OPEN ACCESS
IOP Publishing

Plasma Physics and Controlled Fusion

Plasma Phys. Control. Fusion **67** (2025) 015005 (9pp)

https://doi.org/10.1088/1361-6587/ad8ef2

Static actuator-sharing algorithm for concurrent control of multiple plasma properties

Sai Tej Paruchuri[1,*], Vincent Graber[1], Andres Pajares[2] and Eugenio Schuster[1]

[1] Department of Mechanical Engineering and Mechanics, Lehigh University, Bethlehem, PA 18015, United States of America
[2] General Atomics, San Diego, CA 92121, United States of America

E-mail: saitejp@lehigh.edu

Received 4 September 2024, revised 21 October 2024
Accepted for publication 5 November 2024
Published 28 November 2024

Abstract

Simultaneous regulation of multiple properties in next-generation tokamaks like ITER and fusion pilot plant may require the integration of different plasma control algorithms. Such integration requires the conversion of individual controller commands into physical actuator requests while accounting for the coupling between different plasma properties. This work proposes a tokamak and scenario-agnostic actuator-sharing algorithm (ASA) to perform the above-mentioned command-request conversion and, hence, integrate multiple plasma controllers. The proposed algorithm implicitly solves a quadratic programming (QP) problem formulated to account for the saturation limits and the relation between the controller commands and physical actuator requests. Since the constraints arising in the QP program are linear, the proposed ASA is highly computationally efficient and can be implemented in the tokamak plasma control system in real time. Furthermore, the proposed algorithm is designed to handle real-time changes in the control objectives and actuators' availability. Nonlinear simulations carried out using the Control Oriented Transport SIMulator illustrate the effectiveness of the proposed algorithm in achieving multiple control objectives simultaneously.

Keywords: actuator-sharing algorithm, concurrent plasma control, plasma control integration, actuator allocation

1. Introduction

Next-generation tokamaks like ITER and the fusion pilot plant will require precise regulation of multiple plasma properties to achieve a predefined scenario with a high fusion gain. The regulation of individual plasma properties has been studied extensively in the existing tokamaks. Recent years have seen a shift towards simultaneous control of multiple plasma properties [1–5]. Most of these studies on simultaneous control have relied on designing a single complex controller that can regulate multiple plasma properties. The effectiveness of this approach towards simultaneous control has been demonstrated for the regulation of multiple plasma scalars [2] and the control of plasma profiles with some scalars [3–5]. As more plasma properties are incorporated into the control problem, designing a single controller to achieve all the objectives can be challenging. In recent years, an alternative approach that uses actuator-sharing algorithms (ASAs) has been proposed as an effective way to integrate multiple plasma controllers [6–12]. In this approach, the single complex controller is replaced by multiple simple controllers. These controllers prescribe virtual commands to satisfy distinct plasma control objectives.

[*] Author to whom any correspondence should be addressed.

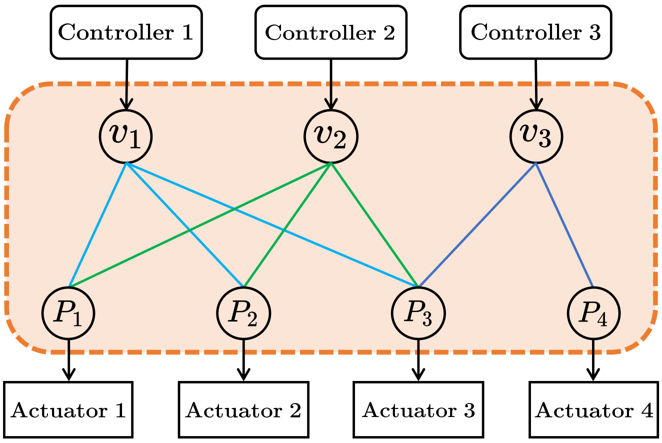

Figure 1. Illustration of the ASA block in the PCS.

These virtual controller commands vary based on the specific plasma control algorithm, and they may not correspond to actual actuator requests. An ASA then converts these controller commands into physical actuator requests. Figure 1 illustrates how an ASA can integrate different controllers in a tokamak plasma control system (PCS). Since the ASA decouples the actuators from the controller and accounts for the actuator constraints and saturation limits, this approach vastly reduces the complexity of the design of the individual controllers. Furthermore, since the performance of the different controllers and the actuators are coupled, sudden changes in the plasma control objectives or failure of the actuators can affect a plasma discharge in a tokamak. ASAs can be designed to handle such events in real time.

ASAs use a model to represent the relation between the controller commands and actuator requests. Depending on the model used, ASAs can be classified into static and dynamic algorithms. Static ASAs use algebraic equations to model the command-request relation in a system. Most existing ASAs developed for the tokamaks fall under this category [8, 9]. These ASAs generally rely on online optimization, in which the command-request model is imposed as a constraint, to convert the controller commands into physical actuator requests. For instance, the ASA in [8] uses mixed-integer quadratic programming (QP). Static ASAs based on QP and tailored to specific simultaneous plasma control problems have also been developed [2, 4, 9]. Since factors like actuator saturation limits can be easily incorporated into the optimization problem, static ASAs are easy to formulate and develop. However, depending on the algebraic command-request model, type of optimization problem, and number of actuators, static ASAs can be computationally expensive. The second category of ASAs, called dynamic ASAs, uses differential equations to model the command-request relation. These ASAs rely on integrating a differential equation to solve the allocation problem and, hence, can be more computationally inexpensive. Furthermore, dynamic ASAs consider the evolution of the actuator trajectories inherently. Hence, the actuator evolution can be designed to be optimal over a time horizon instead of a single time step. Furthermore, factors like lag in actuator response can be incorporated into the dynamic ASAs. Despite these advantages, dynamic ASAs have not been extensively explored since formulating them for a broad class of plasma control problems is challenging, often requiring specific assumptions on the plasma controllers. A generic dynamic ASA based on minimax optimization has been proposed in [12]. Another example of a dynamic ASA is the one proposed in [9] for the burn control problem.

