Fusion Engineering and Design 198 (2024) 114085

Contents lists available at ScienceDirect

Fusion Engineering and Design

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

Nonlinear, real-time optimization for actuator management in tokamaks

Andres Pajares [a,b,*], Kathreen E. Thome [b], Eugenio Schuster [a], Michael L. Walker [b], David A. Humphreys [b]

[a] Lehigh University, Bethlehem, PA 18015, USA
[b] General Atomics, San Diego, CA 92121, USA

ARTICLE INFO

Keywords:
Actuator management
Tokamak control
Nonlinear optimization

ABSTRACT

Experiments in DIII-D have been carried out to test a novel actuator management approach in tokamaks. The actuator-management scheme is posed as a nonlinear-optimization problem in which the actuator commands are calculated in real time according to the changing control priorities, plasma state, and actuator availability. Such optimization problem is solved using the augmented Lagrangian method, combined with a gradient projection method and a conjugate-gradient iteration algorithm. The algorithmic approach followed in this work does not depend on the particular control objectives or actuators considered, which facilitates its integration with other independently-designed control components within a plasma-control system. In addition, the actuator-management algorithm is able to handle the optimization problem in a computationally efficient manner, making it suitable for real-time implementations. Initial DIII-D results in the steady-state high-q_{min} scenario have demonstrated the capabilities of the actuator manager to perform both simultaneous multiple mission and repurposing sharing, which will be required in ITER.

1. Introduction

In ITER, a large number of control objectives will need to be attained by means of a finite set of shared actuators [1]. Some examples are the use of Electron-Cyclotron Heating and Current Drive (ECH&CD) for pre-ionization, Tearing Mode (TM) suppression, and current-profile control, or the utilization of pellet injection for burn control, edge-localized mode pacing, and disruption mitigation. Such high degree of actuator sharing poses a fundamental limitation in the design of the ITER Plasma Control System (PCS). A single actuator will, in general, simultaneously receive multiple commands from individual, independently-designed controllers. For example, ECH&CD may receive one aiming command from a controller for TM suppression, but a different aiming command from a current-profile controller. Therefore, additional information is required by the actuator regarding when and which commands must be executed. Moreover, the mission of an actuator during a discharge may need to change due to multiple reasons. The first reason is a change in the discharge phase, e.g. ECH&CD may be used for pre-ionization but for current-profile control later in the ramp-up and/or flat-top phases. The second reason is a potential variation in the plasma state, e.g. ECH&CD may be used for current-profile control, but a TM may develop that needs ECH&CD suppression. Finally, changes in the actuator availability (e.g. if a reduction in the available ECH power occurs due to launcher or transmission line failures) may require a change in the actuator mission. Generally speaking, two main types of actuator sharing are expected in ITER: Simultaneous Multiple Mission (SMM) sharing and Re-Purposing (RP) sharing [1]. With SMM sharing, a continuous, synchronous use of an actuator for several control objectives is considered. On the other hand, RP sharing is understood as a fast switch in the use of an actuator from one objective to another. The high number of individual control tasks and actuator commands that will be found in ITER has no precedent in current devices, making the actuator-sharing problem even more complicated. In order to deal with these advanced-control aspects, which are applicable not only to ITER but also to future fusion-power plants, there is an increasing need for the development of actuator-management schemes and algorithms that can decide, in real time, how to command the available actuators while fulfilling the required control objectives.

Recent work on actuator management includes [2–5]. In [2], some actuator-management functionalities are developed and tested in simulations for TM suppression and safety-factor profile (q) control. The work in [3] proposes and analyzes, from a general, functional perspective, various architectural designs for actuator management, and proposes a mixed-integer programming algorithm. In [4], an actuator

* Corresponding author at: General Atomics, San Diego, CA 92121, USA.
E-mail address: pajaresa@fusion.gat.com (A. Pajares).

https://doi.org/10.1016/j.fusengdes.2023.114085
Received 28 August 2023; Received in revised form 26 October 2023; Accepted 27 November 2023
Available online 9 December 2023

manager for ECH&CD sharing is presented to carry out simultaneous β and TM control in ASDEX-U. In [5], a generic integrated-control architecture with a task-based actuator-management scheme is presented. The approach is experimentally tested using up to three ECH&CD sources and one Neutral Beam Injector (NBI) in TCV, as well as in ITER simulations using more actuators.

In the present work, a novel actuator-management approach is proposed based on solving a nonlinear-optimization problem in real time. Previous efforts along this line can be found in [6]. The optimization problem is solved by means of the augmented Lagrangian method [7], together with a nonlinear gradient-projection scheme composed of a standard gradient-projection method and a conjugate-gradient method. Although the algorithm presented in this paper does not allow for treating some variables as integers, it provides the flexibility to include any nonlinear constraint (whereas, for example, the work in [3] allows for using integer variables but restricts the constraints to be linear). In addition, the design of this actuator-management algorithm is independent of the number and type of actuators considered, as well as of the control-synthesis procedure of the feedback controllers employed. The requests sent by the feedback controllers are embedded into the actuator-management scheme as constraints, which may be relaxed when the associated controller request cannot be completely fulfilled. Such relaxation is done according to the relative control priorities (e.g. TM suppression is more urgent than profile control, plasma-current control has the same priority as β control, etc.) that the actuator manager receives from a supervisory and exception-handling system [8]. The control priorities are included by means of a cost function that needs to be minimized. Other objectives, such as minimizing the control effort and/or prioritizing the use of certain actuators, can be included within the cost function. In addition, physical saturation limits are imposed on the actuator commands. The objective of the actuator-management optimization is to find the actuator commands that minimize the aforementioned cost function while satisfying as many controller requests as possible within the physical saturation limits of the actuators. In the specific experimental application shown in this work, the overall control scheme is composed of the actuator-management algorithm plus several feedback controllers [9,10] for the plasma magnetics, kinetics, and Magneto-HydroDynamic (MHD) instabilities. DIII-D experiments have been carried out to test the capabilities of the actuator manager, like the SMM and RP sharing functions described above, while maintaining a desired plasma evolution despite plasma-state changes and unexpected actuator trips.

This paper is organized as follows. The actuator-management optimization problem is described in Section 2. The algorithm used to solve the optimization problem is described in Section 3, where Section 3.1 presents the augmented Lagrangian method, Section 3.2 describes the nonlinear gradient-projection method, and Section 3.3 presents the standard gradient projection + conjugate-gradient algorithm. Section 4 presents some initial DIII-D experimental results. Finally, conclusions and potential future work are stated in Section 5.

2. Overview of the actuator-management scheme based on nonlinear, real-time optimization

First, some notation and definitions are introduced in order to facilitate the understanding of the optimization scheme that forms the basis of the actuator-management algorithm. The plasma state is denoted by $x \in \mathbb{R}^{n_x}$, where n_x is the number of plasma states. It is assumed that an estimation of x is available in real time either through direct diagnostic measurements or through a state estimator. The vector of actuator commands is denoted by $u \in \mathbb{R}^n$, where n is the total number of actuators. The vector u is determined by the actuator manager. The vector of feedback-controller requests is denoted by $R \in \mathbb{R}^N$ where N is the total number of feedback (FB) controllers, i.e. each FB controller sends one request. The FB-controller requests are physically meaningful magnitudes computed by the FB controllers, like for example, the desired total injected power, NBI torque, localized current deposition, ECH&CD aiming location, etc. Also, it must be noted that the components of R are computed by FB controllers that are independently designed for particular control goals (e.g. normalized β control, rotation control, current-profile control, TM suppression, etc.). Finally, a vector of virtual-input functions is denoted by $F \in \mathbb{R}^N$, and described next. In general, F may depend on x, u, and other additional parameters, $p \in \mathbb{R}^p$, so F is a mapping from $(u, x, p) \in \mathbb{R}^{n+n_x+n_p}$ into $R \in \mathbb{R}^N$. This mapping is necessary for the actuator manager to have a relationship between controller requests R and actuator commands u. For example, if a FB controller regulates β and its request R_1 is the total NBI power, and u_i are the power commands sent to the NBIs ($i = 1, \ldots, N_{NBI}$, where N_{NBI} is the total number of NBIs), then the associated component of F, denoted by F_1, is given by $F_1 = \sum_{i=1}^{i=N_{NBI}} u_i$. Another example would be a FB controller that sends a NBI torque request, R_2. In such case, the associated virtual input F_2 could be expressed as $F_2 = \sum_{i=1}^{i=N_{NBI}} k_i u_i$, where $k_i \subset p$ are parameters that characterize the NBI-torque injection (e.g. $k_i > 0$ for co-current NBIs and $k_i < 0$ for counter-current NBIs). In these examples, F is linear in u. Other virtual-input functions may be very simple (e.g. fixing a particular actuator command, $F_j = u_i$, where $j \in \{1, \ldots, N\}$) or more complex (e.g. a nonlinear function $F_j = F_j(u, x, p)$). In any case, the actuator-management scheme presented in this work does not require a specific mathematical form for F.

