ACTUATOR-SHARING ALGORITHM FOR SIMULTANEOUS REGULATION OF MULTIPLE PLASMA PROPERTIES WITH COUPLED DYNAMICS

S. T. PARUCHURI, V. GRABER, E. SCHUSTER
Lehigh University
Bethlehem, USA
Email: saitejp@lehigh.edu

ANDRES PAJARES
General Atomics
San Diego, USA

Abstract

The plasma control system (PCS) of next-generation tokamaks like ITER and the Fusion Pilot Plant (FPP) must simultaneously regulate multiple plasma properties. Control solutions to drive individual plasma properties to their targets have been and are being developed over the years. The input commands of each control algorithm must be translated into physical actuator requests such as the auxiliary drive powers and deposition locations before implementation. Any such conversion must account for the complex coupling between the different plasma properties. The possibility of having an individual physical input affecting multiple plasma properties makes the input conversion even more challenging. This work proposes a tokamak and scenario-agnostic actuator-sharing algorithm that can convert individual controller commands into physical actuator requests while accounting for the constraints introduced by coupled plasma properties. The proposed algorithm does not rely on real-time optimization, the most commonly used method for actuator sharing and allocation, which can be computationally expensive. Furthermore, the algorithm is designed to handle real-time changes in controller objectives and actuator availability. The effectiveness of the proposed algorithm has been demonstrated using nonlinear DIII-D tokamak simulations in the Control Oriented Transport SIMulator (COTSIM).

1. INTRODUCTION

Advanced tokamak scenarios in next-generation tokamaks like ITER and Fusion Pilot Plant (FPP) will require simultaneous regulation of multiple plasma properties by prescribing physical actuator requests, such as the auxiliary drive powers and deposition locations, to achieve pre-determined target values. Synthesizing a single control algorithm to regulate all the plasma properties can be challenging due to the high dimensionality of the control problem and the complex coupling between the different plasma properties. An alternative approach is to design multiple control algorithms, each regulating one or more plasma properties, and integrate them by using an actuator-sharing algorithm (ASA). In this approach, each control algorithm prescribes a virtual command to achieve its control objective. The ASA combines these virtual commands and converts them into physical actuator requests. Figure 1 gives an illustration of the role of the ASA block in the PCS. This two-step approach considerably simplifies the control synthesis since the coupling between the different properties and actuator constraints is not incorporated into the individual control problem. Instead, these aspects are handled by the ASA. Furthermore, since the tokamak-specific constraints are incorporated into the ASA, certain control algorithms tested on present-day tokamaks could be implemented on next-generation tokamak with an ASA. In addition, ASAs could be designed to handle off-normal events and actuator failures, which is essential for next-generation tokamaks. Thus, next-generation tokamaks could benefit from the development of robust ASAs.

Emphasis on actuator-sharing algorithms and management strategies has increased in recent years [1, 2, 3]. A review of existing solutions reveals two distinct classes of ASAs - static and dynamic. Static ASAs treat the relation between virtual commands and physical actuator requests as algebraic equations [4, 5]. Most existing static ASAs generally rely on real-time optimization to compute the physical actuator requests. For instance, the algorithm given in [4] uses mixed-integer quadratic programming. Some existing works on simultaneous control of plasma scalars and profiles have also implemented a static actuator-sharing algorithm, sometimes tailored to a specific control problem [5, 6, 7]. For example, linear quadratic programming is used in [5] to convert virtual commands into physical requests for the plasma burn control problem. Similarly, quadratic programming is also used to integrate two specific plasma controllers in [6, 7]. Even though the optimization algorithms in these articles are tailored to particular plasma control problems, they can be generalized and treated as static ASAs.

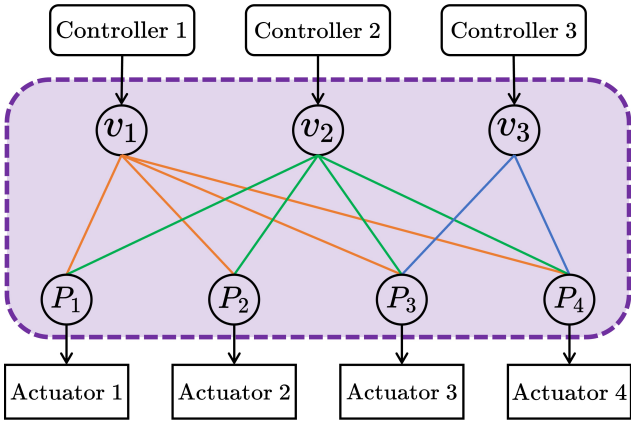

FIG. 1. *Illustration of the ASA block in the PCS.*

Unlike static ASAs, dynamic algorithms use differential equations to prescribe physical actuator requests. An example of dynamic ASA available in the plasma control literature is [5].

