

SYSTEM ARCHITECTURE FOR ACTUATOR MANAGEMENT IN ITER PCS

O. Kudlacek, W. Treutterer, I. Gomez-Ortiz, P. Lang, M. Reich
Max Planck Institute for Plasma Physics
Garching, Germany
Email: ondrej.kudlacek@ipp.mpg.de

S. Stanek
Czech Technical University in Prague, Faculty of Nuclear Sciences and Physical Engineering
Prague, Czechia

J. Schmalz
Saarland University, Saarland Informatics Campus
Saarbrücken, Germany

L. Pangione, T. Ravensbergen, P. de Vries, A. Vu, L. Zabeo
ITER Organization
St Paul Lez Durance Cedex, France

R. Nouailletas, D. Weldon
IRFM, CEA
Saint Paul lez Durance, France

Abstract

ITER, the largest tokamak currently under construction, will rely on a sophisticated control system to coordinate multiple plasma control tasks using tens of actuators. During the SRO phase, where ITER reaches full plasma current of 15 MA and full field and H-mode at half field and half of nominal current, there will be 40 MW of ECRH power at 9 different mirrors, 10 MW of ICRH power on 1 antenna, 4 pellet injectors, and 60 gas valves. A key element of this system is actuator management (AM), which orchestrates actuator usage to avoid conflicts, enhance performance, and increase robustness. Central to AM are virtual actuators, which are presented to the controllers as single entities combining actuators with comparable effects on the plasma. Up to now, the mapping between actuators and virtual actuators has been static, but future operation requires extending this concept to dynamic mappings. In addition, AM must be capable of adjusting pellet injection properties, both by varying pellet size and by modifying injector-flight tube connections. To meet these requirements, the AM architecture has been extended accordingly. This paper discusses these extensions, presents illustrative examples of their application, and provides an outlook on future developments.

1. INTRODUCTION

ITER is the world's largest tokamak, currently under construction in the south of France. Its objective is to generate 500 MW of fusion power for up to 500 seconds while achieving a fusion gain of $Q = 10$. Plasma pulses will be controlled by the Plasma Control System (PCS), utilizing several heating, fueling, and magnetic actuators to regulate many plasma parameters [1]. A critical component of the PCS is the Actuator Manager (AM), which coordinates actuator operations to enhance the system's flexibility, performance and robustness. This contribution will present the AM architecture considered by the ITER Organization for implementation in the PCS.

The PCS development follows a flexible strategy that enables continuous optimization throughout the project's evolution. In alignment with this approach, each PCS component is designed to fulfill requirements in a specific stage of ITER operation and undergoes integrated testing for that phase. Lessons learned from these tests are used to refine each component for subsequent operational stages [2]. This contribution details the AM architecture which covers the needs of the Start of Research Operation (SRO) phase [3], and is extendable beyond this phase.

During the SRO phase, ITER is expected to achieve a plasma current of 15 MA with a toroidal field of 5.3 T in L-mode and demonstrate H-mode operation with a plasma current of 7.5 MA and a toroidal field of 2.65 T. The system will utilize 40 MW of ECRH heating delivered through nine movable mirrors covering various locations of the plasma and one ICRH antenna providing 10 MW of power [4].

The pellet injection system consists of four pellet injectors with adjustable pellet sizes, which can be directed through various flight tubes. These injectors are connected to two casks, each supplying fuel to a pair of injectors. Within each cask, the two injectors share access to three flight tubes: one leading to the low-field side and two to the high-field side of the tokamak. At any given time, each flight tube can be connected to only one injector.

The Actuator Management (AM) shall provide the following functionalities for SRO:

1. Establish an interface between controllers and actuators such that the controllers remain actuator-agnostic. Controllers output their desired cumulative actuation amplitude (e.g., total heating power or gas flow), while the AM distributes this demand among the available physical actuators. At the same time, the AM provides each controller with the cumulative capacity of the actuators assigned to it.
2. Adjust, wherever possible and relevant, the power deposition location of actuators.
3. Orchestrate the pellet firing sequence for both fueling and ELM pacing.
4. Enable dynamic reassignment of actuators to different virtual actuators, allowing an actuator to change the task it serves as needed.
5. Optimize pellet injection performance by adjusting the pellet size and modifying injector–flight tube connections. This is accompanied by a downtime of the actuator, or by its operation at lower performance, which has to be accounted for when making decision.