Within the actuator manager, an optimization problem is solved at each sampling time as given by

$$\min(\{u\} - u_{ref})^T \mathbf{Q}(\{u\} - u_{ref}) + \{s\}^T \mathbf{T} \{s\}, \tag{1}$$

subject to

$$\textbf{Controller-request constraints}:\ F(\{u\}, x, p) + \{s\} = R, \tag{2}$$

$$\textbf{Saturation limits}:\ \{u\} \subset \mathscr{U}, \tag{3}$$

where the brackets $\{\cdot\}$ are used within (1)–(3) to denote to-be-optimized variables (i.e. u and s), $\mathbf{Q} > 0 \in \mathbb{R}^{n\times n}$ and $\mathbf{T} > 0 \in \mathbb{R}^{N\times N}$ are symmetric, positive-definite design matrices that weigh the relative importance of actuators (through $\mathbf{Q}$) and controller requests (through $\mathbf{T}$), $u_{ref} \in \mathbb{R}^n$ is a vector of reference actuator commands, $s \in \mathbb{R}^N$ is a vector of slack variables that characterize the fulfillment of the different controller requests, and $\mathscr{U}$ the set of feasible actuator commands, i.e. a characterization of the actuator saturation limits. The newly introduced variables $\mathbf{Q}$, $\mathbf{T}$, u_{ref}, and s are thoroughly described below.

By stating the actuator-management problem as in (1)–(3), one goal is to satisfy R as closely as possible through the mapping F which depends on u in (2). This is because s is minimized through $\mathbf{T}$ within (1), so if $s \to 0$ then $F(u, x, p) \to R$, i.e. the controller requests R are fulfilled. Controller requests R of higher priority must employ higher terms within $\mathbf{T}$, so that the associated components of s are prioritized. In fact, the components of s have physical relevance, as they represent a lack or excess in the realization of the related controller request. For example, if a controller request given by the total NBI power is equal to 5 MW, but only 4 MW are available (e.g. as a result of saturation limits, or as a compromise to fulfill other control tasks), then $s_j \geq 1$ MW, which means that there will be a lack of at least 1 MW of NBI power. An analogous physical interpretation would be given to other slack variables associated with lack/excess in the achievement of torque requests, localized current requests, etc. From a mathematical perspective, solving for s in addition to u in (1)–(3) makes the optimization problem always feasible, so there is always at least one optimal solution. On the other hand, the actuator commands u_i for which smaller actions are desired must employ higher terms within $\mathbf{Q}$. The vector u_{ref} may be understood as a feedforward actuator command which is computed offline, and fulfills objectives such as closeness to a particular x evolution or increased actuator reliability. For example, if an actuator's command needs to be as close as possible to its associated u_{ref} component (for the example of an NBI, to achieve a

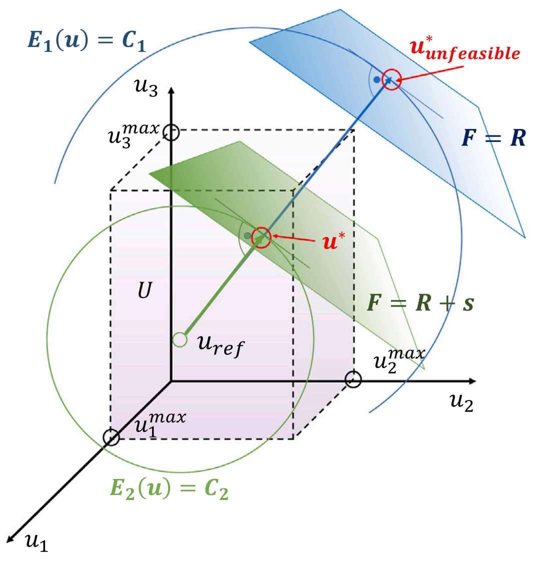


Fig. 1. Graphical representation of the nonlinear optimization problem for actuator management in tokamaks. A particular case with $N = 1$, $n = 3$, linear constraints, and diagonal $\mathbf{Q}$ and $\mathbf{T}$ is illustrated when: (1) R is attained through F with $s = 0$ (shown in blue), with unfeasible optimal solution $u^*_{unfeasible}$, and (2) R is attained through F with $s \neq 0$ (shown in green), with optimal solution u^*. The feasible set $\mathscr{U}$ is the cube shown in faded purple, whereas the optimal solutions u^* and $u^*_{unfeasible}$ are shown in red. The ellipsoids $E_1(u) = C_1$ and $E_2(u) = C_2$ (depicted as circumferences to simplify the diagram) are tangent to $F = R$ and $F = R + s$, respectively, and the intersections correspond to the optimal solutions $u^*_{unfeasible}$ and u^*.

particular delivered power, or as a protective measure against excessive power modulation), the terms in $\mathbf{Q}$ related to the particular actuator can be increased. Finally, it must be noted that the general actuator-management framework described here allows for carrying out both SMM and RP sharing by means of a single optimization scheme.

Fig. 1 shows a graphical representation that illustrates the optimization problem (1)–(3) in the u subspace at a particular time instant. To help visualize the problem, a 3D representation is utilized (i.e. $n = 3$) with a single, linear actuator-request (i.e. $N = 1$ and F is linear with u) and diagonal $\mathbf{Q}$ and $\mathbf{T}$ matrices. However, the ideas exposed in this paragraph apply to any n and N values, and can be generalized for any F, $\mathbf{Q}$, and $\mathbf{T}$. The feasible set $\mathscr{U}$ (shown in faded purple within Fig. 1) is defined by the maximum values of the components of u, denoted by u_k^{max} ($k = 1, 2, 3$), whereas the minimum values of u are set to 0. If all controller requests are fulfilled, then R can be attained exactly through F. In such case, $F = R$ with $s = 0$. However, in general, $F = R$ with $s = 0$ may lie out of the feasible set $\mathscr{U}$, as represented by the blue hyperplane in Fig. 1. The optimal solution $u^*_{unfeasible}$ (shown in red in Fig. 1) would be the point of $F = R$ that is tangent to the ellipsoids defined by $E_1(u) = (u - u_{ref})^T \mathbf{Q}(u - u_{ref}) = C_1$, where C_1 is a constant. Such ellipsoids are centered at u_{ref}. Therefore, $F + s = R$ with $s \neq 0$ (represented by the green hyperplane in Fig. 1) may be required in order to have a feasible solution. The feasible optimal solution u^* (shown also in red in Fig. 1) is defined by the point of $F = R + s$ that is tangent to the sphere $E_2(u) = (u - u_{ref})^T \mathbf{Q}(u - u_{ref}) = C_2$, where $C_2 < C_1$. The slack variable s makes the hyperplane $F = R$ move toward the interior of $\mathscr{U}$ so that a feasible solution for u can be found, i.e. a finite deviation in the execution of the control request must be allowed in order to find control commands within saturation limits.

3. Algorithmic approach to AM via nonlinear, real-time optimization

3.1. Augmented Lagrangian method

The optimization problem (1)–(3) has both general nonlinear constraints (given by (2)) and boundary constraints (given by (3)).[1] In this work, (1)–(3) is reformulated by means of the augmented Lagrangian method [7] so that only boundary constraints are present. This yields an optimization problem that, in principle, can be more easily solved.

The augmented Lagrangian function, denoted by $\mathscr{L}_A$, is defined as

$$\begin{aligned}\mathscr{L}_A(u, s, \lambda, \mu) \triangleq{}& [u - u_{ref}]^T \mathbf{Q}[u - u_{ref}] + s^T \mathbf{T} s \\ &- \lambda^T [F + s - R] + \frac{\mu}{2} \|F + s - R\|_2^2,\end{aligned} \tag{4}$$

where $\lambda^T \in \mathbb{R}^N$ is a vector that contains the Lagrange multipliers λ_i ($i = 1, \ldots, N$), and $\mu \in \mathbb{R}$ is a new parameter of the optimization problem which must be positive. The significance of μ is explained next. From (4), it can be seen that $\mathscr{L}_A(u, s, \lambda, 0)$ corresponds to the usual Lagrangian function, so the addition of the term $\frac{\mu}{2}\|F + s - R\|_2^2$ makes the Lagrangian function be "augmented". Also, the inclusion of such term $\frac{\mu}{2}\|F + s - R\|_2^2$ makes the minimization of $\mathscr{L}_A$ with $\mu > 0$ achieve both the minimization of the cost function in (1) and reinforces the fulfillment of the constraint (2) [7].