Both static and dynamic ASAs offer unique sets of advantages. Static ASAs are much simpler to develop since they rely on algebraic constraints. Furthermore, static ASAs can be adapted to a broad class of plasma control problems and designed to handle events like actuator failure. However, depending on the optimization algorithm used and the number of available physical actuators, static ASAs can be computationally expensive. On the other hand, dynamic ASAs are more computationally efficient, as their implementation relies on solving differential equations at every time step. Further, dynamic ASAs can be designed to be optimal over the entire discharge rather than a single-time instant. However, dynamic ASAs are much more challenging to design and implement for a broad class of plasma control problems. The development of an ASA combining the flexibility of static ASAs with the computational efficiency of dynamic ASAs could prove beneficial for next-generation tokamaks.

In this work, an improved static actuator-sharing algorithm (ASA) has been developed to prescribe physical actuator requests based on individual input commands generated by multiple controllers designed to regulate different plasma properties. The proposed ASA is designed to (i) handle a wide variety of plasma controllers and tokamaks, (ii) not rely on solving an optimization problem in real-time, (iii) account for the coupling between different plasma parameters and actuators, and (iv) deal with control objective & actuator availability changes. The ability to handle multiple combinations of plasma controllers while being tokamak-independent allows the algorithm to be implementable in different scenarios across distinct machines. Thus, a new ASA does not need to be designed each time the scenario objective is changed. Also, the proposed ASA relies on evaluating at each time step a closed-form equation that represents the solution of an optimization problem. In the optimization problem, the cost function measures the total physical actuators' effort, and the constraints are the equations that account for the coupling between the different virtual and physical inputs. During the optimization problem formulation, it is assumed that the coupling between virtual and physical inputs can be represented in the form of linear or linearizable algebraic equations. To ensure the resulting physical input values computed by the ASA satisfy the saturation limits, the algorithm iteratively evaluates a modified version of the closed-form equation until a feasible solution is achieved. The resulting solution may be suboptimal; however, the algorithm is computationally efficient because the number of iterations is bounded by the number of physical actuators. In addition, the ASA accounts for the coupling of plasma properties (or the virtual inputs) and physical actuators. For instance, a heating and current drive simultaneously affects the evolution of the safety factor profile and the electron temperature profile. The algorithm accounts for such coupling by treating the coupling relation as algebraic constraints in the optimization problem. Besides accounting for coupling, the ASA is designed to accommodate changes in the number of virtual commands and physical actuators. Such a design allows the ASA to be implementable in a scenario where the control objectives and actuator availability can change in real time.

The sections in this paper are organized as follows. Section 2 explains the static actuator allocation problem and formulates the corresponding optimization problem. The closed-form solution to the optimization problem when the actuator saturation limits are ignored is derived. The methodology and the algorithm to incorporate saturation limits is discussed. Section 3 goes over the modifications necessary to the algorithm proposed in Section 2 to handle changes in control objectives or actuator availability. The simulations results of three different test cases are presented in Section 4. Finally, Section 5 summarizes the contributions of this work.

2. ACTUATOR ALLOCATION PROBLEM FORMULATION AND ALGORITHM DESIGN

2.1. Problem Formulation

Consider a tokamak scenario in which there are multiple controllers that are trying to regulate n different plasma states. Suppose that the control algorithms prescribe a vector $\boldsymbol{v} \in \mathbb{R}^m$ of virtual commands to regulate the n plasma states at each time step t_k. Furthermore, suppose that there are l physical actuators, whose physical values are represented by the vector $\boldsymbol{p} \in \mathbb{R}^l$, available for control at time instant t_k. Each physical actuator request p_i, where i represents the index of the vector $\boldsymbol{p}$, must satisfy the saturation limits defined by the set $\left[\underline{p}_i, \bar{p}_i\right]$. Suppose the relation between the virtual commands $\boldsymbol{v}$ and physical actuator requests $\boldsymbol{p}$ is given by the nonlinear 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. Note that the number of algebraic constraints in (1) is equal to the number of virtual commands. Then, the problem of actuator sharing can be expressed as the selection of the physical actuator request vector $\boldsymbol{p}$ in order to minimize the total actuator effort, subject to the constraints of satisfying the saturation limits and the constraint (1). Mathematically, this can be formulated as an optimization problem as follows. At each time t, given a vector of virtual inputs $\boldsymbol{v}(t)$,

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

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 formulation, the term $Q = diag(q_1, \ldots, q_l)$ is a diagonal matrix of weights $q_1, \ldots, q_l$. Thus, the cost function f is a measure of the actuator efforts with the relative importance of each actuator defined by Q.

