









Dynamics-incorporated Modeling Framework for Stability Constrained Scheduling Under High-penetration of Renewable Energy

Jinning Wang , Graduate Student Member, IEEE, Fangxing Li , Fellow, IEEE, Xin Fang , Senior Member, IEEE, Hantao Cui , Senior Member, IEEE, Buxin She , Member, IEEE, Hang Shuai , Member, IEEE, Qiwei Zhang , Member, IEEE, Kevin Tomsovic , Fellow, IEEE.

Abstract—In this paper, a modularized modeling framework is designed to enable a dynamics-incorporated power system scheduling under high-penetration of renewable energy. This unique framework incorporates an adapted hybrid symbolic-numeric approach to scheduling models, effectively bridging the gap between device- and system-level optimization models and streamlining the scheduling modeling effort. The adaptability of the proposed framework stems from four key aspects: extensible scheduling formulations through modeling blocks, scalable performance via effective vectorization and sparsity-aware techniques, compatible data structure aligned with dynamic simulators by common power flow data, and interoperable dynamic interface for bi-direction data exchange between steady-state generation scheduling and time-domain dynamic simulation. Through extensive benchmarks with various usage scenarios, the framework's accuracy and scalability are validated. The case studies also demonstrate the efficient interoperation of generation scheduling and dynamics, significantly reducing the modeling conversion work in stability-constrained grid operation towards high-penetration of renewable energy.

Index Terms—Power system scheduling, symbolic modeling, stability constraints, open-source tool, power system digital twin, high-penetration renewable energy.

I. INTRODUCTION

THE societal goals of the decarbonized power grid drive the increase in renewable penetration. As a result, various renewable energy resources and dedicated management techniques are needed. To facilitate research and development needs, CURENT Large-scale Testbed (LTB) platform was developed as a virtual power grid for researchers to prototype and validate their algorithms [1]. This LTB platform includes a dynamics simulator ANDES [2], a distributed messaging environment DiME, and a geographical visualizer AGVis [3] to enable dynamic device modeling and intuitive understanding of grid status.

Jinning Wang, Fangxing Li, Hang Shuai, and Kevin Tomsovic are with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996 USA.

Xin Fang is with the Department of Electrical and Computer Engineering, Mississippi State University, Starkville, MS 39762, USA.

Hantao Cui is with the Department of Electrical and Computer Engineering, NC State University, Raleigh, NC 27606.

Buxin She is with Pacific Northwest National Laboratory, Richland, WA 99352, USA.

Qiwei Zhang is with the Department of Environmental Health and Eng., Johns Hopkins University, Baltimore, MD 21205.

Corresponding author: F. Li, email: fli6@utk.edu.

As the power grid evolves to integrate a high share of renewable energy sources, the complexity of scheduling operations increases, demanding more advanced tools and methodologies for efficient generation scheduling and production cost modeling [4]. Optimal Power Flow (OPF) becomes crucial in this context, serving as a fundamental scheduling scheme to address the intricate dynamics of production cost modeling. The research community has responded to these challenges by developing a variety of OPF tools. As the basic scheduling scheme for the power market, OPF has been well-addressed by the research community and led to various tools. MATPOWER, an extensive and rigorous OPF solution, was developed and released as an open-source tool [5]. Later, as Python gained more popularity in the scientific computation community, MATPOWER was ported to Python as PYPOWER [6]. Pandapower was developed to cover distribution networks with improved performance using the just-in-time compilation technique [7]. To include multi-period operation and planning, along with coupled multi-energy systems, PyPSA is developed and validated [8]. Later, in Julia, a flexible modeling framework, Sienna, formally known as SIIP, was developed to facilitate the modeling and simulation of energy systems [9], [10]. As open-source projects, the aforementioned tools benefit the research community by facilitating various usage scenarios and further development. In addition to the research tool development, the stability-constrained scheduling problem has drawn attention. In low-inertia systems, a locational rate-of-change-of-the-frequency (RoCoF) constrained unit-commitment model ensures frequency stability under contingencies by embedding frequency change limits and leveraging virtual inertia to support the frequency [11]. For AC/DC hybrid grids, a short-term voltage stability-constrained unit-commitment model mitigates voltage instability by retaining dynamic reactive power resources through a decomposition-based approach [12]. Furthermore, an ensemble sparse oblique regression tree approach enhances voltage stability in dispatch operations, converting stability margins into interpretable rules within an optimal power flow model [13]. These strategies advance stability-constrained scheduling, accommodating the evolving needs of renewable-rich power systems.

