

Article

Feasible Actuator Range Modifier (FARM), a Tool Aiding the Solution of Unit Dispatch Problems for Advanced Energy Systems

Haoyu Wang ^{1,*}, Roberto Ponciroli ¹, Andrea Alfonsi ², Paul W. Talbot ³, Thomas W. Elmer ¹, Aaron S. Epiney ³ and Richard B. Vilim ¹

¹ Argonne National Laboratory, Lemont, IL 60439, USA; rponciroli@anl.gov (R.P.); elmer@anl.gov (T.W.E.); rvilim@anl.gov (R.B.V.)

² Nucube Energy, Inc., Pasadena, CA 91103, USA; aalfonsi@nucube.energy

³ Idaho National Laboratory, Idaho Falls, ID 83415, USA; paul.talbot@inl.gov (P.W.T.); aaron.epiney@inl.gov (A.S.E.)

* Correspondence: haoyuwang@anl.gov; Tel.: +1-630-252-1548

Abstract: Integrated energy systems (IESs) seek to minimize power generating costs in future power grids through the coupling of different energy technologies. To accommodate fluctuations in load demand due to the penetration of renewable energy sources, flexible operation capabilities must be fully exploited, and even power plants that are traditionally considered as base-load units need to be operated according to unconventional paradigms. Thermomechanical loads induced by frequent power adjustments can accelerate the wear and tear. If a unit is flexibly operated without respecting limits on materials, the risk of failures of expensive components will eventually increase, nullifying the additional profits ensured by flexible operation. In addition to the bounds on power variations (explicit constraints), the solution of the unit dispatch problem needs to meet the limits on the variation of key process variables, including temperature, pressure and flow rate (implicit constraints). The FARM (Feasible Actuator Range Modifier) module was developed to enable existing optimization algorithms to identify solutions to the unit dispatch problem that are both economically favorable and technologically sustainable. Thanks to the iterative dispatcher–validator scheme, FARM permits addressing all the imposed constraints without excessively increasing the computational costs. In this work, the algorithms constituting the module are described, and the performance was assessed by solving the unit dispatch problem for an IES composed of three units, i.e., balance of plant, gas turbine, and high-temperature steam electrolysis. Finally, the FARM module provides dedicated tools for visualizing the response of the constrained variables of interest during operational transients and a tool aiding the operator at making decisions. These techniques might represent the first step towards the deployment of an ecological interface design (EID) for IES units.

Keywords: unit dispatch problem; integrated energy system; command governor; ecological interface design



Citation: Wang, H.; Ponciroli, R.; Alfonsi, A.; Talbot, P.W.; Elmer, T.W.; Epiney, A.S.; Vilim, R.B. Feasible Actuator Range Modifier (FARM), a Tool Aiding the Solution of Unit Dispatch Problems for Advanced Energy Systems. *Energies* **2024**, *17*, 2945. <https://doi.org/10.3390/en17122945>

Received: 11 May 2024

Revised: 29 May 2024

Accepted: 13 June 2024

Published: 14 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Along with the benefits of reducing the carbon footprint and lessening the dependence on fossil fuels, the increasing penetration of renewable energy sources (RESs) constitutes an issue for the stability of power grids. As a result of the limited dispatchability of solar power plants and wind farms, significant fluctuations in the net demand might lead to blackouts and interruption of service [1,2]. The deployment of RESs, combined with the low cost and accessibility of natural gas, has led to a decrease in electricity prices, even resulting in negative values in some regions during periods of high RES production and low demand. Since the energy storage systems are not mature for grid-scale deployment [3], the only available approach is to enhance the operational flexibility of other generating units

connected to the grid. Even traditional base-load units, e.g., nuclear power plants [4], need to be operated according to unconventional paradigms. The concept of integrated energy systems (IESs) aligns with this perspective. IESs are cooperatively controlled systems that dynamically apportion thermal and/or electrical energy to provide responsive generation to the power grid while also supporting the production of other energy products. Research activities are currently carried on world wide to develop the IES technology [5,6]. An IES unit combines conventional power plants with different characteristics in synergistic ways. For most units, electricity is considered the primary output, ensuring that grid demand is reliably met within the analysis region. Once electricity demand is met, the IES is designed to dynamically apportion any surplus primary energy to energy storage or to the production of industrial products. In some cases, the industrial product may be an intermediate energy carrier, such as hydrogen, or an intermediate chemical feedstock, such as methanol.

The problem of evaluating the mixed outputs of the units that minimize the generation costs or maximize the revenue is called unit dispatch. With regard to IES units, the goal of the unit dispatch problem consists of evaluating the power outputs and the heat flows that maximize the profitability of the whole system over the specified time horizon (days, weeks). It is a constrained optimization problem since limits on the minimum and the maximum power output along with the maximum ramp rate are imposed on all the power plants that constitute the IES unit. In this work, these limits are referred to as explicit constraints. Traditionally, unit dispatch problems are solved by adopting Dynamic Programming (DP) algorithms [7] or Mixed-Integer Linear Programming (MILP) algorithms [8]. These methods are effective for power plants that adopt current operational paradigms. The adoption of new operational paradigms requires dynamic coupling between the power plants that constitute an IES unit. Thermomechanical loads caused by frequent power adjustments can accelerate the performance degradation and the wear and tear of components. If a unit is flexibly operated without respecting operational limits, the risk of expensive component failures will eventually increase, along with the frequency of maintenance interventions, nullifying the additional profits ensured by flexible operation. Let us consider the case of a nuclear unit. In Pressurized Water Reactors (PWRs), the condenser is the second most expensive component of the entire plant after the reactor vessel [9]. To modulate the power output and effectively dispose of the heat produced in the core, the control actions governing the steam generator and steam turbine must be coordinated. Along with the constraints on reactor power and turbine power, limits on the variations in pressure and moisture content in the turbine-discharged steam flow rate need to be imposed as well. Otherwise, the stress on the turbine blades and the condenser pipes will exceed design-basis conditions, thus causing damage to the components. These limits are referred to as implicit constraints. Available tools need to be enhanced to formally account for implicit constraints. While necessary, adding numerous constraints on temperatures, pressures and flow rates increases the size of the optimization problem. In this work, an innovative approach for solving the unit dispatch problem for any set of generating units with key variables that need to be constrained during operation (such as temperature, pressure, flow rate, etc.) is presented and demonstrated. The key premise of the proposed approach is that solving the optimization problem by simultaneously considering both explicit and implicit constraints leads to prohibitive computational costs. A more efficient approach consists of addressing the optimization problem as a two-phase process. Similar to a Kalman filter [10], the proposed scheme has a prediction phase and an update phase. For the prediction phase, a traditional unit dispatcher finds a tentative solution that meets the explicit constraints on the power outputs. Such a tentative solution is then assessed by a sanity-check downstream of the unit dispatcher called a “validator” (update phase). Its task consists of determining whether the tentative set-point trajectories are compliant with the implicit constraints on temperatures, pressures and flow rates. The embedded algorithm is the command governor (CG) [11,12]. It is a discrete-time device that is used in the field of feedback control of dynamic systems. Although a CG is typically used as a supervisory control algorithm,

this work proposes an innovative application, i.e., aiding the solution of the unit dispatch problem for complex mixed-generation plants. Thanks to a model of the studied system, the unit response to demanded power transients can be evaluated, and possible violations of the imposed limits are identified and returned to the unit dispatcher. The algorithm is recursive, i.e., the solution of the optimization problem will be approved only when all the constraints are met.

Widely used to describe the safe operation of a power plant, turbomachinery or any engineering systems, the Normal Operation Region (NOR) is defined as the ensemble of operating points at which the system can function without exceeding design-basis conditions of its components [13]. Having identified p process variables whose evolution needs to be limited to preserve the integrity of system components, the NOR can be mathematically defined as a p -dimensional polytope in the phase space [14]. The validator's task then consists of ensuring that the power demanded for the plants constituting the IES unit does not lead any of these p variables to violate the NOR bounds. In Figure 1, a qualitative representation is shown. The dashed green trajectory describes an acceptable unit response during a power transient, i.e., the entire trajectory lies within the NOR bounds (blue region). In case operating conditions exceed the NOR bounds, the components will experience excessive thermomechanical loads, thus increasing the risk of failure over time (orange region). Ultimately, if a transient pushes the system even further, the safety variable limits (solid black line) will be breached, potentially leading to a severe accident. The proposed scheme provides a quantitative estimate of the NOR, i.e., the constraints on these p variables are "translated" into limits on the power set-point trajectories. As a major outcome, it provides a tool that conveys the major information about the plant conditions in a simple and straightforward fashion.

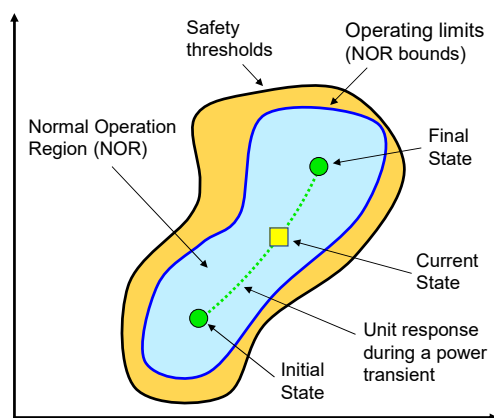


Figure 1. Qualitative illustration of the NOR. The recommended operating conditions are represented by the blue region.

The paper is organized as follows. In Section 2, the new approach to the solution of the unit dispatch problem for IES units and the role of the validator are presented. In Section 3, the modeling framework implementing the proposed scheme is described. In Section 4, the algorithms embedded into the FARM for predicting the system response and detecting the constraint violations and corrective actions are described. In Section 5, the IES unit selected as a test case is described, and the constraints enforced during operation are listed. In Section 6, the steps that lead to the solution of the unit dispatch problem for the studied system are defined, and the obtained results and the associated computational costs are discussed. In Section 7, visualization techniques for displaying the IES unit response during power transients are presented, and a possible role of the FARM as an aid to unit operators is proposed. Finally, the main conclusions are drawn and future steps are discussed.

2. Description of a New Approach to the Solution of Unit Dispatch Problems

As mentioned in the Introduction, when solving the unit dispatch problem for an IES unit, there are two sets of limits that need to be respected, i.e., explicit constraints and implicit constraints. Any solution of the unit dispatch problem that meets both these two sets of constraints is defined as a feasible solution.

- Explicit constraints: the electrical power output can assume any value within a characteristic window, i.e., $P_{min} \leq P(t) \leq P_{max}$ (power output limits). Electrical power variations cannot be too steep either, i.e., $|\frac{dP}{dt}| \leq \frac{dP}{dt}|_{max}$ (power ramp-rate limits). Explicit constraints are also defined as low-resolution physics constraints because they represent fewer physics details.
- Implicit constraints: to prevent thermomechanical loads from surpassing the design limits, constraints on temperature, pressure, and flow rate must be taken into account, i.e., $T_{min} \leq T(t) \leq T_{max}$, $p_{min} \leq p(t) \leq p_{max}$, and $w_{min} \leq w(t) \leq w_{max}$. Implicit constraints are also defined as high-resolution physics constraints since they account for a large number of aspects that govern the processes.

Ideally, all these constraints should be addressed by the unit dispatcher (Figure 2a). Unfortunately, the size of the ensuing optimization problem would entail a large computational burden. The problem could be addressed by splitting the two sets of constraints. Specifically, the unit dispatcher will address the explicit constraints for each unit, and the corresponding control systems will address the implicit constraints by imposing saturation limits on the PID controllers (Figure 2b). By adopting this method, the set-points issued by the unit dispatcher will never cause the violation of implicit constraints of the IES unit components, and the size of the optimization problem will be manageable since the unit dispatcher only addresses the explicit constraints. However, the resulting power transients will be sub-optimal and there is the possibility that the load demand will not be met.

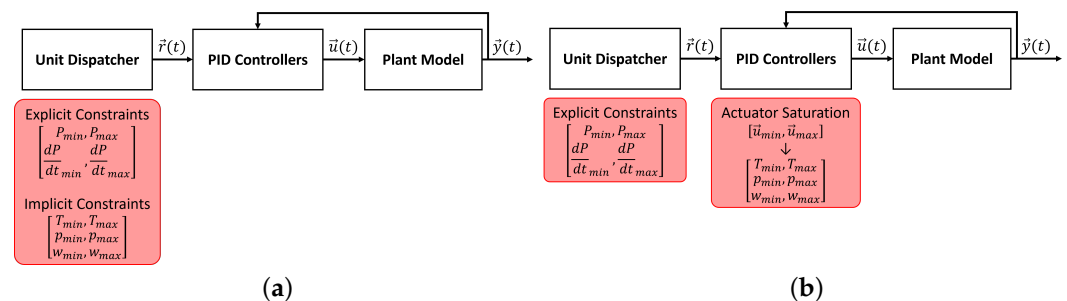


Figure 2. Tentative solutions to account for both explicit and implicit constraints into the optimization algorithm. (a) Addressing all the constraints by using the unit dispatcher; (b) Addressing the implicit constraints by imposing saturation limits on the PID controllers.

The approach proposed in this work consists of solving the constrained optimization problem by placing a sanity-check downstream of the unit dispatcher to ensure the feasibility of the found solutions. The CG was identified as a suitable algorithm to fulfill this task. It is an add-on scheme that enforces pointwise-in-time constraints by modifying the set-point signals fed to a closed-loop, dynamic system. The problem of designing the control system is then addressed as a two-step process. The primal regulators are tasked to stabilize the system and provide tracking properties in the absence of constraints. They can be traditional feedback regulators, e.g., PID controllers. The constraint enforcement task is left to the CG, i.e., a nonlinear device which is added to the primal compensated nonlinear system. Whenever necessary, the CG modifies the reference set-point signals supplied to the primal regulators to enforce the fulfillment of the constraints [15]. Originally conceived as a layer of control system architectures, the CG is used in this work to speed up the constrained optimization process. As mentioned in the Introduction, the CG “translates” the implicit constraints into explicit constraints, thereby imposing more restrictive limits

on the set-point trajectories. Updated limits aid the dispatcher in identifying an optimal solution that conforms to both sets of constraints.

The proposed approach is composed of the following five steps:

1. The capacities of the IES unit subsystems (i.e., maximum power output/consumption) are defined by the user;
2. Given the subsystem capacities, the HERON unit dispatcher (see Section 3) solves the constrained optimization problem by approximating the optimal power output/consumption for each subsystem. These trajectories ($\vec{r}(k)$) satisfy the explicit constraints (Figure 3);
3. The CG determines whether the trajectories calculated by the HERON unit dispatcher satisfy the implicit constraints too. If not, the CG will provide trajectories complying with both explicit and implicit constraints ($\vec{v}(k)$) and return them to the unit dispatcher for re-optimization;
4. From $\vec{v}(k)$ trajectories, the HERON unit dispatcher derives a feedback to adjust the values of the explicit constraints;
5. Steps (2) through (4) are repeated until $\vec{v}(k)$ and $\vec{r}(k)$ are identical; i.e., the optimal trajectories satisfying heat and electrical power demands are also compliant with both explicit and implicit constraints. The “approved” set-points are then issued to the PID controllers that calculate the control actions ($\vec{u}(k)$) to simulate the prescribed power transients.

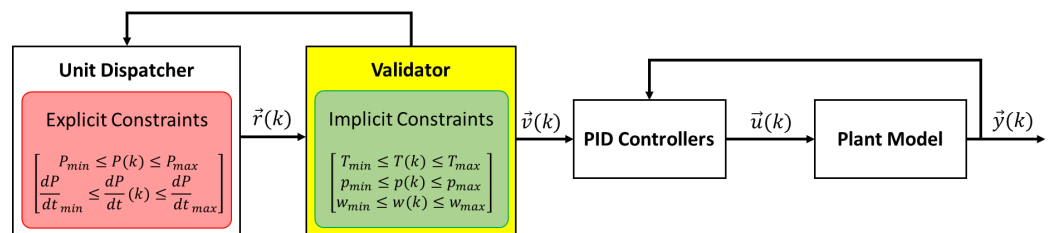


Figure 3. Proposed scheme for efficiently addressing both explicit and implicit constraints.