The first three points are addressed by the architecture described in [5] and briefly summarized in Section 1.1. The final two items constitute the main focus of this paper.

1.1. Summary of Underlying Work

This section summarizes the architecture of the Actuator Management system, which addresses use cases that do not require changing the purpose of actuators and does not include performance optimization associated with actuator downtime. The information flow between individual components of the PCS is illustrated in Figure 1. Three key components of the AM—the virtual actuators, actuator proxies, and launcher proxies—were introduced in [5]. The level of detail provided here is intended to be sufficient for understanding the remainder of the text.

1.1.1. Actuator Proxy

The real actuators within the AM system are represented by **Actuator Proxies**. These proxies translate information from the plant systems and other PCS components into a standardized format compatible with all actuator types and understood by the other parts of the Actuator Management. Each actuator is most importantly characterized by its availability and actuation capacity (for example maximum power).

Specific features or requirements of different actuator types are handled by dedicated plugins. For example, if an actuator provides active control of its power deposition location (such as an ECH mirror), the corresponding actuator proxy is extended by a plugin that monitors and commands this deposition location, and provides the power deposition range.

1.1.2. Launcher Proxy

Launcher Proxies are used to represent launchers when the connection between an actuator and its launcher is not fixed, requiring individual accessibility of both components. This situation occurs only for pellet injectors and their associated flight tubes. While a pellet injector can deliver pellets to the plasma only when properly connected to an available flight tube, the possibility of reconfiguration necessitates separate access to both components. This possibility is further elaborated in part 2.2.

1.1.3. Virtual Actuator

Virtual Actuators represent groups of actuators that are interchangeable with respect to their intended effect on the plasma. From the controller's perspective, a virtual actuator behaves as a single actuator while offering extended capabilities. It computes and aggregates the available actuation capacity of its member actuators and distributes incoming commands among their corresponding actuator proxies. Each virtual actuator is assigned to a specific output channel of a controller, dedicated to a particular purpose.

Special requirements, such as ensuring that all actuators within a virtual actuator track the same rational flux surface for NTM stabilization, are managed through dedicated plugins.

Another special case handled by a plugin is the synchronization of the pellet injection sequence to satisfy two objectives: ensuring that the fueling controller's particle flow request and the ELM controller's minimum pellet frequency are both met. This functionality is implemented in the *Pellet Virtual Actuator*, which aggregates all pellet injectors. The design choice and its underlying motivation are discussed in details in [5].

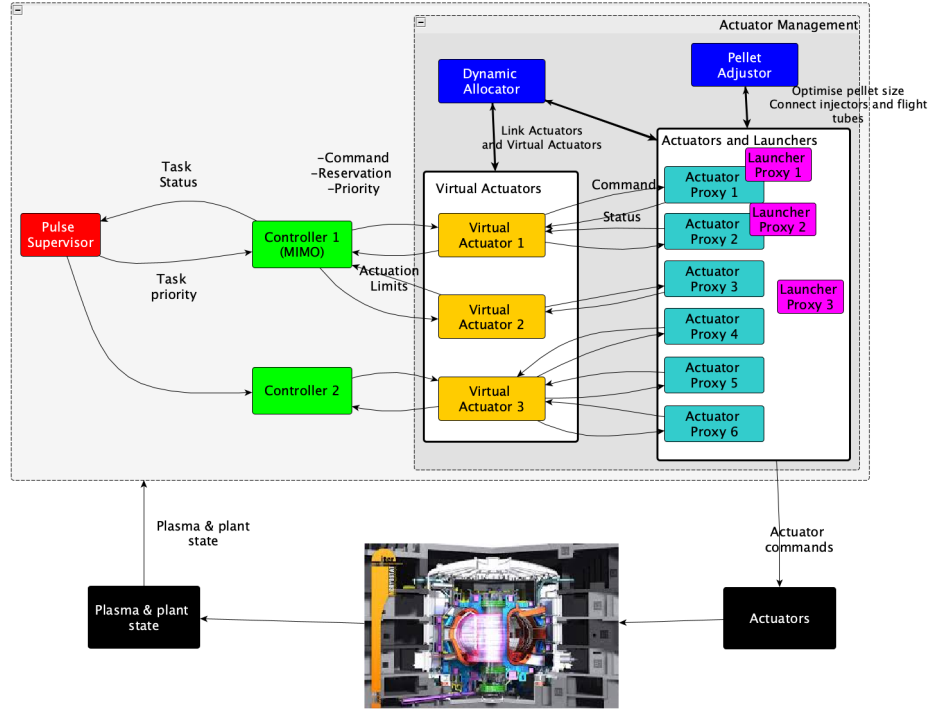


FIG. 1. The information flow between individual parts of the control system with the focus on the Actuator Management

