controlled while other actuators such as the ohmic coils and gas valves are not considered.

3.1. First test: β control with NTM stabilization

In this test, the main goal of the discharge is to achieve β control and NTM stabilization is also considered in order to keep the discharge from disrupting. A central co-CD heating phase,(central co-CD), at the beginning of the discharge is used to establish the plasma in an operational equilibrium and eventually trigger the mode, before switching on β control and NTM stabilization tasks. For the NTM stabilization task, (NTM2/1 stab), the NTM controller will move the assigned actuator(s) to the target surface (e.g. surface $q = 2$) to stabilize the mode with full allocated power. During the time when the EC deposition is changing from the current position to the target, the NTM controller requests a lower power in order to limit perturbing the plasma.

Three tasks are thus considered as well as three controllers: a feedforward, an NTM and a performance controller [22]. These are listed in Table 4. Note that NTM2/1 stab is activated only if a 2/1 mode is detected by the plasma state monitor.

task	priority	activation time (s)	activation condition	controller
central co-CD	0.2	[0.4, 0.55]		feedforward
NTM2/1 stab	1	[0.5, 2.5]	2/1 NTM detection	NTM
β control	0.8	[0.5, 2.5]		performance

Table 4: TCV first test: table of tasks, controllers and the mapping between them

Two EC launchers (indicated as $L4$ and $L6$) are available for the considered tasks. Note that $L4$ and $L6$ have the same EC power supply, thus they always have (approximately) the same power, and can not be controlled with different power commands. However, their deposition location can be different and changed in real-time. This means that, if the higher priority task receives one (or both) of these actuators, this task controls the powers of both of them, whereas the other task, if it receives the remaining actuator, can only control the position of that actuator. As a result, the power range of the assigned resource for the second task is fixed at the actual power of the assigned actuators (i.e: $P_{min\,assigned} = P_{max\,assigned} = P_{actual}$). The AM interface will also take this fact into account when distributing power to each actuator.

Based on other experimental results on TCV, it is known that 2/1 NTMs can be triggered with these two EC launchers [15, 17]. On the other hand, β control request is fixed at $[0,\, 1MW]$ as the amplitude range for the EC power, the task can receive both $L4$ and $L6$ (with maximum $0.5MW$ each) when available. These task activations and priorities are managed in real-time by the supervisory controller.

In Fig. 3, the results of shot 58821 are presented. Initially, in the interval ①-②, central co-CD is the only activated task, hence it has the highest priority and is assigned both $L4$ and $L6$ (see Fig. 3a and b), after that it receives the lowest priority due to the activation of NTM2/1 stab and β control. At ②, and ⑥, a 2/1 NTM is detected, the supervisory controller sets the highest priority to NTM2/1 stab, and the second-highest priority to β control. Accordingly, the AM resource allocation allocates $L4$ to NTM2/1 stab and $L6$ to β control, and with nothing left for central co-CD. During the phases without NTM, starting at ③, β control always gets the highest priority and thus is assigned both $L4$ and $L6$, given the high power request from this task ($[0,\, 1MW]$).

Regarding the EC powers and launcher positions controlled by the two considered controllers (Fig. 3c and e), in the interval ①-②, maximum power of $L4$ and $L6$ as well as the central position are required by central co-CD. During NTM2/1 stab, $L4$ power is adapted by the NTM controller according to the distance between it and the target ($\rho\,(q = 2) \approx 0.55$). During the β control-only period, the central position and the power of both launchers are decided by the performance controller. One can notice

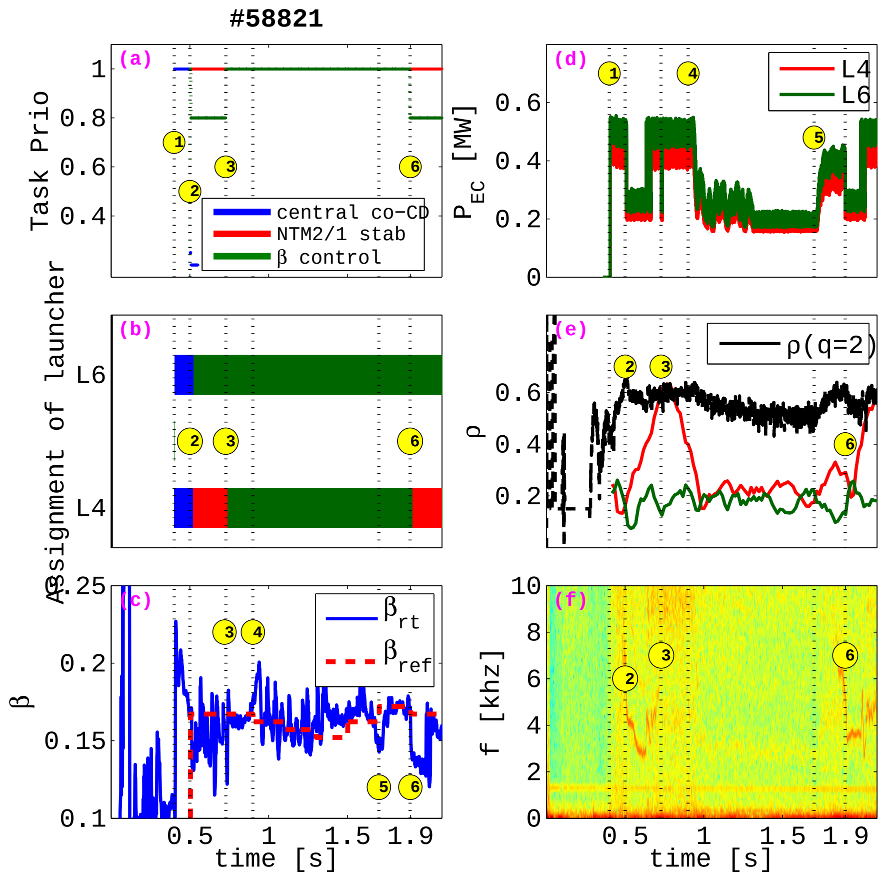

Figure 3: TCV shot 58821: result of integrated control experiment with three control tasks and two EC actuators in TCV, showing a sequence of events: ① central co-CD is activated; ② An NTM is coincidentally detected at the same time as activation of β control. L4 is assigned to stabilize the mode and moved to the $q = 2$ surface. ③ The NTM is stabilized and β control has the highest priority. ④ β_{ref} is decreased, the power command reaches the actuator limits and β oscillates. ⑤ β_{ref} is increased, β recovers and triggers an NTM at ⑥. (a) Task priorities decided by the supervisory controller. (b) Launcher allocation from the AM resource allocation, same legend as (a). (c) Estimated real-time β_{rt} vs β_{ref}. (d) EC powers of $L4$ and $L6$. (e) Launcher $L4$ and $L6$ positions vs NTM position. ρ corresponding to $q = 2$ surface, same legend as (d). (f) NTM freq spectrum shows the presence of the mode.

that the performance controller sends the high power request to the AM resource allocation in order to get both launchers, but it can use lower power as commands sent to these launchers (in the interval ④-⑤).