The economic and reliable operation of the power grid requires an adaptable scheduling modeling framework to leverage grid flexibility resources. In this context, the scheduling

tool might not only need to collect dynamics information in optimization formulation, but also broadcast scheduling results to the dynamics engine for dynamics performance assessment. For example, variable generation providing deliverable secondary frequency regulation is developed and validated through scheduling-dynamics co-simulation [14]. Furthermore, the small real-time generation dispatch interval turned out to be inferior to adaptive frequency control [15]. In addition, electric vehicles that provide the deliverable provision of secondary frequency regulation are modeled using state space modeling and validated through a co-simulation of real-time economic dispatch and time-domain simulation [16]. Inverter-based resources (IBR) providing inertia support through virtual synchronous generator (VSG) control is developed and assessed in both economics and dynamics [17]–[19]. A unified framework is developed to incorporate a comprehensive set of stability constraints into power system optimization problems, where the constraints are formulated in the second-order cone form and validated through dynamics simulations [20]. In these studies, researchers made considerable manual efforts to implement the algorithms and verify the results in both scheduling and dynamic simulations.

As discussed above, the scheduling modeling challenges emerging today include: 1) incorporating new dispatchable devices and elements introduced by IBRs; 2) integrating dynamics information into scheduling to ensure stability; and 3) addressing modeling gaps between the device level and the system scheduling level. To address these challenges, this paper presents the development of an adaptable scheduling modeling framework, called AMS, which also complements the CURENT LTB with scheduling functionality, thereby enabling scheduling-dynamics co-simulation and facilitating the creation of a full-timescale digital twin for the power grid with significant renewable energy.

Unlike [2], which used a hybrid symbolic-numeric approach for the time-domain transient simulation, this work uses this hybrid symbolic-numeric approach for the generation scheduling optimization with dynamics stability constraints and interfaces. In transient stability simulation, the problem is typically formulated as a set of differential-algebraic equations. The modeling efforts focus on the device-level, such as synchronous generator GENROU and renewable generator REGCA. This work, however, extends this concept to a scheduling modeling framework primarily targeting the generation scheduling optimization problems. Compared to the transient stability modeling framework, this one not only involves device-level modeling but also scheduling-level formulation, such as IBRs with virtual synchronous generators providing virtual inertia services. This hybrid symbolic-numeric approach for the generation scheduling optimization provides unique interfaces with dynamics models and parameters which can facilitate stability-constrained optimization. Specifically, in transient stability simulation, the formulation is explicitly defined, and efforts are concentrated on device-level modeling. In contrast, scheduling requires both device-level modeling and experimentation with different formulations to achieve economic optimality. In comparison with tools for power flow analysis and transient stability simulation, such

as PSD-BPA [21] and PSASP [22], this work mainly solves the generation scheduling optimization problems such as unit commitment and economic dispatch with stability constraints without modeling differential algebraic equations. This work models the stability constraints through the linear constraints of dynamics parameters such as inertia, damping, etc. The major contributions of this paper are summarized below.

- 1) Adapted hybrid symbolic-numeric approach for scheduling modeling and simulation. Unlike the approach used in transient stability modeling frameworks, the adapted approach bridges the modeling gap between the device-level and scheduling-level, and thus allows for rapid prototyping of IBRs grid services provision.
- 2) Proposed a modular design to streamline the scheduling modeling process, improving efficiency and ensuring compatibility with a wide range of IBRs and renewable energy sources.
- 3) Developed a dynamics interface to facilitate interoperation with transient simulators and development of transient stability-constrained scheduling algorithms.
- 4) Conducted case studies with various development scenarios to demonstrate the adaptability, including extensibility, scalability, compatibility, and interoperability.
- 5) Provided an open-source, ready-to-use implementation to enable further research and practical applications.

The remainder of this paper is organized as follows. Section II outlines the design philosophy and innovative aspects of the framework design; Section III elaborates on the modeling foundations of the framework and demonstrate the modeling of a frequency stability-constrained scheduling; Section IV examines the framework's adaptability, focusing on its extensibility, scalability, compatibility, and interoperability; Section V performs case studies to provide credibility and demonstrate the versatile applications including high-penetration IBR scenarios; and Section VI concludes this paper with potential future directions.