In Figure 4, a detailed view of the feedback provided by the validator to the unit dispatcher is shown. For simplicity, the optimization of a single power set-point is depicted. Once the system’s evolution in response to $r(k)$ has been simulated, the margins with respect to the lower and upper bounds on the implicitly constrained variables are evaluated, “translated” into margins on the bounds of production variables (i.e., electrical power, in this case) and returned to the unit dispatcher. In Figure 4, they are outlined in red as $[(\delta P^{min})_{Imp}, (\delta P^{max})_{Imp}]$. These contributions will be added to the explicit constraints $((P^{min})_{Exp}$ and $(P^{max})_{Exp}$). In this way, a more comprehensive characterization of the feasible system capabilities is obtained. The final constraints (P^{min} and P^{max}) will include both the inherent limits on standalone subsystem performance (expressed by the explicit constraints) and the additional, more restrictive limits resulting from the interactions with other interconnected subsystems typical of the IES unit configuration (expressed by the translated margins). The unit dispatch problem being addressed is the traditional one. The inclusion of the validator provides dynamic corrections to the constraints on the optimized variables as function of their sensitivity to changes in the monitored process variables as temperatures, pressures, and flow rates. Specifically, it reduces the space of feasible solutions in which to search for the optimal one.

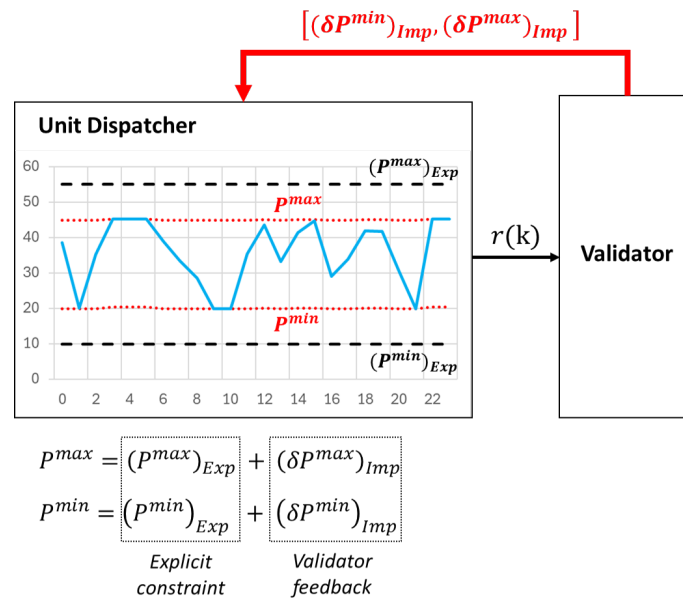


Figure 4. Graphical representation of the feedback system formed by the unit dispatcher and the validator. The limits on implicitly constrained variables are translated into constraints on production variables and returned to the unit dispatcher.

3. Description of the Modeling Framework for Solving the Unit Dispatch Problem for IES Units

In this work, the scheme outlined in Section 2 was implemented to address the unit dispatch challenge for an IES. The IES units are constituted by thermally coupled power plants designed to operate in synergistic ways. Due to their complex configuration and the necessity for component coordination, IES units incur higher initial and operational costs compared to standard energy systems. Assessing the financial viability of such investments involves analyzing the economic performance over their operational lifespan, which spans several decades. Such analysis requires an unprecedented modeling and simulation effort to ensure the proposed operational paradigm will not cause damage to components or materials. To achieve this goal, a research program was started in 2015 [16], focusing on integrated modeling, simulation and experimental facility development. Consequently, a comprehensive framework named FORCE (Framework for Optimization of Resources and Economics) was established [17], integrating and enhancing tools from various software to cater to the specific requirements mentioned.

- RAVEN (Risk Analysis Virtual Environment) [18] is a multipurpose uncertainty quantification (UQ), regression analysis, data analysis and optimization framework [19]. RAVEN is the central engine for the deployment of all the analysis needs, providing the tools for the construction of the analysis workflows. RAVEN is charged with deploying the optimization methodology and algorithms that are used for the capacity optimization. In addition, the framework delivers the techniques for the generation of synthetic scenarios starting from available datasets via its time series generators [20]. Thanks to this capability, hypothetical scenarios characterized by a larger penetration of renewable energy sources or higher level of variability in the demand profile can be investigated.
- HYBRID software [21] is a collection of component models developed primarily using the Modelica language and deployed using the Dymola 2020x software suite [22]. An IES unit is constituted by traditional power plants whose dynamics are characterized by very different time constants; i.e., along with a nuclear power plant, there are gas turbines that can be started up from cold shutdown conditions in 20 min. To obtain a realistic description of the system performance, the capability of accurately simulating the dynamic response of the IES components over different time scales

is crucial. HYBRID contains low-fidelity and high-fidelity models for all the energy systems and components that have been selected as part of the IES framework.

- HERON (Holistic Energy Resource Optimization Network) [23] is a plug-in that allows performing stochastic full-system techno-economic analysis of grid energy-resource systems with economic drivers [24]. The development targets analysis of electricity and secondary product generation and consumption in regional balancing areas, including flexibility to include arbitrary resources, as well as arbitrary resource consumers and producers. HERON is the tool for the construction of the overall IES/FORCE workflow, providing tools for deploying systems' dispatching strategies and capacity optimization.
- TEAL (Tool for Economic AnaLysis) [25] is a discounted cash flow analysis plug-in that allows for a generic definition of cash flows, with drivers provided by RAVEN. It includes flexible options to deal with taxes, inflation, discounting and offers capabilities to compute a combined cash flow for components with different component lives [26]. TEAL is tasked with deployment of the economic analysis within the IES methodological framework.

In Figure 5, the graphical representation of the FORCE workflow is shown. At the top, the HERON module is represented. The result of the techno-economic analysis includes the dispatching strategy, i.e., the set of power output and the heat flux set-points for the energy systems that constitute the unit. At the bottom of Figure 5, the HYBRID module is represented. The embedded Dymola model of the IES unit is used to simulate the power transients that are prescribed by the set-point trajectories provided by the HERON. In addition, the Dymola model can be run off-line to generate datasets characterizing the dynamics of the IES unit. Between these two modules is the validator assessing the tentative dispatching strategies. The developed validator module is named FARM (Feasible Actuator Range Modifier) [27], which bridges the high-fidelity (HYBRID) and low-fidelity (HERON) modeling of the IES unit dynamics. In Figure 6, a graphical representation is shown.

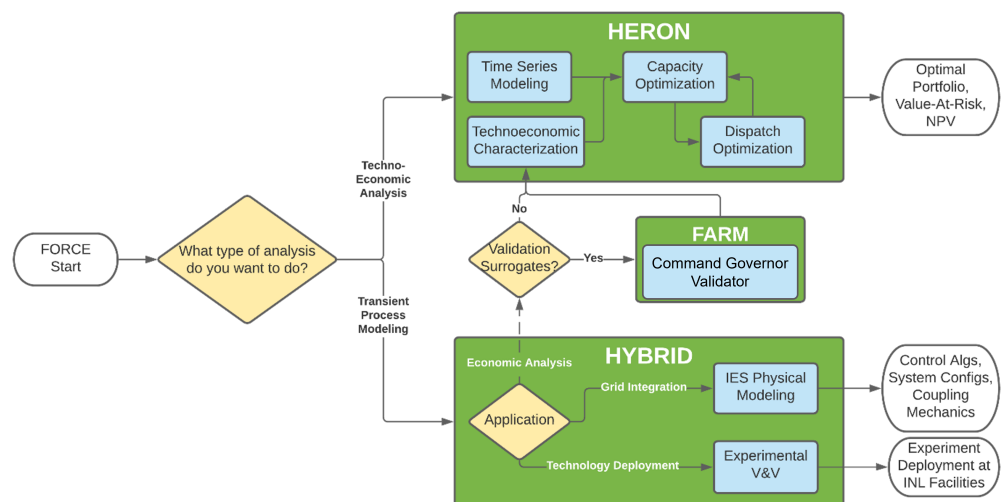


Figure 5. Workflow of the Framework for Optimization of Resources and Economics ecosystem (FORCE) [17]. The FARM serves as the bridge between high-fidelity and low-fidelity description of the IES unit.

To detect possible violations of implicit constraints, every time a power or heat flux set-point changes, the response of the IES unit components needs to be estimated by adopting a reduced order model of the unit and the corresponding control system. For this application, linear time-invariant (LTI) state-space models are often adopted. In LTI models, the evolution of system state variables \vec{x} and system output variables \vec{y} with respect to the system input vector \vec{v} in the discrete time domain is shown in Equations (1) and (2).

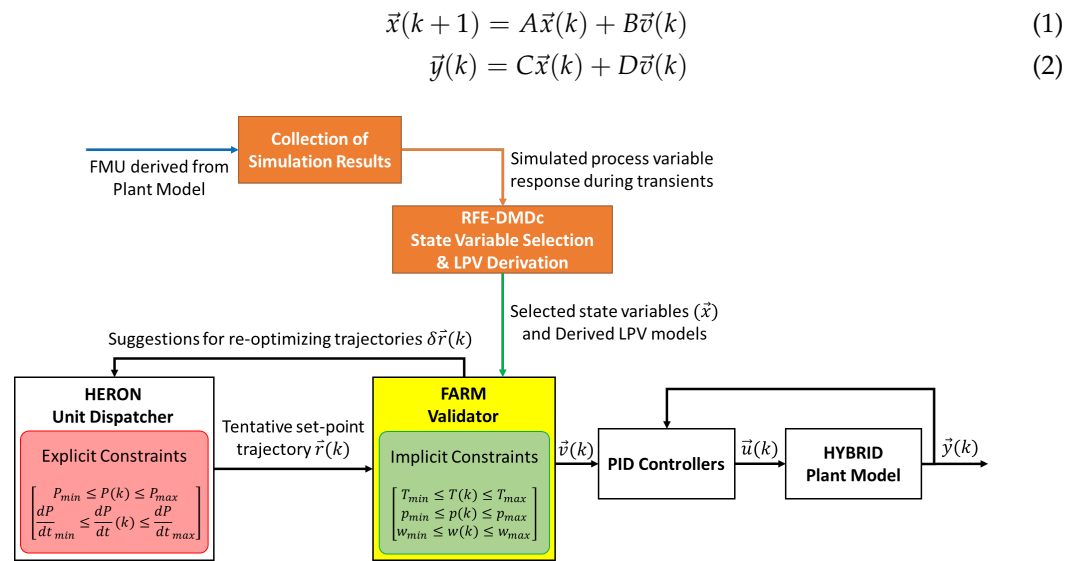


Figure 6. Detailed view of the FARM-based workflow.

In this work, the unit response to tentative set-points is estimated using a linear parameter-varying (LPV) model [28]. An LPV model is constituted by a set of LTI models obtained by linearizing the set of Differential-Algebraic Equations (DAEs) [29] describing the dynamics of the studied system around anchor points in the scheduling space called scheduling parameters (Equations (3) and (4)). In the current implementation, the set of matrices corresponding to the LTI models are indexed by the corresponding production level.

$$\vec{x}(k+1) = A(p)\vec{x}(k) + B(p)\vec{v}(k) \quad (3)$$

$$\vec{y}(k) = C(p)\vec{x}(k) + D(p)\vec{v}(k) \quad (4)$$

where $A(p)$, $B(p)$, $C(p)$ and $D(p)$ are the state-space matrices parameterized by the scheduling parameter vector (p).

To build the LPV model, a data-driven method was implemented in FARM. As a first step, a dataset containing the IES unit responses during multiple power transients must be generated. These data may be either measurements collected from the actual unit or simulation data generated by running a high-fidelity model of the unit. These latter solutions can be significantly facilitated by the use of the FMI/FMU standard [30]. The Functional Mock-up Interface (FMI) defines a standardized interface to be used in simulations that simplifies the creation, storage, exchange, and (re-)use of dynamic system models. With such an interface implemented, the Dymola model can be exported as an executable called an FMU (Functional Mock-up Unit), which can be imported into other supported environments for simulation. An FMU may either encapsulate its own solvers (FMU for Co-Simulation) or require the simulation environment to perform numerical integration (FMU for Model Exchange) [22,31]. The high-fidelity model used in this work to generate the simulation data was developed in Dymola simulation environment. After building the model of the studied IES unit by assembling the components models from the HYBRID library, it was exported into a Co-Simulation FMU and used within the FARM (Figure 6). The FMU receives set-point trajectories and returns the responses of all the available process variables from the simulation. The simulation results were then collected in a database and used to derive the LPV model.

To help solve the unit dispatch problem, the FARM operates in two consecutive phases. The described procedure for deriving the LPV model is performed during an off-line phase called self-learning. After generating the LPV model, the unit dispatch problem is solved during the on-line phase called dispatch. HERON first issues power set-point trajectories meeting the explicit constraints, which represents low-resolution physics. The FARM

simulates these power transients using the LPV model and evaluates the margins of the implicitly constrained variables relative to the corresponding implicit constraints, which represents high-resolution physics. These margins are then “translated” into margins on the explicit constraints ($\delta\vec{r}(k)$) and returned to the HERON (Figure 6). In this way, the information on constraint enforcement can be used during the next iteration of the dispatch optimization process. These iterations between the HERON and the FARM during the dispatch phase are necessary to find a solution that meets both explicit and implicit constraints. In this scheme, the task of the LPV model consists of predicting the response of the studied system during transients and notifying possible constraint violations. In addition, the procedure implemented in the FARM module enables the LPV model to “learn” during the unit dispatch problem-solving process. In case the FMU is coupled with the HERON-FARM optimization scheme, real-time data are collected during the dispatch phase, and new matrices are derived and added to the LPV model developed during the self-learning phase. In this way, the latest, most accurate mathematical description of the system dynamics is available. Inevitably, this procedure results in an additional computational burden. In this work, this capability was not utilized, i.e., no FMU was integrated with the HERON-FARM optimization scheme during the dispatch phase, and the LPV model developed during the self-learning phase was not improved thereafter. Finally, when demonstrating the capabilities of the scheme by solving a unit dispatch problem (Section 6), the response of the IES unit is simulated by using the LPV model.

4. Description of the Algorithms Embedded in the FARM

In this section, a brief overview of the algorithms embedded into the FARM module is provided. For a more complete description of these tools, the reader is referred to dedicated references.

- Command Governor: it performs a multi-dimensional optimization of set-point trajectories fed by the unit dispatcher (Section 4.1).
- Convex Hull: it removes the redundant constraints (Section 4.2).
- Dynamic Mode Decomposition with Control: it derives the matrices that constitute the state-space model (Section 4.3).
- Recursive Feature Elimination with DMDc: it provides the list of state variables of the derive state-space model (Section 4.4).

4.1. Command Governor (CG) for Enforcing Implicit Constraints

The CG algorithm adjusts the vector reference signal (\vec{r}) to produce a feasible set-point signal (\vec{v}) to be fed to the feedback regulators. The goal is deriving set-point trajectories that will start power transients meeting the imposed constraints. In this section, a description of this algorithm is provided. For further information, the reader may refer to [32,33]. The CG is based on an iterative, finite-horizon optimization scheme. At time $t = k$, the current state of the system is sampled and a feasible set-point value is calculated for a relatively short time horizon $t \in [k, k + g]$, where g is a positive integer representing the number of time-steps that constitute the time horizon. An online calculation is used to explore state trajectories from the current state ($t = k$) to the end of time horizon ($t = k + g$). After the first step of the optimized set-point trajectory is fed to the PI controllers, the state is sampled again and the calculation is repeated, yielding a new value. The key concept behind the CG algorithm is the definition of an admissible region for the adjusted set-point values at time $t = k$. Any vector lying within this region can be imposed as system set-point at time $t = k$ and remain constant over the time horizon, without causing the implicitly constrained variables to violate the imposed limits over the same time horizon. The CG algorithm shares several similarities with the Model Predictive Control (MPC) [34]. Although both algorithms optimize the actions governing the dynamic system over a finite time horizon that shifts forward at every time-step, they operate at different levels. The MPC replaces traditional PID controllers by optimizing the actions executed by the actuators to meet desired specifications, e.g., tracking capabilities, asymptotic stability, and

constraint enforcement; the CG “collaborates” with the PID controllers, i.e., it optimizes the set-point trajectories to ensure constraint enforcement, whereas the PID controllers guarantee good tracking capabilities and asymptotic stability. It is a two-step approach that does not alter the traditional control system architecture.