This work proposes a computationally efficient static ASA that can integrate a wide variety of plasma control algorithms, allocate a broad class of actuators, and deal with real-time changes in the control objectives and actuators' availability. The proposed ASA relies on a linear/linearized command-request model to formulate the allocation problem as a QP problem. The linear nature of the model enables efficient computation of the allocation problem's solution using iterative evaluation of closed-form equations. Furthermore, the allocation problem is formulated to consider a generic class of control algorithms and tokamak actuators, thus making it tokamak and scenario agnostic. From this perspective, the proposed ASA can be considered as a generalization of the existing problem-specific solutions used in [1, 2, 4, 9]. Besides generalizing existing solutions, the proposed allocation problem is formulated to incorporate real-time changes in the available actuators or plasma control objectives. In any given scenario, the plasma control objectives can change as the plasma evolves. For instance, a minimum safety-factor $q_{\min}$ controller may be turned on only when the $q_{\min}$ value reaches a threshold. In addition, actuators could undergo failures or be repurposed for a different control objective (for example, an electron cyclotron (EC) used for total energy control could be repurposed for NTM suppression). The proposed ASA is designed to handle such turning on/off of plasma controllers and changes in available actuators while maintaining the QP problem framework. In other words, whenever there are changes in the available actuators or plasma control objectives, the ASA updates the cost function and the constraints in the QP problem without altering its underlying structure. Retaining the QP problem framework is beneficial since it obviates the need for comparatively complex optimization algorithms. The contributions of this paper include: (i) formulation of a QP problem based on an algebraic command-request model to solve the allocation problem, (ii) development of a methodology to incorporate changes in the control objectives and actuators' availability into the QP problem, (iii) synthesis of a computationally inexpensive algorithm that solves the allocation problem, (iv) illustration of the effectiveness of the proposed ASA using nonlinear simulations in the Control-Oriented Transport SIMulator (COTSIM) using three different test cases.

The sections in this paper are organized as follows. Section 2 mathematically defines the allocation problem and formulates the corresponding QP problem. The section also goes over the modifications necessary to the QP problem to handle changes in the control objectives or actuators' availability. The results of numerical simulations carried out to test the proposed ASA are presented in section 3. Section 4 concludes the work and lists potential extensions. Finally, a computationally inexpensive algorithm that solves the allocation problem formulated in section 2 is presented in appendix.

2. Allocation problem formulation and algorithm design

2.1. Problem formulation

In this subsection, the allocation problem is described, and the QP problem associated with the allocation problem is formulated. Consider a tokamak scenario in which multiple controllers try to regulate different plasma properties using l physical actuators. The assumption is that these controllers prescribe virtual commands based on the deviation of the corresponding plasma properties from their targets, and the relation between the virtual controller commands and the physical actuator requests is known. Let $\boldsymbol{v} \in \mathbb{R}^m$ be the vector of stacked virtual inputs prescribed by the controllers at each time step t_k, and $\boldsymbol{p} \in \mathbb{R}^l$ be the vector of physical actuator requests. The actuator request p_i corresponding to the ith actuator must lie within the range $\left[\underline{p}_i, \bar{p}_i\right]$, where $\underline{p}_i$, $\bar{p}_i$ are the lower and upper saturation limits of the actuator, respectively. Suppose that the relation between $\boldsymbol{v}$ and $\boldsymbol{p}$ is given by the equation

$$A(\boldsymbol{v})\boldsymbol{p} = b(\boldsymbol{v}), \tag{1}$$

where $A : \boldsymbol{v} \mapsto A(\boldsymbol{v}) \in \mathbb{R}^{m\times l}$ is a nonlinear matrix function and $b : \boldsymbol{v} \mapsto b(\boldsymbol{v}) \in \mathbb{R}^m$ is a nonlinear vector function.

Remark 1. Note that the above command-request relation is assumed to be linear in $\boldsymbol{p}$. In certain cases, the relation could be nonlinear. Linearization with respect to a reference vector $\boldsymbol{p}^r$ could be used to obtain a model of the form given in (1). At any given time step t_k, the actuator request $\boldsymbol{p}_{t_{k-1}}$ at the previous time step t_{k-1} could be used as the reference to linearize the command-request relation. Considering that the changes in actuator request values between consecutive time steps are small, the linearized model obtained using $\boldsymbol{p}_{t_{k-1}}$ should be sufficiently accurate to represent the command-request relation. But the model obtained using $\boldsymbol{p}_{t_{k-1}}$ is time-dependent. However, since static ASAs deal with instantaneous values, the time dependence of the model should not affect the allocation algorithm.

The problem of actuator allocation can be stated as follows: at each time step t_k, given the vector of virtual commands $\boldsymbol{v}(t_k)$, compute $\boldsymbol{p}(t_k)$ such that (1) holds and $p_i \in \left[\underline{p}_i, \bar{p}_i\right]$ for $i = 1, \ldots, l$. In other words, the problem involves selecting $\boldsymbol{p}$ such that (1) and the saturation limits are satisfied. The allocation problem may have multiple solutions (for example, multiple values of $\boldsymbol{p}$ could satisfy (1) and the saturation limits) or may not have a solution (when the controllers are trying to track distant targets and the actuators are constrained by the saturation limits). In the former case, one methodology to determine a unique $\boldsymbol{p}$ is to solve a QP optimization problem, which, at each time t, involves the minimization of the cost function f, i.e.