II. DESIGN PHILOSOPHY

This section first elaborates on the development considerations to ensure adaptability. Next, the adapted hybrid symbolic-numeric modeling approach is proposed for scheduling modeling.

A. Development Considerations

To achieve an adaptable scheduling modeling framework, the following issues should be considered.

1) *Compatibility*: Compatibility with external tools is essential for interoperation with power grids that integrate more emerging technologies. Specifically, the interoperation with transient simulators requires compatibility from file format to simulation manipulation. Thus, the scheduling tool is expected to have two features to support efficient interoperation with the transient engine. These features are a compatible file format and an automated data exchange application programming interface.

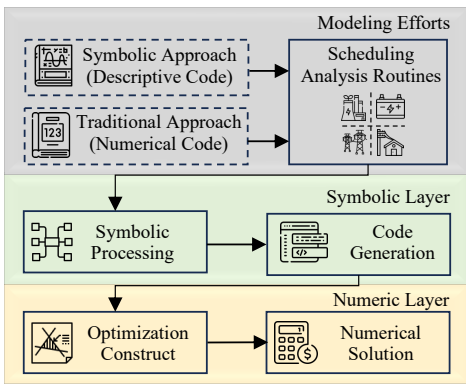


Fig. 1. Architecture of the adapted hybrid symbolic-numeric approach for scheduling modeling and simulation.

2) *Vectorization*: Vectorization is a common and useful parallelism technique for improving data manipulation efficiency. It enables efficient utilization of modernized computing hardware by continuously storing and accessing numerical data. In the scheduling formulation, the implementation of multiple data in one instruction [23] can be applied to data storage and the numerical optimization formulation.

3) *Modularity*: An appropriate modular design can be beneficial for the developer to use and maintain the tool in the long-run perspective. It can save considerable effort after the initial development stage, although it usually takes longer in the early stage due to the frequent necessary refactoring [24].

4) *Usability*: Numerical modeling can result in unignorable learning costs for development and maintenance due to its non-intuitive nature, although it is widely adopted in power system simulation and usually has good computation performance. The descriptive modeling method means using descriptive codes for modeling. This approach was introduced in transient stability simulation aiming at reducing development efforts for new dynamics devices [2].

B. Adapted Hybrid Symbolic-Numeric Approach

In our previous work ANDES [2], the hybrid symbolic-numeric modeling is proposed to reduce the dynamics device modeling efforts. In contrast, scheduling modeling involves not only emerging devices, but also their integration into new formulations. Thus, an adapted hybrid symbolic-numeric modeling approach is proposed to bridge the gap between device-level and scheduling-level modeling. It symbolizes device-level elements and renders them as accessible elements for problem formulation at the scheduling-level.

Figure 1 illustrates the architecture of the hybrid symbolic-numeric framework adapted for scheduling modeling. In the proposed approach, the gray block represents the hybrid modeling approach, while the green block represents the symbolic layer, and the yellow block represents the numeric layer. In the modeling part, the symbolic approach is primarily used to formulate the problem intuitively. In addition, the numerical approach helps in areas where the symbolic one is less applicable, such as in building system-level matrices. Upon modeling

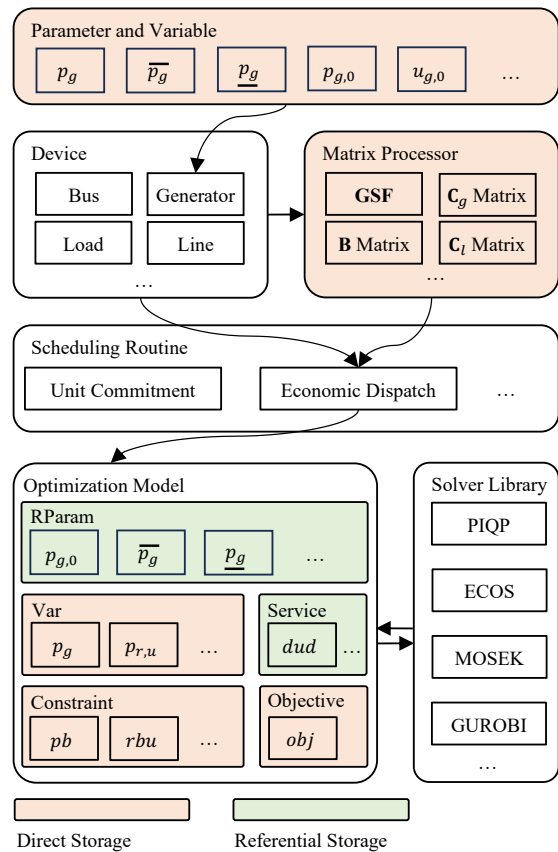


Fig. 2. Modularized modeling scheme for scheduling.

the problem, the symbolic layer parses the descriptive scheduling models with corresponding numerical counterparts. Lastly, the numeric layer solves the resulting numeric optimization model to perform the simulation. This approach is effective in integrating IBRs into scheduling models, ensuring compatible handling of their unique characteristics and dynamics. More details of the proposed framework will be further elaborated in the following section.