Let us consider a system characterized by p output variables ($\vec{y} = [y_1, y_2, \dots, y_p]^T$) and assume that the lower and the upper limits on these variables can be represented by a set of linear inequalities, i.e., $y_j \geq y_j^{min}$ and $y_j \leq y_j^{max}, j \in [1, p]$ (Equation (5)).

$$\begin{bmatrix} +1 & 0 & \dots & 0 \\ -1 & 0 & \dots & 0 \\ 0 & +1 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & +1 \\ 0 & 0 & \dots & -1 \end{bmatrix} \cdot \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix} \leq \begin{bmatrix} +y_1^{max} \\ -y_1^{min} \\ +y_2^{max} \\ -y_2^{min} \\ \vdots \\ +y_p^{max} \\ -y_p^{min} \end{bmatrix} \Rightarrow S \cdot \vec{y} \leq \vec{s} \tag{5}$$

First, let us impose that the constraints are not violated at the current time-step ($t = k$). Equations (4) and (5) can be written as Equation (6).

$$S \cdot \vec{y}(k) = SC^d \vec{x}(k) + SD^d \vec{v}(k) \leq \vec{s} \tag{6}$$

Then, the feasibility of the set-point trajectory $\vec{v}(k)$ needs to be assessed for the following time-steps $t = k + 1, k + 2, \dots, k + g$ as well. Once K_x and K_r matrices are defined (Equation (7)), the linear inequalities can be re-written as shown in Equation (8).

$$K_x = SC^d, \quad K_r = SD^d \tag{7}$$

$$K_x \cdot \vec{x}(k + m) + K_r \cdot \vec{v}(k) \leq \vec{s}, \quad m = 1, 2, \dots, g \tag{8}$$

The admissible region is defined by all the $g + 1$ linear inequalities in Equations (6) and (8). Equation (3) is adopted to express Equation (8) in terms of the current state variables (Equation (9)).

$$\begin{aligned} SC^d \cdot \vec{x}(k + m) + SD^d \cdot \vec{v}(k) &\leq \vec{s} \\ SC^d (A \cdot \vec{x}(k + m - 1) + B^d \cdot \vec{v}(k)) + SD^d \cdot \vec{v}(k) &\leq \vec{s} \\ K_x A \cdot \vec{x}(k + m - 1) + (Lim - K_x AT) \cdot \vec{v}(k) &\leq \vec{s} \end{aligned} \tag{9}$$

where

$$Lim = S \cdot (C^d \cdot (I - A^d)^{-1} \cdot B^d + D^d), \quad T = (I - A^d)^{-1} \cdot B^d \tag{10}$$

Equation (8) can be expressed in terms of the current state variables values ($\vec{x}(k)$) by iteratively performing the same derivation (Equation (11)).

$$K_x A^m \cdot \vec{x}(k) + (Lim - K_x A^m T) \cdot \vec{v}(k) \leq \vec{s}, \quad m = 1, 2, \dots, g \tag{11}$$

In Equation (12), the constraints expressed in Equation (5) evaluated between $t = k$ and $t = k + g$ are assembled. They constitute the MOAS (Maximal Output Admissible Set), i.e., the admissible region for the vector of adjusted set-points at time $t = k$. The corresponding matrix equation is reported in Equation (13).

$$\begin{bmatrix} K_x & K_r \\ K_x A & Lim - K_x AT \\ K_x A^2 & Lim - K_x A^2 T \\ \vdots & \vdots \\ K_x A^g & Lim - K_x A^g T \end{bmatrix} \cdot \begin{bmatrix} \vec{x}(k) \\ \vec{v}(k) \end{bmatrix} \leq \begin{bmatrix} \vec{s} \\ \vec{s} \\ \vec{s} \\ \vdots \\ \vec{s} \end{bmatrix} \tag{12}$$

$$H_v \cdot \vec{v}(k) \leq \vec{h}_v(k) \tag{13}$$

Geometrically, the MOAS defines an m -dimensional convex polytope. Each vector laying within this polytope is a feasible solution to the constraints enforcement problem (Equation (5)) [35]. In the Introduction, the concept of the NOR was presented. The NOR was defined as a p -dimensional polytope defined by the set of constraints imposed on the identified p variables. Given the meaning of the MOAS, the relationship with the NOR can be immediately identified. The NOR is the feasible region for the constrained process variables of the system, and it is defined in a p -dimensional phase space. The MOAS is the admissible region for the adjusted set-points that do not lead to forbidden responses of the output variables, and it is defined in an m -dimensional space. The CG translates the constraints on the implicitly constrained variables into limits on the set-point trajectories. The NOR defined in the p -dimensional space is replaced by a region in the m -dimensional space. With respect to the description of the system dynamics, the two regions are equivalent. As a major outcome, the CG allows deriving a procedural definition of the NOR that can be implemented into an algorithm for optimizing the operation of a unit such as a power plant. To find the optimal solution, the CG solves a Quadratic Programming (QP) optimization problem [36]. As shown in Equation (14), the goal consists of calculating an adjusted set-point trajectory that is as close as possible to the reference one.

$$\vec{v}(k) = \underset{\vec{v}(k)}{\operatorname{argmin}} [\vec{v}(k) - \vec{r}(k)]^2 \quad (14)$$

With regard to the application of the FARM to the solution of the unit dispatch problem, in the current formulation, the power and heat flux set-points are adjusted on an hourly basis. The length of the prediction horizon should then cover the whole hour. The time constants governing the dynamics of the power plants that constitute the IES unit are not so long, i.e., the operational transients initiated by set-point variations are considered completed before the end of the hour. Still, the prediction horizon is imposed equal to one hour to be conservative. The CG will adjust the set-points to enforce constraints throughout the entire hour.

4.2. Convex Hull Method for Removing Redundant Constraints

When defining the unit dispatch problem, implicit constraints are imposed to limit the thermomechanical loads. Since the components that constitute the IES unit of interest are thermally coupled, the constraints on the process variables of one component might affect the operation of another component. As a result, some constraints defining the MOAS might be redundant. If they are not removed, they will increase the computational burden and cause the QP solver to return a sub-optimal solution. To address this issue, a procedure borrowing the “convex hull” concept from geometry [37] was implemented in the FARM module. As a starting point, the MOAS for m -dimensional actuator signal with q constraints (Equation (13)) is normalized by dividing each element of H_v matrix and \vec{h}_v vector by the Euclidean norm of the corresponding row in H_v matrix. This normalized MOAS is reported in Equation (15).

$$\begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1m} \\ H_{21} & H_{22} & \cdots & H_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ H_{q1} & H_{q2} & \cdots & H_{qm} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \leq \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_q \end{bmatrix} \quad (15)$$

As a result of normalization, all the rows of H matrix in Equation (15) share the same Euclidean norm (Equation (16)).

$$\sqrt{H_{i1}^2 + H_{i2}^2 + \cdots + H_{im}^2} = 1, (i = 1, 2, \cdots, q) \quad (16)$$

Equation (15) defines an m -dimensional convex polytope. Once an inner point (\vec{c}) is found meeting all the q constraints, a translatory motion is performed to include the origin point (Equation (17)).

$$\begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1m} \\ H_{21} & H_{22} & \cdots & H_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ H_{q1} & H_{q2} & \cdots & H_{qm} \end{bmatrix} \cdot \begin{bmatrix} v_{c,1} \\ v_{c,2} \\ \vdots \\ v_{c,m} \end{bmatrix} \leq \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_q \end{bmatrix} - \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1m} \\ H_{21} & H_{22} & \cdots & H_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ H_{q1} & H_{q2} & \cdots & H_{qm} \end{bmatrix} \cdot \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{bmatrix} \quad (17)$$

where $v_{c,j} = v_j - c_j$ ($j = 1, 2, \dots, m$).

Equation (17) can then be simplified as shown in Equation (18).

$$\begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1m} \\ H_{21} & H_{22} & \cdots & H_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ H_{q1} & H_{q2} & \cdots & H_{qm} \end{bmatrix} \cdot \begin{bmatrix} v_{c,1} \\ v_{c,2} \\ \vdots \\ v_{c,m} \end{bmatrix} \leq \begin{bmatrix} h_{c,1} \\ h_{c,2} \\ \vdots \\ h_{c,q} \end{bmatrix} \quad (18)$$

where $h_{c,i} = h_i - \sum_{j=1}^m H_{ij} \cdot c_j$ ($i = 1, 2, \dots, q$).

It is worth stressing that the origin point ($0 \in \mathbb{R}^m$) is within the m -dimensional polytope defined by Equation (18), i.e., $h_{c,i} \geq 0$. A new matrix M is defined as the element-wise division between H_{ij} and $h_{c,i}$ (Equation (19)),

$$M = \begin{bmatrix} \frac{H_{11}}{h_{c,1}} & \frac{H_{12}}{h_{c,1}} & \cdots & \frac{H_{1m}}{h_{c,1}} \\ \frac{H_{21}}{h_{c,2}} & \frac{H_{22}}{h_{c,2}} & \cdots & \frac{H_{2m}}{h_{c,2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{H_{q1}}{h_{c,q}} & \frac{H_{q2}}{h_{c,q}} & \cdots & \frac{H_{qm}}{h_{c,q}} \end{bmatrix} \quad (19)$$

The i^{th} row of M matrix is an m -dimensional vector. Its direction follows the m -dimensional normalized vector composed by the numerators (Equation (20)). Its magnitude is inversely proportional to the value of $h_{c,i}$.

$$\vec{H}_{i,Direction} = [H_{i1}, H_{i2}, \dots, H_{im}] \quad (20)$$

The convex hull is built by using all the q rows of m -dimensional vectors. Among all the m -dimensional vectors that share the same direction, the one with largest magnitude (smallest values of $h_{c,i}$) corresponds to the tightest constraint. This property can be used to define a ranking criterion. The convex hull is the smallest convex set that contains all the provided points, i.e., the points on the vertices of such convex polytope correspond to the tightest constraints in the MOAS. For visualization purposes, the convex hull corresponding to a two-input test case was built. In this test case, a constant flow rate of high-temperature nitrogen was distributed between a 30 kW_e Nitrogen Turbine (NT) unit for electricity production and a heat exchanger to store the surplus thermal energy in a 200 kW_{th} Thermal Energy Storage (TES) unit. The TES can discharge to supply electricity under high demand period. The unit dispatch problem consists of optimizing the corresponding power set-points at the beginning of each hour to meet the load demand for electrical power. In a discrete-time domain (10 s time-steps are adopted), the MOAS is built by imposing upper and lower bounds on the five monitored variables (including the flow rate of NT and charging level of TES) for each of the 361 time-steps that constitute the prediction horizon. It is an impressively high number of constraints, and most of them add no value. Overall, the M matrix has 3610 rows of two-dimensional vectors (blue points in Figure 7). The 14 blue points on the vertices of the convex polygon correspond to the 14 tightest constraints in the MOAS, whereas all the other 3596 blue points inside the polygon correspond to constraints that can be neglected. The ConvexHull function is available from the scipy python package [38]. It allows constructing a convex hull

from a given M matrix and returning the indexes of the points on the vertices which correspond to the non-redundant constraints. The final set of constraints determines the same MOAS as the initial set. Consider the results reported in Figure 8. The original 3610 constraints and the reduced 14 non-redundant constraints define the same region of multi-dimensional actuators. Besides the reduced computational costs, the convergence to the optimal solution has improved. When too many redundant constraints are provided, the QP solver is challenged, and a sub-optimal solution is occasionally returned (black square in Figure 8a). Thanks to the convex hull method, the solution is within the feasible region and has a minimized distance from the original actuator (black square in Figure 8b).

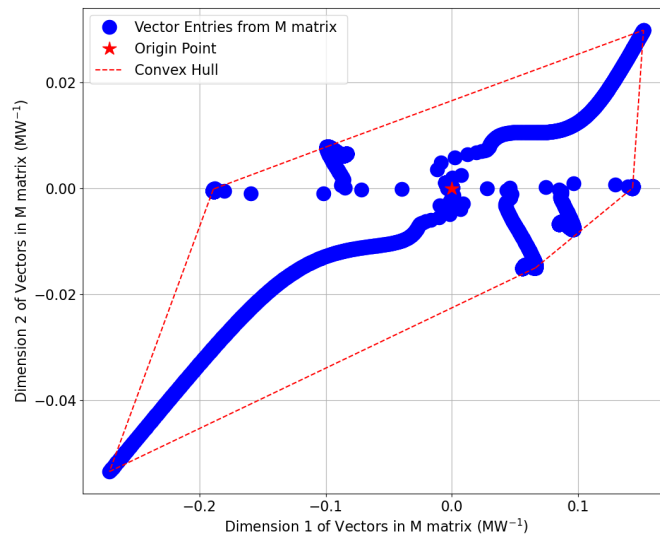


Figure 7. Graphical representation of the convex hull containing 3610 points.

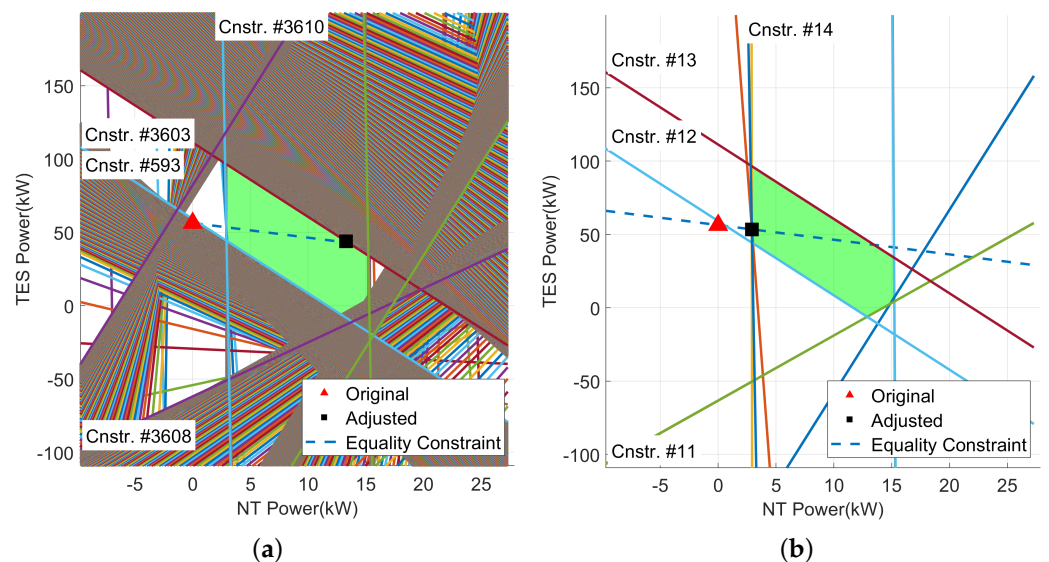


Figure 8. Illustration of the role of the convex hull in the optimization process. The original MOAS (green region) defined by 3610 constraints and the sub-optimal solution (a) and the final MOAS (green region) defined by 14 non-redundant constraints calculated by the convex hull method and the optimal solution (b) are shown.

4.3. Dynamic Mode Decomposition with Control for Deriving the State-Space Matrices

As mentioned in Section 3, FARM needs a state-space model to predict the IES unit response during operational transients and to avoid recommending dispatch variations that could compromise the integrity of components. Models based on the solution of

first-principle equations are preferred as they can accurately describe a wide range of operating conditions without any training datasets. However, when a power plant is operated, physical parameters (e.g., thermal resistances, friction factors, etc.) evolve in time because of inevitable degradation phenomena. As a result, the dynamics of the system is impacted, leading the model to become outdated over time. The effectiveness of the constraint enforcement algorithm, which depends on the accuracy of model predictions, will consequently decline. In the process of developing tools to aid the solution of the unit dispatch problem in real-world scenarios, the latest, most accurate description of the system conditions and dynamics, i.e., an actual Digital Twin (DT) that mimics the system behavior in response to performed control actions, is necessary. FARM uses a system identification method that allows deriving the state-space model of the studied system from the measurements of the main process variables. In particular, the state matrix (\tilde{A}^d) and the input matrix (\tilde{B}^d) of the state-space model in the discrete-time domain (Equations (1) and (2)) are calculated by adopting the Dynamic Mode Decomposition with Control (DMDc) [39]. The matrices are constructed by collecting temporal snapshots of the system inputs ($\vec{v}(k)$) and state variables ($\vec{x}(k)$) over discrete time-steps ($k \in [1, l]$), as shown in Equation (21).

$$\begin{aligned} X' &= \begin{bmatrix} | & | & \cdots & | \\ \vec{x}(2) & \vec{x}(3) & \cdots & \vec{x}(l) \\ | & | & \cdots & | \end{bmatrix} \\ X &= \begin{bmatrix} | & | & \cdots & | \\ \vec{x}(1) & \vec{x}(2) & \cdots & \vec{x}(l-1) \\ | & | & \cdots & | \end{bmatrix} \\ V &= \begin{bmatrix} | & | & \cdots & | \\ \vec{v}(1) & \vec{v}(2) & \cdots & \vec{v}(l-1) \\ | & | & \cdots & | \end{bmatrix} \end{aligned} \quad (21)$$

Since the derived matrices represent the best-fit solution for the data contained in the training dataset, the relationship in Equation (1) does not exactly hold. It can be rewritten in a matrix form to include the new data matrices.

$$X' \approx \tilde{A}^d X + \tilde{B}^d V \quad (22)$$

To calculate \tilde{A}^d and \tilde{B}^d , Equation (22) is rewritten as:

$$X' \approx G \Omega \quad (23)$$

where $G = [\tilde{A}^d, \tilde{B}^d]$ and $\Omega = [X, V]$. The operator $G \in \mathbb{R}^{n \times (n+m)}$ can be calculated as:

$$G = X' \Omega^\dagger \quad (24)$$

To solve Equation (24), the Singular Value Decomposition (SVD) is applied to the augmented data matrix Ω (Equation (25)).

$$\Omega = U \Sigma W^* \approx \tilde{U} \tilde{\Sigma} \tilde{W}^* \quad (25)$$

where $\tilde{U} \in \mathbb{R}^{(n+m) \times q}$, $\tilde{\Sigma} \in \mathbb{R}^{q \times q}$ and $\tilde{W} \in \mathbb{R}^{(l-1) \times q}$ are the truncated SVD components. The truncation value q is the number of non-zero elements in Σ . In this work, the truncation value q is set to limit the condition number of Σ matrix to below 10^9 . Equation (26) provides an approximation of G .

$$G \approx X' \tilde{W} \tilde{\Sigma}^{-1} \tilde{U}^* \quad (26)$$

By breaking the linear operator \tilde{U} into two components according to the dimensions of $\vec{x}(k)$ and $\vec{v}(k)$, the estimates of \tilde{A}^d and \tilde{B}^d matrices can be found (Equation (27)).

$$[\tilde{A}^d, \tilde{B}^d] \approx [X' \tilde{W} \tilde{\Sigma}^{-1} \tilde{U}_1^*, X' \tilde{W} \tilde{\Sigma}^{-1} \tilde{U}_2^*] \quad (27)$$

where $\tilde{U}_1 \in \mathbb{R}^{n \times q}$ and $\tilde{U}_2 \in \mathbb{R}^{m \times q}$.