$$\underset{\boldsymbol{p}(t)}{\operatorname{argmin}} f(\boldsymbol{p}(t)) = \underset{\boldsymbol{p}(t)}{\operatorname{argmin}} \boldsymbol{p}^{\mathrm{T}}(t) Q \boldsymbol{p}(t) \tag{2}$$

subject to the constraints

$$A(\boldsymbol{v}(t))\boldsymbol{p}(t) = b(\boldsymbol{v}(t)), \tag{3}$$

$$\boldsymbol{p}(t) \in \left[\underline{p}_1, \bar{p}_1\right] \times \ldots \times \left[\underline{p}_l, \bar{p}_l\right]. \tag{4}$$

In the above optimization problem, $Q = \operatorname{diag}(q_1, \ldots, q_l)$ is a diagonal matrix with weights $q_1, \ldots, q_l$. The reformulated problem involves minimizing the cost function f, which measures the total actuator effort while satisfying the constraints mentioned in the original allocation problem. Note that the allocation problem does not have a solution when no $\boldsymbol{p}$ within the saturation limits can satisfy (1). If such a situation arises at a certain time step, slack variables can be introduced into the optimization problem, as discussed in appendix, to solve it. The resulting $\boldsymbol{p}$ violates (1) but satisfies the saturation limits. If the allocation problem does not have a solution through extended periods of the plasma discharge, then redefining the controller targets might be the only possible solution.

2.2. Incorporating controller and actuator changes into the QP problem

Real-time changes in the control objectives and actuators' availability require adaptation of the above-presented QP optimization problem. This subsection introduces the control objective adjustment matrix C and actuator status matrix S, which can be used to redefine the original QP problem whenever controller and actuator changes occur during a plasma discharge. The approach presented below relies on first defining a generic QP problem that incorporates all the controllers and actuators in any given plasma scenario. The matrices C and S are used to eliminate terms in the generic QP problem corresponding to the inactive controllers and actuators, respectively.

In a given tokamak scenario, let r represent the maximal number of plasma controllers that can be activated during a plasma discharge. Suppose the ith controller prescribes m_i virtual commands represented by the vector $\boldsymbol{v}_{mi}$ for $i = 1, \ldots, r$. Note that $m_1 + \ldots + m_r = m \geqslant r$ by definition. Furthermore, suppose that the command-request relation of each controller is independent of the other controllers, i.e. the command-request relation given in (1) can be written in the from

$$A(\boldsymbol{v})\boldsymbol{p} = \begin{bmatrix} A_{m_1}(\boldsymbol{v}_{m_1}) \\ \vdots \\ A_{m_r}(\boldsymbol{v}_{m_r}) \end{bmatrix} \boldsymbol{p} = \begin{Bmatrix} b_{m_1}(\boldsymbol{v}_{m_1}) \\ \vdots \\ b_{m_r}(\boldsymbol{v}_{m_r}) \end{Bmatrix} = b(\boldsymbol{v}). \tag{5}$$

To define the control objective adjustment matrix C corresponding to the time step t_k, define the matrix $\hat{C}$ as

$$\hat{C} = \begin{bmatrix} c_1 I_{m_1} & & \\ & \ddots & \\ & & c_r I_{m_r} \end{bmatrix} \in \mathbb{R}^{m\times m}, \tag{6}$$

where, for $i = 1,\ldots,r$, $c_i \in \mathbb{R}$ is equal to 1 is the ith controller is active and 0 otherwise. The term I_{m_i} is an identity matrix of dimension m_i. The matrix C can now be constructed by eliminating the rows of $\hat{C}$ with only zeros. At each time t_k, C is $\hat{m}\times m$ dimensional matrix, where $\hat{m} = \sum_{i=0}^{r} c_i m_i \leqslant m$. This matrix, as shown below, will be used to redefine the QP problem whenever there is a change in the control objectives.

To account for changes in the actuators' availability, the actuator status matrix S is defined using an approach similar to the one presented above. Define the diagonal matrix $\hat{S}$ as

$$\hat{S} = \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_l \end{bmatrix} \in \mathbb{R}^{l\times l}, \tag{7}$$

where l represents the maximal number of physical actuators available for plasma control in a given tokamak scenario, the term $s_j, j = 1,\ldots,l$, is a positive constant and takes a value of 1 when the actuator j is available and 0 otherwise. The matrix S is built by combining the non-zero columns of $\hat{S}$ without changing the column order. Note that S is a $l \times \hat{l}$ dimensional matrix, where $\hat{l} = \sum_{j=0}^{l} s_j \leqslant l$.

The matrices C and S defined above can now be used to reformulate the QP optimization problem whenever there is a change in the controller objectives or actuators' availability. In particular, the terms A, Q, b, and $\boldsymbol{p}$ arising in the QP problem defined in subsection 2.1 are replaced by $\check{A}$, $\check{Q}$, $\check{b}$, and $\check{\boldsymbol{p}}$, which are defined as

$$\check{A} = CAS, \qquad \check{Q} = S^{\mathrm{T}}QS, \qquad \check{b} = Cb, \qquad \check{\boldsymbol{p}} = S^{\mathrm{T}}\boldsymbol{p}. \tag{8}$$

Intuitively, the introduction of C and S, as shown in the above equation, eliminates terms relevant to inactive controllers and actuators from the QP problem. The definition of these matrices provides a computationally inexpensive and straightforward method to handle controller and actuator changes in real time. Note that the C and S matrices are updated and the QP problem is redefined only when there is a change in the controller or actuator status.