III. MODELING FRAMEWORK FOUNDATION FOR STABILITY-CONSTRAINED SCHEDULING

This section presents the foundation of the proposed scheduling modeling framework. It provides a brief overview of the modularized scheduling modeling scheme and then elaborates on the essential functionalities of the framework. Finally, it demonstrates the integration of virtual inertia scheduling (VIS) into real-time economic dispatch (RTED) using the proposed framework as an example of the stability constrained-scheduling.

A. Modularized Scheduling Modeling Scheme

Figure 2 illustrates the modularized scheduling modeling scheme of the framework. First, the parameters and variables are encapsulated in devices. Second, the system matrices are calculated and stored in the matrix processor. Following it, devices and system matrices are rendered as accessible elements

for scheduling routines. In the scheduling routines, optimization models are defined given the descriptive codes. Next, the framework yields a solvable numerical model through symbolic processing. Finally, the optimization model can be solved using the solver library.

B. Memory-Efficient Data Storage: Direct and Referential Methods

For minimal memory usage, two data storage patterns are developed for value providers, namely, direct storage and referential storage. Direct storage means that the object has its value attribute v for storing data. In contrast, referential storage means that the object has attributes that point to the data provider rather than having its value attribute. A lazy evaluation pattern is implemented in referential storage with a property method v , where the value is presented when calling and cleared afterward. This way avoids duplicate data storage because the referenced data are delivered upon calling only.

The structure of the descriptive scheduling modeling scheme is as follows. First, parameters and variables are defined within devices as the basic system data source. Second, the system matrices are calculated and stored within the system.

C. Device-Level Data Management

A power system *Device* provides source data and defines algebraic variables, and the attributes of devices are accessible by scheduling routines. The *Parameter*, sourced or calculated from input, describes a power system device. The *Variable*, particularly algebraic variable in the optimization problem, is a solvable element within a power system device. These are then symbolized by the symbolic processor to further describe the scheduling routine.

To alleviate development and maintenance efforts, AMS revised the basic device modeling elements, as described in Section III-A of [2], deprecating the expression code of algebraic variables and state variables e_str . Additionally, the defined topology data can be easily reused in scheduling routine. Consequently, development efforts at the device level focus on grid operational data, such as generation costs, reserve requirements, and load profile.

D. Efficient Handling of System Matrices in Scheduling

In scheduling formulation, system matrices refer to the matrices commonly used by multiple scheduling routines, such as generation shift factors **GSF**, generator connection matrix **C_g**, susceptance matrix **B**, and load connection matrix **C_l**. These system matrices are calculated and stored in the *MatProcessor*, and it is similar to a *Device* where its attributes are also accessible by scheduling routines.

As an exemplification of modular design, the encapsulation of *MatProcessor* can avoid repetitive computations because the system matrices are the same for different scheduling routines for one single case.

E. Assistive Services for Intricate Scheduling Formulations

An assistive procedure, *Service* is developed to describe formulations or math operations that are complicated for the symbolic methods. As symbolic operation brings about expensive computation costs on a large scale, complicated operations can be formulated with numerical coefficients. For example, the ramping constraints of a synchronous generator in a multi-period scheduling can be formulated in this manner. In multi-period economic scheduling, a service *RampSub* is introduced to build the coefficients matrix and thus to simplify the ramping constraints expression. Additionally, element-wise operations can be done similarly, for instance, by introducing a service *NumOpDual* to calculate the total zonal regulation up requirements through element-wise multiplication, complemented by output reshaping.

F. Symbolic Code Generation

The symbolic processing in AMS can be quite simplified leveraging existing optimization languages. As discussed in previous subsections, *Constraint* and *Objective* have the e_str attribute to store the descriptive code as strings. In the symbolic layer as depicted in Figure 1, the symbolic processor translates these strings using the operation of regular expressions [25]. This translation process generates executable code by substituting symbols with their corresponding value providers.