To complete the state-space model, C^d and D^d matrices need to be derived. Given the nature of the studied systems, the input vector has no direct impact to the system output (D^d matrix is zero). To find the best-fit solution of the C^d matrix using the system state vectors $\vec{x}(k)$ and system output vectors $\vec{y}(k)$ in the training dataset, the matrix of temporal snapshot of system output vector is built (Equation (28)).

$$Y = \begin{bmatrix} | & | & \cdots & | \\ \vec{y}(1) & \vec{y}(2) & \cdots & \vec{y}(l-1) \\ | & | & \cdots & | \end{bmatrix} \quad (28)$$

Equation (2) can be rewritten in a matrix form by inserting the snapshots corresponding to state and output variables (X and Y) (Equation (29)). \tilde{C}^d matrix can be calculated by using the Moore-Penrose pseudo-inverse algorithm [40], as shown in Equation (30).

$$Y = C^d X + D^d Y \approx \tilde{C}^d X \quad (29)$$

$$\tilde{C}^d \approx YX^{-1} \quad (30)$$

4.4. Recursive Feature Elimination for Selecting the State Variables

To derive the matrices that constitute the state-space model, the DMDc tracks the responses of the state variables during operational transients. The state variables of a dynamic system are defined as the minimum set of process variables that allows predicting its future behavior in response to any given set of inputs. If an analytical model is not available, identifying the state variables from the pool of process variables is not trivial. Many commonly used algorithms for deriving state-space models only reconstruct the relationship between input and output variables (regression models). The main limit of these methods is that the state variables of the derived models have no physical meaning, i.e., their number is determined by the quality of the curve-fitting on the training dataset [41]. To address this issue, Recursive Feature Elimination (RFE) was adopted to identify a single, exhaustive set of process variables for a system constituted by multiple subsystems interacting with each other. This automated procedure for state variable selection was implemented into FARM. As shown in Figure 6, the optimal set of process variables is identified and their evolution is tracked to derive the LPV state-space model. In this section, a brief overview of the scheme is reported.

The cost function accounting for the accuracy of both state variable and output variable predictions is defined as follows. Let us consider n candidate state variables and p output variables. The known values of the candidate state variables ($x_i(k \geq 1)$) and the output variables ($y_j(k \geq 1)$) are contained in the training dataset. The predicted values of the candidate state variables ($\hat{x}_i(k \geq 1)$) and output variables ($\hat{y}_j(k \geq 1)$) are calculated by using the trained matrices (\tilde{A}^d , \tilde{B}^d and \tilde{C}^d), the input variable trajectory ($\vec{v}(k \geq 0)$), and the initial values of the candidate state variables ($x_i(k = 0)$). $\sigma(x_i)$ and $\sigma(y_j)$ represent the standard deviations of the i^{th} dimension of the known state variables and the j^{th} dimension of the known output variables, and they are used for normalization purposes. The cost function is obtained by summing the weighted mean square error (MSE) between the predicted values and the known values of the candidate state variables and the weighted MSE between the predicted values and the known values of the output variables calculated over l time-steps (Equation (31)).

$$\text{CostFunction} = \frac{1}{n \cdot l} \sum_{i=1}^n \sum_{k=1}^l \left(\frac{\hat{x}_i(k) - x_i(k)}{\sigma(x_i)} \right)^2 + \frac{1}{p \cdot l} \sum_{j=1}^p \sum_{k=1}^l \left(\frac{\hat{y}_j(k) - y_j(k)}{\sigma(y_j)} \right)^2 \quad (31)$$

The RFE is a well-known tool that has been extensively adopted to develop feature-ranking methodologies for dimensionality reduction applications [42]. The procedure adopted in the FARM for selecting the state variables is iterative, and it consists of the following steps:

1. Construct an evaluation model trained on a complete feature space dataset.
2. Rank the features by using one of the following methods:
 - (a) Sensitivity ranking method (ranking of the features based on their sensitivity coefficients);
 - (b) Correlation ranking approach (ranking of the features based on their correlation coefficients);
 - (c) A model-based importance ranking method exploiting the mathematical formulation of the evaluation model (e.g., linear model coefficients and others).
3. Remove the features with the lowest scores calculated by the selected ranking criterion.
4. Repeat the process until the user-defined number of variables is selected.

The RFE method is an instance of backward feature elimination. To reduce the computational time, multiple features are removed during each iteration. The method produces a feature subset ranking, as opposed to a feature ranking. Feature subsets are nested $F_1 \subset F_2 \subset F_3 \subset \dots \subset F$. To achieve the higher efficiency while avoiding the fitting issues caused by the collinearity between candidate variables, a hierarchical clustering pre-filtering [43] was then deployed. The development of an automated approach to state variable selection based on RFE was driven by the observation that the \tilde{C}^d matrix is a transfer operator from the state space to the output space of the evaluation model. Once normalized by using a min-max scaling approach, the $\tilde{C}^d \approx YX^{-1}$ matrix is used to assess the importance of the candidate state variables (features) with respect to the output variable(s). The importance matrix is defined as follows:

$$I(i, j) = \frac{C(i, j) - \min(C(i, :))}{\max(C(i, :)) - \min(C(i, :))} \quad (32)$$

where:

- $C(i, j)$ is the original value of the element in the i^{th} row and j^{th} column of \tilde{C}^d ;
- $C(i, :)$ is the i^{th} row of \tilde{C}^d ;
- $\min(C(i, :))$ and $\max(C(i, :))$ are the minimum and maximum values in the i^{th} row of \tilde{C}^d , respectively.

Once importance matrix is evaluated, a “score” for each state variable is derived by calculating the expected value of the importance matrix along each column (Equation (33)). The candidate state variables are then ranked by this score; i.e., the variables with the lowest scores are eliminated. The importance matrix and the scores are updated at each iteration of the feature elimination process, until the desired number of variables is reached.

$$\bar{I}(j) = \sum_{i=1}^p \frac{I(i, j)}{p} \quad (33)$$

Let us consider a dynamic system constituted by two subsystems, A and B (Figure 9) and assume that N process variables (\vec{z}^A) are recorded for subsystem A , and M process variables (\vec{z}^B) are recorded for subsystem B . The goal is to select the state variables of the coupled system, i.e., the minimum set of $(n + m)$ process variables describing its dynamics ($\vec{z}^{opt} \equiv \vec{x}^{System}$), where $n < N$ and $m < M$.

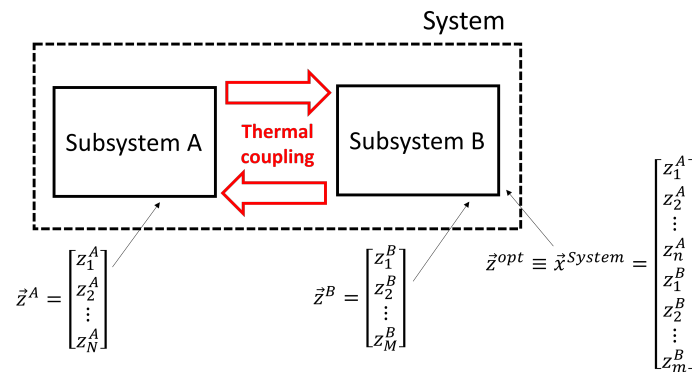


Figure 9. Simplified representation of interacting subsystems. The process variables for subsystems A and B as well as the state variables of the coupled system are outlined.

Initially, the RFE-based scheme was applied to the entire set of process variables, without discriminating between the subsystems. The results were not satisfactory, i.e., most of the selected variables belonged to a single, high-impact subsystem. To ensure an unbiased selection, an alternative importance-ranking methodology was proposed. The main steps are described below.

- **Step I**
The RFE selection is performed on each subsystem, i.e., A and B are independently studied. At the end of this stage, the two sets of candidate state variables for subsystems A and B are selected (\bar{x}^A and \bar{x}^B).
- **Step II**
The two sets of recorded process variables are analyzed with a cross-correlation technique. This step is crucial even when the subsystems are not interacting since the process variables of a subsystem can be “privileged”, being characterized by higher scores through the importance metrics (Equation (33)). The influence that \bar{z}^A variables perform on the candidate state variables (\bar{x}^B) and the output variables (\bar{y}^B) of B is evaluated and vice versa. It is worth stressing that the impact on both the candidate state variables and the output variables needs to be accounted for (Equations (34) and (35)). As a result, two new sets of candidate state variables are obtained ($\bar{x}^{A \rightarrow B}$ and $\bar{x}^{B \rightarrow A}$).

$$\bar{x}^{A \rightarrow B} := \bar{z}^A \rightarrow [\bar{x}^B, \bar{y}^B] \quad (34)$$

$$\bar{x}^{B \rightarrow A} := \bar{z}^B \rightarrow [\bar{x}^A, \bar{y}^A] \quad (35)$$

- **Step III**
A cross-system set of candidate state variables is generated ($[\bar{x}^A, \bar{x}^B, \bar{x}^{A \rightarrow B}, \bar{x}^{B \rightarrow A}]$). The optimal set ($\bar{z}^{opt} \equiv \bar{x}^{System}$) is evaluated by maximizing the accuracy of the model in reconstructing the state variable and the output variables spaces. A parallel searching is then performed over all the combinations of the cross-system set of candidate state variables by minimizing the cost function (Equation (31)).

5. Description of the Test-Case System and Operational Constraints

To demonstrate the capabilities of FARM module, an IES comprising three subsystems was adopted as a test case. As shown in Figure 10, a constant superheated steam flow rate (w_{Steam}) is generated by a nuclear reactor and shared between a balance of plant (BOP) and a high-temperature steam electrolysis (HTSE) process through the energy manifold (EM). The primary purpose of this flow rate is to generate electricity in the BOP (P_{BOP}). A significant portion of the steam flow rate (w_{BOP}) is then channeled towards the BOP to be expanded in the steam turbine. After meeting the load demand ($D(t)$), any excess steam (w_{HTSE}) and any excess power (P_{HTSE}) are used to feed the hydrogen production process in the HTSE. The rate value of the steam flow rate (67.53 kg/s) is not sufficient for operating both the BOP and the HTSE at full capacity simultaneously, i.e., the steam distribution

between the two thermally coupled subsystems needs to be properly coordinated. To avoid abrupt pressure variations, a single directional steam sink with prescribed pressure (not represented in Figure 10) is connected to the EM (w_{Sink}). In addition, a gas turbine (GT) is used to supply the grid with electricity during high demand scenarios (P_{GT}). In the graphical representation of the studied system, the blue and the red arrows represent the steam pipes and electricity lines, respectively, where the steam mass flow rate balance and the electrical power balance are satisfied (Equations (36) and (37)).

$$w_{Steam} = w_{BOP}(t) + w_{HTSE}(t) + w_{Sink}(t) \quad (36)$$

$$D(t) = P_{BOP}(t) + P_{HTSE}(t) + P_{GT}(t) \quad (37)$$

Details and reference operating conditions of the three subsystem models are provided in Sections 5.1–5.3.

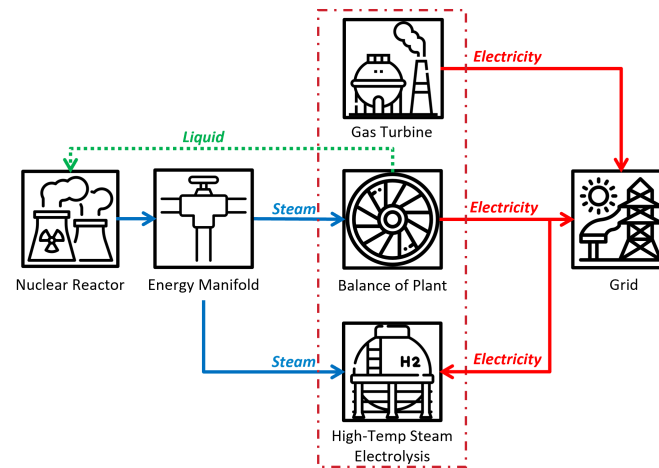


Figure 10. A graphical representation of the IES unit selected as a test case. The thermal coupling between subsystems is outlined [44].

5.1. Gas Turbine (GT)

The GT is a 52-MW_e natural gas-burning turbine. A detailed view of the component in the Dymola graphical user interface is shown in Figure 11. Two dedicated system buses, i.e., the “sensorBus” and the “actuatorBus”, are implemented. The former collects the process variables to be monitored (red dashed trajectories), while the latter collects the signals that are sent to the system actuators (green dashed trajectories). The control system is represented by the block labeled as “control system” at the top. It is constituted by a PID controller that generates a signal adjusting the gas flow rate in response to the desired power output variations. The thermo-mechanical constraints to be met during GT operation are (1) the bounds on electrical power output to avoid cold start and (2) the bounds on the gas turbine firing temperature to avoid inefficient combustion or turbine structural damage (Table 1).

Table 1. List of input and constrained output variables for each subsystem. The adopted values for the upper and lower bounds are reported.

Subsystem	Input (v)	Outputs (\bar{y})	Lower Bounds	Upper Bounds
GT	Power set-point (v_1)	Electrical power output (y_1) Firing temperature (y_2)	20.0 MW 1100 K	50.0 MW 1500 K
BOP	Power set-point (v_2)	Electrical power output (y_3) Turbine inlet pressure (y_4)	25.0 MW 21.0 bar	45.0 MW 31.0 bar
HTSE	Power set-point (v_3)	Electrical power consumption (y_5) Hydrogen production rate (y_6)	0.2 MW 2.0 g/s	1.2 MW 10.0 g/s

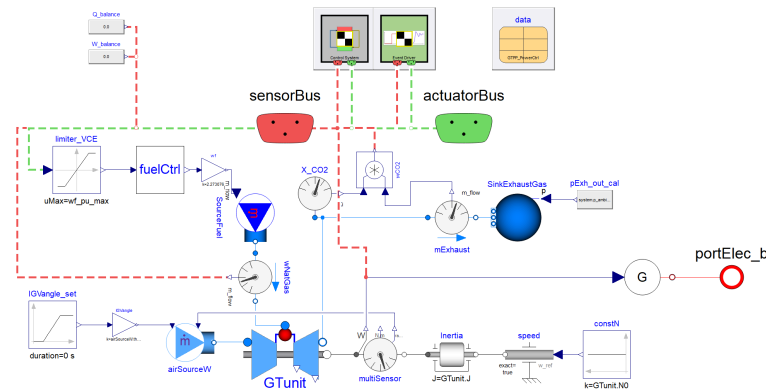


Figure 11. Graphical user interface of the Dymola GT model [22].

5.2. Balance of Plant (BOP)

The BOP is a 47 MW_e Rankine energy conversion cycle. A detailed view of the component in the Dymola graphical user interface is shown in Figure 12. Similar to the GT model, the process variables to be monitored (i.e., inlet steam pressure, steam temperature, generated electric power, feed water temperature and mass flow rate) are returned to the PID controllers through the “sensorBus”, and the actuation signals controlling the opening of turbine control valve “TCV”, bypass valve “InternalBypass”, feedwater valve “SHS_charge_control” and feedwater pump are fed into the system through the “actuatorBus”. As shown in Figure 12, a portion of the steam flow rate is first fed into the high pressure turbine (HPT), then into the low pressure turbine (LPT) for electric power generation, and finally discharged to the condenser. The steam left is used to heat up the condensed water and the auxiliary feed water that exits through “port_b”. The thermo-mechanical constraints to be met during BOP operation are (1) the bounds on the electrical power output to avoid cold start and (2) the upper bound on the steam turbine inlet pressure variations to avoid excessive loads on the turbine blades (Table 1).