2.3. Actuator sharing algorithm

The ASA can now be summarized as follows. At any given time-step t_k,

(i) compute the virtual commands using the plasma control algorithms and define the vector $\boldsymbol{v}$ of virtual commands,
(ii) compute the terms $A(\boldsymbol{v}(t_k))$ and $b(\boldsymbol{v}(t_k))$ arising in the command-request model,
(iii) define the control objective adjustment matrix C and actuator status matrix S based on the control objectives and actuators' availability at time t_k,
(iv) compute the terms $\check{A}$, $\check{Q}$, $\check{b}$, and $\check{\boldsymbol{p}}$ using (8),
(v) solve the QP problem defined by $\check{A}$, $\check{Q}$, $\check{b}$, and $\check{\boldsymbol{p}}$.

The most critical step in the above algorithm is solving the QP problem. A wide-array of algorithms, including active set and interior point methods, exist to solve QP optimization problems. Some of these techniques have already been implemented in the PCSs of existing tokamaks like EAST for real-time optimization purposes [13]. Appendix summarizes an alternative algorithm that iteratively uses closed-form equations to arrive at the solution of the allocation problem. The maximum number of iterations in the algorithm presented in appendix is restricted by the number of physical actuators. Since the iteration primarily involves evaluation of a closed-form equation, the algorithm presented in appendix is computationally inexpensive.

3. Numerical testing of the ASA

This section summarizes the results of numerical simulations that were carried out to test the effectiveness of the proposed static ASA in a DIII-D tokamak scenario using the COTSIM [14]. In particular, three test cases were studied: (i) simultaneous control of the safety factor gradient at a rational surface and total thermal energy of the plasma [15], (ii) simultaneous control of the minimum safety factor and total thermal energy [16], (iii) simultaneous control of the minimum safety factor and total thermal energy with an arbitrary actuator failure.

3.1. Case 1: Safety factor gradient and total energy control

In certain scenarios, neoclassical tearing modes can appear at regions where the safety factor q takes rational values. In such scenarios, regulating the slope of the safety factor profile at the rational surfaces may prevent or delay the onset of such instabilities [17]. This test case considers the problem of simultaneous regulation of the safety factor gradient and total thermal energy of the plasma, which is another plasma property that is critical for MHD stability.

The safety factor gradient controller is built using the model developed in [15]. In this work, the gradient at the (3,2) rational safety factor surface is approximated by the difference between the safety factor values at two points around the rational surface. Refer to figure 2 for a graphical illustration of this approach. Given a rational safety factor surface, gradient control can be achieved by controlling $q_{\mathrm{D}} = q_{\mathrm{R}} - q_{\mathrm{L}}$, where q_{R} and q_{L} are the safety factor values at the right and left control points ρ_{R} and ρ_{L}, respectively. Suppose the objective is to track a target $\bar{q}_{\mathrm{D}}$, the evolution of the error $\tilde{q}_{\mathrm{D}} = q_{\mathrm{D}} - \bar{q}_{\mathrm{D}}$ is given by [15]

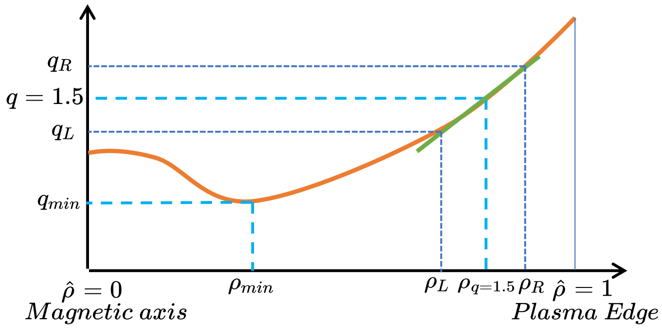

Figure 2. Illustration of the minimum safety factor and safety factor gradient approximation.

$$\dot{\tilde{q}}_{\mathrm{D}} = \underbrace{\hat{c}_{\mathrm{R}}\left(t, P_{\mathrm{tot,fb}}, q_{\mathrm{R}}\right) - \hat{c}_{\mathrm{L}}\left(t, P_{\mathrm{tot,fb}}, q_{\mathrm{L}}\right) - \dot{\bar{q}}_{\mathrm{D}}}_{\check{c}_q} + \boldsymbol{g}_{\mathrm{D}}\left(t, P_{\mathrm{tot,fb}}\right)^{\mathrm{T}} \boldsymbol{u}_{\mathrm{fb}}\left(t\right). \quad (9)$$

In the above equation, $\hat{c}_i$, $(i = R, L)$, is a nonlinear function of the total feedback power $P_{\mathrm{tot,fb}}$ and the safety factor value q_i. The term u_{fb} is the vector of feedback auxiliary current drive powers available for feedback control, and $\boldsymbol{g}_{\mathrm{D}}$ is the vector that accounts for the spatial deposition of the auxiliary drives. The steps involved in the derivation of the above model can be found in [15]. In simulations, two NBIs and two ECs were assumed to be available for control. Then, the vector u_{fb} can be written as $u_{\mathrm{fb}} = [P_{\mathrm{NBI},1}, P_{\mathrm{NBI},2}, P_{\mathrm{EC},1}, P_{\mathrm{EC},2}]^{\mathrm{T}}$. In addition, the total feedback power can be expressed as $P_{\mathrm{tot,fb}} = P_{\mathrm{NBI},1} + P_{\mathrm{NBI},2} + P_{\mathrm{EC},1} + P_{\mathrm{EC},2}$.

The evolution of the error between the total plasma energy W and a given target $\bar{W}$ is governed by

$$\dot{\tilde{W}} = \underbrace{-\frac{W}{\tau_E\left(t, P_{\mathrm{tot,fb}}\right)} + P_{\mathrm{tot,ff}} - \dot{\bar{W}}}_{\check{c}_W} + P_{\mathrm{tot,fb}}, \quad (10)$$

where τ_E is the energy confinement time and $P_{\mathrm{tot,ff}}$ is the feedforward total power. The derivation of the above-given 0D model can be found in [3].