Remark 1 *The command-request relation presented in* (1) *assumes that this relation is linear in the physical actuator request vector* $\boldsymbol{p}$*. In cases where this relation is nonlinear, it can be linearized with respect to the reference vector* $\boldsymbol{p}_{t_{k-1}}$*, which represents the physical actuator requests computed at the previous time step. This linearization results in an equation of the form* (1)*. It is important to note that this linearized relationship is updated at each time step to account for changes in the reference values. This approach is effective because it aligns with the nature of static ASA proposed in this work, which handles instantaneous values.*

2.2. Actuator Allocation Without Saturation Limits

In the optimization problem formulated above, the command-request relation (1) defines an equality constraint while the saturation limits define a series of inequality constraints. Ignoring the inequality constraints simplifies the optimization problem to a form where the solution can be explicitly derived. The following analysis discusses the derivation of this solution, which will serve as the basis for solving the original problem formulated in Subsection 2.1.

For a given vector $\boldsymbol{v}(t)$ at time t, define g as $g(\boldsymbol{p}) = A\boldsymbol{p} - b$ and g_i as the i^{th} component of the vector. Define the Lagrangian as $\mathcal{L}(\boldsymbol{p}, \boldsymbol{\lambda}) = f(\boldsymbol{p}) - \boldsymbol{\lambda}^T\, g(\boldsymbol{p})$, where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m]^T \in \mathbb{R}^m$ is the vector of Lagrange multipliers. According to the theory of Lagrange multipliers [8], if a local minimum $\boldsymbol{p}^*$ exists, and the vectors $\nabla_{\boldsymbol{p}} g_1, \ldots, \nabla_{\boldsymbol{p}} g_m$ are linearly independent, then there exists $\boldsymbol{\lambda}^*$ such that

$$\nabla_{\boldsymbol{p}} f(\boldsymbol{p}^*) - \nabla g(\boldsymbol{p}^*)^T \boldsymbol{\lambda}^* = 0, \tag{5}$$

$$g(\boldsymbol{p}^*) = 0. \tag{6}$$

In the above discussion, the notation $\nabla_{\boldsymbol{p}}$ denotes the gradient vector with respect to $\boldsymbol{p}$, and $\nabla g(\boldsymbol{p}^*)$ is matrix whose rows are the gradient vectors $\nabla_{\boldsymbol{p}} g_1(\boldsymbol{p}^*), \ldots, \nabla_{\boldsymbol{p}} g_m(\boldsymbol{p}^*)$, i.e.,

$$\nabla g(\boldsymbol{p}^*) = \begin{bmatrix} \nabla_{\boldsymbol{p}} g_1(\boldsymbol{p}^*) & \ldots & \nabla_{\boldsymbol{p}} g_m(\boldsymbol{p}^*) \end{bmatrix}^T. \tag{7}$$

From the definitions of f and g given above, it is evident that $\nabla_{\boldsymbol{p}} f(\boldsymbol{p}^*) = 2Q\boldsymbol{p}^*$ and $\nabla g(\boldsymbol{p}^*) = A$. Substituting these terms into (5) yields $\boldsymbol{p^*} = \frac{1}{2}Q^{-1}A^T\boldsymbol{\lambda}^*$. Substituting the expression for $\boldsymbol{p}^*$ into (6) results in $\boldsymbol{\lambda}^* = 2\left(AQ^{-1}A^T\right)^{-1} b$. Substituting the expression for $\boldsymbol{\lambda}$ into the expression for $\boldsymbol{p}^*$ gives

$$\boldsymbol{p}^* = Q^{-1}A^T\left(AQ^{-1}A^T\right)^{-1} b, \tag{8}$$

which is the optimal vector physical actuator values that satisfy the command-request relation (1).

2.3. Incorporating Saturation Limits Using Iterative Approach

To incorporate saturation limits, analysis can be carried out with inequality constraints. However, this makes the overall analysis convoluted. Alternatively, a simple iterative algorithm, that is suitable for PCS implementation, is proposed in this section. Before proceeding further, note that the optimization problem formulated in Subsection 2.1 may not have a feasible solution. In other words, no combination of physical actuator values $\boldsymbol{p}$ satisfies the constraint (3) while staying within the saturation limits. To relax the constraints and avoid such cases, slack variables $\boldsymbol{s} = [s_1, \ldots, s_m]^T$ are introduced into the optimization problem. Define $\hat{Q} = diag(q_1, \ldots, q_l, q_{s_1}, \ldots, q_{s_m})$ with $q_{s_j} >> q_i > 0$ for $i = 1, \ldots, l$ and $j = 1, \ldots, m$. In addition, define $\hat{A} = [A, I]$, where $I \in \mathbb{R}^{m \times m}$ is an identity matrix, and $\boldsymbol{p}_a = [\boldsymbol{p}^T \boldsymbol{s}^T]^T$. The modified optimization problem is framed as follows. At each time t, given a vector of virtual inputs $\boldsymbol{v}(t)$,