2. EXTENSIONS TO THE ACTUATOR MANAGEMENT

This section focuses on the extensions to the Actuator Management that address the last two requirements listed in Section 1: (i) the dynamic allocation of actuator proxies to different virtual actuators, and (ii) the optimization of actuator properties associated with downtime when the actuator cannot be used, based on the plasma and plant state.

Dynamic allocation is particularly relevant for EC mirrors, whose deposition location may need to shift—e.g., from NTM 2/1 control to NTM 3/2 control, or from NTM control to central heating and back—depending on plasma conditions (such as NTM growth) or actuator failures. While EC mirrors are a prime example, the dynamic allocation strategy is applicable to any actuator type.

Optimization of actuator properties associated with downtime is relevant only for pellet injectors, which, however, **do not change their virtual actuator when reconfigured**. Consequently, the dynamic allocation and the downtime optimization represent two distinct, non-overlapping use case sets. For this reason, these functionalities are implemented in two separate components of the AM: the **Dynamic Allocator** and the **Pellet Adjustor**.

Both of these components need additional information about the virtual actuators and actuator proxies. First of all, they must be aware of the **Priority** of the task associated with each virtual actuator. This value shall be defined by the pulse supervisor and propagated via the controller that utilizes the virtual actuator. The pellet virtual actuator receives both the priority of density control and ELM pacing.

Second, each virtual actuator must be provided with a lower limit on its cumulative actuation capacity: the **Reservation**. This value is provided by the controller, and does not need to match the controller's actual request. In general, it should be significantly higher. Separating the reservation from the controller request offers several advantages:

- The controller request may fluctuate rapidly due to oscillatory behavior or input noise. These fluctuations should not directly affect dynamic allocation.
- The reservation can be used to proactively allocate resources where they are likely to be needed. For example, in scenarios prone to NTM instabilities, a reservation can preemptively secure additional resources to enable rapid response if an instability arises.

The pellet virtual actuator receives the reservation both for the particle flow from the fueling controller, and frequency from the ELM controller.

Last, each virtual actuator shall be aware of its **purpose**: This defines the intended use of the virtual actuator and is used to identify suitable matching actuators for it. For heating systems, two primary purposes are distinguished:

- **Bulk heating:** Refers to heating applied anywhere within the plasma volume, without specific localization requirements.
- **Local heating:** Refers to heating targeted at a specific region within the plasma. In this case, the virtual actuator is provided with input parameters defining the desired deposition location and acceptable deposition error.

For the pellet virtual actuator, the purpose is always the fueling and ELM pacing.

In addition to the inputs described above, each virtual actuator also reports its **allocation status**. Three distinct statuses can be identified:

- **Fully satisfied:** The virtual actuator has been assigned sufficient resources to meet its reservation.
- **Blocked access:** The required resources exist but are not available to the virtual actuator—typically because they are already allocated to another virtual actuator with higher priority.
- **Resource shortage:** The system lacks the necessary resources to fulfill the virtual actuator's reservation.

For the pellet virtual actuator, the status is computed separately both for the fueling and frequency reservation.

There is an important distinction between the “Blocked Access” and “Resource Shortage” statuses from the perspective of the pulse supervisor. In the case of Blocked Access, the pulse supervisor has the ability to take corrective action—for example, by increasing the priority of the virtual actuator that requires resources and has “Blocked Access” status. By contrast, in the case of a Resource Shortage, where no resources are available to fulfill a virtual actuator's reservation request, the pulse supervisor cannot directly resolve the issue. Instead, it must decide either to proceed with reduced performance or to terminate the pulse. It is important to note that the pulse supervisor makes these decisions based on input from both the controller (regarding reference tracking) and the dynamic allocation status corresponding virtual actuator(s).