Therefore, the optimization problem (1)–(3) can be rewritten as

$$\min_{u,s,\lambda} \mathscr{L}_A(u, s, \lambda, \mu), \text{ subject to } \textbf{saturation limits}: u \subset \mathscr{U}. \tag{5}$$

Given initial values for u, s, λ, and μ denoted by u_0, s_0, λ_0, and μ_0, respectively, and constraint and solution-convergence tolerances, denoted by TOL_{const} and TOL_{sol}, respectively, the optimization problem in (5) (and, therefore, the original problem (1)–(3)) can be solved with the iterative procedure described by Algorithm 1, which is based on that presented in [7].

Algorithm 1 (*Augmented Lagrangian Method*).

```
* Step 0. Set initial values: $u_0$, $s_0$, $\lambda_0$, $\mu_0$
- Start loop: for $j = 0, 1, 2, \ldots$
* Step 1. Solve (5) with fixed $\mu = \mu_j$ and $\lambda = \lambda_j$, and initial iterates $u_j$ and $s_j$ → Obtain $u_{j+1}$, $s_{j+1}$
* Step 2. Check constraints:
    – If $\|F(u_{j+1}) + s_{j+1} - R\|_2 \leq \text{TOL}_{const}$ → Set $\lambda_{j+1} = \lambda_j - \mu_j [F(u_{j+1}) + s_{j+1} - R]$, $\mu_{j+1} = \mu_j$, and go to Step 3
    – Else → Set $\lambda_{j+1} = \lambda_j$, $\mu_{j+1} > \mu_j$, increase $j$, and start over at Step 1
* Step 3. Check convergence of the solution $u_{j+1}$, $s_{j+1}$:
    – If $\|\nabla \mathscr{L}_A(u_{j+1}, s_{j+1}, \lambda_j, \mu_j)\|_2 \leq \text{TOL}_{sol}$ → Stop: optimal solution is $u_{j+1}$, $s_{j+1}$
    – Else → Increase $j$, start over at Step 1
```

In Algorithm 1, Step 1 requires obtaining u_{j+1} and s_{j+1} by solving (5) with fixed $\lambda = \lambda_j$ and $\mu = \mu_j$. This subproblem is solved by means of a nonlinear gradient-projection method that utilizes a line-search approach (see Section 3.2).

[1] For a generic variable x, a boundary constraint is a limit on the lower and upper values that such variable can take. Therefore, a boundary constraint has the form $l \leq x \leq u$, where l and u are the lower and upper bounds for x, respectively. On the other hand, a general constraint has a broader form given by $f(x) = 0$.

3.2. Nonlinear gradient-projection method

The nonlinear gradient-projection method described in this section is used to solve (5) with fixed $\mu = \mu_j$ and $\lambda = \lambda_j$, as stated in Step 1 of Algorithm 1 (see Section 3.1). This method starts from the initial values given by u_j, s_j, which are defined in Step 1 of Algorithm 1. For convenience, $\hat{u} \triangleq [u, s]^T$ is defined so that it contains the optimization variables of this section, and $\hat{u}_0 = [u_j, s_j]^T$ is the initial point (the index j is fixed in this section). The main idea of this method is to approximate $\mathscr{L}_A$ for a quadratic cost function around the neighborhood of an iterate $\hat{u}_k$ ($k = 0, 1, 2, \ldots$, where $\hat{u}_0 = [u_j, s_j]^T$ is the initial iterate), until the gradient of the approximate quadratic cost function is smaller than a given tolerance. Then, u_{j+1}, s_{j+1} is found (see Algorithm 1, Step 1).

For convenience, the last two terms of $\mathscr{L}_A$ in (4) are grouped and defined as f, i.e.

$$f \triangleq -\lambda^T [F + s - R] + \frac{1}{2}\mu \|F + s - R\|_2^2. \tag{6}$$

Because F may be nonlinear with $\hat{u}$, it is approximated at the kth iteration using a Taylor's series expansion up to the second order, as given by

$$f \approx f_k + \frac{1}{2}\nabla_{\hat{u}} f_k^T [\hat{u} - \hat{u}_k] + \frac{1}{2}[\hat{u} - \hat{u}_k]^T \mathbf{B_k} [\hat{u} - \hat{u}_k], \tag{7}$$

where $f_k \triangleq -\lambda_k^T \left[F(u_k) + s_k - R\right] + \frac{1}{2}\mu_k \|F(u_k) + s_k - R\|_2^2$ is f evaluated at $\hat{u}_k$, $\nabla_{\hat{u}}$ denotes the gradient with respect to $\hat{u}$, and the matrix $\mathbf{B_k}$ is the quadratic approximation of $\nabla^2_{\hat{u}} f_k$. Using (7), $\mathscr{L}_A$ in (4) is approximated by a cost function J given by

$$\begin{aligned}\mathscr{L}_A \approx J &\triangleq [u - u_{ref}]^T \mathbf{Q} [u - u_{ref}] + s^T \mathbf{T} s \\ &+ f_k + \frac{1}{2}\nabla_{\hat{u}} f_k^T [\hat{u} - \hat{u}_k] + \frac{1}{2}[\hat{u} - \hat{u}_k]^T \mathbf{B_k} [\hat{u} - \hat{u}_k],\end{aligned} \tag{8}$$

so that the nonlinear program (5) is approximated by a quadratic program with boundary constraints only, as given by

$$\min_{\hat{u}} J, \text{ subject to } \textbf{saturation limits}\colon \hat{u} \subset \hat{\mathscr{U}}. \tag{9}$$

where $\hat{\mathscr{U}}$ is the feasible set for $\hat{u}$, i.e. the set $\mathscr{U}$ generalized to include $\pm\infty$ bounds for the components of s. The optimal solution to (9) at the kth iteration is calculated as detailed in Section 3.3, and is denoted by $\hat{u}_k^*$. For numerical stability reasons, the next iterate $\hat{u}_{k+1}$ within the nonlinear gradient-projection method is computed from $\hat{u}_k^*$ using a line-search approach, i.e. $\hat{u}_{k+1} = \hat{u}_k + \alpha[\hat{u}_k^* - \hat{u}_k]$, where $0 < \alpha \leq 1$ is the step length that makes $J(\hat{u}_{k+1}) < J(\hat{u}_k)$. The process is repeated until $\|\nabla J(\hat{u}_{k+1})\| \leq \text{TOL}_{GP}$, for some tolerance TOL_{GP}. The overall iterative scheme of the nonlinear gradient-projection method is summarized in Algorithm 2. As introduced above, Algorithm 2 is only used within Step 1 of Algorithm 1.

Algorithm 2 (*Nonlinear Gradient Projection (Used Within Step 1 of Algorithm 1)*).

```
* Step 0. Set initial value: $\hat{u}_0 = [u_j, s_j]$
- Start loop: for $k = 0, 1, 2, \ldots$
* Step 1. Solve (9) with initial iterate $\hat{u}_k \rightarrow$ obtain $\hat{u}_k^*$
* Step 2. Line-search method: $\hat{u}_{k+1} = \hat{u}_k + \alpha(\hat{u}_k^* - \hat{u}_k)$
* Step 3. Check convergence of the solution $\hat{u}_{k+1}$:
    – If $\|\nabla J(\hat{u}_{k+1})\|_2 \leq \text{TOL}_{GP} \rightarrow$ Stop: optimal solution is $\hat{u}_{k+1} = [u_{j+1}, s_{j+1}]^T$
    – Else $\rightarrow$ Increase $k$, start over at Step 1
```

In Algorithm 2, Step 1 requires obtaining $\hat{u}_k^*$ by solving (9). This subproblem is solved by means of a standard gradient-projection method for quadratic programs equipped with a conjugate-gradient algorithm (see Section 3.3).