It should be noted that in the presence of the NTM, β cannot reach the reference since its available power is insufficient. Without NTM (③-④ and ⑤-⑥), the real-time β matches quite well the reference. However, during ④-⑤, the real-time β oscillates and cannot track the reference which is now so low that the lower limits of the EC power are reached. After ⑤, the β reference increases again, so that successful control can again be achieved. The higher β however triggers an NTM at ⑥, which is again detected and handled by the system.

3.2. Second test: β control in L and H-mode

The main objective of this test is to make the plasma β to track a reference trajectory, both in L and H-mode regimes. To implement this, both parallel tasks (feedforward FF control and β control) and sequential tasks (the LH-mode tasks: be in L-mode and be in H-mode) are required. The β control task is accomplished by feedback controlling the power of some actuators based on the tracking error. The be in L-mode task and the be in H-mode task (see Appendix B for details) are alternatively active during the discharge. These LH-mode tasks are accomplished by constraining the total input power to the plasma to a maximum/minimum in order to keep it below/above the LH transition threshold respectively. If the total power required by the other control tasks is already within the specified limits by the LH-mode tasks, then no change to the power is requested. Otherwise, the be in H-mode task will request additional power in order to reach its threshold, and the be in L-mode task will request to subtract the exceeding power.

This example serves to illustrate clearly the difference between parallel and sequential tasks (see Sec. 2.2 for details). The LH-mode tasks are sequential since their commands depend on the commands of all the other controllers, hence they are executed (sequentially) after the others. The other tasks are parallel, since they do not depend on each other and can be executed independently.

The considered tasks and controllers in this test are listed in Table 5. In this discharge, three EC launchers $L7$, $L8$ and $L9$ and one NBI are available with different power limits described in Table 6.

task	priority	activation time (s)	controller
FF control	0.45	[0.6, 0.7]	feedforward
β control	0.6	[0.7, 1.8]	performance
be in L-mode	0.8	[0, 0.9] & [1.3, 1.8]	LH-mode
be in H-mode	1	[0.9, 1.3]	

Table 5: TCV second test: table of tasks, controllers and the mapping between them

actuator kind	name	power limits [MW]
EC	$L7$	[0.1, 0.5]
	$L8$	[0.1, 0.2]
	$L9$	[0.1, 0.5]
NBI	NBI	[0.05, 1.05]

Table 6: TCV second test: available actuators and their power limits

Fig. 4 shows the results of shot 62441. The be in H-mode task is active in the interval ③-⑥, and the be in L-mode task is active during the rest of the discharge. At first, FF control (feedforward) is activated (①-②) and is assigned $L9$. Then from ② until the end of the discharge, β control is switched on and FF control is switched off. The NBI is assigned to β control thanks to its high reliability, but it only fires $0.1s$ later for some technical reasons that will be discussed later. Since this is not known to the controller, the integrator term in the β controller continues integrating the error, eventually causing an overshoot on β when the NBI fires (Fig. 4c). The control of β is then regained and β is decreased to reach the reference. The power of $L9$, when not in use, is asked to decrease to its minimum, but then the corresponding gyrotron trips and is unavailable for the rest of the discharge. When the be in H-mode task is active at ③, the LH-mode controller compensates the missing power in order to bring the plasma to H-mode and to remain in H-mode during ③-④. Consequently in this interval, β can not reach the low reference which is unachievable in H-mode. During ④-⑤, β matches very well the reference in H-mode. In the interval ⑤-⑥, the situation is similar to the phase ③-④ when the β reference is again decreased, the heating power is thus decreased to the threshold which maintains the plasma in H-mode. At ⑥, the be in L-mode task is activated again, the β controller can freely