$$\underset{\boldsymbol{p}_a(t)}{\operatorname{argmin}} f(\boldsymbol{p}_a(t)) = \underset{\boldsymbol{p}_a(t)}{\operatorname{argmin}} \boldsymbol{p}_a^T(t) \hat{Q} \boldsymbol{p}_a(t) \tag{9}$$

subject to the constraints

$$\hat{A}(\boldsymbol{v}(t)) \boldsymbol{p}_a(t) = b(\boldsymbol{v}(t)), \tag{10}$$

$$\boldsymbol{p}_a(t) \in \left[\underline{p}_1, \bar{p}_1\right] \times \ldots \times \left[\underline{p}_l, \bar{p}_l\right] \times (-\infty, \infty) \times \ldots \times (-\infty, \infty). \tag{11}$$

In the above formulation, one slack variable is added to the left hand side of each constraint equations prescribed by the controller. Unlike the problem given in Subsection 2.1, a solution always exists for the above optimization problem since the slack variables are allowed to take any value in $(-\infty, \infty)$. The closed-form equivalent of (8) for the minimization of 9 while satisfying 10 is given by

$$\boldsymbol{p}_a^* = \hat{Q}^{-1} \hat{A}^T \left(\hat{A} \hat{Q}^{-1} \hat{A}^T \right)^{-1} b, \tag{12}$$

To ensure $\boldsymbol{p}_a^*$ satisfies the saturation limits given by (11), an iterative approach, summarized below, is used. While the resulting solution may sometimes be suboptimal, the iterative algorithm is computationally efficient and straightforward, relying solely on evaluating closed-form equations, with maximum iterations limited to the number of physical actuators.

(i) Determine the physical actuator values by solving an unconstrained optimization problem defined in (12). This provides us with an initial solution candidate for $\boldsymbol{p}_a^*$.

(ii) The candidate solution is then checked to determine whether any of the computed values exceed the saturation limits. If any actuator value breaches these limits, their respective value is set to its corresponding saturation limit, ensuring that it does not exceed the prescribed constraints.

(iii) After adjusting the over-limit values, the optimization problem is redefined. In this revised problem, only the physical actuator requests corresponding to the unsaturated terms are taken into account in the optimization, excluding those set to their saturation limits.

(iv) The optimization process is continued iteratively until a solution that satisfies both the optimization objective and the saturation constraints is reached.

Remark 2 *The value of the slack variable s_i obtained from the above algorithm can be viewed as the extent to which the i^{th} constraint equation in* (1) *is violated. By default, the optimization problem tries to set the slack variable values as low as possible because of their high cost. However, if the algorithm generates high values of slack variables, then it can be concluded that the individual controllers are placing demands that are beyond the capabilities of the physical actuators.*

3. HANDLING CHANGES IN CONTROL OBJECTIVES AND ACTUATOR AVAILABILITY

The algorithm described in Subsection 2.3 does not accommodate changes in control objectives or actuator availability. However, the ensuing discussion demonstrates that such variations can be effectively managed by updating the terms within the original optimization problem presented in Subsection 2.1 in certain plasma control scenarios. Intuitively, this update process involves eliminating terms in Q, A, b and $\boldsymbol{p}$ that correspond to inactive objectives and actuators using the control objective adjustment matrix C and actuator status matrix S. The following discussion describes how these matrices are constructed.

Suppose that the term r represents the maximal set of controllers available in a tokamak scenario. Each controller prescribes m_i virtual commands $\boldsymbol{v}_{m_i}$ $(i = 1, \ldots, r)$ such that $m_1 + \ldots + m_r = m \geq r$. Also, assume that the command-request relation corresponding to each controller does not depend on commands prescribed by other controllers. In other words, the command-request relation in (1) has the form

$$A(\boldsymbol{v})\boldsymbol{p} = \begin{bmatrix} A_{m_1}(\boldsymbol{v}_{m_1}) \\ \vdots \\ A_{m_r}(\boldsymbol{v}_{m_r}) \end{bmatrix} \boldsymbol{p} = \begin{Bmatrix} b_{m_1}(\boldsymbol{v}_{m_1}) \\ \vdots \\ b_{m_r}(\boldsymbol{v}_{m_r}) \end{Bmatrix} = b(\boldsymbol{v}). \tag{13}$$

Define the diagonal matrix $\hat{C}$ as $\hat{C} = diag(c_1 I_{m_1}, \ldots, c_r I_{m_r}) \in \mathbb{R}^{m \times m}$, where c_i is a scalar and I_{m_i} is an identity matrix of dimension m_i. Specifically, the scalar c_i assumes a value of 1 when the controller is active and 0 when it is not. The control objective adjustment matrix C is then constructed by combining only the non-zero rows of $\hat{C}$ without changing their order. Thus, if the number of control objectives at time t_k is equal to $m^k \leq m$, then C is a $m^k \times m$ dimensional matrix.