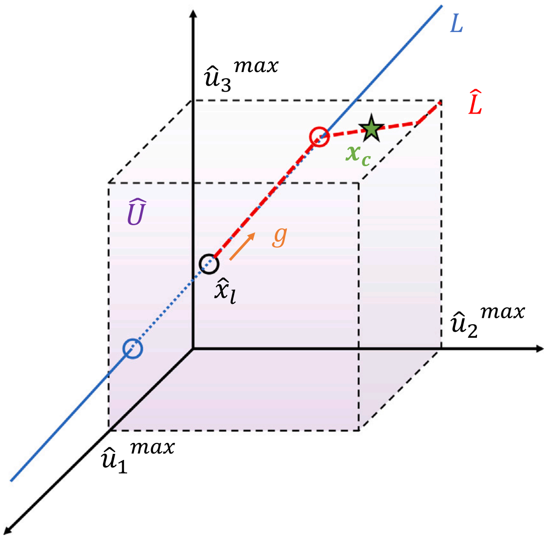


Fig. 2. Graphical representation of the standard gradient-projection method in 3D. The point $\hat{x}_l$ is inside $\hat{\mathscr{U}}$, and the maximum descent direction of J is given by g. The straight, continuous line L is defined by $\hat{x}_l$ and g. Also, L is plotted as blue solid when it lies out of $\hat{\mathscr{U}}$, and blue dotted when it lies within $\hat{\mathscr{U}}$. The piecewise continuous line $\hat{L}$ (red dashed) is defined by the projection of L onto $\hat{\mathscr{U}}$. It can be noted that $\hat{L}$ starts at $\hat{x}_l$, and bends first at the point where L overcomes $\hat{u}_3^{max}$, i.e. where it leaves $\hat{U}$. Then, $\hat{L}$ bends again when L overcomes $\hat{u}_2^{max}$. Finally, the Cauchy point x_c (green) can have components on the boundary ($\hat{x}_{c,1}$ in the represented case) or interior ($\hat{x}_{c,2}$ and $\hat{u}_{c,3}$ in the represented case) of $\hat{\mathscr{U}}$. Therefore, the index sets within (12)–(14) are $\mathscr{I} = 1$, and $\mathscr{E} = \{2, 3\}$.

3.3. Gradient-projection method and conjugate gradient method

The quadratic program (9) is solved using a standard gradient-projection method that uses a conjugate-gradient algorithm to solve the arising subproblems. This method starts from the initial value given by $\hat{u}_k$, which is defined in Step 1 of Algorithm 2. In order to clearly differentiate the iterates of this section from the iterates of previous sections, the iterates of this section are denoted by $\hat{x}_l$ (where $l = 0, 1, 2, \ldots$, and $\hat{x}_0 \triangleq \hat{u}_k$, so k is fixed in this section). The main idea of this method is to search for the optimal solution of (9) along the direction of maximum descent of the quadratic cost function J, and then explore the boundary of the feasible set where this solution lies. This is done in an iterative fashion, as explained next.

First, for convenience, the definition of J in (8) is rewritten as

$$J = \frac{1}{2}\hat{x}^T G \hat{x} + c^T \hat{x} + b, \tag{10}$$

where $G \in \mathbb{R}^{(n+N)\times(n+N)}$, $c \in \mathbb{R}^{n+N}$, and $b \in \mathbb{R}$ depend on $\mathbf{Q}$, $\mathbf{T}$, f_k, $\mathbf{B_k}$, and $\hat{u}_k$ as given by (A.3) and (A.5) (see Appendix A). At each iteration with initial point $\hat{x}_l$, the direction of maximum descent for J is given by its negative gradient, i.e.

$$g = -\left(G_l \hat{x}_l + c_l\right), \tag{11}$$

where G_l and c_l denote that G and c are evaluated at $\hat{x}_l$. With the direction defined by g in (11) and the point $\hat{x}_l$, a continuous, straight line L can be defined. At each iteration with initial point $\hat{x}_l$, the standard gradient-projection method has two stages. In the first stage, the objective is to find the first local minimizer of J along L and within $\hat{\mathscr{U}}$ (see Fig. 2 for a 3D representation of the standard gradient-projection method). The projection of L onto $\hat{\mathscr{U}}$ is denoted by $\hat{L}$. The first local minimizer of J along $\hat{L}$ is denoted by x_c (in the literature, this point is known as "Cauchy point" [7]). It must be taken into account that x_c may have some components that are on the boundary of $\hat{\mathscr{U}}$, and others in the interior of $\hat{\mathscr{U}}$.

In the second stage, the objective is to find a solution to the problem given by

$$\min_{\hat{u}} J, \tag{12}$$

subject to:

$$\hat{x}_i = x_{c,i},\ \forall i \in \mathscr{I}, \tag{13}$$

$$\hat{x}_i \subset \hat{\mathscr{U}},\ \forall i \in \mathscr{E} \tag{14}$$

where $\hat{x}_i$ and $x_{c,i}$ are the ith components of $\hat{x}$ and x_c, respectively (for $i = 1, \ldots, n+N$), $\mathscr{I}$ is the set of indexes corresponding to $x_{c,i}$ that lie on the boundary of $\hat{\mathscr{U}}$, and $\mathscr{E}$ is the set of indexes corresponding to $x_{c,i}$ that are in the interior of $\hat{\mathscr{U}}$. It can be noted that (12)–(14) is equivalent to finding the local minimizer on a subset of the boundary of $\hat{\mathscr{U}}$ (i.e. on the face of $\hat{\mathscr{U}}$ where x_c lies).

In some cases, the problem in (12)–(14) can be very similar to that in (9) and, therefore, equally hard to solve. The idea is to solve (12)–(14) in an approximate fashion by relaxing the second constraint, i.e. $\hat{x}_i \subset \hat{\mathscr{U}}, \forall i \in \mathscr{E}$. Such approximate solution is denoted by $\bar{x}$. If any of the components of $\bar{x}$ falls out of the feasible set $\hat{\mathscr{U}}$, then such component is projected back onto $\hat{\mathscr{U}}$. Therefore, if $\bar{x}_i > \hat{u}_i^{max}$ or $\bar{x}_i < \hat{u}_i^{min}$ (where $\bar{x}_i$ is the ith component of $\bar{x}$, $\forall i \in \mathscr{E}$, and $\hat{u}_i^{max}$ and $\hat{u}_i^{min}$ are the maximum and minimum values for $\hat{x}_i$, respectively), then $\bar{x}_i = \hat{u}_i^{max}$ or $\bar{x}_i = \hat{u}_i^{min}$ are set. The projection of $\bar{x}$ onto $\hat{\mathscr{U}}$ provides the next iterate, $\hat{x}_{l+1}$. The overall gradient-projection scheme is summarized in Algorithm 3, where TOL_{GP}^* is the associated tolerance. As introduced above, Algorithm 3 is only used within Step 1 of Algorithm 2.

Algorithm 3 (*Gradient Projection Method (Used Within Step 1 of Algorithm 2)*).

```
* Step 0. Set initial value: $\hat{x}_0 = \hat{u}_k$
  Start loop: for $l = 0, 1, 2, \ldots$
* Step 1 (stage 1). Calculate $g$, $\hat{L}$ and $x_c$
* Step 2 (stage 2). Solve (12)–(14) in an approximate fashion with
  initial iterate $\hat{x}_l$ → Obtain $\bar{x}$, and calculate $\hat{x}_{l+1}$ by projecting $\bar{x}$
  onto $\hat{\mathscr{U}}$smm
* Step 3. Check for convergence:
    – If $\|\nabla J(\hat{x}_{l+1})\|_2 \le \mathrm{TOL}_{GP}^*$ → Stop: optimal solution is $\hat{x}_{l+1}$,
      set $\hat{u}_k^* = \hat{x}_{l+1}$
    – Else → increase $l$ and start over at Step 1
```

In Algorithm 3, Step 2 is tackled by means of the conjugate-gradient algorithm presented in Appendix B.

4. Experimental testing of the actuator manager in DIII-D

The capabilities of the actuator-management algorithm have been tested in DIII-D experiments. The objective was assessing the actuator-manager performance despite actuator trips, plasma-state variations, and control-priority changes, while trying to satisfy as many FB controller requests as possible in order to maintain a particular plasma evolution. The DIII-D high-q_{min} scenario was employed, as it is a relevant scenario for the successful demonstration of steady-state operation in ITER [11]. Some scenario parameters of interest are the plasma current ($I_p \approx 1.05$ MA), toroidal field ($B_T \approx 1.65$ T), normalized plasma β ($\beta_N \approx 3.5$), line-average electron density ($\bar{n}_e \approx 4.5 \times 10^{19}$ m^{-3}), and non-inductive current fraction ($f_{NI} \approx 75\%$).