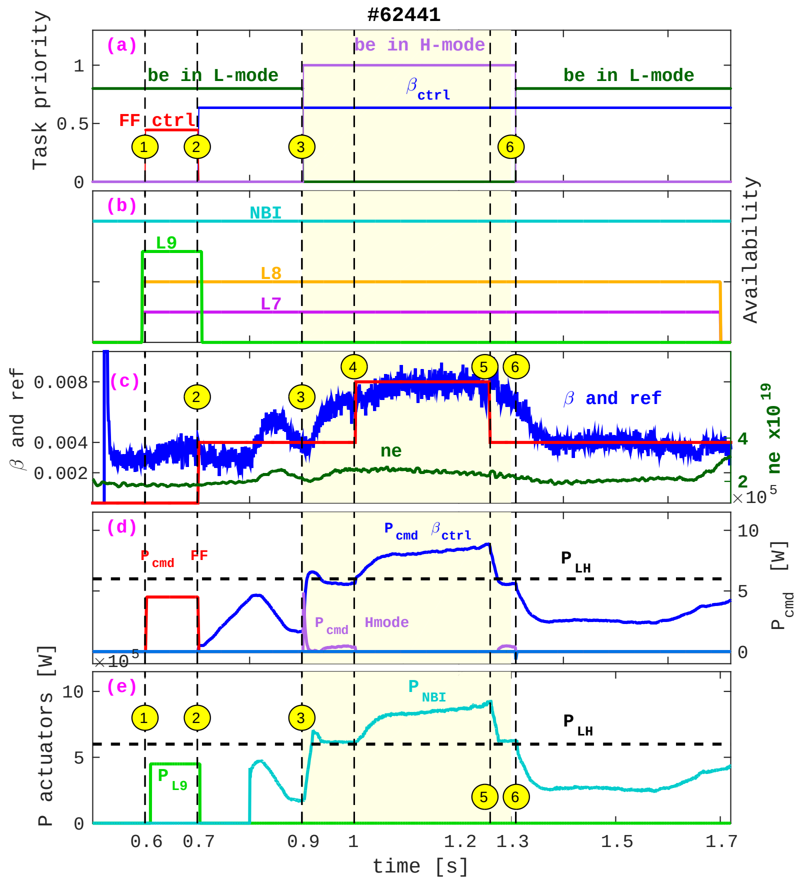

Figure 4: TCV shot 62441: result of an integrated control experiment with four control tasks and four actuators in TCV, showing a sequence of events: ① FF control is activated and is assigned $L9$; ② β control is on and FF control is off. β control is assigned NBI. NBI switches on some time later and the reference β is reached after an overshoot. ③ be in H-mode is active. LH-mode controller puts more power to NBI to go to and maintain H-mode. ④ β_{ref} is increased, the power command for NBI increases accordingly and β reaches the reference in H-mode. ⑤ β_{ref} is decreased, but since the plasma should still stay in H-mode phase, NBI power is decreased but still higher than the limit $P_{LHHmode}$, β decreases but can not reach the reference. ⑥ the be in L-mode is active, the NBI power can decrease and β reaches the reference in L-mode. (a) Task priorities decided by the supervisory controller. (b) Actuator availability. (c) Estimated real-time β vs β_{ref} and plasma density ne showing the L and H-mode transition. (d) Power commands from each task. (e) Readback power from actuators.

decrease the NBI power and β tracks very well the reference in L-mode.

Note that we have not aligned the time evolution of the β references with the

L- and H-mode phases on purpose, to test the supervisory controller, the actuator manager and the sequential tasks. Therefore when a low β is requested but the be in H-mode task is still active with a higher priority, the system is forced to provide the lowest power while staying in H-mode, as expected.

Via these examples, we have shown that the supervisory controller and actuator manager can adequately manage tasks and allocate actuators in a proper way. Note in particular that the successful test of generic sequential tasks, which is a first demonstration to our knowledge. All scenarios were executed using the same architecture (Fig.1) but with different configuration parameters of the pulse schedule (number of tasks and controllers). We stress that the time-behavior of the controllers was not programmed. The time-evolution in the experiments is solely a result of the high-level specification of the objectives of the discharge, and the automated actions of the control system. This also means that all necessary knowledge of the plasma state and actuator states need to be available in RT. For example in the second test, the starting time for the NBI was set before the shot by the NBI operator to $0.8s$, but allowed to be controlled by the RT-system over the whole shot. Therefore the actuator manager received the information that NBI was available before it was actually the case. This is why NBI started only at $0.8s$ although it was requested before. This will be corrected for future experiments, but illustrates well the degree of integration of all the systems that need to be obtained and included in the overall RT control strategy.

In the following section, the scheme will be applied to the more complex actuator set of ITER.

4. Integrated control test scenario and simulation for ITER

In order to further illustrate the capabilities of the integrated control scheme proposed in this paper, it is applied to a simulation of the ITER control system. This system is chosen due to the complexity of its actuator systems [23] and the necessity to handle a multitude of tasks simultaneously. In this section, we describe a test scenario for ITER and make some assumptions for the actuator system for the sake of simplicity. This simulation does not include any model of the plasma, instead, the task activation from the supervisory controller will be specified by hand, mimicking a likely evolution of the plasma, in order to demonstrate that the AM resource allocation can appropriately allocate the actuator resources to relevant tasks in such a case.

4.1. Description of the test scenario

In this test scenario for ITER, we aim at simultaneous control of β and q-profile in an H-mode regime with NTM stabilization at different q surfaces ($q = 2/1$ and $q = 3/2$). A sawtooth control (at $q = 1$ surface) is also considered since long period sawtooth can readily destabilize NTMs [24]. In this example, a fixed power onto the target surface is desired for sawtooth control; thus this task is considered functionally equivalent to NTM preemption at the $q = 1$ surface and is handled by the same controller, now referred to as MHD controller.

A sequential task be in H-mode (or H-mode, see Appendix B) is also active during the discharge. This task ensures that the total power injected in the plasma is larger than a threshold power P_{LH} (fixed at $52\,MW$ for this example) which is needed to bring and maintain the plasma in H-mode.

All considered tasks and their activation times are listed in Table 7. To fulfil these tasks, three controllers are employed: an MHD controller, a performance controller, and an LH-mode controller. The corresponding task-controller mapping is also shown in Table 7.

Task index	task name	priority	activation time (s)	controller
1	NTM2/1 stab	1	[2.5, 3.5] and [5, 7]	MHD
2	NTM3/2 stab	0.8	[2, 4] and [6, 7.5]	
3	sawtooth control	0.2	[1.5, 8.5]	
4	q-profile control	0.6	[1.2, 8]	performance
5	β control	0.5	[1, 8.3]	
6	H-mode	0.15	[0.1, 10]	LH-mode

Table 7: Task list of the ITER test

For simplicity, we assume constant resource requests from each task (specified in Table 8). Their resource requests, which are sent to the AM resource allocation,

Task index	Request index	$P_{req}\,(MW)$	ρ_{req}	icd_{req}
1	1	[6, 10]	[0.5, 0.6]	[0.5, 1]
2	2	[3, 5]	[0.3, 0.4]	[0.5, 1]
3	3	[1, 5]	[0.1, 0.2]	[0.5, 1]
4	4	[1, 4]	[0, 0.2]	[−1, −0.5]
	5	[5, 8]	[0, 0.2]	[0.5, 1]
5	6	[10, 15]	[0, 1]	[−1, 1]
6	7	[0, 0]	[0, 1]	[−1, 1]

Table 8: Controllers, tasks and fixed requests of ITER test

consist of a range of requested power P_{req}, normalized position ρ_{req} and type icd_{req}. This last quantity allows us to classify the current drive effect of each actuator, as follows:

- $icd \in [0.2, 1]$: Co-current
- $icd \in [-1, -0.2]$: Counter-current
- $icd \in (-0.2, 0.2)$: Heating

4.2. Assumptions on actuator and controller systems

Of the approximately 20 actuators that will eventually be connected to the ITER PCS, we will consider here only the Electron Cyclotron (EC), Ion Cyclotron (IC) and Neutral Beam Injection (NBI) heating and current drive systems. We propose the following simplifying assumptions on the actuator systems for the purpose of this illustration (though the proposed solution is equally applicable to the full system):

- The EC actuators are grouped into 4 EC groups which deliver the total power of $20MW$ into the plasma:
  - 1 Equatorial Launcher (EC_EL) with 3 mirrors is split into 2 groups: the first group has 1 mirror for counter-CD (CD in the opposite direction from the plasma current), named -1EC_EL (-1 stands for counter-CD); and the second group has 2 mirrors for co-CD, named +1EC_EL (+1 stands for co-CD).
  - 4 Upper Launchers (EC_UL) with 2 mirrors of each launcher for off-axis co-CD are split into 2 groups of +1EC_UL.