The actuator status matrix is built using a similar method. Recall that the term l defined in Subsection 2.1 is the maximum number of actuators available for control. Define the matrix $\hat{S}$ as $\hat{S} = diag(s_1, \ldots, s_l) \in \mathbb{R}^{l \times l}$, where $s_j = 1$ when the actuator j is available, and $s_j = 0$ otherwise. The actuator status matrix S is built by combining non-zero columns of the matrix $\hat{S}$ without changing the column order. Thus, if the number of active actuators at time t_k is $l^k \leq l$, the matrix S has dimensions $l \times l^k$.

To incorporate the effects accounted for by the control objective adjustment matrix and actuator status matrix into the static ASA, we introduce new terms denoted as $\check{A}$, $\check{Q}$, $\check{b}$, and $\check{\boldsymbol{p}}$, defined as follows:

$$\check{A} = CAS, \qquad \check{Q} = S^T Q S, \qquad \check{b} = Cb, \qquad \check{\boldsymbol{p}} = S\boldsymbol{p}. \tag{14}$$

By solving the optimization problem defined with the above set of new terms, the ASA can effectively adapt to changes in control objectives and actuator availability. Note that the matrices C and S are updated whenever there is a change in control objectives or actuator availability. Thus, the transformation defined in the above equations is not carried out at every time step.

4. ALGORITHM TESTING USING NUMERICAL SIMULATIONS

The effectiveness of the ASA has been demonstrated using nonlinear DIII-D tokamak simulations in the Control Oriented Transport SIMulator (COTSIM) [9]. Three cases were considered: (i) simultaneous control of the safety factor gradient at a rational surface and total thermal energy of the plasma [10], (ii) simultaneous control of the safety factor minimum and total thermal energy [11], (iii) simultaneous control of the safety factor minimum and total thermal energy with an arbitrary actuator failure. The following discussion presents the simulations results.

4.1. Case 1: Safety Factor Gradient and Total Energy Control

The first case considers the problem of simultaneous control of the safety factor q profile gradient at a given rational surface and the total thermal energy W of the plasma. It is well known that instabilities like neoclassical tearing modes can appear at regions where the q profile takes rational values. In scenarios where rational values cannot be avoided, controlling the slope of the safety factor profile might prevent or delay the onset of such instabilities [12]. Similarly, total energy is another plasma property that is related to the MHD stability of the plasma. The evolution of both properties is coupled. Using the ASA proposed in this work, the simultaneous control of these two plasma properties can be made possible.

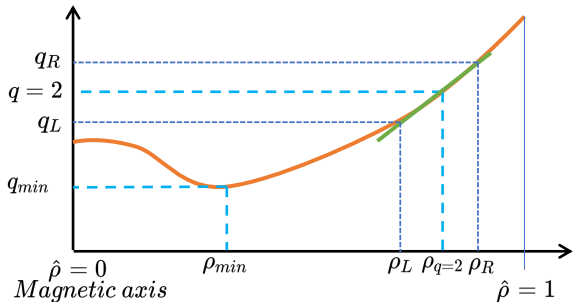

FIG. 2. *Illustration of the safety factor minimum and safety factor gradient approximation*

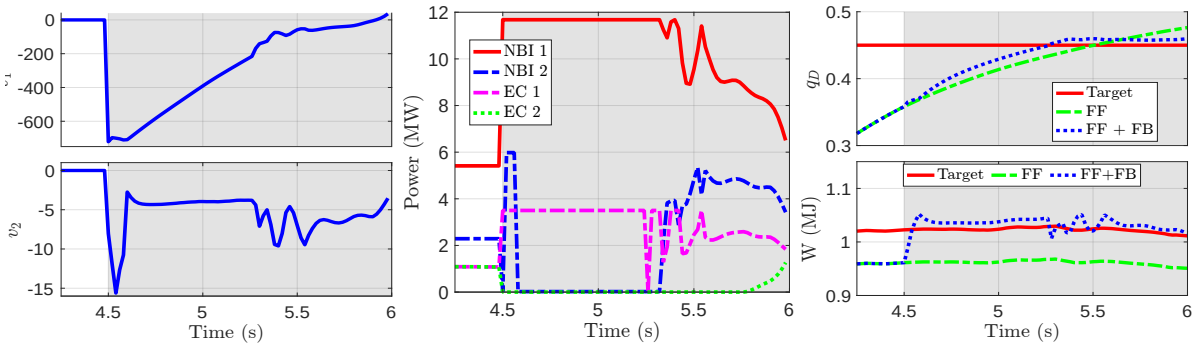

FIG. 3. *Case 1: (a) - Virtual inputs (left), (b) - Actuator requests (center), (c) - Closed-loop trajectories (right).*