It is clear from the above models that the two plasma properties are coupled. To design the controllers and integrate them using ASA, virtual inputs v_1 and v_2 are defined as $v_1 = \boldsymbol{g}_{\mathrm{D}}(t, P_{\mathrm{tot,fb}})^{\mathrm{T}} \boldsymbol{u}_{\mathrm{fb}}(t)$ and $v_2 = P_{\mathrm{tot,fb}}$. With this definition, the command-request model of the form given in (1) can be written as

$$\begin{bmatrix} g_{\mathrm{D},1}\left(t, v_2\right) & g_{\mathrm{D},2}\left(t, v_2\right) & g_{\mathrm{D},3}\left(t, v_2\right) & g_{\mathrm{D},4}\left(t, v_2\right) \\ 1 & 1 & 1 & 1 \end{bmatrix} \times \begin{Bmatrix} P_{\mathrm{NBI},1}\left(t\right) \\ P_{\mathrm{NBI},2}\left(t\right) \\ P_{\mathrm{EC},1}\left(t\right) \\ P_{\mathrm{EC},2}\left(t\right) \end{Bmatrix} = \begin{Bmatrix} v_1\left(t\right) \\ v_2\left(t\right) \end{Bmatrix}. \quad (11)$$

With virtual inputs, controllers can be designed independently of the auxiliary drive constraints, making control synthesis much easier. As mentioned in subsection 2.1, these constraints are handled by the ASA.

Feedback linearization-based controllers are used for the stabilization of $\tilde{q}_{\mathrm{D}}$ and $\tilde{W}$ [15]. These controllers negate the nonlinearities in the models and introduce stabilizing linear terms. One choice of feedback linearizing virtual inputs is

$$\begin{aligned} v_1 &= -\check{c}_q - k_{q,p}\tilde{q}_{\mathrm{D}} - k_{q,I}\int_0^t \tilde{q}_{\mathrm{D}}, \\ v_2 &= -\check{c}_W - k_{W,p}\tilde{W} - k_{W,I}\int_{t_0}^t \tilde{W}\mathrm{d}t, \end{aligned} \quad (12)$$

where $k_{q,p}, k_{q,i}, k_{W,p}, k_{W,i} > 0$ are controller gains. Substituting the above virtual inputs into (9) and (10) results in linear equations, which can be proved to be stable using Lyapunov analysis [18]. The above virtual commands can then be converted into physical actuator requests using the ASA.

The above controllers and the static ASA presented in subsection 2.3 were tested in COTSIM for a DIII-D scenario. The DIII-D configuration information corresponding to shot number 147 634 was used [19]. The saturation limits of the actuators were selected as 12 MW, 6 MW, 3.5 MW, and 3.5 MW, respectively. While these limits differ from the current saturation limits of the DIII-D auxiliary drives, they were chosen to test the controllers' ability to track large errors. The simulation results are shown in figure 3. The gray background in the figure corresponds to the time period when the controllers are active. The virtual commands prescribed by the controllers are presented in figure 3(a). The ASA converted these virtual commands into physical actuator requests presented in figure 3(b). The evolution of the plasma states, q_{D} and W, when these requests were used are shown in figure 3(c). The closed-loop trajectories demonstrate that the controllers and the ASA can achieve the desired plasma control objectives. The steady-state error in the trajectories of q_{D} and W indicates that the integral gains $k_{q,i}$ and $k_{W,i}$ could be tuned further.

3.2. Case 2: Minimum safety factor and total energy control

Another critical property that is linked to the MHD stability of the plasma is the minimum safety factor [16]. Figure 2 also gives a graphical illustration of the minimum safety factor $q_{\min}$ at $\rho_{\min}$. This case considers the control of the minimum safety factor and total plasma energy simultaneously. Since the gradient θ of the poloidal flux is related to the safety factor, minimum-safety-factor regulation can be achieved by regulating the value of θ profile at $\rho_{\min}$ [16]. The control model, developed in [16], is given by

$$\dot{\tilde{\theta}}_{q_{\min}}\left(t\right) = \hat{c}\left(t, P_{\mathrm{tot,fb}}, \theta_{q_{\min}}\right) - \dot{\bar{\theta}}_{q_{\min}}\left(t\right) + \boldsymbol{h}\left(t, P_{\mathrm{tot,fb}}\right)^{\mathrm{T}} \boldsymbol{u}_{\mathrm{fb}}\left(t\right), \quad (13)$$

where $\theta_{q_{\min}}$ is polidal flux gradient at $\rho_{\min}$, $\bar{\theta}_{q_{\min}}$ is the target, $\tilde{\theta}_{q_{\min}} = \theta_{q_{\min}} - \bar{\theta}_{q_{\min}}$, $\hat{c}$ is a nonlinear function of $P_{\mathrm{tot,fb}}, \theta_{q_{\min}}$. In

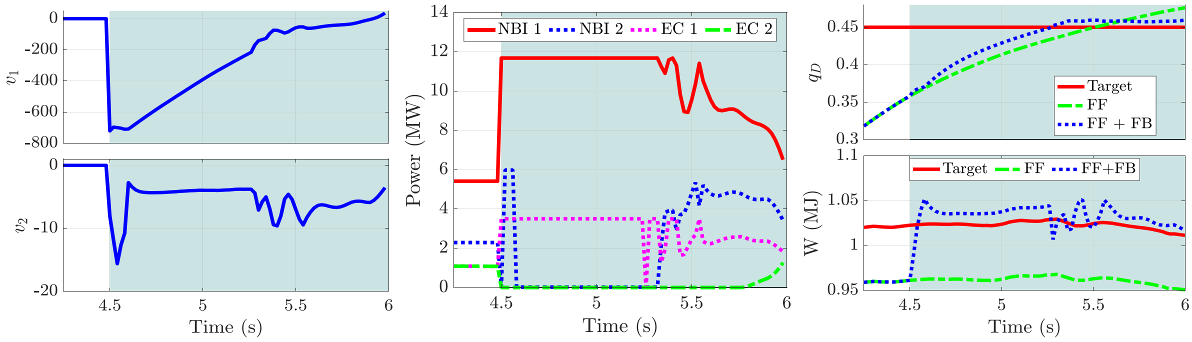

Figure 3. Case 1: (a) - virtual inputs (left), (b) - actuator requests (center), (c) - closed-loop trajectories (right).