Four actuators ($n = 4$) were managed during these experimental tests, namely, (i) central solenoid (E-coil), (ii) on-axis NBI, (iii) two off-axis NBIs (which were controlled as a single actuator), and (iv) ECH&CD. The associated actuator commands u are given by (i) plasma current, I_p, (ii) on-axis NBI power, $P_{NBI}^{\text{ON-axis}}$, (iii) the off-axis NBI power, $P_{NBI}^{\text{OFF-axis}}$, and (iv) EC power, P_{EC}.[2] These actuators were shared for four different control tasks ($N = 4$) which have kinetic, magnetic, and MHD-instability control purposes: (i) stored thermal-energy (W) control, (ii) edge safety-factor (q_e) control, (iii) central safety-factor (q_0) control, and (iv) TM suppression. Independently designed FB controllers [9,10] sent their requests R_i ($i = 1, \ldots, 4$) to the actuator manager. The controller requests are: (1) total injected-power request (R_1) for W control, (2) total plasma-current request (R_2) for q_e control, (3) local current-density request (R_3) for q_0 control, and (4) total EC-power request (R_4) for TM suppression.[3] Therefore, the associated virtual-input functions that map these controller requests R_i ($i = 1, \ldots, 4$) into the actuator commands are: (1) $F_1 = P_{NBI}^{\text{ON-axis}} + P_{NBI}^{\text{OFF-axis}} + P_{EC} \triangleq P_{tot}^{AM}$, (2) $F_2 = I_p$, (3) $F_3 = F_3(q_0, I_p, P_{NBI}^{\text{ON-axis}}, P_{NBI}^{\text{OFF-axis}}, P_{EC})$ (F_3 is a nonlinear function [9]), and (4) $F_4 = P_{EC}$. Because F is nonlinear in u, the problem (1)–(3) considered in these experiments is a nonlinear-optimization problem. Table 1 includes a summary of R_i and F_i ($i = 1, \ldots, 4$).

Table 1
Summary of controller requests, R_i, and virtual inputs, F_i.

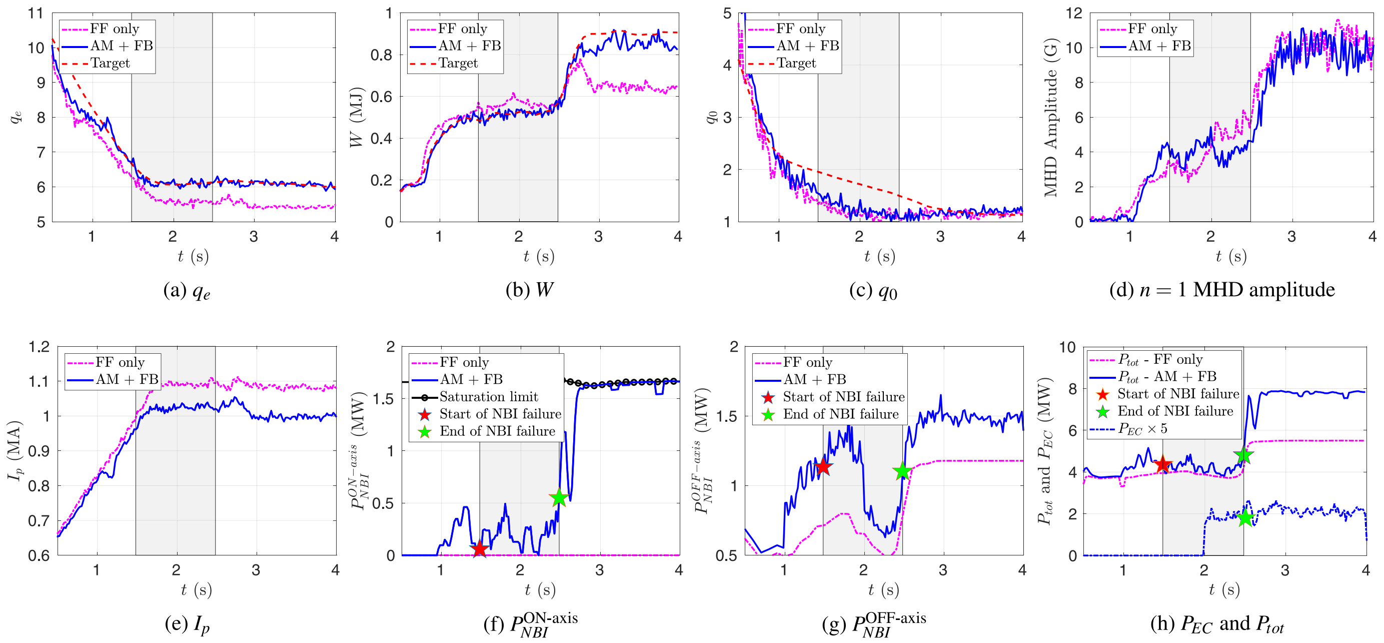
i	Request (R_i)	Virtual input (F_i)
1	Total injected power	$P_{NBI}^{\text{ON-axis}} + P_{NBI}^{\text{OFF-axis}} + P_{EC}$
2	Total plasma current	I_p
3	Local current density	$F_3(q_0, I_p, P_{NBI}^{\text{ON-axis}}, P_{NBI}^{\text{OFF-axis}}, P_{EC})$
4	Total EC power	P_{EC}

During these experiments, the physical saturation limits changed over time based on the actuators status. Without actuator trips, they were approximately given by $I_p \in [0.3, 2.0]$ MA, $P_{NBI}^{\text{ON-axis}} \in [0.0, 1.70]$ MW, $P_{NBI}^{\text{OFF-axis}} \in [0.0, 2.90]$ MW, and $P_{EC} \in [0.0, 1.00]$ MW. In addition, some actuator trips were introduced that were not known beforehand by the actuator manager (more details about these trips are specified later in Sections 4.1 and 4.2). Some NBIs were employed during this experiment for diagnostic-data acquisition. The total power delivered by these NBIs is denoted by P_{tot}^{diag}, whereas the total power delivered by the NBIs managed by the actuator-management algorithm is denoted by P_{tot}^{AM}. The total injected power is denoted by $P_{tot} \triangleq P_{tot}^{AM} + P_{tot}^{diag}$. Also, ECH&CD actuation was enabled only after $t = 2$ s, thus introducing a change in the availability of the actuators that is unexpected to the actuator-management scheme.

The controller requests were prioritized as follows: (1st) q_e control (highest priority), (2nd) TM suppression (only if a TM exists), and (3rd) W and q_0 control (lowest priority). By default, there is no need for TM suppression at the beginning of a shot. The matrix $\mathbf{T}$ within (1)–(3) was set up a priori to reflect such priorities. However, the Off-Normal Fault-Response (ONFR) system [8] was employed in these experiments to monitor $n = 2$ MHD modes. Therefore, $\mathbf{T}$ was updated in real time to allow for such RP sharing of the ECH&CD system. Due to the present monitoring capabilities of ONFR, only $n = 2$ modes led to changes in $\mathbf{T}$. However, the actuator manager could allow for other real-time changes in controller-request priorities (e.g. prioritizing q_0 regulation above 1 if the sawtooth instability becomes an issue). Finally, the matrix $\mathbf{R}$ did not prioritize the use of any particular actuator. Within the actuator manager, $u_{ref} = u_{FF}$ was employed, where u_{FF} is a pre-defined feedforward

[2] Other actuator commands could be employed by the present actuator-management algorithm, such as the E-coil current/voltage, the NBI modulation-timing and voltages, or the EC timing and mirror angles. However, in this work, such lower-level actuator commands are considered to be controlled by other PCS components that have control functions related to the actuators hardware. Only commands with a particular physical meaning (i.e. plasma current, power, etc.) are considered by the actuator manager.

[3] The ECCD deposition location, and therefore the EC aiming, play a critical role for TM suppression [10]. Although the actuator manager can handle EC aiming commands, such aspect is not included in this paper due to both hardware and software issues with EC aiming during the reported experiments.



(a) q_e (b) W (c) q_0 (d) $n = 1$ MHD amplitude (e) I_p (f) $P_{NBI}^{\text{ON-axis}}$ (g) $P_{NBI}^{\text{OFF-axis}}$ (h) P_{EC} and P_{tot}

Fig. 3. Time evolutions for q_e, W, q_0, $n = 1$ MHD amplitude, I_p, $P_{NBI}^{\text{ON-axis}}$, $P_{NBI}^{\text{OFF-axis}}$, P_{EC}, and P_{tot} in FF-only (magenta dashed-dotted) and AM + FB (blue solid) shots during DIII-D experiments, together with the targets (red dashed). The FF-only shot is 185 362 and the AM + FB shot is 185 372. During the FF-only shot, $P_{NBI}^{\text{ON-axis}}$ and P_{EC} are zero. Also, the start and end of the diagnostic-NBI failure are marked by a red and green star, respectively, and the period when such NBI fails is shaded in gray in all figures.