The safety factor gradient at a given rational surface can be controlled by regulating two points around the surface [10]. Figure 2 gives a graphical illustration of this idea. Suppose the gradient has to be controlled at $q = 2$. It can be achieved by controlling the difference $q_D = q_R - q_L$, where q_R and q_L are the safety factor values at the right and left control points ρ_R and ρ_L respectively. Given a target gradient $\bar{q}_D$, the error $\tilde{q}_D = q_D - \bar{q}_D$ satisfies

$$\dot{\tilde{q}}_D = \underbrace{\hat{c}_R(t, P_{tot}, q_R) - \hat{c}_L(t, P_{tot}, q_L) - \dot{\bar{q}}_D}_{\check{c}_q} + \boldsymbol{g}_D(t, P_{tot})^T \boldsymbol{u}_{fb}(t), \tag{15}$$

where $\hat{c}_i$, $i = R, L$, are nonlinear functions of total power P_{tot} and corresponding safety factor value q_i, u_{fb} is the vector of feedback auxiliary drive powers, $\boldsymbol{g}_D$ is the vector that account for the spatial deposition of the auxiliary drives. A detailed discussion on the derivation of the above model can be found in [10]. Suppose there are two NBI and two EC clusters available for control, then $u_{fb} = [P_{NBI,1}, P_{NBI,2}, P_{EC,1}, P_{EC,2}]^T$ and $P_{tot} = P_{NBI,1} + P_{NBI,2} + P_{EC,1} + P_{EC,2}$ is the feedback total power. On the other hand, given a target total energy $\bar{W}$, the model for total energy error $\tilde{W}$ is given by

$$\dot{\widetilde{W}} = \underbrace{-\frac{W}{\tau_E(t, P_{tot})} + P_{tot,ff} - \dot{\overline{W}}}_{\check{c}_W} + P_{tot}, \tag{16}$$

where τ_E is the energy confinement time and $P_{tot,ff}$ is the feedforward total power. A detailed discussion of the above model can be found in [13]. Define virtual inputs v_1 and v_2 as $v_1 = \boldsymbol{g}_D(t, P_{tot})^T \boldsymbol{u}_{fb}(t)$ and $v_2 = P_{tot,fb}$. The command-request relation can be written as

$$\begin{bmatrix} g_{D,1}(t, v_2) & g_{D,2}(t, v_2) & g_{D,3}(t, v_2) & g_{D,4}(t, v_2) \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{Bmatrix} P_{NBI,1}(t) \\ P_{NBI,2}(t) \\ P_{EC,1}(t) \\ P_{EC,2}(t) \end{Bmatrix} = \begin{Bmatrix} v_1(t) \\ v_2(t) \end{Bmatrix}, \tag{17}$$

which has the form shown in (1). To drive the errors $\tilde{q}$ and $\tilde{W}$ to 0, feedback linearization is used [13], a control approach where the virtual inputs are selected such that the nonlinear terms in (15) and (16) are canceled while stable linear terms are added. The choice of feedback linearizing inputs that drive the error to 0 is given by

$$v_1 = -\check{c}_q - k_{q,p}\tilde{q}_D - k_{q,I}\int_0^t \tilde{q}_D, \qquad v_2 = -\check{c}_W - k_{W,p}\widetilde{W} - k_{W,I}\int_{t_0}^t \widetilde{W} dt, \tag{18}$$

where $k_{q,p}, k_{q,i}, k_{W,p}, k_{W,i} > 0$ are controller gains. Substitution of the above virtual inputs into (15) and (16) cancels the nonlinear terms $\check{c}_q$ and $\check{q}_W$ and introduces stable linear terms. Lyapunov analysis can be used to prove that the closed-loop system is stable [14].

Closed-loop simulations were carried out in COTSIM for a DIII-D tokamak scenario using the controllers defined in (18) and the static ASA proposed in Subsection 2.3. The DIII-D configuration information corresponding to shot number 147634 was used. To evaluate the controller and ASA's capacity to handle large errors, the saturation limits of the auxiliary powers $P_{NBI,1}, P_{NBI,2}, P_{EC,1}, P_{EC,2}$ were set as 12 MW, 6 MW, 3.5 MW, 3.5 MW, respectively. Figure 3 presents the simulation results for this case. The gray background in the subfigures denotes the period of plasma discharge when the feedback controller and the ASA are active. Figure 3 (a) shows the virtual inputs v_1, v_2 generated (independently) by the safety factor gradient and total energy controllers, respectively. The actuator requests (NBI and EC H&CD powers) that are computed from the virtual inputs by the ASA are shown in Figure 3 (b). Finally, Figure 3 (c) shows the performance of closed-loop trajectories (safety factor gradient q_D and