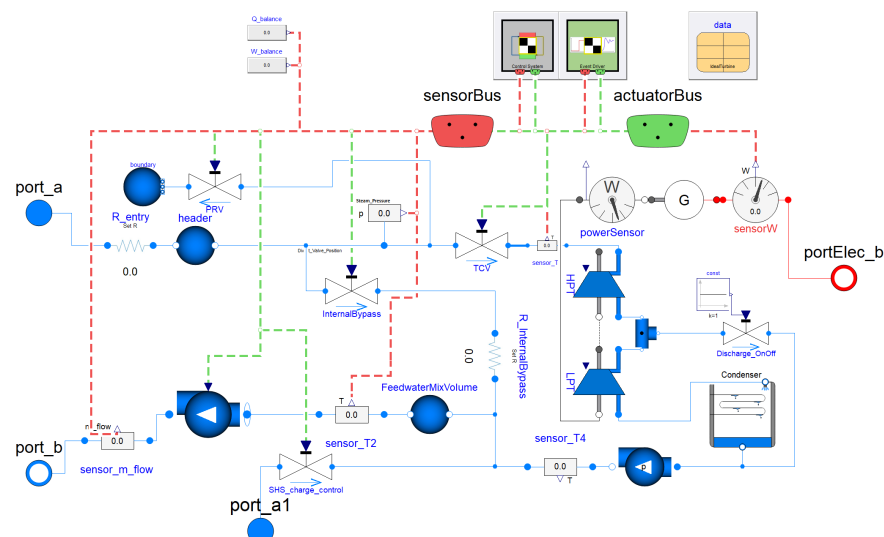


Figure 12. Graphical user interface of the Dymola BOP model [22].

5.3. High-Temperature Steam Electrolysis (HTSE)

The HTSE is a hydrogen production plant. The nominal electricity consumption rate is 1.3-MW_e. A detailed view of the component in the Dymola graphical user interface is shown in Figure 13 [45]. The main component is a Solid Oxide Electrolysis Cell (SOEC), which uses electrical energy as the driving force to split a water/steam mixture to produce oxygen and hydrogen on anode and cathode, respectively. The produced hydrogen exits

the SOEC with the remaining steam and is purified by a mixed-gas condenser that separates hydrogen from steam, while the produced oxygen is removed by a sweep gas flow rate. The external steam from “SteamIn_Port” is regulated to have a maximum flow rate of (0.37 kg/s) by the “SteamValve”. The electrical power determines the hydrogen production to keep the steam utilization rate below 24%. The maximum value of the hydrogen production rate is determined by the external steam supply. As a quick summary, the electrical power consumption is limited to avoid depletion or breakdown of SOEC, and the hydrogen production rate is limited to avoid the HTSE from cold start or overload (Table 1).

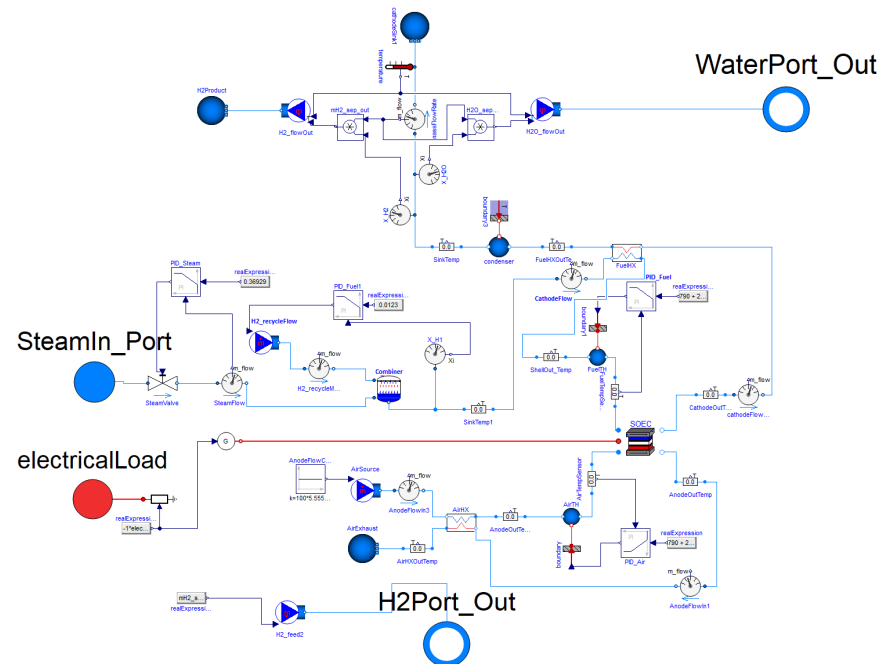


Figure 13. Graphical user interface of the Dymola HTSE model [22].

6. Solution of the Unit Dispatch Problem for the Studied IES Unit

In this section, the derivation of the state-space model for predicting the system response is described (Section 6.1), and the solution of the unit dispatch problem are presented. In practical applications, the boundary conditions that drive the electricity and hydrogen markets are continuously changing. To demonstrate the capability of the HERON-FARM in managing these fluctuations at reasonable computational costs, the results corresponding to two different scenarios are shown. The former scenario is characterized by a time-varying electrical power demand trajectory and time-varying generation costs (Section 6.2); the latter scenario is characterized by fluctuating hydrogen market prices (Section 6.3).

6.1. Derivation of the LPV State-Space Model

As discussed in Sections 4.3 and 4.4, the selection of state variables and the derivation of state-space models are conducted using data-driven techniques that utilize the simulated responses of the IES unit process variables during multiple operational transients as inputs. Data can be either manually collected through an external simulator or automatically generated if an FMU is provided to FARM (Figure 6) during the self-learning phase. Since the subsystems that constitute the IES unit can be addressed as closed-loop systems with a single input (i.e., the power set-point), a square-wave-shaped trajectory is issued to each of them. Trajectories issued to the GT, the BOP and the HTSE include 5, 4 and 3 power levels, respectively (Table 2). For each of the obtained 60 power scenarios, 6-h-long transients were simulated, with the values of process variables sampled every 10 s. In this process, a numerical issue was observed. When synchronized square-wave-shaped trajectories are applied to multiple input variables at the same time, collinearity occurs among the rows of the V matrix (Equation (21)). Because the resulting data matrix Ω is singular (Equation (25)),

the DMDc algorithm fails to converge when performing the SVD. To address this issue, the trajectories for different set-points were slightly shifted relative to one another. Prior knowledge of the IES characteristic time constants is required to determine the length of the set-point shift. In this work, a shift equivalent to two sampling periods was applied (Figure 14). The objective is to ensure that no new step-wise transients are imposed on the next input variable until the previous subsystem has settled into a steady-state condition. It is worth noting that the shift on set-points is required when deriving the LPV model, such as during the self-learning phase, or the dispatch phase in case the LPV update is demanded (Section 3). In the simulated unit dispatch problems, the LPV model is not expected to be updated during the dispatch phase, and then the set-points are not shifted.

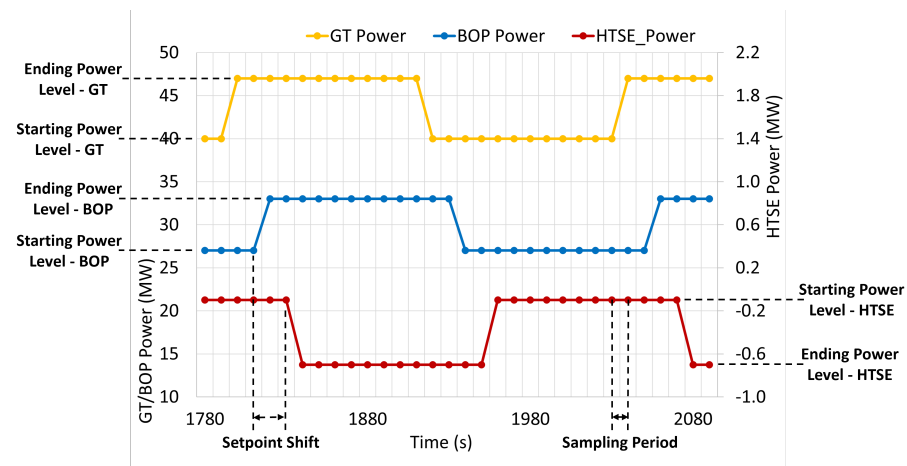


Figure 14. Representation of the shifted, square-wave-shaped set-point trajectories supplied to BOP, GT and HTSE.

Table 2. Starting and ending values of the power set-point trajectories supplied to BOP, GT and HTSE.

Subsystem	Starting Power Level (MWe)	Ending Power Level (MWe)
GT	19.0	26.0
	26.0	33.0
	33.0	40.0
	40.0	47.0
	47.0	19.0
BOP	27.0	33.0
	33.0	39.0
	39.0	45.0
	45.0	27.0
HTSE	−0.1 ¹	−0.7 ¹
	−0.7 ¹	−1.3 ¹
	−1.3 ¹	−0.1 ¹

¹ Negative values indicate power consumption.

The RFE-DMDc algorithm selects the state variables by minimizing the summation of the cost function over the 60 simulated power transients (Equation (31)). In Table 3, the selected variables are reported. In Figure 15, the responses of output and state variables and the corresponding LPV model predictions for one of the identified power transients are shown.

As a result, an LPV model containing 60 sets of state-space matrices is built. A k -nearest neighbor classifier is trained to index the matrices using the ending power level of the three subsystems as the scheduling parameter (Table 2). In Figure 16, the use of the LPV model and the k -nearest neighbor classifier when solving the unit dispatch problem is shown. When the HERON dispatcher provides the tentative set-points ($\vec{r}(k)$ in Figure 3), the classifier selects the set of matrices from the LPV model that best approximates the

system dynamics at the current power level. These matrices are then used by the CG to predict the system response and assess the tentative set-points.

Table 3. List of the state variables selected by the RFE-DMDc algorithm.

State Variable	Name and Physical Meaning
x_1	BOP.sensor_T1.port_a.p (Steam pressure at high-pressure turbine inlet (Pa))
x_2	GT.GTunit.combChamber.E (Total gas energy within the combustion chamber (J))
x_3	HTSE.AirHX.volume_2[4].medium.T_degC (Air temperature at node 4 of air heater exchanger hot side volume (°C))

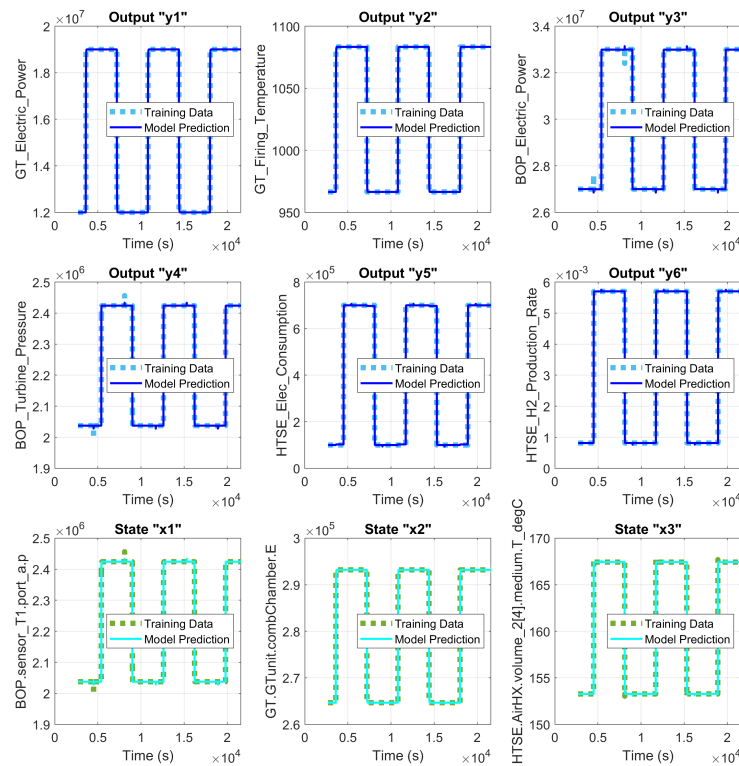


Figure 15. Validation of the state variable selection. The comparison between the predicted and the expected responses of output variables (top two rows) and selected state variables (bottom row) is shown.

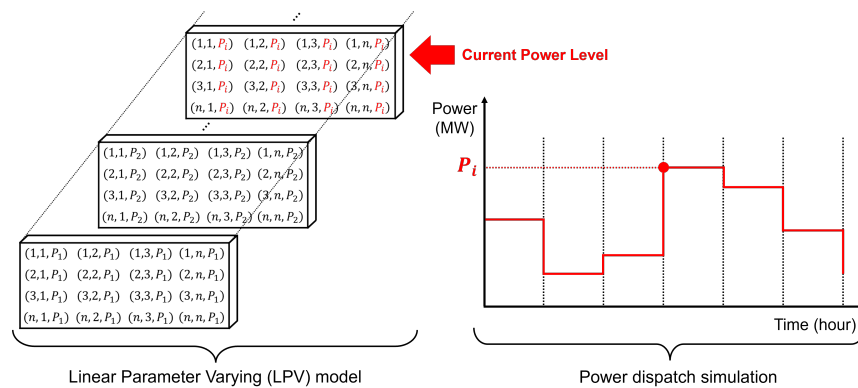


Figure 16. Graphical representation of using an LPV model to solve the unit dispatch problem. In this simplified illustration, a scalar scheduling parameter is used.

6.2. Unit Dispatch Problem 1: Time-Varying Power Generation Cost for GT

Usually, MILP algorithms for solving the unit dispatch problem assume the commodity prices and operational costs of the committed generating units are constant. On the other hand, in real applications, the commodity prices (electricity and hydrogen) and the trajectory of the net electricity demand are updated by the market on an hourly-basis or even at smaller timescales. The power generation costs are time-varying as well. For example, consider the power generation cost of a gas turbine. This parameter is mainly determined by the price of natural gas that is adjusted on an hourly basis. In this work, we assume the IES unit is operated in a market that is continuously evolving. The first unit dispatch problem is characterized by two time-varying boundary conditions, i.e., the load demand trajectory and the GT power generation costs. The former was obtained by scaling the values retrieved from the New York Independent System Operator—Load Data [46]. The adopted trajectory ranges between 51.9 MW_e and 88.9 MW_e over the 24 h dispatch period (purple line in Figure 17). As for the latter, a time-varying curve was obtained by scaling a 24 h Henry Hub natural gas price history [47] to mimic a representative day. The calculated trajectory ranges between USD 177.9/MWh and USD 191.4/MWh (blue triangles in Figure 18), where the average cost matches the Levelized Cost of Electricity (LCOE) of gas peaking units (USD 183.0/MWh) [48]. The other parameters, including the BOP generation cost, the electricity sales price, and the hydrogen sales price, are kept constant throughout the dispatch period (Table 4).

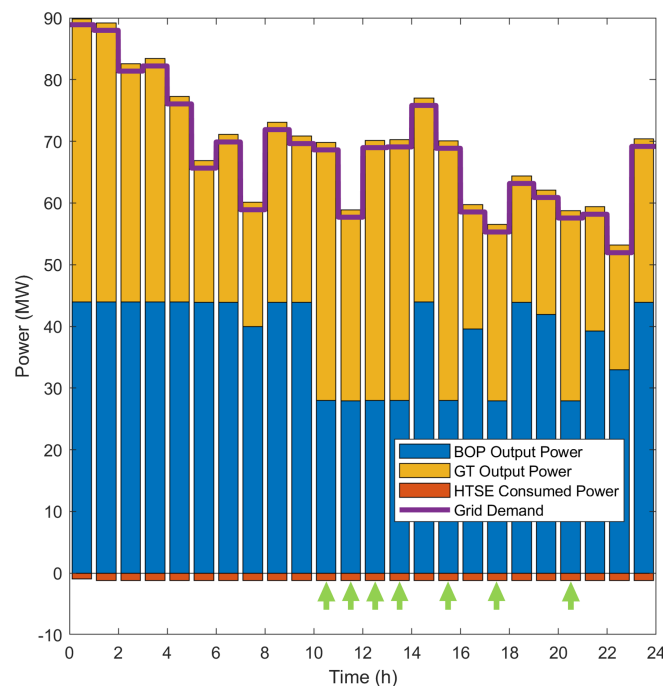


Figure 17. Solution of Unit Dispatch Problem 1 (time-varying GT generation cost) calculated by HERON-FARM scheme.