  - The EC system has in total 24 gyrotrons (power supply) of about $0.83MW$ each (effective power delivered to the plasma). In this example, we assume fixed connections between gyrotrons and launchers such that each EC group connects to 6 gyrotrons and thus can deliver $5MW$ into the plasma. This is a simplification with respect to the real situation, whereby some gyrotrons can connect to 2 mirrors and some others to 3 mirrors (detail in [25]).

- The IC system, for central heating purpose, has 2 antennas which deliver $20MW$ into the plasma, thus $10MW$ max per antenna.
- The NBI system has 2 sources of $16.5MW$ each, which inject current in the co-current direction as well as heating the plasma.. The NBI power can be only *on* or *off* ($P_{NBI} \in \{0, 16.5MW\}$) and the deposition location, which is assumed to be near the center in this scenario, cannot be changed.

Notice that all actuators can in principle be assigned to any considered tasks. However, NTM stabilization and sawtooth control is proved more efficient using co-ECCD [26], [24]. Consequently, only co-ECCD will be used for the tasks performed by the MHD controller and all others are marked as *excluded* actuators.

In this test, all controllers (except for the H-mode controller) give a command equal to the maximum assigned power from the AM resource allocation. The H-mode task computes its required additional power Appendix B (which may be zero if the other controllers already request sufficient power).

The results of actuator allocation for this test case are shown in Sec. 4.3. In this test scenario we consider only a $200s$ period during the H-mode flat-top for simplicity. First, we show a result for the normal case when all actuators work perfectly as intended. In the second example, we demonstrate the system's ability to compensate for to the loss of one NBI source during the discharge.

4.3. ITER test results

Fig. 5 presents the result for the first test: the prescribed time-varying task priorities (top panel) and computed actuator allocation for each task (middle panel) are shown.

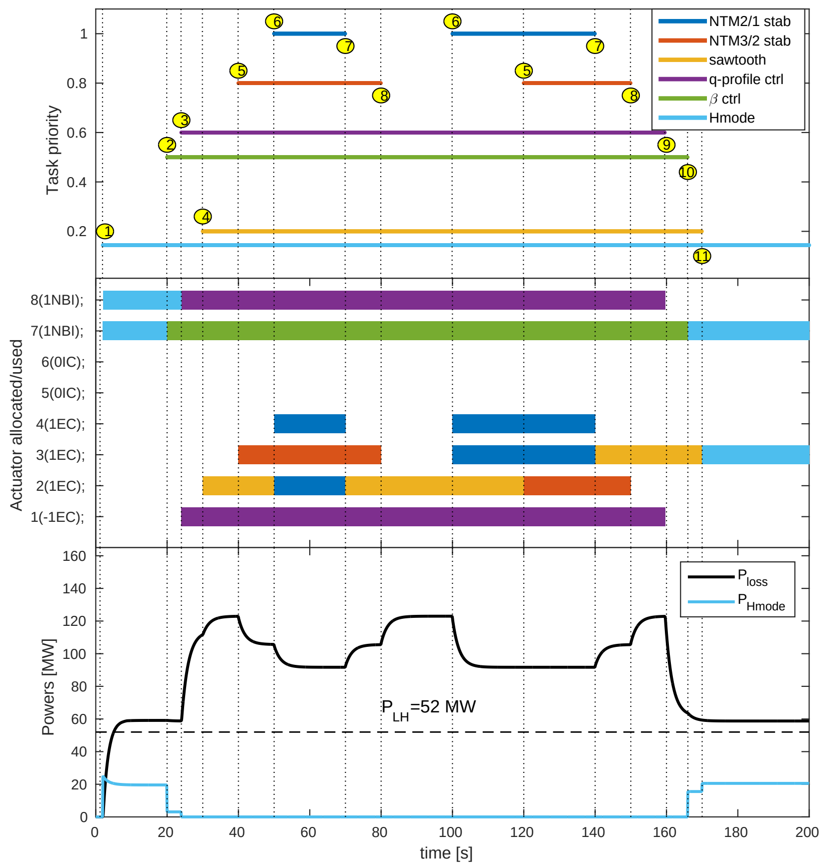

Figure 5: ITER test scenario 1. Time-varying task priorities are prescribed and shown in (top). Corresponding actuator allocation, computed by the actuator management for each task (middle). Power across the separatrix P_{loss}, (assumed equal to the input power here), and H-mode task power command (bottom). ① H-mode is activated, power is increased to the level required for H-mode. ② β control is activated; ③ q-profile control is activated; ④ sawtooth control is activated; ⑤ a 3/2 NTM is detected, various actuators are assigned to suppressing it; ⑥ a 2/1 NTM is detected, actuators are re-prioritized the higher-priority task of suppressing it; ⑦ 2/1 NTM is stabilized; ⑧ 3/2 NTM is stabilized, all actuators returned to regular control tasks; ⑨ q-profile control is turned off; ⑩ β control is turned off; ⑪ sawtooth control is turned off, H-mode task is the only active task and it ensures the total injected power remains above the threshold.