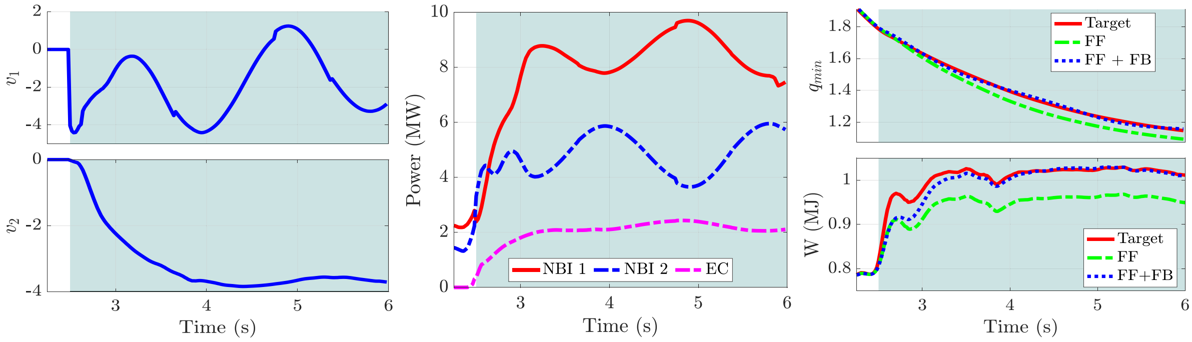

Figure 4. Case 2: (a) - virtual inputs (left), (b) - actuator requests (center), (c) - closed-loop trajectories (right).

the simulations, 2 NBI clusters and 1 EC cluster were assumed to be available for control. Thus, $\boldsymbol{u}_{\text{fb}}$ takes the form $\boldsymbol{u}_{\text{fb}} = [P_{\text{NBI},1}, P_{\text{NBI},2}, P_{\text{EC}}]^{\text{T}}$ and $\boldsymbol{h}(t, P_{\text{tot,fb}})$ is the vector that accounts for the current drive deposition profiles. The virtual input for this case can be defined similarly to how it was defined for the safety-factor-gradient control discussed in the previous case.

Feedback linearzing controllers were designed to drive $\tilde{\theta}_{q_{\text{min}}}$ to 0. Simulations were carried out using COTSIM to test the controllers. The DIII-D configuration utilized for the simulations discussed in subsection 3.1 was used. To test the ability of the controllers to stabilize large errors, the saturation limits of the actuators $P_{\text{NBI},1}, P_{\text{NBI},2}, P_{\text{EC}}$ were set as 13 MW, 13 MW, 4.5 MW, respectively. These values differ from the actual DIII-D auxiliary drive limits and were selected specifically for simulation purposes. Figure 4(a) shows the virtual inputs generated by the controllers, and figure 4(b) presents the actuator requests computed by the ASA. When these actuator requests were used to close the control loop, the plasma properties evolved as shown in figure 4(c). The gray background in the subfigures depicts the period of active feedback control. It is clear that the controllers and the ASA are able to achieve the desired control objectives.

3.3. Case 3: Minimum safety factor and total energy control with actuator failure

This case considers the simultaneous regulation of the minimum safety factor and total energy and tests the ability of the ASA to handle sudden actuator failures. The simulation configurations used for carrying out the simulations presented in subsection 3.2 were also used here. At 3.5 s, the EC cluster was assumed to fail. The ASA will continue assigning physical actuator requests to the EC cluster if this information is not considered in real time. However, once the failure is detected, if the actuator status matrix S is redefined and the QP problem is updated as discussed in subsection 2.2, the ASA should increase the powers of the other actuators. This expected behavior is observed in the simulation results presented in figure 5. The light gray background in the figure corresponds to the period when all actuators are available. On the other hand, the dark gray background corresponds to the period when the EC is inactive. The subfigures (a) and (b) present the virtual inputs and the physical actuator requests, respectively. The subfigure (c) shows the closed-loop trajectories. As evident from the physical actuator requests plot, the ASA increases the NBI powers as soon as the EC fails. As a result, q_{min} continues to track the target. On the other hand, it can be seen that the actuator failure causes the total energy to deviate from

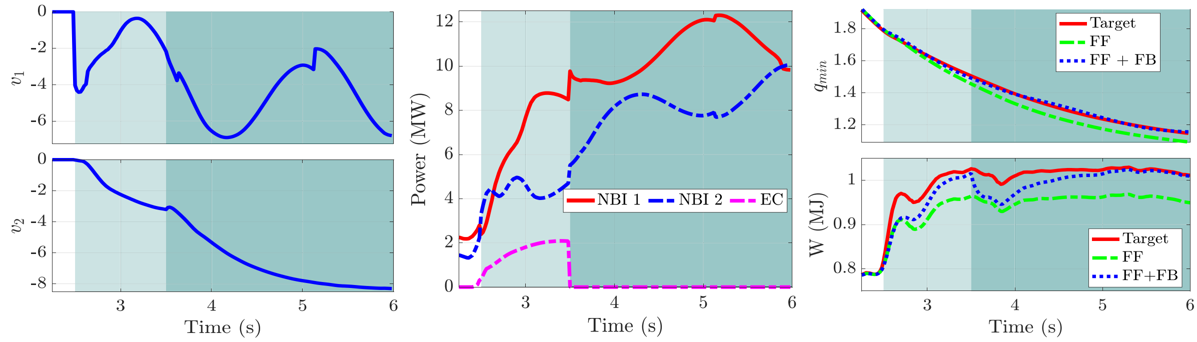

Figure 5. Case 3: (a) - virtual inputs (left), (b) - actuator requests (center), (c) - closed-loop trajectories (right).

the target. However, the power corrections made by the ASA cause the total energy to converge again to the target.