In Figure 17, the solution of the unit dispatch problem calculated by the HERON-FARM scheme is shown. The positive power outputs of BOP and GT and the negative power consumption of HTSE are plotted as stacked bars above and below the time axis, respectively. An explanation of the obtained results is provided here below:

1. Due to higher sales price, selling electricity is more profitable than selling hydrogen. Thus, the primary focus is the electrical power generation. Because of the insufficient electricity supply, the hydrogen production is interrupted at $t = 0$ h.

2. Since the sales price of hydrogen could cover the generation cost of the consumed electricity, HTSE tends to generate hydrogen at maximum rate given the steam and electrical power surplus.
3. During the hours when GT power generation costs are lower than those of BOP (10 h \leq t \leq 13 h, and t = 15 h, 17 h, 20 h, as marked by the green arrows), it is more profitable to generate electricity using GT, which results in the BOP producing at its minimum rate.

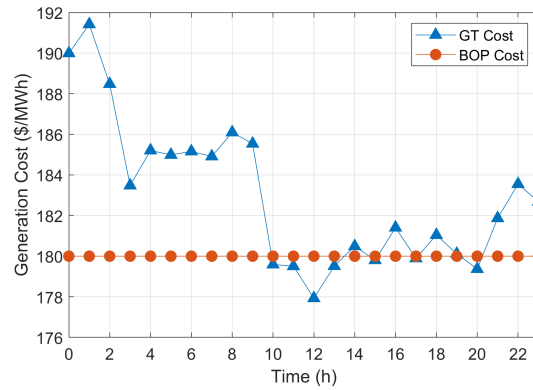


Figure 18. Trajectories for BOP and GT power generation costs throughout the unit dispatch period.

In Figure 19, the responses of output variables over the 24 h dispatch period are shown. Thanks to the adoption of FARM, the calculated solution is also feasible, and all the imposed constraints on output variables listed in Table 1 are met.

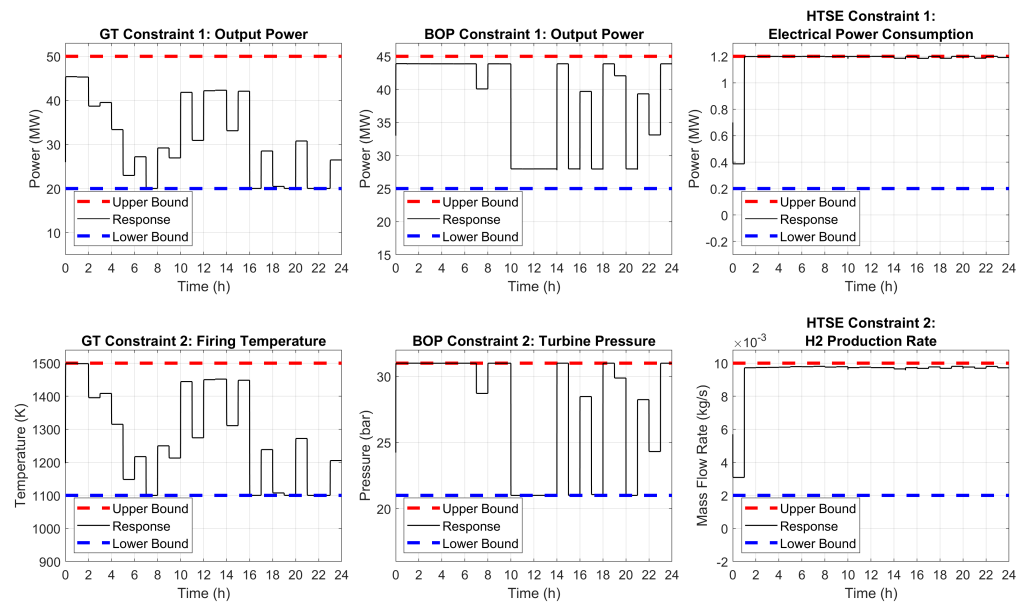


Figure 19. Responses of output variables for Unit Dispatch Problem 1.

The proposed scheme solves the unit dispatch problem through multiple iterations between FARM and HERON, as shown in Figure 20. In the first iteration, HERON provided FARM with the explicit constraint-compliant trajectories over the entire 24 h dispatch period (blue circles), and FARM returned to HERON the implicit constraint-compliant trajectories over the 24 h period (orange triangles) and the margins with respect to the bounds. In the second iteration, HERON re-optimized the explicit constraint-compliant trajectories based on FARM feedback. These trajectories were accepted as they matched the implicit constraint-compliant trajectories provided by FARM.

Table 4. Adopted economic parameters in two unit dispatch problems.

Parameter	Dispatch Problem 1	Dispatch Problem 2
BOP Generation Cost	USD 180.0/MWh	USD 180.0/MWh
GT Generation Cost	Time-varying USD 177.9–USD 191.4/MWh	Constant USD 200.0/MWh
Hydrogen Sales Price	Constant USD 14.0/kg (USD 216.0/MWh of consumed electricity)	Time-varying USD 13.0–USD 14.2/kg (USD 186.9–USD 221.2/MWh of consumed electricity)
Electricity Sales Price	USD 400.0/MWh	USD 400.0/MWh

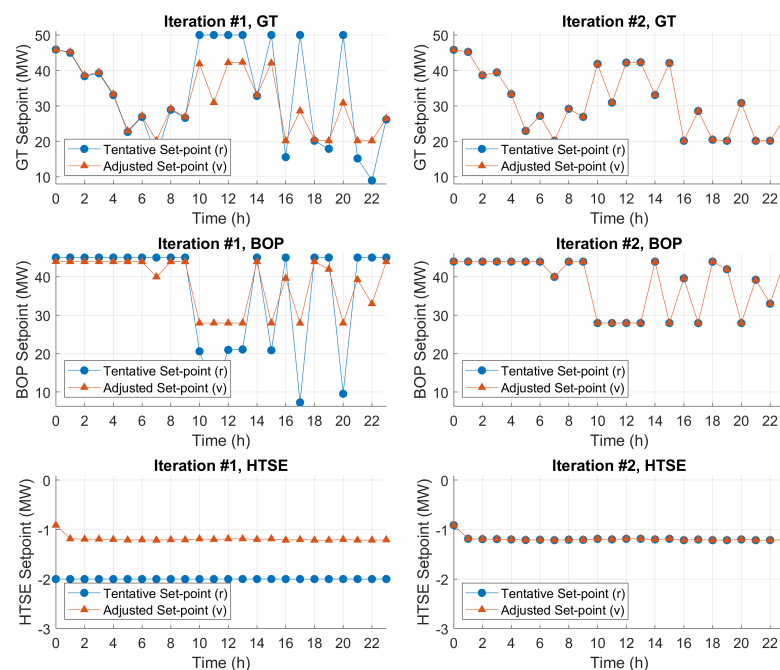


Figure 20. Iterative feedback between FARM and HERON for Unit Dispatch Problem 1.

6.3. Unit Dispatch Problem 2: Time-Varying Hydrogen Sales Price

The second unit dispatch problem differs from the first one in one of the time-varying boundary conditions; i.e., the sales price of hydrogen is characterized by a 24 h trajectory. Due to the lack of hydrogen prices on an hourly basis, this trajectory is obtained by scaling a random consecutive 24 h NYISO electricity price [49] to mimic a representative day. The final trajectory covers the range from USD 13.0/kg to USD 14.2/kg, as shown in Figure 21 (blue triangles). This range matches the hydrogen sales price (USD 13.0–USD 16.0/kg) suggested by the U.S. Department of Energy [50]. The other economic parameters, including the BOP and the GT power generation costs and electricity sales price are kept constant over the 24 h dispatch period (Table 4). The same electricity demand trajectory used in Unit Dispatch Problem 1 (Section 6.2) is adopted.

To compare the profit from selling hydrogen with the cost of electricity used in hydrogen production, GT and BOP power generation costs are also converted into equivalent hydrogen sales price as shown by the red and yellow circles in Figure 21. For instance, when the hydrogen sales price exceeds the threshold of USD 13.44/kg (equivalent to the GT generation cost of USD 200.0/MWh) during the periods of $0 \text{ h} \leq t \leq 11 \text{ h}$, $11 \text{ h} \leq t \leq 13 \text{ h}$, and $22 \text{ h} \leq t \leq 23 \text{ h}$, using the electricity generated by GT to supply the HTSE for hydrogen production is economically favorable, as the revenue from hydrogen sales surpasses the expenses in generating the consumed electricity via GT.

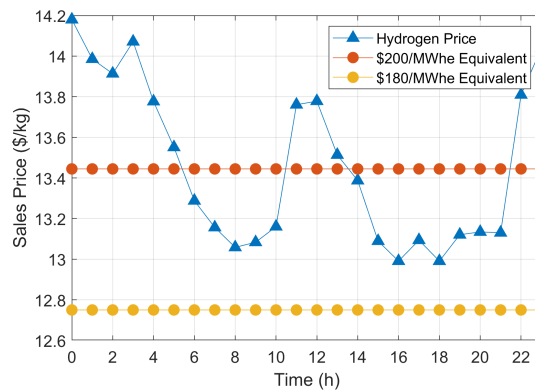


Figure 21. Hydrogen sales price trajectory throughout the unit dispatch period.

In Figure 22, the solution of the unit dispatch problem calculated by the HERON-FARM scheme is shown (the meaning of the color bars is consistent with Section 6.2). An explanation of the obtained results is provided here below:

1. When the revenue from hydrogen sales surpasses the expenses incurred in electrical power production by the GT and BOP ($0 \text{ h} \leq t \leq 5 \text{ h}$, $11 \text{ h} \leq t \leq 13 \text{ h}$, and $22 \text{ h} \leq t \leq 23 \text{ h}$, as marked by the green arrows), it is preferable to produce as much hydrogen as possible using the excess steam and electricity once grid demand has been met. As stressed in Section 6.2, generating electricity has higher priority due to the higher sales price, which suppressed the hydrogen production at $t = 0 \text{ h}$. BOP tends to generate at its maximum rate due to its lower generation cost than GT.
2. When the income from hydrogen sales does not exceed the costs associated with GT power generation:
 - (a) During high demand periods ($t = 6 \text{ h}$, $8 \text{ h} \leq t \leq 11 \text{ h}$, and $14 \text{ h} \leq t \leq 15 \text{ h}$, as marked by the orange arrows), the BOP output is maximized to supply to the grid, GT addresses the demand fluctuations, and HTSE operates at minimum rate to avoid further loss.
 - (b) During low demand periods ($t = 7 \text{ h}$, and $16 \text{ h} \leq t \leq 21 \text{ h}$, as marked by the blue arrows), the GT output is minimized, the BOP addresses demand fluctuations, and the HTSE operates at maximum rate to maximize the profit.

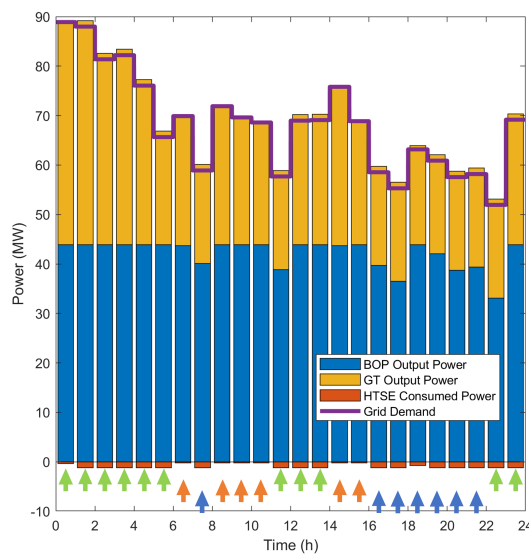


Figure 22. Solution of the Unit Dispatch Problem 2 (time-varying hydrogen sales price) calculated by HERON-FARM scheme.

6.4. Computational Efficiency of the Proposed Scheme for Solving the Unit Dispatch Problem

Having the FARM validator downstream of the unit dispatcher to assess the solution found inevitably complicates the optimization scheme's structure and increases the associated computational costs. It is the price paid to ensure feasible power trajectories that do not compromise the integrity of the components. It should be noted that the computational costs significantly depend on the load demand scenario under consideration. Specifically, the larger the expected power variations from the committed power plants, the more frequent the interventions by the FARM validator and the more iterations required to reach convergence. In many occasions, meeting the limits on production outputs and the corresponding rates of variation (i.e., explicit constraints) is enough to secure a feasible solution, i.e., accounting for the limits on other process variables is not strictly necessary. In these cases, FARM does not suggest any adjustments and simply approves the tentative set-point trajectories from the unit dispatcher.

In this section, the performance of the HERON-FARM scheme in terms of computational efficiency is evaluated by solving as a representative test case (the Unit Dispatch Problem 1 described in Section 6.2). The HERON dispatcher enforces three explicit constraints (y_1 , y_3 and y_5 in Table 1) in addition to meeting market demand; the FARM validator enforces three implicit constraints (y_2 , y_4 and y_6 in Table 1). The problem was solved multiple times using different configurations of the optimization algorithm. In particular, a sensitivity analysis on the sample time (T_s) adopted by the FARM validator is performed. The choice of the appropriate value for this parameter depends on the time constants that govern the dynamics of the studied system. For an IES unit with fast dynamics, smaller values should be used to ensure all dynamics are captured accurately. The risk of adopting larger values is that transients may occur between two consecutive samplings, leading to potential overshoots in process variables that go undetected. For these reasons, an exhaustive characterization of the system dynamics is recommended before solving the unit dispatch problem.

The computational cost is determined by the costs associated with completing the three tasks outlined below:

- Data-driven procedure for deriving the LPV model from simulation data during self-learning phase ("LPV Derivation" task)
The computational time associated with this task is affected by the complexity of the FMU or other high-fidelity model of the studied system used to generate the training data, i.e., the time for running simulations and collecting data may significantly differ. In this analysis, the computational time to complete this task was measured from loading the simulation data from hard drive to generating the database of state-space representation matrices.
- Solution to the unit dispatch problem by HERON, addressing only the explicit constraints during the dispatch phase ("HERON dispatcher" task)
The evaluation of the computational time to perform the tasks during the dispatch phase is more complicated. Since the HERON dispatcher and the FARM validator constitute a feedback loop (Figure 6), it is challenging to distinguish the individual computational time from the overall time of the dispatch phase. To this end, the computational time for the HERON dispatcher was estimated by solving the unit dispatch problem without using the FARM validator. The measured value serves as a baseline.
- Validation of the HERON-provided solution by the FARM validator during the dispatch phase ("FARM validator" task)
The computational time for the FARM validator was calculated by subtracting the HERON dispatcher contribution from the overall computational time for the dispatch phase. The values reported below include the set-point validation using the FARM validator and the prediction of the system response using the LPV model. The computational time may significantly increase if the capability to update the LPV model via FMU simulation is utilized during the dispatch phase (Section 3).

Simulations were run on a 2.7 GHz Intel Xeon Platinum 8168 CPU. The results of the analysis are reported in Table 5 and graphically represented in Figure 23a. It can be observed that the use of smaller sample times by FARM increases the computational costs. This trend can be easily explained. As mentioned above, smaller values for the sample time are used when the system dynamics is characterized by short time constants. To prevent undetected dangerous oscillations in the response of process variables, the FARM validator evaluates the system response over a greater number of discrete steps, causing the dimensions of the associated MOAS (Equation (12)) to grow proportionally.

Table 5. Computational costs for solving Unit Dispatch Problem 1 using different configurations of HERON-FARM scheme. All the reported values for the task duration are expressed in seconds.

Sample Time	LPV Derivation	HERON Dispatcher	FARM Validator
30.0	18.7 ± 0.2	8.0 ± 0.2	9.2 ± 0.5
60.0	11.0 ± 0.4	8.0 ± 0.2	3.6 ± 0.2
100.0	7.6 ± 0.2	8.0 ± 0.2	1.9 ± 0.2
150.0	6.4 ± 0.2	8.0 ± 0.2	1.4 ± 0.2
225.0	5.2 ± 0.1	8.0 ± 0.2	1.2 ± 0.2
300.0	4.7 ± 0.2	8.0 ± 0.2	1.0 ± 0.2

In Figure 23b, the contributions to the computational time for a representative case ($T_s = 60.0$ s) is shown. Solving the entire 24 h unit dispatch problem using the HERON-FARM scheme took approximately 23 s. This duration includes both the off-line computational time for characterizing the system dynamics (“LPV derivation” task) and the on-line computational time for solving the optimization problem (“HERON dispatcher” and “FARM validator” tasks). The efficiency of the HERON-FARM optimization scheme is determined solely by the online computational time. In this regard, the validator’s presence downstream of the unit dispatcher only increases the online computational time by less than 50%. Specifically, an additional 3.6 ± 0.2 s is required on top of the 8.0 ± 0.2 s taken by the standalone HERON dispatcher.