As can be seen in the top panel, H-mode is enabled for the whole discharge (①). This task requests sufficient heating power to stay above the power threshold (Fig. 5, bottom, see Appendix B for the computation of P_{Loss}), but this is necessary only at the very beginning and the end of the discharge when it is the only active task. For this purpose, it receives actuators 7, 8 at the beginning and actuators 3, 7 at the end. After a small amount of time, β control and q-profile control are also activated (② and ③) and are assigned their actuators. In particular q-profile control, which requires both co-CD and counter-CD, is assigned actuators 8 and 1, which have this opposite current drive capability. β control is given actuator 7 which fully satisfies its requests. NTM2/1 stab, NTM3/2 stab and sawtooth control share three co-ECCD actuators 2, 3 and 4. NTM2/1 stab, which has the highest priority when a 2/1 NTM is detected (⑥), gets two EC actuators; while NTM3/2 stab and sawtooth control receive one actuator each (⑤ and ④). sawtooth control gets nothing if the 2/1 and 3/2 NTMs happen at the same time, since it has a lower priority.

The second test, Fig. 6, shows the case when actuator 8, the second NBI, which was being used for q-profile control, suddenly trips (⑫, black stripe, middle panel). Immediately, a new actuator allocation is calculated for q-profile control, β control as well as for H-mode to again satisfy all their requests. The actuator 7 is given to q-profile control to replace the role of the tripped actuator 8, thus both IC actuators 5 and 6 are allocated to β control, and finally more actuators are assigned to H-mode at the end of the discharge than the previous case (Fig. 5) to fulfil this task.

5. Conclusion

A generic architecture for task-based integrated plasma control is developed in this work. Using a *task-based* approach for integrated control, we have built a *task layer* composed of a supervisory controller, an actuator manager and controllers, which is completely tokamak-agnostic. As a consequence, the proposed *task layer* can be widely and easily applied to different tokamaks. The developed supervisory controller and actuator manager can deal with the actuator sharing issue for multiple plasma control

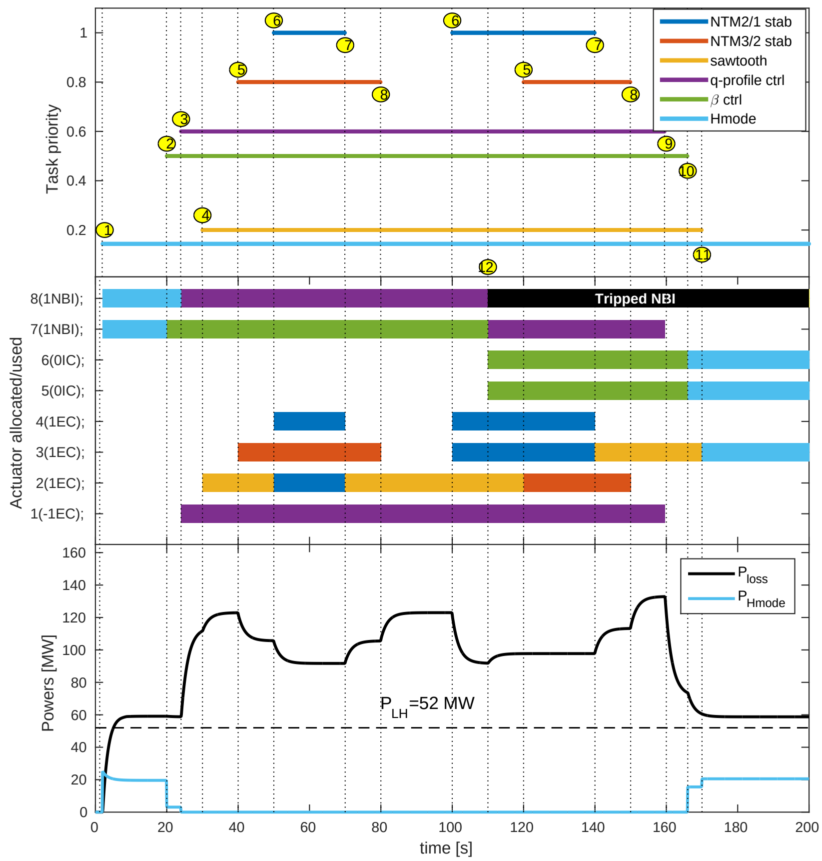

Figure 6: ITER test with a tripped NBI source: Task priority and actuator allocation for each task with its corresponding color ①-⑪ same as Fig. 5; ⑫ second NBI trips

tasks in the same discharge. This work has shown the design, implementation and experimental demonstration of actuator management for integrated control. Generic, standardized interfaces between the various components were developed. The scheme has been implemented in TCV and successfully used in experiments. Simulations for ITER confirm the ability to systematically and dynamically allocate shared actuators to simultaneously active tasks of the proposed work also for large amounts of actuators and tasks.

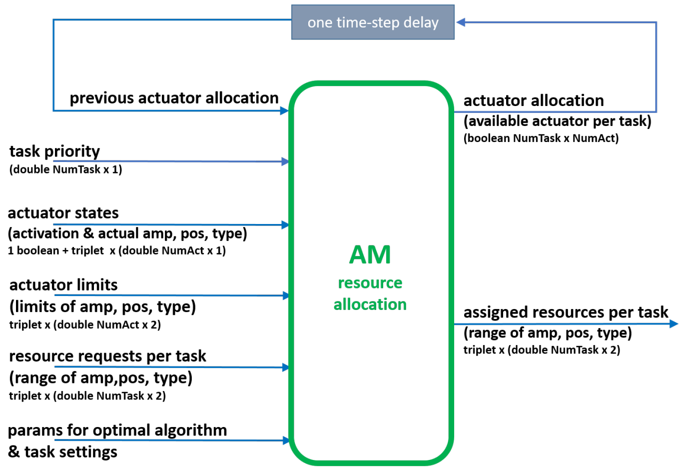

Figure A1: AM resource allocation inputs and outputs including signal names and sizes corresponding to the H&CD control cases discussed in this paper.

In the future, the actuator allocation algorithm will be improved based on [3] to minimize the computation time when the number of control tasks and actuator constraints is larger, as in the full ITER case. We also aim to extend the developed scheme to other tokamaks, in particular those where plasma monitoring, actuator sharing and integrated control are crucial to fulfil operational requirements.

Appendix A. Task layer components and interfaces

In this section the components of the *task layer* are described in more detail, also clarifying the interface of each component in Fig. 2. We present the various interfaces composed of AM resource allocation, AM interface, controllers, task-to-controller mapping, and controller-to-task mapping components, while the plasma state monitor and supervisory controller interface is elaborated in [9]. Each interface is presented with its inputs and outputs, the meaning and their associated signal types and sizes.