4. Conclusion

A novel static ASA has been developed to convert virtual commands from plasma controllers into physical actuator requests that satisfy the saturation limits. The allocation problem and the associated QP problem are formulated in such a way that the ASA is computationally efficient and can handle real-time changes in the control objectives and actuators' availability. An iterative algorithm that solves the allocation problems is also presented. The effectiveness of the ASA has been demonstrated using nonlinear simulation in COTSIM using three different test cases. Future extensions of this work could focus on incorporating actuator dynamics into the ASA and experimental validation of the simulation results.

Data availability statement

The data cannot be made publicly available upon publication because they are owned by a third party and the terms of use prevent public distribution. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

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 Awards DE-SC0010661, DE-SC0021385 and DE-FC02-04ER54698.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Appendix. Iterative algorithm to solve the allocation problem

This section presents an iterative algorithm that solves the allocation problem formulated in section 2.1. The development of the algorithm is split into two steps. First, the minimization of the cost function f defined in (2) subject to the command-request model (3) is considered. In this step, the saturation limits of the actuators are ignored. A closed-form solution of this simplified problem is derived. In the second step, this closed-form expression is implemented iteratively to arrive at a vector of physical actuator requests that satisfy the saturation limits.

A.1. Actuator allocation without saturation limits

Consider the cost function given in (2) and the constraint imposed by the command-request model (3). Define the function g as $g(\boldsymbol{p}) = A\boldsymbol{p} - b$ and the Lagrangian as

$$\mathcal{L}(\boldsymbol{p}, \boldsymbol{\lambda}) = f(\boldsymbol{p}) - \boldsymbol{\lambda}^{\mathrm{T}} g(\boldsymbol{p}), \tag{A.1}$$

where $\boldsymbol{\lambda}$ represents the vector of Lagrange multipliers. The Lagrange multiplier theorem states that if a local minimum $\boldsymbol{p}^*$ of the cost function f exists, and $\nabla_{\boldsymbol{p}} g_1, \ldots, \nabla_{\boldsymbol{p}} g_m$ are linearly independent, then there exists $\boldsymbol{\lambda}^*$ such that

$$\nabla_{\boldsymbol{p}} f(\boldsymbol{p}^*) - \boldsymbol{\nabla} g(\boldsymbol{p}^*)^{\mathrm{T}} \boldsymbol{\lambda}^* = 0, \tag{A.2}$$

$$g(\boldsymbol{p}^*) = 0. \tag{A.3}$$

The notation $\nabla_{\boldsymbol{p}}$ represents the gradient with respect to $\boldsymbol{p}$, and

$$\boldsymbol{\nabla} g(\boldsymbol{p}^*) = \left[\nabla_{\boldsymbol{p}} g_1(\boldsymbol{p}^*) \quad \ldots \quad \nabla_{\boldsymbol{p}} g_m(\boldsymbol{p}^*)\right]^{\mathrm{T}}. \tag{A.4}$$

From (2) and the definition of g, it is clear that $\nabla_{\boldsymbol{p}} f(\boldsymbol{p}^*) = 2Q\boldsymbol{p}^*$ and $\nabla g(\boldsymbol{p}^*) = A$. Substituting these expressions into (A.2–A.4) and rearranging the terms results in a closed-form equation for the optimal physical actuator request vector $\boldsymbol{p}^*$ of the form

$$\boldsymbol{p}^* = Q^{-1}A^{\mathrm{T}}\left(AQ^{-1}A^{\mathrm{T}}\right)^{-1}b. \tag{A.5}$$

A.2. Incorporating saturation limits using iterative approach

The closed-form equation (A.5) derived above can be used iteratively to enforce the saturation limits constraints of the actuators. However, before presenting the iterative algorithm, it is important to note that the allocation problem or the associated QP problem, defined by (2–4), may not have a solution, i.e. no combination of actuator values can satisfy the controller commands. To handle such cases, slack variables $\boldsymbol{s} = [s_1, \ldots, s_m]^{\mathrm{T}}$ are introduced into the QP problem. In other words, the cost function is minimized with respect to $\boldsymbol{p}$ and $\boldsymbol{s}$, where $\boldsymbol{s}$ is allowed to arbitrary real-number values. The modified QP problem can be stated as, at each time t, given $\boldsymbol{v}(t)$,

$$\operatorname*{argmin}_{\boldsymbol{p}_a(t)} f(\boldsymbol{p}_a(t)) = \operatorname*{argmin}_{\boldsymbol{p}_a(t)} \boldsymbol{p}_a^{\mathrm{T}}(t)\,\hat{Q}\boldsymbol{p}_a(t) \tag{A.6}$$

subject to the constraints

$$\hat{A}(\boldsymbol{v}(t))\boldsymbol{p}_a(t) = b(\boldsymbol{v}(t)), \tag{A.7}$$

$$\boldsymbol{p}_a(t) \in \left[\underline{p}_1, \bar{p}_1\right] \times \ldots \times \left[\underline{p}_l, \bar{p}_l\right] \times (-\infty, \infty) \times \ldots \times (-\infty, \infty), \tag{A.8}$$

where $\hat{Q} = \mathrm{diag}(q_1, \ldots, q_l, q_{s_1}, \ldots, q_{s_m})$ with $q_{s_j} >> q_i > 0$ for $i = 1, \ldots, l$ and $j = 1, \ldots, m$, $\hat{A} = [A, I]$, $I \in \mathbb{R}^{m\times m}$ is the identity matrix, and $\boldsymbol{p}_a = [\boldsymbol{p}^{\mathrm{T}}\boldsymbol{s}^{\mathrm{T}}]^{\mathrm{T}}$. The closed-form equivalent of (A.5) in this case can be written as

$$\boldsymbol{p}_a^* = \hat{Q}^{-1}\hat{A}^{\mathrm{T}}\left(\hat{A}\hat{Q}^{-1}\hat{A}^{\mathrm{T}}\right)^{-1}b, \tag{A.9}$$