The **effectiveness evaluation** of each actuator proxy has been added to assess its capability in fulfilling the objectives of each virtual actuator.

In the area of heating, all available actuators are considered effective for bulk heating, as this purpose does not require localization. For the local heating, a heating actuator is considered effective only if it can reach the desired power deposition zone specified in the virtual actuator and is capable of depositing power locally within that region. For example, an EC mirror with a deposition range from $\rho = 0.4$ to $\rho = 0.9$ will always be considered effective for bulk heating or for local heating at $\rho = 0.8$. However, it will not be effective for local heating at $\rho = 0.2$, as it cannot heat in this location.

For pellet injectors, each injector is considered effective for both fueling and ELM pacing. However, the effectiveness depends on pellet size and injection geometry represented by connected flight tube: large pellets injected from the high-field side are more effective for fueling, whereas small pellets and low-field side injection are more suitable for ELM pacing. In the other words, the effectiveness is defined by combination of the pellet injector and the flight tube it is connected to.

For future developments, the architecture is designed to allow actuator proxies to easily incorporate additional information, such as power deposition width, current drive efficiency, actuator reliability, or the “travel time” required to reach a desired location. This information can be leveraged when determining actuator allocation strategies or optimizing pellet adjustments.

2.1. Dynamic Allocation

As previously mentioned, the reassignment of an actuator proxy from one virtual actuator to another is handled by the dynamic allocator. This component has access to all virtual actuators and actuator proxies within the AM, collects the necessary information, and makes decisions regarding reallocation.

The actuator allocation policy is a configurable parameter of the AM. In the long term, it should be possible to select different policies for different stages of the discharge. If no policy is specified, the AM assumes that the initial mapping between virtual actuators and actuator proxies is fixed and cannot be modified.

All strategies that dynamically link actuators to virtual actuators take into account the time required for an actuator to become effective after reassignment. For example, when an ECH mirror is reassigned from the central heating virtual actuator to the NTM control virtual actuator, it requires a significant amount of time to rotate and align with the NTM location. The virtual actuator responsible for NTM control must recognize that the resource has already been allocated to it and should not issue additional requests during this transition period.

The first implemented policy, presented in this work, aims to fully satisfy as many reservation requests as possible from the highest-priority virtual actuators. This approach is referred to as the **SatisfyTop** policy.

2.1.1. SatisfyTop Allocator

This strategy fully satisfies the reservation of the virtual actuators with the highest priority, and sacrifices the reservation of the virtual actuators with the lowest priority. It works in three steps:

1. Each of the virtual actuators frees actuators that are not needed to accomplish its reservation, and the actuators that are not effective any longer.

- In case the reservation request drops, the virtual actuator frees the actuators that are not needed to satisfy the reservation. Those actuators become idle (the actuators that are not in any virtual actuator)
 - In case that the actuator is not effective any longer, for example NTM position cannot be reached by certain EC mirror any more, the actuator becomes idle.
2. The allocator tries to assign idle actuators to the virtual actuators that need additional resources. Naturally, the assigned actuators must be effective for the virtual actuator's purpose. Virtual Actuators with the highest priority get the resources first.
 3. In case that there are virtual actuators that still require additional resources, the allocator tries to move actuators from the virtual actuators with lower priority to the served virtual actuator.

A demonstration of the SatisfyTop dynamic allocator is shown part 3.1.

2.2. Pellet Adjustor

As described above, the configuration of the pellet injectors can be adjusted in three possible ways in real time:

- **Injection geometry adjustment:** The injector can be connected to a different flight tube to change the point from which the pellet enters the plasma. This action takes approximately 1 s, during which the injector cannot fire a pellet.
- **Pellet diameter change:** The pellet diameter can be switched between 3 mm and 5 mm. This change requires about 30 s, during which the injector is unavailable for firing.
- **Pellet length adjustment:** The pellet length at a given diameter can be modified. This process takes up to 3 s depending on the magnitude of the change and does not involve injector downtime. However, the reconfiguration is expected to reduce pellet quality during the process, increasing the fraction of missed pellets.