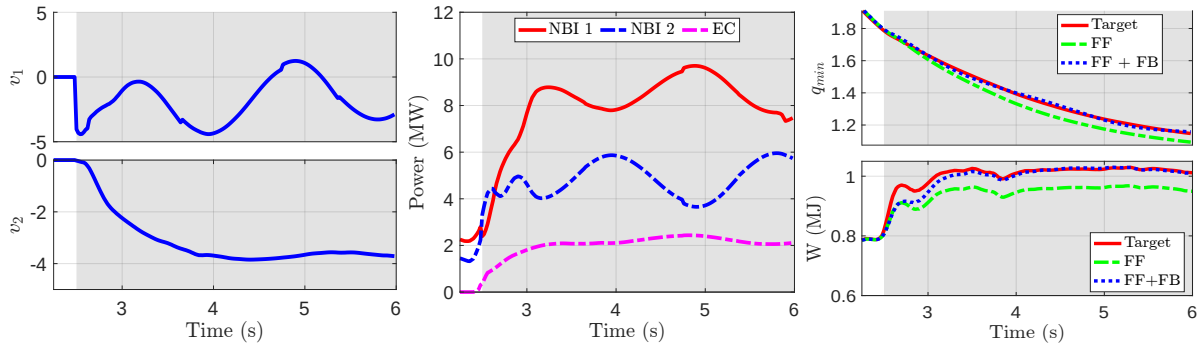

FIG. 4. *Case 2: (a) - Virtual inputs (left), (b) - Actuator requests (center), (c) - Closed-loop trajectories (right).*

total energy W) when the actuator requests generated by the ASA are implemented. The simulation results show that the ASA is effective in allocating the necessary actuation to achieve both control objectives. However, the steady-state error in the closed-loop trajectories shows that the integral gains $k_{q,i}$ and $k_{W,i}$ could be tuned further.

4.2. Case 2: Safety Factor Minimum and Total Energy Control

This case considers the simultaneous control of q_{min} and W. The safety factor minimum q_{min} is another plasma property that is linked to the MHD stability of the plasma [11]. Figure 2 gives a illustration of the safety factor minimum q_{min} and its location ρ_{min}. Safety factor minimum control can be achieved by regulating $\theta_{q_{min}}$, the poloidal flux gradient value at ρ_{min}, since both are related [11]. Thus, the model for safety factor minimum is

$$\dot{\tilde{\theta}}_{q_{min}}(t) = \hat{c}(t, P_{tot}, \theta_{q_{min}}) - \dot{\bar{\theta}}_{q_{min}}(t) + \boldsymbol{h}(t, P_{tot})^T \boldsymbol{u}_{fb}(t), \tag{19}$$

where $\bar{\theta}_{q_{min}}$ is the target, $\tilde{\theta}_{q_{min}} = \theta_{q_{min}} - \bar{\theta}_{q_{min}}$, $\hat{c}$ is a nonlinear function of $P_{tot}, \theta_{q_{min}}$. In this case, 2 NBI clusters and 1 EC cluster were used. Thus, $\boldsymbol{u}_{fb}$ is defined as $\boldsymbol{u}_{fb} = [P_{NBI,1}, P_{NBI,2}, P_{EC}]^T$ and $\boldsymbol{h}(t, P_{tot})$ is the vector that accounts for the current drive deposition profiles. The derivation of the above model is given in [11].

Using the procedure discussed in Subsection 4.1, a feedback linearizing controller was designed for safety factor minimum control. Detailed step-by-step procedure can be found in [11]. Nonlinear simulations were carried out using COTSIM. The DIII-D configuration utilized for the simulations discussed in Subsection 4.1 was used. In the simulations, the saturation limits for $P_{NBI,1}, P_{NBI,2}, P_{EC}$ were set as 13 MW, 13 MW, 4.5 MW, respectively, to test the closed-loop performance in scenarios with significant deviations from the target. Figure 4 (a) and (b) show the virtual inputs computed by the controller and the physical inputs computed by the proposed ASA, respectively. Figure 4 (c) shows the closed-loop trajectories of the safety factor minimum and the total energy. The gray background depicts the period of active feedback control. As evident, the controllers and the static ASA are able to achieve the desired targets.

4.3. Case 3: Safety Factor Minimum and Total Energy Control With Actuator Failure

The ability of the proposed ASA to handle the actuator failure was tested for the safety factor minimum and total energy control problem. All model and control parameters used for carrying the simulations discussed in Subsection 4.2 were used for this case. An artificial EC failure is introduced at 3.5 seconds. The simulations results obtained from COTSIM are shown in Figure 5. The gray background in the figure shows the time when the controller is active. The light gray background 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. Figure 5 (a) and (b) show the virtual inputs generated by the controller and the phyical powers generated by the ASA, respectively. The figures show that the ASA compensates for EC failure by increasing the NBI powers. Figure 5 (c) shows the closed-loop safety factor minimum and total energy trajectories. It is clear that the safety factor minimum follows the target even when the actuator fails. On the other hand, actuator failure causes deviation of the total energy from the target. However, the closed-loop trajectory immediately recovers and converges to the target.