Appendix A.1. Actuator Manager resource allocation

In this subsection, we describe in detail the AM resource allocation algorithm and its interfaces Fig. A1.

Inputs

- *previous actuator allocation*: the result of actuator allocation (refer to AM resource allocation outputs) at the previous time step. It allows the AM resource allocation algorithm to skip new computation if the previous solution still fully satisfies the requirement at the actual time step.
- *task priority*, defined by the supervisory controller, represents the normalized priority (values between [0, 1]) of each considered task. Disabled tasks have priority 0. Using a hierarchical approach, to avoid ambiguity in allocating actuators to tasks, the supervisory controller has to ensure that active tasks never have the same priority value.
- *actuator states* represent the present state of each actuator including the activation signal and the triplet of (actual amplitude, position and type). Specifically, the amplitude gives absolute values in the appropriate unit; the radial position is normalized values (e.g.$\in$ [0, 1]), and if we consider the heating and current drive actuators, the type is the direction of current drive (in the range [-1, 1], negative value is for counter-current, positive value for co-current and value near 0 for heating). These states come from the local actuator control system and diagnostics via the plasma and actuator state reconstruction in the interface layer.
- *actuator limits* give the operational limits (min-max) per actuator, for each state of (amplitude, position and type). They may be ranges of power, position or direction for H&CD actuators, or the range of amplitude for gas valves, etc. Other important information such as the maximum rates of changing of actuator output, delays and bandwidth of the actuator can also be considered. These limits are given by the local actuator control system.
- *resource requests per task* are the requests for actuation resources (again specified by the range of amplitude, position and type) ideally needed to execute each task. In case where the range is not critical for some requests, a full range is given; for example, the β control task is insensitive to whether an H&CD

actuator also performs co- or counter-current drive, so the requested type range is $[-1, 1]$. The resource requests per task form the main requirement that the AM resource allocation has to attempt to satisfy within the limits of actuators and the constraints of the global physics goals.

- *parameters for optimal algorithm and task settings* include pre-defined settings for the AM resource allocation optimization algorithm (such as the cost function weights), as well as some specific settings for tasks (such as preferred or excluded actuators). These parameters come from the pulse schedule via the user interface.

Outputs

- *actuator allocation* represents the optimal choice of assigned resources per task, as determined by the allocation algorithm. The signal specifies which actuators are assigned to each task, and which are not (*true* or *false* respectively). An actuator is only assigned to one task. Some active tasks can receive nothing if there are not enough actuators.
- *assigned resources per task* give the limits of resources allocated to each task (range of power, position and type per task). The assigned resource limits are computed from the actuator allocation solution and the actual limits of the actuators; for example the lower power limit is the minimum power of all assigned actuators to a task, and the upper power limit is the sum of their maximum powers; while the limits of position and type are the largest common range of all assigned actuator limits. The assigned resources must either fully (i.e. the assigned resources covers the maximum of the request) or partially (i.e. the assigned resources only covers the minimum but not the maximum of the request) satisfy the resource requests per task, otherwise, nothing is given.

Algorithm for actuator allocation The actuator allocation algorithm is the core of the Actuator Management resource allocation block. It determines the optimal actuator allocation for all the control tasks while considering the actuator availability and limits.

The underlying principles of the algorithm, which is inspired by earlier works [2, 3], are described in Fig. A2.

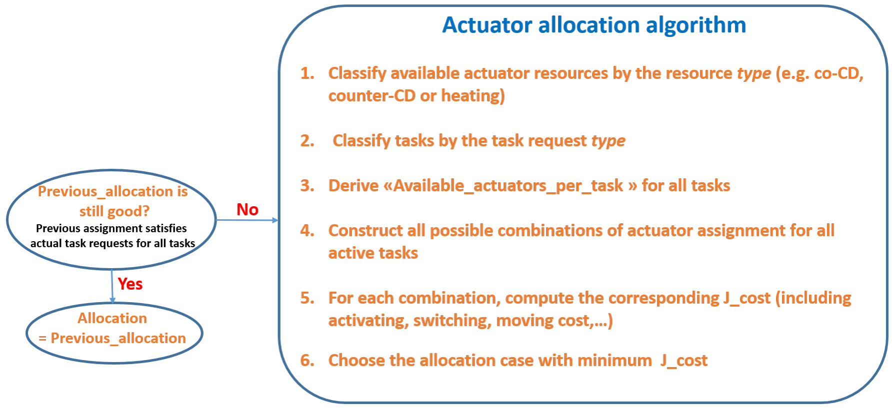

Figure A2: Algorithm for actuator allocation in the AM resource allocation

In the present implementation, a quasi-brute-force method is used, whereby the cost function is evaluated for all possible allocation options that partially or fully satisfy the resource requests. After all active tasks have been considered, the best allocation is chosen as the one with the lowest cost. The cost function includes the task priorities, a penalty cost for activating actuators, a penalty cost for switching from one task to another, and a penalty cost for moving actuators to new positions. Furthermore, a term is added to promote a solution satisfying as many tasks as possible as well minimizing the number of used actuators.

The algorithm allows additional requirements for actuators to be specified, such as as *preferred actuators* and *excluded actuators* for specific tasks (due to technical constraints or practical purposes). The algorithm removes excluded actuators from the available actuator lists of these relevant tasks and tries to allocate first the preferred actuators if they are still available. These requirements are pre-set by the user. The present algorithm was sufficient for the first tests described in this paper, but we highlight that it can be replaced by e.g. a Mixed Integer Programming algorithm as in [3], which may be more efficient for large numbers of tasks, actuators and constraints.