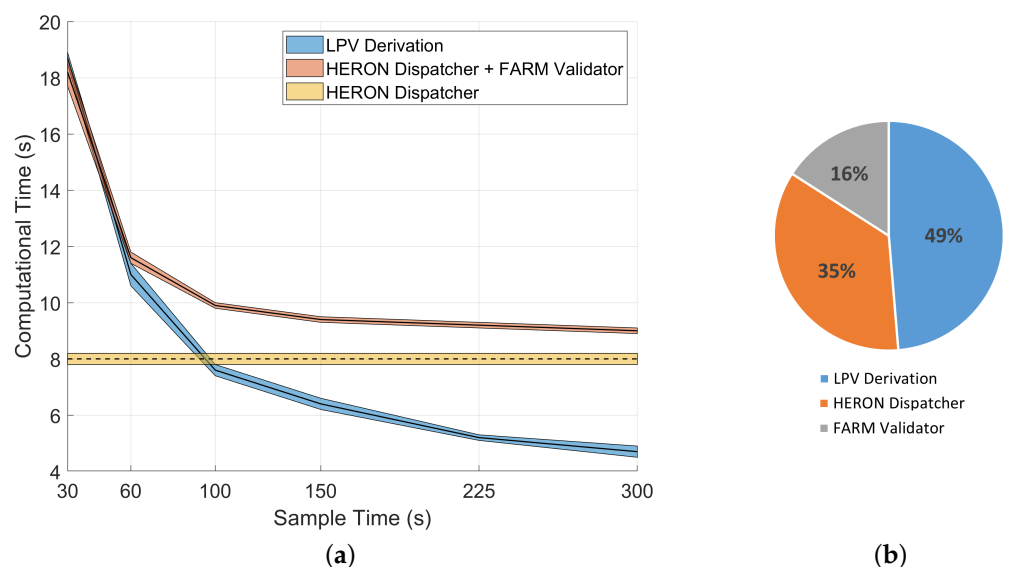


Figure 23. Efficiency of the HERON-FARM optimization scheme. The computational costs associated with different scheme configurations (with and without FARM validator and for various sample times) (a) and the contributions for a specific case ($T_s = 60.0$ s) (b) are shown.

7. Visualization Techniques for Displaying the Solution of the Unit Dispatch Problem

In addition to the algorithms aiding the solution of the unit dispatch problem, the FARM module provides tools for visualizing the response of monitored variables during transients. In the context of actual plant operations, “monitored variables” refer to relevant process variables that are either measured by dedicated sensors or predicted by appropriate models. Although this sophisticated framework is applicable to any type of generating units, its benefits are particularly evident when managing multi-asset systems like IES units. With the current techniques available, human operators may find it challenging to handle the large volumes of real-time data from the plant and power grid and to identify a solution minimizing the power generation costs and respecting all the limits on materials and components. These techniques can be used for both monitoring the evolution of process variables and assessing component conditions (Section 7.1) and to support the operator at adjusting the set-points to be fed to the controllers (Section 7.2).

7.1. Graphical Representation of the Normal Operation Region for Process Variables

In this section, two novel techniques for visualizing the responses of process variables are presented. Thanks to the capability of estimating in real-time the margins with respect to the operation limits, these tools might represent the first step towards the deployment of an Ecological Interface Design (EID) for IES units. An EID is a framework for creating human-machine interfaces for complex systems [51]. EIDs can be extremely beneficial in aiding operators by improving the situational awareness and supporting the decision-making process, especially in unfamiliar and unanticipated complicated situations. First, an EID provides a systematic view of the whole process instead of simple representation of individual process parameters and supports knowledge-based behavior of the operator. The second feature is a special kind of graphics that transfer simple mental operations to the level of perception. Implementation of such graphical patterns provides “visualization” of mental calculations and reduces cognitive workload [52]. A recent study demonstrated that EID interfaces ensure better control task performance and greater control stability [53]. As for the application to the nuclear industry, the U.S. Nuclear Regulatory Commission (U.S. NRC) hesitated to adopt the EID display for the nuclear power industry because of technical and research issues [54]. The tools proposed in this work will help addressing some of these issues, e.g., (1) inconsideration on the task and temporal-based displays, (2) information volume and density, (3) compatibility of user mental models and (4) training and qualification implication.

The former visualization technique is called the Dynamic Spider Chart and represents an extension of the Spider Chart (or Radar Chart) concept. A Spider Chart is a two-dimensional plot where three or more variables are represented on equiangular axes, starting from the same point by spokes [55]. The data on each spoke are scaled such that the limits are proportional to the length of the spokes. The multivariate data are represented by an irregular polygon, which is obtained by connecting the tips of adjacent spokes. These charts are mainly used to identify possible correlations, outliers, trade-offs, and several other comparative measures. In [56], an application of Spider Charts to the development of EIDs for nuclear power plant operation can be found. In this work, a dynamic version of the method for visualizing the IES unit response within the NOR during transients was developed. As mentioned in the Introduction, the NOR is defined by the ensemble of values that can be assumed by the process variables of an engineering system. A Dynamic Spider Chart is essentially an animated two-dimensional plot that can simultaneously display the evolution of multiple process variables using multiple axes. Any number of variables can be displayed at the same time as long as the readability of the chart is not compromised. Additionally, violations of operational constraints can be easily identified, which makes such a method suitable to make decisions about power transient planning. For demonstration purposes, the Unit Dispatch Problem 1 described in Section 6.2 was solved again but without coupling the FARM Validator. In Figure 24, the problem solution at time $t = 9$ h is displayed. Two concentric hexagons are drawn in black

color to represent the upper and the lower bounds of all variables, whereas the grey-shaded ring in between represents the NOR. As long as the constraints are met and the plant is operated in a safe and efficient manner, the blue-shaded, irregular polygon representing the current operating conditions will lie within the NOR bounds. Finally, the pink and the green contours represent the maximum and the minimum values ever assumed by the process variables during the considered transient, respectively. Without the enforcement of implicit constraints during the dispatch, the process variables can assume any values, even undesired ones. The Dynamic Spider Chart marks historical violations of either the upper or lower bounds by highlighting the names of the constrained variables in yellow. As shown in Figure 24, in the solved problem, the limits on the GT firing temperature (GT_Firing_Temp) and on the BOP inlet pressure (BOP_Inlet_Pressure) were exceeded before time $t = 9$ h. In case both the bounds were violated, the variable name would be highlighted in red.

The Dynamic Spider Chart provides an exhaustive overview of all the monitored process variables, since it allows depicting as many axes as desired. At the same time, tracking the evolution of the irregular polygon is not straightforward, i.e., only past constraint violations can be displayed. To assist the operator in closely monitoring the conditions of a specific portion of the plant or a specific piece of equipment, a second visualization technique that tracks the recent history of a subset of three variables was developed. It is called the Phase Space Diagram. As an extension of classic 3D trace plots, in this representation, the dynamic response of a IES unit is displayed by a 3D curve, where each point represents the values of three selected process variables. In Figure 25, the application of this method to the same transient described for the demonstration of the Dynamic Spider Chart is shown. The NOR is represented by the grey-colored parallelepiped, which is enclosed by six planes representing the lower and the upper bounds on the three selected variables. The thickness and the color of the piece-wise curve that represents the response of the IES unit indicate the progression of time. Specifically, the line is thin and pale blue at the start of the transient, and it gradually thickens and darkens to blue as it approaches the current time. The gold star is a marker indicating the current position in the phase space. Similar to the Dynamic Spider Chart, the Phase Space Diagram also assists in quickly identifying constraint violations. In this example, the constraint violations occur in correspondence of the front and the top faces, and the corresponding segments of the trajectory will be highlighted in red.

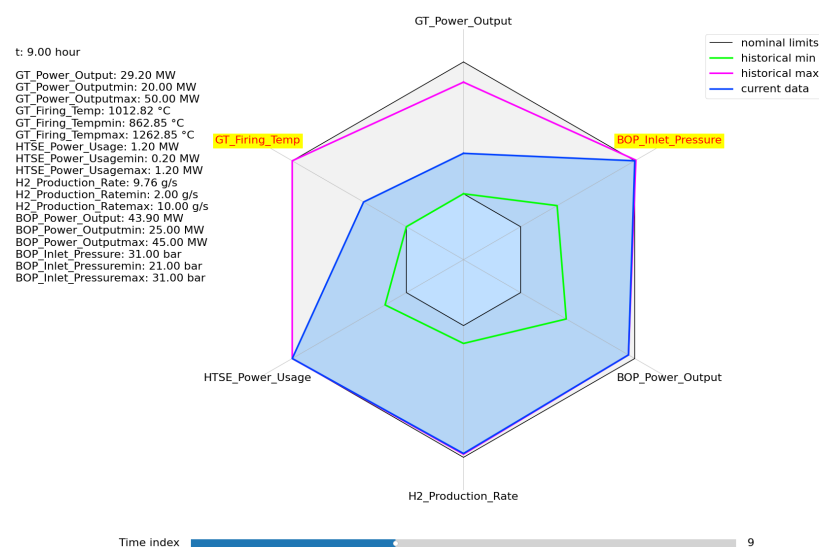


Figure 24. Dynamic Spider Chart displaying the NOR corresponding to Unit Dispatch Problem 1 at time $t = 9$ h.

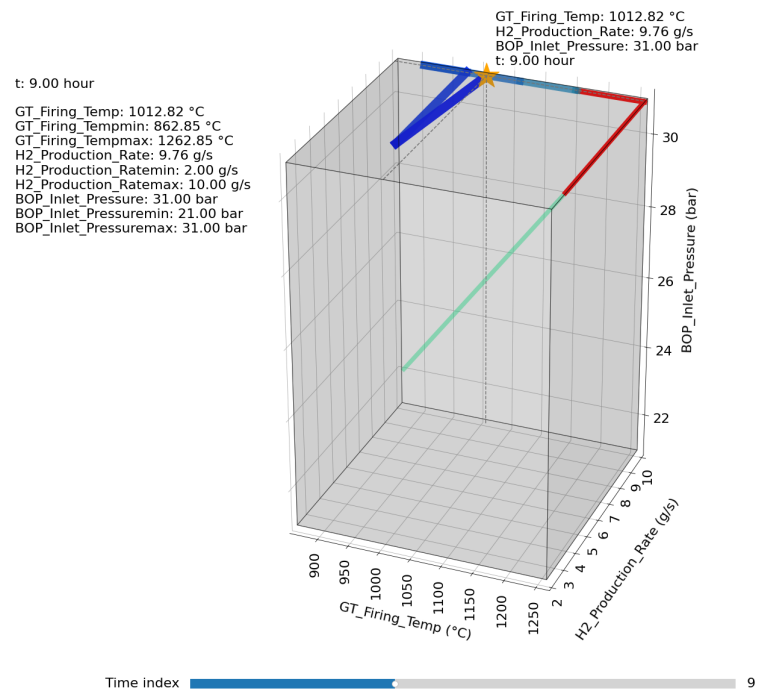


Figure 25. Phase Space Diagram displaying the NOR corresponding to Unit Dispatch Problem 1 at time $t = 9$ h. The evolution history of the system response is shown by the piece-wise curve from thin to thick, and the current system response is shown by the gold star.

7.2. Graphical Representation of the Admissible Region for the Power Set-Point Trajectories

In addition to methods for visualizing the NOR, the FARM module provides a tool displaying the Admissible Region, i.e., the set of acceptable values for the power set-points. This multi-dimensional region (also known as the MOAS) is calculated by the CG algorithm embedded into FARM (Equation (13)) when validating the tentative set-points. Between the NOR and the Admissible Region, there is a direct connection as the CG translates the limits on the process variables (i.e., NOR bounds) into limits on the power set-points (Section 4.1). From this standpoint, the visualization of the Admissible Region might play an important role for the safe operation of an IES unit. Basically, these translated limits are suggestions that an operator can make a direct use of. By displaying both the tentative (\vec{r}) and the adjusted (\vec{v}) set-point trajectories, the operator can identify the gap between the power transient demanded to the plant and the one recommended by the algorithm, which ensures that none of the constrained variables exceed the NOR bounds. In Figure 26, the three-dimensional region rendered in yellow represents the Admissible Region calculated during the solution of Unit Dispatch Problem 1 (Section 6.2) at time $t = 9$ h (first iteration). It is worth stressing that the volume representing the Admissible Region is not a parallelepiped as the NOR shown in Figure 25. As described in Section 4.2, the MOAS is defined by a much larger number of constraints since its task is to enforce the operational constraints over the entire prediction horizon. For this reason, the Admissible Region is represented by a volume in the 3D space demarcated by a large number of planes. In addition, while the size and the shape of the NOR remain constant once the bounds on the constrained variables are established, the size and the shape of the Admissible Region change at every time step of the unit dispatch problem. In Figure 26, both the tentative (blue dot) and the adjusted (orange triangle) set-points are displayed, along with the recommended values. Since this view corresponds to the first iteration of the optimization process and the implicit constraints are not enforced yet, the blue dot lies outside the region. In these cases, the orange triangle lies on the border of the region, as it represents the closest power scenario to the original one that remains within the operational limits. Mathematically, it constitutes

the closest approximation of the original trajectory that meets all the imposed constraints (it is calculated by minimizing the distance from \vec{r} (Equation (14))).

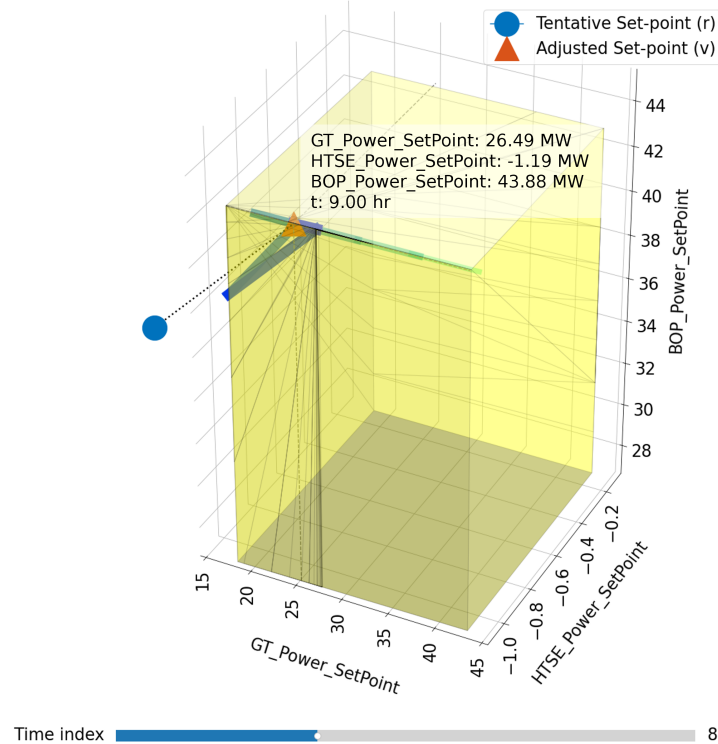


Figure 26. Representation of the Admissible Region calculated by the CG algorithm during the solution of Unit Dispatch Problem 1 at time $t = 9$ h (first iteration).

8. Conclusions

The deployment of advanced energy systems as IES units along with the adoption of new operational paradigms require dedicated tools. Thermomechanical loads induced by frequent power adjustments can accelerate the wear and tear of components. If a unit is flexibly operated without respecting operational limits on materials, the risk of failures of expensive components will eventually increase along with the frequency of maintenance interventions, nullifying the additional profits ensured by flexible operation. The FARM module was developed to enable existing optimization algorithms to identify solutions to the unit dispatch problem that are both economically favorable and technologically sustainable. The main capabilities are summarized below:

- Design of a dedicated tool for solving the unit dispatch problem for advanced energy systems
When solving the unit dispatch problem for an advanced energy system as an IES unit, it is necessary to consider the operational limits on the power outputs and the main process variables for all the subsystems that constitute the multi-asset system. The iterative dispatcher–validator scheme presented in this work permits addressing all the imposed constraints without excessively increasing the computational costs. The scheme was tested on IES unit constituted by a BOP, a GT and an HTSE that produce both electrical power and hydrogen. One of the key reasons for the inherent efficiency of this scheme is that the validator intervention is not always required. Often, meeting the limits on power outputs and the corresponding rates of variation is sufficient to obtain a feasible solution. In these cases, there are no iterations, and the validator simply approves the trajectories estimated by the dispatcher.
- Computationally-efficient enforcement of constraints on process variables
The central premise of the developed scheme is that attempting to solve the optimization problem by simultaneously addressing both explicit and implicit constraints

would result in prohibitively high computational costs. A more efficient approach consists of tackling the problem as a two-phase process. Traditionally used as a control algorithm, the CG was proposed as a tool to favor the convergence to the feasible solution maximizing the cost function. Nonetheless, when too many redundant constraints are provided, the QP solver is challenged, and a sub-optimal solution is occasionally returned. To address this issue, the CG algorithm was modified by adding the convex hull method that permits dropping redundant constraints.