Eventually, all three actions should be orchestrated by a single algorithm. The orchestration algorithm will access all the pellet injectors and flight tube proxies as well as the pellet virtual actuators. It uses a configurable policy which shall be changeable throughout the discharge. Here, only the initial version addressing injection geometry adjustments have been developed so far [6] and is presented.

By default, the pellet adjustor handles failed injectors or flight tubes as follows:

- If a flight tube becomes unavailable, the connected injector is disconnected and reconnected to another available flight tube, if possible.
- If an injector becomes unavailable, its flight tube is disconnected and kept as a backup.

This default strategy introduces only potential benefits of increasing the number of injectors that can fire pellets to the plasma, without any risk of degrading the current situation.

Optionally, the pellet adjustor can also modify the connections between flight tubes and pellet injectors to maximize the system's effectiveness, either for fueling or for ELM pacing. This optimization is performed under the constraint that reservations for both pellet frequency and particle flow remain satisfied during the reconfiguration process. If the reservation is not satisfied even before reconfiguration, this constraint does not apply.

The advantage of this strategy is the potential to optimize the overall pellet system performance based on the plasma state. The main drawback, however, is that the reconfiguration process requires a significant amount of time (approximately 1 s), during which the actuation capacity of the pellet injection system is temporarily reduced.

3. RESULTS

3.1. Dynamic Allocation

This section demonstrates the use of the SatisfyTop allocator described in part 2.1.1. It involves 5 EC mirrors and 1 IC antenna. It is assumed that all the EC mirrors are moveable, and all can reach ρ between 0.1 and 0.9, and that all actuators involved in this example have the same power. The IC power deposition location cannot move.

In our scenario, the PCS is dealing with three control tasks: NTM 3/2 and NTM 2/1 preemption/stabilization, and the bulk heating of the plasma. Table 1 shows the priority, initial reservation and actuator allocation, and the requested deposition location for each virtual actuator. The EC mirrors are effective for all three virtual actuators, while the IC antenna can be used only for the bulk heating.

The role of the dynamic allocator is demonstrated on the scenario described below with the actuator reallocation steps shown in Figure 2 and time evolution of requests and deposition in Figure 3:

Virtual Actuator	Priority	Reservation	Initial Allocation	Requested Deposition Location
Bulk Heating	0.2	3 actuators	IC, EC1, EC2	preferred at $\rho = 0.1$
NTM 3/2	0.5	2 actuators	EC3, EC4	$\rho = 0.6 \pm 0.05$
NTM 2/1	1.0	1 actuator	EC5	$\rho = 0.8 \pm 0.05$

TABLE 1. Priorities, initial actuator allocation and reservations, and requested deposition locations for virtual actuators.

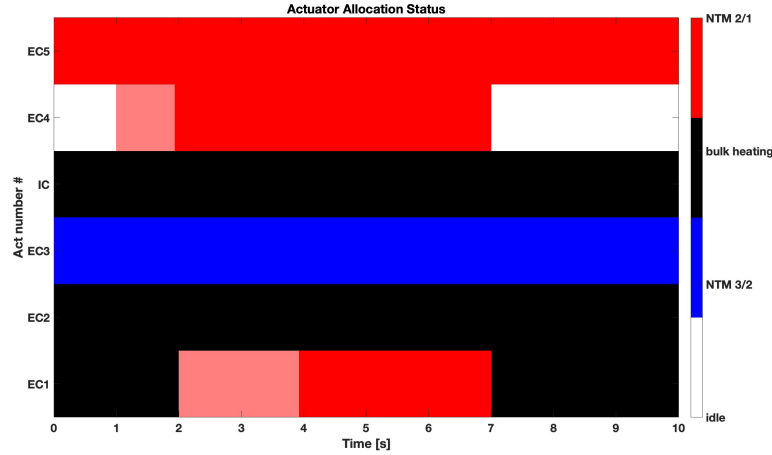


FIG. 2. The allocation of actuators to virtual actuators. The areas with less bright color mean that the actuator is transiting and cannot be used by the virtual actuator yet.