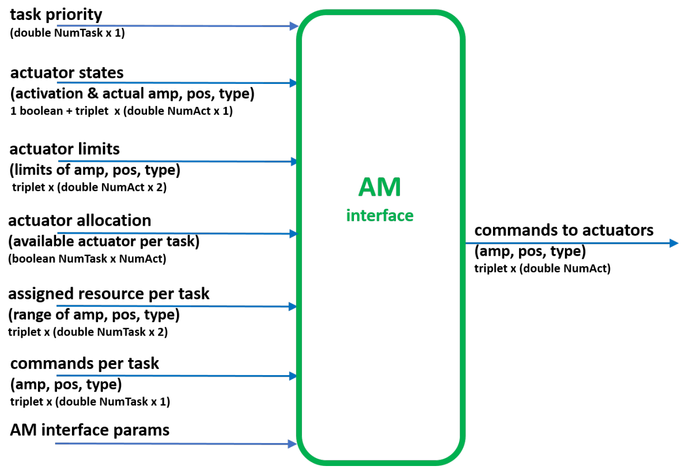

Figure A3: AM interface inputs and outputs including signal names and sizes

Appendix A.2. AM interface

The AM interface (Fig. A3) merges the commands per task into commands to the actuators. This component receives the task priority from the supervisory controller, assigned resources from the AM resource allocation and commands per task from the controllers. It uses this information to map the controller commands from each task (specified as amplitude, position and type) into the commands for each actuator, and send them to the actuator interface in the interface layer. In previous works [3], this component was referred to as the Low-level Actuator Manager but we have changed this name for increased clarity.

Inputs Some inputs are the same as the AM resource allocation inputs and outputs (see Appendix A.1), the others are specified hereafter

- *commands per task* contains commands from the controllers, specified terms of the triplet (amplitude, range, type).
- *AM interface params* include pre-set parameters for AM interface from the pulse schedule.

Output

- *commands to actuators* are the effective amplitude, position and type for each individual actuator. This signal is sent to the actuator interface in order to be converted into the inputs of the actuator systems.

Appendix A.3. Task-to-controller interface

The inputs and outputs of the task-to-controller mapping and controller-to-task mapping components are shown in Fig. 2. These components act as an interface layer between the controllers and the *task layer*. The task-to-controller mapping functions as a filter to distribute tasks per controller, following the mapping similarly defined in Table 3. On the contrary, an inverse mapping of Table 3 allows the controller-to-task mapping to group all controller outputs into task-dependent signals.

Appendix A.4. Controller interface

The generic interface of the controllers (Fig. A4) is elaborated in this subsection. This interface allows to integrate various controllers into the proposed scheme in a standardized way and facilitates the addition of new controllers or replacement of existing ones.

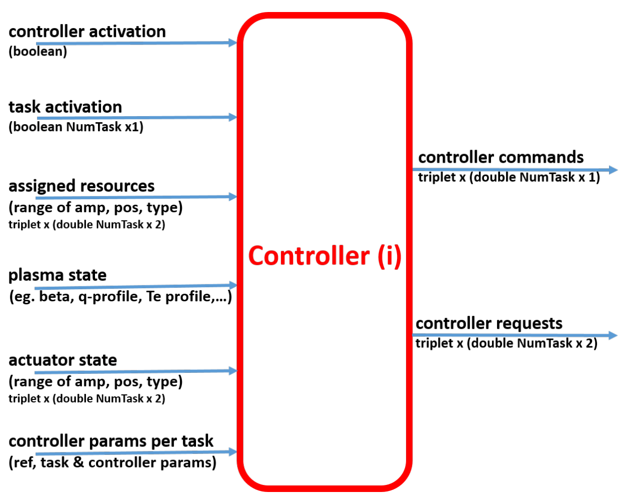

Figure A4: Generic controller inputs and outputs including signal names and sizes

Inputs

- *controller activation* is the activation signal for this controller.
- *task activation* is the activation signal for all tasks associated with this controller.
- *assigned resources* are the resources assigned to each task that is handled by this controller. This signal comes from the AM resource allocation via the task-to-controller mapping (see Appendix A.1).
- *plasma state* is the full representation of the plasma state (such as q-profile, β, T_e profile, shape, etc.). This input is necessary to execute the (feedback) control laws. To be generic, all controllers receive the same plasma state information and they can each extract the information they require.
- *actuator state* is the representation of the combination states (e.g. range of amplitude, position and type) of assigned actuators.
- *control params per task* are all parameters of the concerned tasks, including references, and task parameters (such as controller gains per task, etc.). They are defined in the pulse schedule and can be modified in real-time by the supervisory controller based on rules provided by the pulse schedule when necessary.

Outputs

- *controller commands* are the commands of each controller for each task in terms of the resources, within the limit of the assigned resources, represented (per task) by the triplet signals of amplitude, position and type. This is sent to the actuators via the controller-to-task mapping and AM interface.
- *controller requests* are the requests (e.g. range of amplitude, position and type) of each controller for the given tasks, sent to the the AM resource allocation (see Appendix A.1) via the controller-to-task mapping for the next time step.

This controller interface applies to the *parallel* controllers (see controller structure in Sec. 2.2, controller bullet). For the *sequential* controllers, the extra input *other controller commands* must be considered. The controller requests of the sequential controllers are not taken into account. In fact, the sequential tasks do not need a resource allocation specifically dedicated to them, because their commands, which aim to compensate the commands of the other controllers so-that the sequential tasks are achieved, should not be limited by the assigned resources and can also be negative. The commands of the sequential tasks will be distributed directly to the relevant actuators via the AM interface.

The sequential task with the highest priority (as determined by the supervisory controller) is the last to be executed, meaning it has the final decision on the total commands sent to the AM interface. Therefore, the *other controller commands* input of the sequential controller number m will include the commands of all parallel controllers, as well and those of the sequential controllers numbered 1 to $(m-1)$.

Appendix B. Details of the be in H-mode task

It is necessary to discuss further details about the be in H-mode task as an example of sequential tasks since its commands depend on the power commanded of the other controllers. Its mission is to ensure that the loss power P_{Loss} (or the total plasma heating power) is higher than a critical threshold P_{LH}. The threshold power of H-mode depends on the plasma state (e.g. plasma density, plasma size, magnetic field, etc.) and on whether the plasma is currently in H-mode or L-mode. Hence, this value is given to the LH-mode controller as a time varying task parameter by the supervisory controller. In [14] some threshold power estimations for ITER are proposed, however in the demonstrative example in Sec. 4, it is fixed at $P_{LH} = 52MW$. The loss power P_{Loss} has to include the total auxiliary power P_{aux} $(= P_{NBI} + P_{IC} + P_{EC}\ )$ and the radiation power P_α in case of fusion reaction. The latter is related to the total auxiliary power P_{aux} by a non-linear factor Q [27], such that:

$$P_{Loss} = P_{aux} + P_\alpha = \left(1 + \frac{Q}{5}\right) P_{aux} \tag{B.1}$$

For the sake of simplicity, we propose a linearized relation between Q and $P_{EC}\,[MW]$ based on different values of HH, the factor represents any confinement improvement or degradation, in Fig. 1 in [27] (three lines corresponding to $HH = 0.75$, 0.85 and 1), such that:

$$Q = \zeta\,(HH)\,P_{EC} + \xi\,(HH) \tag{B.2}$$

for three cases in the Table B1.