The symbolic processor, a predefined function within a scheduling routine, is inherited by all newly developed routines. The symbolic code generation is case-independent, ensuring efficient processing regardless of the size of the case.

G. Optimization Model Construction

Within a scheduling routine, all elements of an optimization model are first evaluated using the generated code and then assembled into a solvable optimization problem. An optimization model includes elements such as *RParam*, *Var*, *Service*, *Constraint*, and *Objective*. A *RParam* indicates the source model and elements. A *Var* defines an optimization variable in the size of the source model. A *Constraint* provides the descriptive expressions and constraint type. An *objective* stores the descriptive objective expressions and the type. After symbolic processing, all these elements are converted into executable code. By sequentially evaluating this code, the executable optimization counterpart is generated. Finally, these components are assembled into a solvable optimization problem.

H. Automated Documentation Generation

Auto-generated documentation is crucial for disseminating the source code and its clearly elaborated functionality.

In scheduling routines, human-readable documentation are essential for power system researchers to be aware of the formulations and methods that they are utilizing. AMS presents formulations by utilizing the tex_name symbol attribute of the modeling elements. Consequently, documentation tools can automatically generate detailed source files and render them in conjunction with modeling efforts.

I. Descriptive Modeling Instance for Stability-Constrained Scheduling

Power system stability is categorized into five categories: resonance stability, converter-driven stability, rotor angle stability, voltage stability, and frequency stability [26]. This subsection presents a frequency stability-constrained generation scheduling model with inertia and damping constraints. The goal is to maintain frequency nadir above the under-frequency load shedding thresholds and ensure the RoCoF follows grid standards after a N-1 generation trip contingencies. In the future, more stability indices such as short-circuit ratio can be investigated using the proposed framework.

$$\min \sum_{g \in \Omega_G} (c_2 P_g^2 + c_1 P_g + c_0 u_g + c_g^U P_{g,r}^U + c_g^D P_{g,r}^D) \quad (1)$$

s.t.

$$\Delta D^U = \sum_{g \in \Omega_G} P_{g,r}^U \quad (2)$$

$$P_{g,r}^U + P_g \leq \overline{P}_g, \forall g \in \Omega_G \quad (3)$$

Eqns. (1) - (3) describe a portion of the RTED scheduling. In these equations, the upper subscripts U and D represent RegUp and RegDn, respectively. The lower subscript g represents a generator and the lower subscript r represents regulation. The objective function (1) minimizes the total cost of power generation and regulation reserves. Eqn (2) describes the RegUp reserve requirement, while (3) imposes the physical limits on the generators.

Following the problem formulation, we can descriptively develop the RTED model. Listing 1 presents a simplified descriptive formulation of RTED as an example. The descriptive RTED model is conceptually divided into two main sections: the data section and the model section. The data section defines the relevant data and necessary services. *RParam* *pmax* defines the data source of the maximum generator output. *Service* *dud* performs multiplication on two inputs and includes return functions that reshape the output size, serving as the ΔD^U in (2). In the model section, optimization formulations are defined with descriptive code. *Var* *pg* defines the power output variable, and it represent P_g . Additionally, *Var* *pru* and *prd* specify the regulation up and regulation down reserve, as they represent $P_{g,r}^U$ and $P_{g,r}^D$. *Constraint* *rbu* sets the RegUp reserve requirement of (2), and *rru* establishes the RegUp reserve source of (3). In the last, *Objective* *obj* defines the objective function of (1) to minimize the total cost of generation and reserve. In the definition of *rbu* and *obj*, there is a *e_str* that give a symbolic description of the equations. This symbolic representation can then be processed into an executable optimization model.