- $t = 0$ s – The NTM 3/2 mode has just been stabilized, reducing its reservation request from 2 to 1 actuator. Consequently, one actuator is released. The dynamic allocator chooses to keep EC3 and release EC4, which remains idle and unassigned.
- $t = 1$ s – An NTM 2/1 mode emerges, increasing the reservation request for its virtual actuator to 2 actuators. Since EC4 is idle and can reach the required location, it is allocated to NTM 2/1.
- $t = 2$ s – The NTM 2/1 mode remains unstable, prompting an increase in reservation to 3 actuators. As no idle actuators remain, the SatisfyTop allocator must reallocate from the lowest-priority virtual actuator, which is the bulk heating actuator. It chooses to take EC1.
- $t = 7$ s – Once the NTM 2/1 is stabilized, EC1 is returned to the bulk heating virtual actuator, ideally targeting central heating. EC4 is released and remains idle.

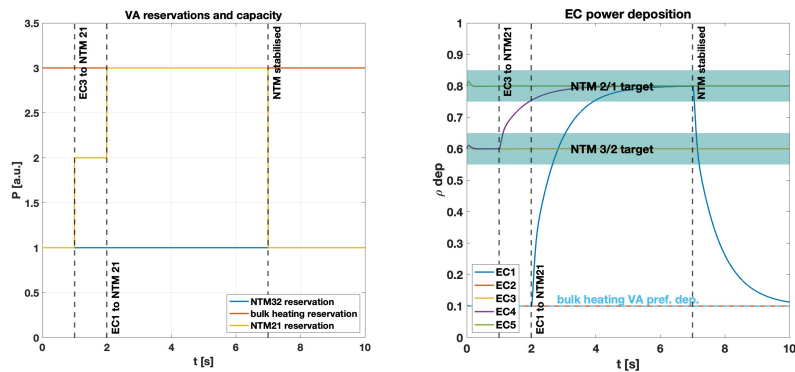


FIG. 3. Time evolution of the reservation request for each virtual actuator and power deposition location of each EC mirror.

3.2. Pellet Adjustor

This section describes the optimization of the flight tube–pellet injector connections for a scenario in which pellet injection is used to simultaneously control plasma density and pace ELMs. The functionality of the controller and the pellet adjustor is demonstrated using a very simplified model of plasma density evolution and ELM behavior. It should be emphasized that these models are intended solely to illustrate the operation of the pellet adjustor; **no physics conclusions can be drawn**.

The plasma density evolution is modeled by the following equation:

$$\frac{dn}{dt} = -\frac{n}{\tau} + c \cdot \Gamma_{pel}(t), \quad (1)$$

where n is the plasma density, τ the particle confinement time, c a constant, and Γ_{pel} the particle flow delivered by the pellets, defined as

$$\Gamma_{pel} = \sum_{i=1}^N [\Theta(t - \tau_i) - \Theta(t - \tau_i - W)] \frac{N_i}{W} \cdot f_i^{fuel}, \quad (2)$$

where N_i is the number of particles in pellet i , τ_i the injection time, W the pellet ablation duration, Θ the Heaviside function, and f_i^{fuel} the fueling efficiency of pellet i . The equation for Γ_{pel} stems from an assumption constant pellet ablation rate over time W .

The basic mechanism of ELM triggering relies on the fact that pellet injection temporarily increases the pedestal pressure gradient, thereby raising the likelihood of an ELM crash. The ELM size in this model is assumed to be proportional to the time elapsed since the previous ELM event. In the model, the probability of an ELM crash (p_{ELM}) increases linearly with time after the last crash. When pellet arrives to the plasma, a random variable $a \in [0, 1]$ is drawn. If a pellet is fired from the low-field side, an ELM is triggered immediately. If a pellet is fired from the high-field side, an ELM occurs only if the random number $a < \frac{1}{2}p_{ELM} + \frac{1}{2}$.

The pellet injector and flight tube model considers the delay between the "fire" command reception and the pellet arrival (approximately 100 ms), the reload time after firing the pellet, and the unavailability of the injector while reconnecting to a different flight tube.

The functionality of the pellet adjustor is demonstrated in a scenario where both plasma density control and ELM pacing are performed by a pellet injection system consisting of four injectors housed in two casks. All pellets are identical, each containing $3 \cdot 10^{20}$ particles. Injectors 1, 2, and 3 are initially connected to the high-field side flight tube, while Injector 4 is connected to the low-field side flight tube. During the first three seconds of the demonstration, fueling is prioritized over ELM pacing, whereas in the last three seconds, ELM pacing is given higher priority.