HH	ζ	ξ	
1	-0.14	10	without NTM
0.85	-0.10	7	with 3/2 NTM
0.75	-0.07	4.7	with 2/1 NTM

Table B1: Parameters for linearized Q factor

In case of two or more NTM appearances at the same time, the lowest Q value is taken into account.

Finally a low-pass filter is applied to P_{Loss} in order to simulate the effect of the energy confinement time on the rate of change of P_{Loss} whenever P_{aux} changes or an NTM appears. The time constant of the filter is considered equal to the expected thermal energy confinement time on ITER ($t_{confinement} = 3.5s$).

Acknowledgment

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 was supported in part by the Swiss National Science Foundation.

References

[1] D. Humphreys and et al., “Novel aspects of plasma control in ITER,” *Physics of Plasmas (1994-present)*, vol. 22, 2015.

[2] C. Rapson and et al., “Actuator management for ecrh at ASDEX Upgrade,” *Fusion Engineering and Design*, vol. 96, pp. 694–697, 2015.

[3] E. Maljaars and F. Felici, “Actuator allocation for integrated control in tokamaks: architectural design and a mixed-integer programming algorithm,” *Fusion Engineering and Design*, vol. 122, pp. 94–112, November 2017.

[4] W. Treutterer and et al., “ASDEX Upgrade discharge control systems real-time plasma control framework,” *Fusion Engineering and Design*, vol. 89 (3), pp. 146–154, 2014.

[5] N. Eidietis and et al., “Implementing a finite-state off-normal and fault response system for disruption avoidance in tokamaks,” *Nucl. Fusion 58 (2018) 056023*.

[6] N. Ravenel and et al., "Architecture of WEST plasma control system," *Fusion Engineering and Design*, vol. 89 (5), pp. 548–552, 2014.

[7] W. Treutterer and et al., "Towards a preliminary design of the ITER plasma control system architecture," *Fusion Engineering and Design*, vol. 115, pp. 33–38, 2017.

[8] C. Rapson and et al., "Experiments on actuator management and integrated control at ASDEX Upgrade," *Fusion Engineering and Design*, vol. 123, pp. 603–606, 2016.

[9] T. Blanken and et al., "Real-time plasma state monitoring and supervisory control on TCV," *Nucl. Fusion (2018), accepted*.

[10] F. Hofmann and et al., "Creation and control of variably shaped plasmas in TCV," *Plasma Phys. Control. Fusion*, vol. 36 B277, 1994.

[11] F. Felici, O. Sauter, S. Coda, B. Duval, T. Goodman, J. Moret, J. Paley, and the TCV Team, "Real-time physics-model-based simulation of the current density profile in tokamak plasmas," *Nucl. Fusion*, vol. 51, 2011.

[12] F. Felici and et al., "Real-time model-based plasma state estimation, monitoring and integrated control in TCV, ASDEX Upgrade and ITER," *26th IAEA Fusion Energy Conf. 2016, Kyoto, Japan*.

[13] E. Maljaars and et al., "Simultaneous control of plasma profiles and neoclassical tearing modes with actuator management in tokamaks," *42nd European Physical Society Conference on Plasma Physics, EPS 2015*.

[14] Y. Martin and et al., "Power requirement for accessing the H-mode in ITER," *J. Phys.: Conf. Ser. 123 012033*, 2008.

[15] M. Kong and et al., "Real-time control of neoclassical tearing modes and its integration with multiple controllers in the TCV tokamak," *44 th EPS Conference on Plasma Physics, Belfast, Northern Ireland*, 2017.

[16] K. Nagasaki and et al., "Stabilization of neoclassical tearing mode by eccd and its evolution simulation on JT-60U tokamak," *Nucl. Fusion*, vol. 45 1608, 2005.

[17] M. Kong and et al., "Control of NTMs and integrated multi-actuator control on TCV," *27th IAEA Fusion Energy Conference, Gandhinagar, India*, 2018.

[18] P. Martin and et al., "Physics, control and mitigation of disruptions and runaway electrons in the EUROfusion Medium Size Tokamaks science programme," *26th IAEA Fusion Energy Conf. 2016, Kyoto, Japan*.

[19] MATLAB and I. N. M. U. S. Statistics Toolbox Release 2012b, The MathWorks.

[20] C. Galperti and et al., "Integration of a real-time node for magnetic perturbations signal analysis in the Distributed Digital Control System of the TCV Tokamak," *20th Real Time Conference, Padova, Italy*, 2016.

[21] F. Felici and et al., "Distributed digital real-time control system for TCV tokamak," *Fusion Engineering and Design*, vol. 89, pp. 155–164, 2014.

[22] D. Kim and et al., "An active feedback plasma profile control approach applied to tcv plasmas & perspectives toward iter," *43nd European Physical Society Conference on Plasma Physics, EPS 2016*.

[23] M. Singh and et al., "Status of heating and current drive systems planned for ITER," *IEEE Transactions on Plasma Science*, vol. 44 (9), p. 15141524, 2016.

[24] O. Sauter and et al., "Control of neoclassical tearing modes by sawtooth control," *Physical review letters*, vol. 88 105001, no. 10, 2002.

[25] M. Henderson and et al., "The targeted heating and current drive applications for the ITER electron cyclotron system," *Physics of Plasmas*, vol. 22 (2), no. 021808, 2015.

[26] D. Humphreys and et al., "Active control for stabilization of neoclassical tearing modes," *Physics of Plasma*, vol. 13, no. 056113, 2006.

[27] O. Sauter, M. Henderson, G. Ramponi, and C. Zucca, "On the requirements to control neoclassical tearing modes in burning plasmas," *Plasma Phys. Control. Fusion*, vol. 52 02502, 2010.