- **Development of a DT of the studied system**
When solving the unit dispatch problem, a DT that mimics the behavior of the studied system is necessary to avoid power transients that could compromise the integrity of the components. In this work, a dedicated procedure for building a DT was embedded into the FARM module. It is constituted by an algorithm for selecting the state variables, an algorithm for deriving the state-space matrices from tracked variables and the protocol for updating the model. The tool is quite versatile and can also be used for other real-time applications. The adopted data-driven approach relies on sampled data to derive the state-space model of the system. In this work, simulation data were utilized, but sensor readings from an actual system could also be used. This capability is crucial to account for the evolution of physical parameters during operation that might make the model outdated over time. In this way, the latest, most accurate description of the system dynamics is always available.
- **Enhanced monitoring capabilities through innovative visualization techniques**
To ensure the desired performance of a complex multi-asset system without damaging system components, an efficient real-time monitoring tool is necessary. Currently available techniques need to be improved to handle the mass of real-time data from the plant and from the power grid and determine a solution minimizing the power generation costs and respecting all the limits on materials and components. In this work, two tools for displaying the response of the process variables of interest during system operation and another tool aiding the operator at making decisions were developed and tested.

Author Contributions: Conceptualization: R.P., H.W. and A.A.; Methodology: R.P. and H.W.; Software: H.W., A.A. and P.W.T.; Validation: H.W. and A.A.; Formal analysis: R.P., H.W. and A.A.; Investigation: R.P., H.W., A.A. and P.W.T.; Resources: H.W., A.A., A.S.E. and P.W.T.; Data curation: H.W. and A.A.; Writing—original draft preparation: R.P. and H.W.; Writing—review and editing: R.P., H.W., P.W.T. and A.S.E.; Visualization: T.W.E. and H.W.; Supervision: R.B.V.; Project administration: R.B.V. and A.S.E.; Funding acquisition: R.B.V. and A.S.E. All authors have read and agreed to the published version of the manuscript.

Funding: This work and the APC were supported by the U.S. Department of Energy, Office of Nuclear Energy. The submitted manuscript was created by UChicago Argonne, LLC, operator of Argonne National Laboratory. Argonne, a DOE Office of Science laboratory, is operated under contract DE-AC02-06CH11357.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request. The examples for the algorithms discussed in this work are available on GitHub at <https://github.com/idaholab/raven> (accessed on 12 June 2024).

Acknowledgments: The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan at <http://energy.gov/downloads/doe-public-access-plan> (accessed on 12 June 2024).

Conflicts of Interest: All the authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BOP	Balance of Plant
CG	Command Governor
DAE	Differential-Algebraic Equations
DP	Dynamic Programming
DT	Digital Twin
DMDc	Dynamic Mode Decomposition with Control
EID	Ecological Interface Design
FARM	Feasible Actuator Range Modifier
FMI	Functional Mock-up Interface
FMU	Functional Mock-up Unit
FORCE	Framework for Optimization of Resources and Economics
GT	Gas Turbine
GUI	Graphical User Interface
HERON	Holistic Energy Resource Optimization Network
HPT	High Pressure Turbine
HTSE	High-Temperature Steam Electrolysis
IES	Integrated Energy System
IRR	Internal Rate of Return
LCOE	Levelized Cost of Electricity
LPT	Low Pressure Turbine
LPV	Linear Parameter-Varying
LTI	Linear Time-Invariant
MSE	Mean Square Error
MILP	Mixed-Integer Linear Programming
MIMO	Multiple Input Multiple Output
MOAS	Maximal Output Admissible Set
NOR	Normal Operation Region
NPV	Net Present Value
NRC	Nuclear Regulatory Commission
NT	Nitrogen Turbine
PI	Profitability Index
PID	Proportional Integral Derivative
PWR	Pressurized Water Reactor
QP	Quadratic Programming
RES	Renewable Energy Sources
RFE	Recursive Feature Elimination
SVD	Singular Value Decomposition
TEAL	Tool for Economic Analysis
TES	Thermal Energy Storage
UQ	Uncertainty Quantification

References

- Javanshir, N.; Syri, S. Techno-Economic Analysis of a Highly Renewable and Electrified District Heating Network Operating in the Balancing Markets. *Energies* **2023**, *16*, 8117. [[CrossRef](#)]
- Nikolewski, R. How hard is it to develop California's electric grid of the future? Like repairing a car while driving. *Los Angeles Times*, 29 January 2024.
- Mulder, S.; Klein, S. Techno-Economic Comparison of Electricity Storage Options in a Fully Renewable Energy System. *Energies* **2024**, *17*, 1084. [[CrossRef](#)]
- Ponciroli, R.; Wang, Y.; Zhou, Z.; Botterud, A.; Jenkins, J.; Vilim, R.B.; Ganda, F. Profitability Evaluation of Load-Following Nuclear Units with Physics-Induced Operational Constraints. *Nucl. Technol.* **2017**, *200*, 189–207. [[CrossRef](#)]
- Vaglio-Gaudard, C.; Dominguez-Bautista, M.D.; Frignani, M.; Fütterer, M.; Goicea, A.; Hanus, E.; Hollands, T.; Lombardo, C.; Lorenzi, S.; Miss, J.; et al. The TANDEM Euratom project: Context, objectives and workplan. *Nucl. Eng. Technol.* **2024**, *56*, 993–1001. [[CrossRef](#)]
- Wu, X.; Dai, H.; Zhang, N.; Kong, W. Development Outlook of Integrated Energy System in China. In Proceedings of the 6th International Conference on Power and Renewable Energy (ICPRE), Shanghai, China, 17–20 September 2021.

7. Xu, Y.; Zhang, W.; Liu, W. Distributed dynamic programming-based approach for economic dispatch in smart grids. *IEEE Trans. Ind. Inform.* **2014**, *11*, 166–175. [[CrossRef](#)]
8. Feng, Z.K.; Niu, W.J.; Wang, W.C.; Zhou, J.Z.; Cheng, C.T. A mixed integer linear programming model for unit commitment of thermal plants with peak shaving operation aspect in regional power grid lack of flexible hydropower energy. *Energy* **2019**, *175*, 618–629. [[CrossRef](#)]
9. Ganda, F.; Kim, T.K.; Taiwo, T.A.; Hansen, J.; Wigeland, R. *Economic Evaluation of Promising Options*; FCRD-FCO-2015-000013; Argonne National Laboratory: Lemont, IL, USA, 2015.
10. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *Trans. Asme J. Basic Eng.* **1960**, *82*, 35–45. [[CrossRef](#)]
11. Garone, E.; Di Cairano, S.; Kolmanovsky, I. Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica* **2017**, *75*, 306–328. [[CrossRef](#)]
12. Bemporad, A.; Mosca, E. Constraint fulfilment in control systems via predictive reference management. In Proceedings of the 1994 33rd IEEE Conference on Decision and Control, Lake Buena Vista, FL, USA, 14–16 December 1994.
13. Passerini, S.; Ponciroli, R.; Vilim, R.B. Impact of Active Control on Passive Safety Response Characteristics of Sodium-Cooled Fast Reactors: I—Theoretical Background. *Nucl. Technol.* **2017**, *199*, 1–15. [[CrossRef](#)]
14. Wang, H.; Ponciroli, R.; Elmer, T.; Ley, H.; Dave, A.J.; Vilim, R.B. Quantitative Estimation and Visualization of the Normal Operation Region for Power Transient Planning. In Proceedings of the 13th Nuclear Plant Instrumentation, Control & Human-Machine Interface Technologies, Knoxville, TN, USA, 15–20 July 2023.
15. Bemporad, A. Reference Governor for Constrained Nonlinear Systems. *IEEE Trans. Autom. Control* **1998**, *43*, 415–419. [[CrossRef](#)]
16. Bragg-Sitton, S.M.; Boardman, R.D.; Rabiti, C.; O’Brien, J. Reimagining future energy systems: Overview of the US program to maximize energy utilization via integrated nuclear-renewable energy systems. *Int. J. Energy Res.* **2020**, *44*, 8156–8169. [[CrossRef](#)]
17. Force of Integrated Energy Systems. Available online: <https://ies.inl.gov/SitePages/FORCE.aspx> (accessed on 28 July 2022).
18. RAVEN: Risk Analysis Virtual Environment. Available online: <https://github.com/idaholab/raven> (accessed on 12 June 2024).
19. Alfonsi, A.; Rabiti, C.; Mandelli, D.; Cogliati, J.; Kinoshita, R.; Naviglio, A. RAVEN and dynamic probabilistic risk assessment: Software overview. In Proceedings of the ESREL European Safety and Reliability Conference, Wroclaw, Poland, 14–18 September 2014.
20. Chen, J.; Rabiti, C. Synthetic wind speed scenarios generation for probabilistic analysis of hybrid energy systems. *Energy* **2017**, *120*, 507–517. [[CrossRef](#)]
21. HYBRID: A Modeling Toolset to Assess the Integration and Economic Viability of Integrated Energy Systems. Available online: <https://github.com/idaholab/HYBRID> (accessed on 12 June 2024).
22. Garcia, H.E.; Chen, J.; Kim, J.S.; Vilim, R.B.; Binder, W.R.; Sitton, S.M.B.; Boardman, R.D.; McKellar, M.G.; Paredis, C.J. Dynamic performance analysis of two regional nuclear hybrid energy systems. *Energy* **2016**, *107*, 234–258. [[CrossRef](#)]
23. HERON: Holistic Energy Resource Optimization Network. Available online: <https://github.com/idaholab/HERON> (accessed on 12 June 2024).
24. Talbot, P.W.; Gairola, A.; Frick, K.; Rabiti, C. HERON As a Tool for Light Water Reactor Market Interaction in a Deregulated Market. In Proceedings of the ASME Power Conference, Virtual, Online, 4–5 August 2020.
25. TEAL: Tool for Economic Analysis. Available online: <https://github.com/idaholab/TEAL> (accessed on 12 June 2024).
26. Epiney, A.; Rabiti, C.; Talbot, P.W.; Alfonsi, A. Economic analysis of a nuclear hybrid energy system in a stochastic environment including wind turbines in an electricity grid. *Appl. Energy* **2020**, *260*, 114227. [[CrossRef](#)]
27. FARM: Feasible Actuator Range Modifier Plugin for RAVEN. Available online: <https://github.com/Argonne-National-Laboratory/FARM> (accessed on 29 May 2024).
28. Bachnas, A.A.; Tóth, R.; Ludlage, J.H.A.; Mesbah, A. A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study. *J. Process Control* **2014**, *24*, 272–285. [[CrossRef](#)]
29. Kunkel, P.; Mehrmann, V. *Differential-Algebraic Equations: Analysis and Numerical Solution*; European Mathematical Society: Helsinki, Finland, 2006; Volume 2.
30. Lawrence Berkeley National Laboratory. *FMU Export of a Python-Driven Simulation Program*; Release 1.0.0rc15; Lawrence Berkeley National Laboratory: Berkeley, CA, USA, 2018.
31. Fiorina, C.; Habtemariam, N.; Lorenzi, S.; Scolaro, A.; Guilbaud, T.; Alfonsi, A. The Functional Mock-Up Interface as a Unified Framework to Enable Multi-Scale, Multi-Fidelity and Control-Oriented Simulations of Nuclear Reactors. In Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Niagara Falls, ON, Canada, 13–17 August 2023.
32. Casavola, A.; Garone, E.; Tedesco, F. Distributed reference management strategies for a networked water distribution system. In Proceedings of the 18th IFAC World Congress, Milan, Italy, 28 August–2 September 2011.
33. Tedesco, F.; Casavola, A.; Garone, E. A distributed parallel command governor strategy for the coordination of multi-agent networked systems. In Proceedings of the 4th IFAC Nonlinear Model Predictive Control Conference, Noordwijkerhout, The Netherlands, 23–27 August 2012.
34. Garcia, C.; Prett, D.; Morari, M. Model predictive control: Theory and practice—A survey. *Automatica* **1989**, *25*, 335–348. [[CrossRef](#)]
35. Glicksman, A.M. *An Introduction to Linear Programming and the Theory of Games*; Dover Publications, INC: Mineola, NY, USA, 2001.
36. cvxpy: A Python-Embedded Modeling Language for Convex Optimization Problems. Available online: <https://www.cvxpy.org/> (accessed on 25 July 2022).

37. Rockafellar, R.T. *Convex Analysis*; Princeton Mathematical Series; Princeton University Press: Princeton, NJ, USA, 1970; Volume 28.
38. scipy/scipy: SciPy 1.9. 0. Available online: <https://doi.org/10.5281/zenodo.6940349> (accessed on 20 September 2023).
39. Proctor, J.L.; Brunton, S.L.; Kutz, J.N. Dynamic Mode Decomposition with Control. *SIAM J. Appl. Dyn. Syst.* **2016**, *15*, 142–161. [[CrossRef](#)]
40. Moore, E.H. On the Reciprocal of the General Algebraic Matrix. *Bull. Am. Math. Soc.* **1920**, *26*, 394–395.
41. Van Overschee, P.; De Moor, B. *Subspace Identification for Linear Systems: Theory—Implementation—Applications*, 1st ed.; Springer Science & Business Media: New York, NY, USA, 2012.
42. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene selection for cancer classification using support vector machines. *Mach. Learn.* **2002**, *46*, 389–422. [[CrossRef](#)]
43. Mullner, D. Modern Hierarchical, Agglomerative Clustering Algorithms. *arXiv* **2011**, arXiv:1109.2378v.
44. Flaticon.com: Vector Icons and Stickers. Available online: <https://www.flaticon.com/> (accessed on 19 February 2024).
45. Kim, J.S.; Boardman, R.D.; Bragg-Sitton, S.M. Dynamic performance analysis of a high-temperature steam electrolysis plant integrated within nuclear-renewable hybrid energy systems. *Appl. Energy* **2018**, *228*, 2090–2110. [[CrossRef](#)]
46. New York Independent System Operator: Energy Market & Operational Data—Load Data. Available online: <https://www.nyiso.com/load-data> (accessed on 7 December 2023).
47. Business Insider: Natural Gas Price Today. Available online: <https://markets.businessinsider.com/commodities/natural-gas-price> (accessed on 2 January 2024).
48. Lazard Ltd. Lazard’s Levelized Cost of Energy Analysis—Version 16.0. Available online: <https://www.lazard.com/> (accessed on 4 January 2024).
49. New York Independent System Operator: Energy Market & Operational Data—Pricing Data. Available online: <https://www.nyiso.com/energy-market-operational-data> (accessed on 7 December 2023).
50. U.S. Department of Energy: Hydrogen Fuel Cell Vehicles. Available online: <https://www.energy.gov/energysaver/hydrogen-fuel-cell-vehicles> (accessed on 10 January 2024).
51. Vicente, K.J.; Rasmussen, J. Ecological Interface Design: Theoretical Foundations. *IEEE Trans. Syst. Man Cybern.* **1992**, *22*, 589–606. [[CrossRef](#)]
52. Anokhin, A.; Ivkin, A.; Dorokhov, S. Application of ecological interface design in nuclear power plant (NPP) operator support system. *Nucl. Eng. Technol.* **2018**, *50*, 619–626. [[CrossRef](#)]
53. Carrasco, C.; Jamieson, G.A.; St-Cyr, O. Revisiting three ecological interface design experiments to investigate performance and control stability effects under normal conditions. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), San Diego, CA, USA, 5–8 October 2014.
54. Kim, S.K.; Suh, S.M.; Jang, G.S.; Hong, S.K.; Park, J.C. Empirical research on an ecological interface design for improving situation awareness of operators in an advanced control room. *Nucl. Eng. Des.* **2012**, *253*, 226–237. [[CrossRef](#)]
55. Porter, M.M.; Niksiar, P. Multidimensional mechanics: Performance mapping of natural biological systems using permuted radar charts. *PLoS ONE* **2018**, *13*, e0204309. [[CrossRef](#)] [[PubMed](#)]
56. Rejas, L.; Parrado, I.; Fernandez, S.; Ortega, F. Operating displays e new concept: High performance displays. In Proceedings of the 10th Nuclear Plant Instrumentation, Control & Human-Machine Interface Technologies, San Francisco, CA, USA, 11–15 June 2017.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.