Figure 4 shows the time evolution of the controlled quantities (plasma density and ELM size), the pellet injection times for each injector, the injector–flight tube connections, and the evolution of request, reservation, and capacity for both particle flow and pellet injection frequency.

Initially, the plasma density is increased by injecting pellets from the injectors connected to the high-field side. At $t = 3$ s, the priority changes and ELM pacing becomes more important. At this point, the pellet virtual actuator prefers to fire pellets from the low-field side, which are more effective at triggering ELMs but have lower fueling efficiency. To improve the overall performance of the system for ELM pacing, the pellet adjustor reconnects one of the pellet injectors (PIS1) to the low-field side flight tube. Note that during the reconnection process, the reservation remains lower than the actuation capacity for both pellet frequency and particle flow. The performance of the system on ELM pacing consequently improves and ELMs become consistently more regular and smaller.

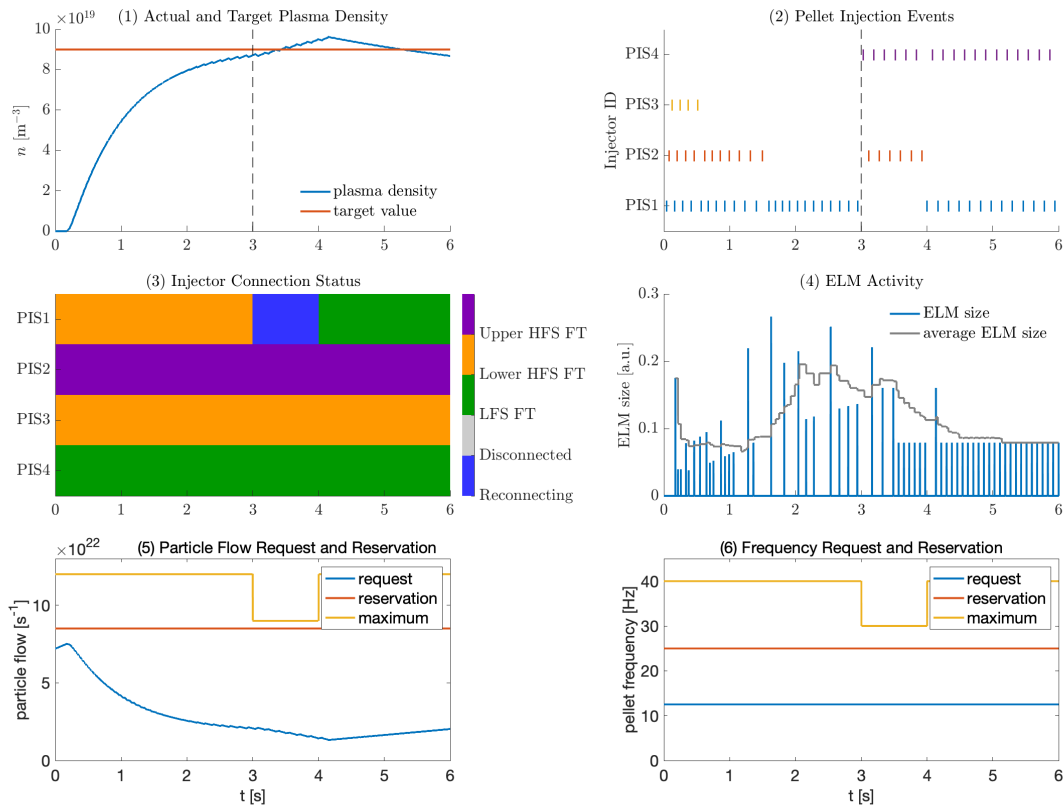


FIG. 4. Time evolution of: (1) measured and reference plasma density, (2) pellet injection times (each line indicating a fire command), (3) injector–flight tube connection status, (4) ELM events and their magnitudes, (5) particle flow request, reservation, and capacity, and (6) pellet injection frequency request, reservation, and capacity. The vertical line at 3 s indicated the moment when ELM pacing becomes more important than fueling.