(FF) evolution. The experimental actuator commands shown in the next subsections are averaged over a 60 ms time span, in order to facilitate their visualization.

4.1. Case 1: Simultaneous multi-mission sharing with partial failure in diagnostic NBI

A demonstration of the actuator manager (AM) capabilities to do SMM sharing can be found in shot 185 372 (see Fig. 3). By means of the AM scheme + FB controllers (AM + FB), the objective was to simultaneously drive q_e, W, and q_0 (solid blue lines in Figs. 3(a), 3(b), and 3(c), respectively) toward particular target evolutions (dashed red lines in Figs. 3(a), 3(b), and 3(c)) starting at $t = 0.9$ s. No TM suppression was considered in this AM + FB shot and, therefore, no RP sharing is carried out. A partial failure in a diagnostic NBI was emulated when $t \in [1.5, 2.5]$ s. Such NBI is delivering about 0.8 MW, but the partial failure limits its maximum power to around 0.4 MW, i.e. P_{tot}^{diag} is reduced by 0.4 MW. The AM scheme was responsible for maintaining the control performance despite such actuator loss. For comparison, shot 185 362 is included, which employed FF-only inputs u_{FF}, i.e. the FB controllers and AM algorithm were turned off. It can be observed that successful W and q_e control, as well as acceptable q_0 regulation, were achieved with the AM + FB scheme. Because q_e control had the highest control priority, I_p was reduced in AM + FB (see Fig. 3(e)) in order to elevate q_e and keep a close tracking during the whole shot. Reducing I_p without modifying P_{tot} would have made the plasma colder (lower W). To avoid that, the W controller increased R_1 and the AM scheme calculated a higher P_{tot} with respect to the FF only case (see Fig. 3(h)) in order to adapt to the I_p change. The modulation of P_{tot} was successfully carried out despite the diagnostic-NBI partial failure at $t = 1.5$ s, when P_{tot} suffers a sudden drop (see the red star in Fig. 3(h)). Excellent W tracking was achieved till $t \approx 2.75$ s. After that, W is kept near its target, but significant $n = 1$ MHD activity (see Fig. 3(d)) contributed to the variations found in W. On the other hand, q_0 was kept slightly closer to its target in AM + FB than in FF-only, but did not converge toward it until $t \approx 3$ s despite very significant increases in $P_{NBI}^{\text{OFF-axis}}$ and P_{EC} (see Figs. 3(g) and 3(h)). This lack of responsiveness in q_0 to the actuation is not related to the AM algorithm or the q_0 controller because these components of the control scheme did command the appropriate actuators (i.e. the off-axis current-drive sources). It is very likely that the $n = 1$ MHD activity hindered the possibility of increasing q_0. Other shots with no significant MHD activity (as shown in Section 4.2) showed better q_0 control.

Figs. 3(f), 3(g), and 3(h) show the evolution of $P_{NBI}^{\text{ON-axis}}$, $P_{NBI}^{\text{OFF-axis}}$, and P_{EC} (the latter is multiplied by 5 within Fig. 3(h) to facilitate visualization). The NBI-power modulation started at $t = 0.9$ s when the AM + FB scheme was turned on. At $t = 1.5$ s, $P_{NBI}^{\text{ON-axis}}$ and $P_{NBI}^{\text{OFF-axis}}$ were increased to compensate for the reduction in P_{tot}^{diag} (not shown). In addition, when P_{EC} came on at $t = 2$ s, the AM scheme determined that $P_{NBI}^{\text{OFF-axis}}$ had to be reduced in order to minimize $[u - u_{ref}]^T \mathbf{Q}[u - u_{ref}]$ in (1) while maintaining the overall control performance (see Figs. 3(a), 3(b), and 3(c)). A big increase in $P_{NBI}^{\text{ON-axis}}$ and $P_{NBI}^{\text{OFF-axis}}$ is found at $t = 2.5$ s due to the high W target, which made $P_{NBI}^{\text{ON-axis}}$ saturate during some periods of time.

4.2. Case 2: Simultaneous multi-mission sharing with total failure in FB-controlled NBI

Shot 185 376 shows the performance of the AM algorithm to do SMM sharing while also dealing with a FB-controlled actuator failure (see Fig. 4). In particular, a total failure in one of the off-axis NBIs was introduced after $t = 3$ s, i.e. the delivered power for $P_{NBI}^{\text{OFF-axis}}$ was reduced to around 0.65 MW. As in Section 4.1, the objective was to simultaneously regulate q_e, W, and q_0 (see Figs. 4(a), 4(b), and 4(c), respectively) around their targets. The targets for q_0 and q_e are the same as those in Section 4.1, but not for the target for W, which is slightly higher. Shot 185 362 (FF-only) is included as well for reference. For clarity, Figs. 4(b) and 4(c) do not include the FF-only evolutions (they can be found in Figs. 3(a) and 3(c)). It can be observed that good tracking for all q_e, W, and q_0 was achieved in the AM + FB case. Modulation of P_{tot} (see Fig. 4(h)) made W attain its target. A drop in both W and P_{tot} can be observed at $t \approx 3$ s (red star) due to the loss of the off-axis NBI, but P_{tot} recovered quickly by means of the FB action. This represents another successful demonstration of the AM scheme to replace tripped actuators. However, W did not recover as quickly, although it was driven toward its target. This slow response in W was

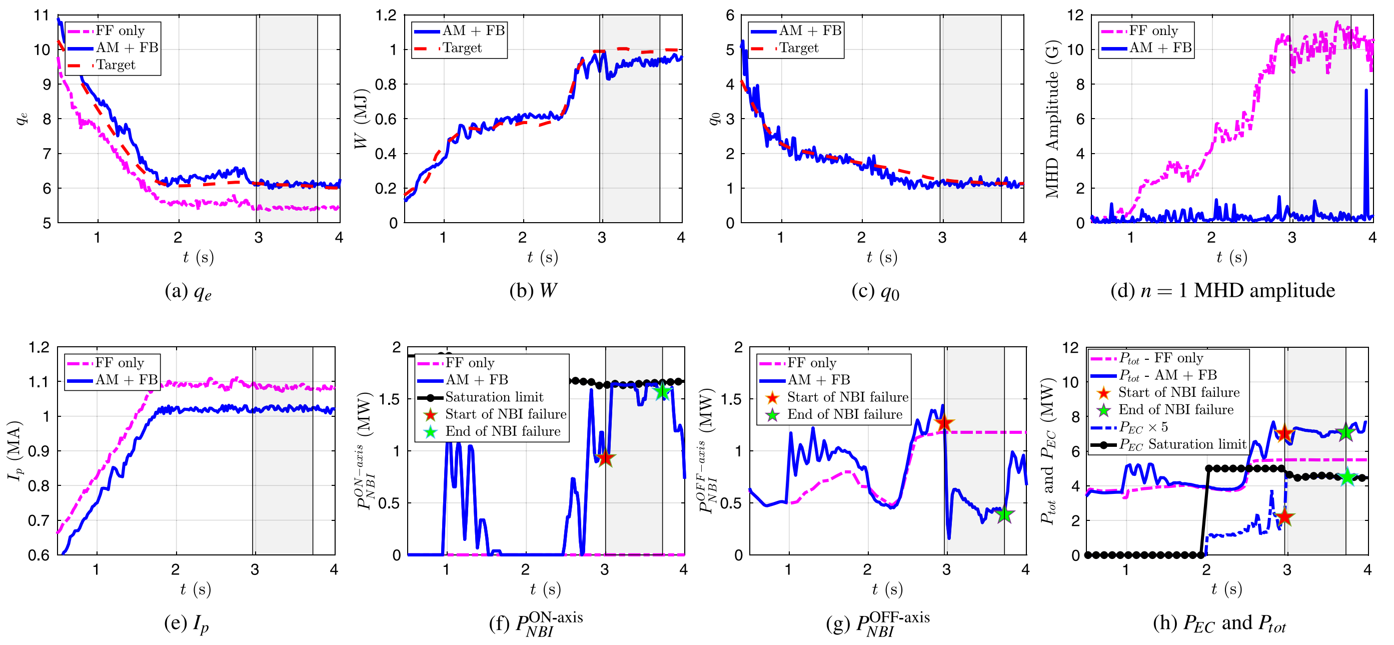


Fig. 4. Time evolutions for q_e, W, q_0, $n = 1$ MHD amplitude, I_p, $P_{NBI}^{\text{ON-axis}}$, $P_{NBI}^{\text{OFF-axis}}$, P_{EC}, and P_{tot} in FF-only (magenta dashed-dotted) and AM + FB (blue solid) shots during DIII-D experiments, together with the targets (red dashed). The FF-only shot is 185 362 and the AM + FB shot is 185 376. The start and end of the FB-controlled NBI failure are marked by a red and green star, respectively, within Figs. 4(f), 4(g), and 4(h), and the period when such NBI fails is shaded in gray in all figures.