Remark 2. Note that the slack variables are included in the modified command-request model (A.7) and are heavily penalized in the cost function. Thus, whenever the original QP problem without the slack variables has a solution, solving the modified QP problem presented above gives a $\boldsymbol{p}$ close to the original solution. When the original problem does not have a solution, solving the modified QP problem results in large values of the slack variables, which can be considered a measure of the command-request model violation.

Remark 3. The weights $q_{s_1}, \ldots, q_{s_m}$ of the slack variables in the matrix $\hat{Q}$ correspond to the m plasma controllers that prescribe the virtual inputs. Depending on the choice of $q_{s_1}, \ldots, q_{s_m}$, specific control objectives can be prioritized over others whenever complete actuator saturation occurs.

The vector $\boldsymbol{p}_a^*$ obtained by (A.9) may not satisfy the saturation limits. The following iterative approach gives a $\boldsymbol{p}_a^*$ that satisfies the saturation limits.

(i) Compute $\boldsymbol{p}_a^*$ using (A.9).
(ii) If any values in $\boldsymbol{p}_a^*$ violate the saturation limits, their respective values are set to the corresponding saturation limit.
(iii) The optimization problem is redefined to only include those components of $\boldsymbol{p}_a^*$ that did not saturate.
(iv) Return to Step (i). This process is continued iteratively until (A.7) and (A.8) are satisfied.

Remark 4. Even if the original QP problem, formulated in subsection 2.1, has a solution, the above iterative algorithm may give a solution that is suboptimal with respect to the QP problem, since the inequality constraints imposed by the saturation limits are not considered while deriving (A.9) [20]. However, it is essential to note that the above algorithm gives a solution that satisfies both (A.7) and (A.8). In other words, it solves the allocation problem even if it only gives a suboptimal QP solution.

ORCID iDs

Sai Tej Paruchuri https://orcid.org/0000-0003-2372-3888
Vincent Graber https://orcid.org/0000-0003-2058-2258
Andres Pajares https://orcid.org/0000-0001-9251-9675

References

[1] Pajares A and Schuster E 2019 Integrated robust control of individual scalar variables in tokamaks 2019 *IEEE 58th Conf. on Decision and Control (CDC)* pp 3233–8
[2] Pajares A, Schuster E, Thome K E, Welander A S, Barr J L, Eidietis N W and Humphreys D A 2022 *Nucl. Fusion* **62** 036018
[3] Pajares A and Schuster E 2021 *Nucl. Fusion* **61** 036006
[4] Pajares A and Schuster E 2020 *IEEE Trans. Plasma Sci.* **48** 1606–12
[5] Al Khawaldeh H, Leard B, Paruchuri S T, Rafiq T and Schuster E 2023 *Fusion Eng. Des.* **192** 113795
[6] Kudlacek O, Treutterer W, Janky F, Sieglin B and Maraschek M 2019 *Fusion Eng. Des.* **146** 1145–8
[7] Vu N T, Blanken T, Felici F, Galperti C, Kong M, Maljaars E and Sauter O 2019 *Fusion Eng. Des.* **147** 111260
[8] Maljaars E and Felici F 2017 *Fusion Eng. Des.* **122** 94–112
[9] Graber V and Schuster E 2022 *Nucl. Fusion* **62** 026016
[10] Treutterer W, Humphreys D, Raupp G, Schuster E, Snipes J, De Tommasi G, Walker M and Winter A 2014 *Fusion Eng. Des.* **89** 512–7
[11] di Grazia L E, Frattolillo D, De Tommasi G and Mattei M 2023 *J. Optim. Theory Appl.* **198** 958–87
[12] Paruchuri S T, Graber V, Khawaldeh H A and Schuster E 2024 *IEEE Trans. on Plasma Science* 1–7
[13] Wang Z, Wang H, Schuster E, Luo Z, Huang Y, Yuan Q, Xiao B, Humphreys D and Paruchuri S T 2023 Implementation and initial testing of a model predictive controller for safety factor profile and energy regulation in the east tokamak 2023 *American Control Conf. (ACC)* pp 3276–81
[14] Pajares A 2019 Integrated control in tokamaks using nonlinear robust techniques and actuator sharing strategies *PhD Thesis* Lehigh University

[15] Paruchuri S T and Schuster E 2023 *Fusion Eng. Des.* **194** 113914

[16] Pajares A and Schuster E 2021 Robust nonlinear control of the minimum safety factor in tokamaks 2021 *IEEE Conf. on Control Technology and Applications (CCTA)* pp 753–8

[17] Turco F, Luce T, Solomon W, Jackson G, Navratil G and Hanson J 2018 *Nucl. Fusion* **58** 106043

[18] Khalil H K 2002 *Nonlinear Systems* 3rd edn (Pearson)

[19] Holcomb C *et al* 2014 *Nucl. Fusion* **54** 093009

[20] Rao S S 2019 *Engineering Optimization: Theory and Practice* 5th edn (Wiley)