5. CONCLUSION

An enhanced static actuator-sharing algorithm (ASA) has been developed to prescribe physical actuator requests based on individual input commands generated by multiple plasma controllers. The proposed ASA is computationally efficient and can accommodate various plasma controllers and tokamak configurations. Furthermore, the ASA can also handle real-time changes in control objectives and actuator availability. The capabilities of the ASA

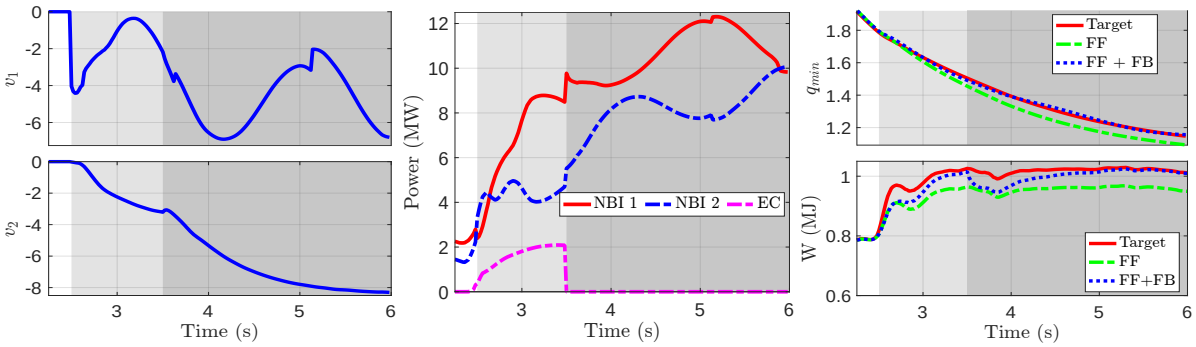

FIG. 5. *Case 3: (a) - Virtual inputs (left), (b) - Actuator requests (center), (c) - Closed-loop trajectories (right).*

have been demonstrated using nonlinear simulations. Future studies can focus on incorporating actuator lag and experimental testing of the proposed ASA.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] KUDLACEK, O., TREUTTERER, W., JANKY, F., SIEGLIN, B., and MARASCHEK, M., Actuator management development on ASDEX-Upgrade, SI:SOFT-30 **146** (2019) 1145.

[2] VU, N. T., BLANKEN, T., FELICI, F., et al., Tokamak-agnostic actuator management for multi-task integrated control with application to TCV and ITER, Fusion Engineering and Design **147** (2019) 111260.

[3] TREUTTERER, W., HUMPHREYS, D., RAUPP, G., et al., Architectural concept for the ITER Plasma Control System, Proceedings of the 9th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research **89** (2014) 512.

[4] MALJAARS, E. and FELICI, F., Actuator allocation for integrated control in tokamaks: Architectural design and a mixed-integer programming algorithm, Fusion Engineering and Design **122** (2017) 94.

[5] GRABER, V. and SCHUSTER, E., Nonlinear burn control in ITER using adaptive allocation of actuators with uncertain dynamics, Nuclear Fusion **62** (2022) 026016.

[6] PAJARES, A. and SCHUSTER, E., Integrated robust control of the global toroidal rotation and total plasma energy in tokamaks, IEEE Transactions on Plasma Science **48** (2020) 1606.

[7] PAJARES, A., SCHUSTER, E., THOME, K. E., et al., Integrated control of individual plasma scalars with simultaneous neoclassical tearing-mode suppression, Nuclear Fusion **62** (2022) 036018.

[8] HESTENES, M. R., *Optimization Theory: The Finite Dimensional Case*, 1975.

[9] PAJARES, A., *Integrated Control in Tokamaks Using Nonlinear Robust Techniques and Actuator Sharing Strategies*, PhD thesis, Lehigh University, Bethlehem, PA, USA, 2019.

[10] PARUCHURI, S. T. and SCHUSTER, E., Nonlinear control of safety factor gradient in tokamaks using spatially variable electron cyclotron current drives, Fusion Engineering and Design **194** (2023) 113914.

[11] PAJARES, A. and SCHUSTER, E., Robust nonlinear control of the minimum safety factor in tokamaks, in *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 753–758, 9.

[12] TURCO, F., LUCE, T., SOLOMON, W., et al., The causes of the disruptive tearing instabilities of the ITER baseline scenario in DIII-D, Nuclear Fusion **58** (2018) 106043.

[13] PAJARES, A. and SCHUSTER, E., Current profile and normalized beta control via feedback linearization and Lyapunov techniques, Nuclear Fusion **61** (2021) 036006.

[14] KHALIL, H. K., *Nonlinear Systems*, Pearson, 3rd edition, 2002.