not due to the MHD activity with $n = 1$ (see Fig. 4(d)) or $n = 2$ (not shown in this paper), which were very low under AM + FB. Instead, saturation of the actuators (see Figs. 4(f), 4(g), and 4(h)), in particular $P_{NBI}^{\text{ON-axis}}$ and P_{EC}, did not allow for reaching higher W. On the other hand, lower I_p was used all throughout the AM + FB shot (see Fig. 4(e)) to attain higher q_e. In this case, the regulation of q_e seems slightly worse than in Section 4.1 because of its dedicated controller, which used a lower proportional gain. Still, convergence of q_e toward its target was eventually achieved. Finally, very good tracking for q_0 was attained in this AM + FB case. Such control performance may have been enabled by the low MHD activity (see Fig. 4(d)) found in the AM + FB shot, as well as the higher W and lower I_p. In addition, the convergence of q_0 toward its target after $t \geq 3$ s seems to be enabled by the compatibility of the q_e, W, and q_0 evolutions, as well as the proper performance of the q_0 controller.

Fig. 4(f), 4(g), and 4(h) show the time evolution of $P_{NBI}^{\text{ON-axis}}$, $P_{NBI}^{\text{OFF-axis}}$, and P_{EC} (the latter is multiplied by ×5 to facilitate visualization). When the off-axis NBI fails at $t = 3$ s, an increase in both $P_{NBI}^{\text{ON-axis}}$ and P_{EC} is found to compensate for such loss. In fact, this made both $P_{NBI}^{\text{ON-axis}}$ and P_{EC} saturate during extended periods of time, making it difficult to achieve the target desired for W (see Fig. 4(b)). When the off-axis NBI failure was over, $P_{NBI}^{\text{ON-axis}}$ decreased to levels similar to those before the failure and $P_{NBI}^{\text{OFF-axis}}$ increased substantially, but the AM scheme determined that P_{EC} had to be kept at saturation levels in order to maintain the control performance.

4.3. *Case 3: Simultaneous multi-mission + re-purposing sharing*

Shot 185 375 shows the ability of the AM algorithm to do both SMM and RP sharing (see Fig. 5). Regulation of q_e, W, and q_0 was attempted (see Figs. 5(a), 5(b), and 5(c), respectively) while the ONFR system was employed to monitor the $n = 2$ MHD activity (see Fig. 5(d)). When the $n = 2$ MHD amplitude was above a 0.75 G detection threshold (shown as a dashed red line) for longer than 10 ms, ONFR detects the mode. At this time, an RP event (i.e. an event that requires the repurposing of an actuator) was generated for ECH&CD, and the TM-suppression controller request, R_4, was activated. This happened at around $t = 2.62$ s in shot 185 375. In addition, a recovery threshold of 0.2 G is set (shown as a dashed green line). If the MHD amplitude falls below the recovery threshold for longer than 10 ms, ONFR determines that there is no further need for TM suppression with another RP event. Although the $n = 2$ MHD amplitude fell below 0.2 G at around $t = 2.9$ s and $t = 3.25$ s, it can be seen that such drop was not long enough to generate another RP event. On the other hand, the control performance while doing SMM sharing is similar to that in Section 4.1. A significant increase in P_{tot} (see Fig. 5(h)) can be found in the AM + FB shot at $t = 2.62$ s because of the changes in the W target and the increase in P_{EC} associated with TM suppression. Also, I_p is reduced and modulated in the AM + FB case (see Fig. 5(e)) to elevate and regulate q_e around its target with the highest control priority. Significant modulation of I_p is found at around $t = 2.62$ s as a result of the MHD activity, which caused a temporary deviation in q_e (see Fig. 5(a)) that was quickly corrected by the AM + FB scheme.

Figs. 5(f), 5(g), and 5(h) shows the evolutions for $P_{NBI}^{\text{ON-axis}}$, $P_{NBI}^{\text{OFF-axis}}$, and P_{EC} (the latter is multiplied by ×5 to facilitate visualization). When the RP event was generated at $t = 2.62$ s, P_{EC} was driven to its maximum value as a result of the TM suppression request, R_4, activated by the MHD activity. Near the time instant of the RP event, both $P_{NBI}^{\text{ON-axis}}$ and $P_{NBI}^{\text{OFF-axis}}$ were adjusted by the AM scheme. On the one hand, $P_{NBI}^{\text{OFF-axis}}$ was decreased by about the same amount as P_{EC} was increased, because the AM scheme recognized both actuators as off-axis current sources. On the other hand, $P_{NBI}^{\text{ON-axis}}$ wiggled slightly before going up and hitting saturation due to the increase in the W target. The resulting variations in $P_{NBI}^{\text{OFF-axis}}$ and $P_{NBI}^{\text{ON-axis}}$ were caused by the variations in W arising from the significant MHD activity.

5. Conclusions and possible future work

An actuator-management scheme using nonlinear, real-time optimization has been developed and implemented within the DIII-D PCS. Experiments have successfully tested the actuator manager working in conjunction with several FB controllers for kinetic, magnetic, and MHD-related plasma variables. The FB controllers have been synthesized independently of the actuator manager. Also, the general design and implementation of the actuator-management algorithm does not change with the type of actuators and FB controllers employed, which allows for an easy control integration and combined testing. Functional

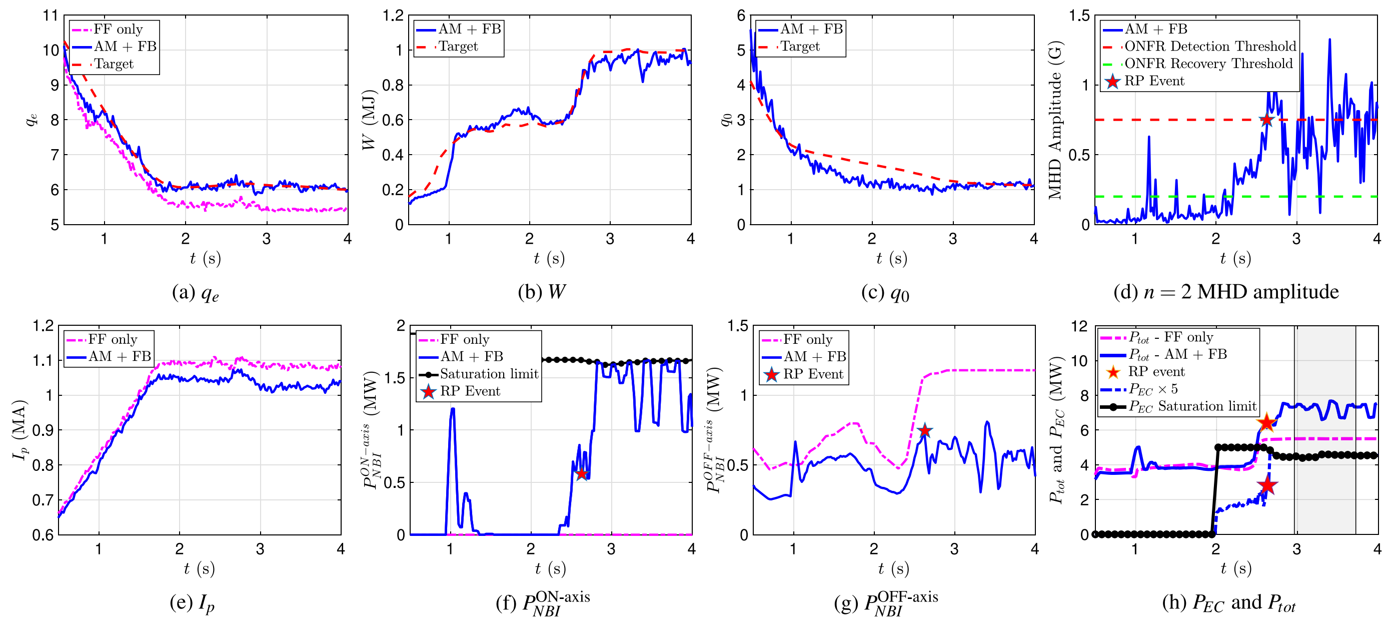


Fig. 5. Time evolutions for q_e, W, q_0, $n = 2$ MHD amplitude, I_p, $P_{NBI}^{\text{ON-axis}}$, $P_{NBI}^{\text{OFF-axis}}$, P_{EC}, and P_{tot} in FF (magenta dashed-dotted) and AM + FB (blue solid) shots during DIII-D experiments, together with the targets and ONFR detection threshold (red dashed). The FF-only shot is 185 362 and the AM + FB shot is 185 375. The RP-sharing event is marked with a red star.

integration is also established with other components of the DIII-D PCS architecture with supervisory and exception-handling functions, like ONFR. Unexpected actuator failures and other off-normal events are correctly handled by the control scheme during the experimental tests. In addition, the control performance of the FB controllers is not affected by the SMM and RP sharing functions carried out by the actuator manager. These advanced-control actuator-management functionalities, as well as the implementation and integration flexibility of the different control components, will be essential during the some stages of the commissioning of the ITER PCS for the burning-plasma operation phases, and will play a critical role in reactor-grade tokamaks. Moreover, in the DIII-D shots shown in this work, the actuator-management algorithm has a calculation time lower than 3 ms, and consistently converges at every sampling time, demonstrating the robustness and efficiency of the numerical scheme in real-time implementations.