4. CONCLUSIONS AND FUTURE WORKS

This contribution presents the extensions of the AM architecture presented in [5] by two additional capabilities:

- The resources can be dynamically assigned to virtual actuators serving different control tasks based on the priority and needs of the tasks
- The pellet injectors and flight tubes can be dynamically connected to each other such that the performance for their purpose is optimized

The decision algorithms that decide about these actions are currently simple and rule based. This approach is well-suited for the initial prototype of dynamic allocation due to its speed, simplicity, and transparency. However, it has several limitations that reduce its effectiveness in more complex or demanding scenarios:

- Simple algorithms do not always converge to an optimal solution. For example, it may be preferable to partially satisfy the requirements of multiple virtual actuators rather than fully satisfying one and sacrificing another—something that is difficult to express in rule-based logic.
- As the number of parameters increases, it becomes progressively harder to incorporate them into the rule set in a coherent and maintainable way.

For these reasons, future development should focus on approaches that allocate actuators and optimize pellet injection system configuration through cost function minimization, enabling more flexible and optimal decision-making.

These approaches must ensure a **deterministic maximum number of iterations** and **guarantee timely convergence** for realistic problem sizes expected at ITER.

4.1. Future Developments on Dynamic Allocation

For dynamic allocation problem, a method suitable for real-time cost function minimisation has to be found. For the maximum problem size, assume 15 actuators (11 EC moveable mirrors, 2 IC, and 2 NBI), and no more than 6 virtual actuators (1 for bulk heating, 2 for central heating/profile control, 2 for NTM control, and 1 serving as a backup). There are several approaches to address this:

- **Brute-force algorithms** evaluate all possible actuator-to-virtual-actuator combinations and select the configuration that minimizes the cost function implemented for example in [7, 8]. However, for the problem size described above, the number of possible combinations is approximately $6^{15} \approx 5 \cdot 10^{11}$. Solving a problem of this magnitude in real time is computationally infeasible and can therefore be ruled out as a practical solution.
- **Quadratic Integer Programming** (proposed for example in [9]), which is a type of mathematical optimization problem where the cost function and/or constraints are quadratic (containing terms with variables squared or multiplied together), and the variables are restricted to integer values. There are existing C++ libraries that solve this problem, for example CPLEX. It is solved in an iterative way, which means that **there is no guarantee of maximum number of operations that will be taken**. The computational time grows exponentially with the number of integer variables, which is unfavorable for real-time usage.
- Formulate the problem as a case of **Linear Integer Programming**. In this framework, **the maximum number of operations required for the algorithm to converge can be explicitly determined, ensuring real-time compatibility**. Nevertheless, further analysis is needed to evaluate whether all desired behaviors can be captured using only a linear cost function and linear constraints.

4.2. Future Developments on Pellet Adjustments

The next topic to be addressed in detail is the complete design of the Pellet Adjustor, including the capability to change pellet diameter and size. In contrast to the dynamic allocation of actuators to virtual actuators, no existing tokamak is currently able to perform actuator reconfiguration that involves downtime. Consequently, there is no prior community experience with this type of task. A key challenge will be to identify suitable methods to properly account for actuator downtime during the reconfiguration process.

ACKNOWLEDGEMENTS

The views and opinions expressed herein do not necessarily reflect those of the ITER Organization. This publication is provided for scientific purposes only. Its contents should not be considered as commitments from the ITER Organization as a nuclear operator in the frame of the licensing process.

REFERENCES

- [1] Humphreys D. et al. 2015 Phys. Plasmas 22 021806
- [2] P.C. de Vries et al, Fusion Engineering and Design, Volume 204, 2024, 114464
- [3] ITER research plan: ITER-19-003 on, <https://www.iter.org/technical-reports>
- [4] A. Vu et al, Progress in the ITER Plasma Control System design, under review in FED
- [5] O. Kudlacek et al, Fusion Engineering and Design, Volume 216, July 2025, 115071
- [6] S. Stanek, Actuator management for tokamaks, bachelor thesis, Czech Technical University in Prague
- [7] C. J. Rapson, et al., Actuator management for ECRH at ASDEX Upgrade, Fusion Eng. Des. 96 - 97 (2015) 694 – 697
- [8] N.M. Trang Vu, et al, Fusion Engineering and Design, Volume 147, 2019, 111260
- [9] E. Maljaars et al, Fusion Engineering and Design, Volume 122, 2017, Pages 94-112