Future work may include the use of additional actuators by the actuator manager, such as poloidal-field coils, 3D coils, and/or gas puffing/pellet injectors, the addition of a higher number of FB controllers to the scheme, and the implementation and application of the actuator-management capabilities in other devices and scenarios.

CRediT authorship contribution statement

Andres Pajares: Software, Methodology, Validation, Formal analysis, Data curation, Writing, Visualization. **Kathreen E. Thome:** Methodology, Resources, Formal analysis, Data curation. **Eugenio Schuster:** Conceptualization, Supervision, Project administration, Funding acquisition. **Michael L. Walker:** Supervision, Resources. **David A. Humphreys:** Supervision, Project administration, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment & Disclaimer

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, using the DIII-D National Fusion Facility, a DOE Office of Science user facility, under Award(s) DE-FC02-04ER54698 and DE-SC0010661.

Appendix A. Cost function matrices and vectors within gradient projection algorithm

Using (7), Eq. (8) can be rewritten as

$$J = u^T \mathbf{Q} u - 2u_{ref}^T \mathbf{Q} u + u_{ref}^T \mathbf{Q} u_{ref} + s^T Q s + f_k + \frac{1}{2}\nabla_u f_k^T \hat{u} - \frac{1}{2}\nabla_{\hat{u}} f_k^T \hat{u}_k + \frac{1}{2}\hat{u}^T \mathbf{B_k} \hat{u} - \hat{u}_k^T \mathbf{B_k} \hat{u} + \frac{1}{2}\hat{u}_k^T \mathbf{B_k} \hat{u}_k, \tag{A.1}$$

which, grouping the constant, linear, and quadratic terms with $\hat{u}$, is rewritten as

$$J = \hat{u}^T \left\{ \begin{bmatrix} \mathbf{Q} & \mathcal{O} \\ \mathcal{O} & \mathbf{T} \end{bmatrix} + \frac{1}{2}\mathbf{B_k} \right\} \hat{u} - \left\{ [2u_{ref}^T \mathbf{Q}, 0, \dots, 0] + \hat{u}_k^T \mathbf{B_k} - \frac{1}{2}\nabla_{\hat{u}} f_k^T \right\} \hat{u} + u_{ref}^T \mathbf{Q} u_{ref} + f_k - \frac{1}{2}\nabla_{\hat{u}} f_k^T \hat{u}_k + \frac{1}{2}\hat{u}_k^T \mathbf{B_k} \hat{u}_k \triangleq \frac{1}{2}\hat{u}^T G \hat{u} + c^T \hat{u} + b, \tag{A.2}$$

where

$$G \triangleq \begin{bmatrix} 2\mathbf{Q} & \mathcal{O} \\ 2\mathcal{O} & \mathbf{T} \end{bmatrix} + \mathbf{B_k}, \tag{A.3}$$

$$c \triangleq -[2u_{ref}^T \mathbf{Q}, 0, \dots, 0] - \hat{u}_k^T \mathbf{B_k} + \frac{1}{2}\nabla_{\hat{u}} f_k^T, \tag{A.4}$$

$$b \triangleq u_{ref}^T \mathbf{Q} u_{ref} + f_k - \frac{1}{2}\nabla_{\hat{u}} f_k^T \hat{u}_k + \frac{1}{2}\hat{u}_k^T \mathbf{B_k} \hat{u}_k, \tag{A.5}$$

and $\mathscr{O}$ denotes the zero matrix with appropriate dimensions.

Appendix B. Conjugate gradient algorithm

The solution $\bar{x}$ is calculated as

$$\bar{x} = Y\bar{x}_Y + Z\bar{x}_Z, \tag{B.1}$$

where $\bar{x}_Y \in \mathbb{R}^M$ contains the components $x_{c,i}$ with $i \in \mathscr{I}$ (which are fully known), the number of $x_{c,i}$ on the boundary of $\hat{\mathscr{U}}$ is denoted by M, $Y \in \mathbb{R}^{(n+N)\times M}$ and $Z \in \mathbb{R}^{(n+N)\times(n+N-M)}$ transform the components of $\bar{x}_Y$ and $\bar{x}_Z$ into the corresponding components of $\bar{x}$ (i.e. they are just matrices with ones and zeros, and appropriate dimensions), and $\bar{x}_Z \in \mathbb{R}^{n+N-M}$ is calculated with the following iterative method:

Algorithm 4 (*Conjugate Gradient Iteration (Used Within Step 2 of Algorithm 3)*).

```
* Step 0. Use an initial value $\bar{x}_Z^0$ and set $W = Z^T G Z$, $c_z = Z^T G Y \bar{x}_Y + Z^T c$
  Calculate $r_0 = Z^T G Z + c_z$, $g_0 = W^{-1} r_0$, $d_0 = -g_0$
  Start loop: for $j = 0, 1, 2, \dots$
* Step 1. Calculate step $\alpha = r_j^T g_j / (d_j^T W d_j)$ and $\bar{x}_Z^{j+1} = \bar{x}_Z^j + \alpha d_j$
* Step 2. Calculate $r_{j+1} = r_j + \alpha W d_j$ and $g_{j+1} = W^{-1} r_{j+1}$
* Step 3. Calculate $\beta = r_{j+1}^T g_{j+1} / (r_j^T g_j)$ and $d_{j+1} = -g_{j+1} + \beta d_j$
* Step 4. Check for convergence:
    – If $\|r_{j+1}^T W r_{j+1}\|_2 \le \mathrm{TOL}_{CG}$ → Stop: solution is $\bar{x}_Z^{j+1}$
    – Else → increase $j$ and start over at Step 1
```

where TOL_{CG} is the tolerance of the conjugate-gradient algorithm.

References

[1] D. Humphreys, et al., Novel aspects of plasma control in ITER, Phys. Plasmas 22 (2015) 021806.
[2] E. Maljaars, et al., Simultaneous control of plasma profiles and neoclassical tearing modes with actuator management in tokamaks, in: Proc. 42nd EPS Conference on Plasma Physics, 2015.
[3] E. Maljaars, et al., Actuator allocation for integrated control in tokamaks: Architectural design and a mixed-integer programming algorithm, Fusion Eng. Des. 122 (2017) 94–112.
[4] C.J. Rapson, et al., Experiments on actuator management and integrated control at ASDEX upgrade, Nucl. Fusion 53 (2013) 063020.
[5] N.M.T. Vu, et al., Tokamak-agnostic actuator management for multi-task integrated control with application to TCV and ITER, Fusion Eng. Des. 147 (2019) 111260.
[6] A. Pajares, et al., Actuator management via real-time optimization for integrated control in tokamaks, in: Proc. 46th EPS Conference on Plasma Physics, 2019.
[7] J. Nocedal, S.J. Wright, Numerical Optimization, second ed., Springer, 2006.
[8] N.W. Eidietis, et al., Implementing a finite-state off-normal and fault response system for disruption avoidance in tokamaks, Nucl. Fusion 58 (2018) 056023.
[9] A. Pajares, et al., Integrated robust control of individual scalar variables in tokamaks, in: Proc. 58th IEEE Conference on Decision and Control (CDC), 2019, pp. 3233–3238.
[10] R.L. Haye, et al., Control of neoclassical tearing modes in DIII-D, Phys. Plasmas 9 (2002) 2051.
[11] R.J. Buttery, et al., DIII-D research to prepare for steady state advanced tokamak power plants, J. Fusion Energy 38 (2019) 72–111.