Upon completing the RTED modeling, a demonstrative scheduling of RTED with VIS is given in (4)-(8). Note that these formulations aim to illustrate the streamlined scheduling modeling process, while detailed formulations can be found in [17], [19]. In this formulation, decision variables are the generator output P_g , IBR virtual inertia M_r , and IBR virtual damping D_r . For simplicity, here we show the additional VIS formulations, but omit the standard RTED formulations.

```

1 class RTED:
2     def __init__(self, system, config):
3         ...
4         # --- Data Section ---
5         self.pmax = RParam(
6             info='Gen_maximum_active_power',
7             name='pmax', tex_name=r'p_{G,max}',
8             unit='p.u.', model='StaticGen',
9             no_parse=False,)
10        self.dud = NumOpDual(
11            u=self.pdz, u2=self.du,
12            fun=np.multiply, rfun=np.reshape,
13            rargs=dict(newshape=(-1,)),
14            name='dud', tex_name=r'd_{u,d}',
15            info='RegUp_reserve_requirement',)
16        # --- Model Section ---
17        self.pg = Var(
18            info='Gen_active_power', unit='p.u.',
19            name='pg', tex_name=r'p_{g}',
20            model='StaticGen', src='p',
21            v0=self.pg0)
22        self.pru = Var(
23            info='RegUp_reserve', unit='p.u.',
24            name='pru', tex_name=r'p_{r,u}',
25            model='StaticGen', nonneg=True,)
26        self.prd = Var(
27            info='RegDn_reserve', unit='p.u.',
28            name='prd', tex_name=r'p_{r,d}',
29            model='StaticGen', nonneg=True,)
30        self.rbu = Constraint(
31            name='rbu', is_eq=True,
32            info='RegUp_reserve_balance',
33            e_str='gs@mul(ug,pru)-dud',)
34        self.rru = Constraint(
35            name='rru', is_eq=False,
36            info='RegUp_reserve_source',
37            e_str='mul(ug,pg+pru)-mul(ug,pmax)',)
38        cost = 'sum(mul(c2,power(pg,2))'
39            cost += '+sum(cl@(t*pg))+sum(ug*c0)'
40            cost += '+sum(cru*pru+crd*prd)'
41        self.obj = Objective(
42            name='obj', unit='$',
43            info='total_cost',
44            sense='min', e_str=cost)

```

Listing 1. Descriptive RTED modeling.

In these Eqns, Ω_G and Ω_R are all generators and all IBR, respectively. The subscripts g and r represent the generator and IBR, respectively. The superscripts M and D represent virtual inertia and damping, respectively. f_g is the generation cost. \overline{M}_r and \overline{D}_r are the upper bounds of virtual inertia and damping, respectively. M_{req} and D_{req} are the virtual inertia and damping reserve requirements, respectively, and can be determined by the desired transient performance.

$$\min \left(\sum_{g \in \Omega_G} f_g(P_g) + \sum_{r \in \Omega_R} (c_r^M M_r + c_r^D D_r) \right) \quad (4)$$

$$\forall g \in \Omega_G, \forall r \in \Omega_R$$

s.t.

$$M_r \leq \overline{M}_r \quad (5)$$

$$D_r \leq \overline{D}_r \quad (6)$$

$$\sum_{r \in \Omega_R} M_r = M_{req} \quad (7)$$


```

1 class VISBase:
2     def __init__(self):
3         ...
4         self.M = Var(
5             info='Emulated_inertia_(M=2H)',
6             name='M', tex_name=r'M', unit='s',
7             model='VSG', nonneg=True,)
8         self.D = Var(
9             info='Emulated_damping',
10            name='D', tex_name=r'D', unit='p.u.',
11            model='VSG', nonneg=True,)
12        self.Mreq = Constraint(
13            name='Mreq', is_eq=True,
14            info='Emulated_inertia_requirement',
15            e_str='-gvsg@M+dvm',)
16        self.Dreq = Constraint(
17            name='Dreq', is_eq=True,
18            info='Emulated_damping_requirement',
19            e_str='-gvsg@D+dvd',)
20
21 class RTEDVIS(RTED, VISBase):
22     def __init__(self, system, config):
23         RTED.__init__(self, system, config)
24         VISBase.__init__(self)
25         gcost = 'sum(mul(c2,power(pg,2)))'
26         gcost += '+sum(c1@(t_dot_pg))+ug*c0'
27         rcost = '+sum(cru*pru+crd*prd)'
28         vsgcost = '+sum(cm*M+cd*D)'
29         self.obj.e_str = gcost+rcost+vsgcost

```

Listing 2. Integration of VIS into RTED as an example of stability-constrained scheduling.

$$\sum_{r \in \Omega_R} D_r = D_{req} \quad (8)$$

Listing 2 demonstrated the virtual inertia support provisions in RTED, using the developed framework. Lines 4-11 define two non-negative decision variables associated with the device "VSG". Next, lines 12-19 define two constraints describing the required emulated inertia and damping. Then, lines 21-24 combine the existing RTED with the newly added lines 4-19, forming the new RTED-VIS scheduling model. Lines 25-29 rewrite the objective functions to include the cost of virtual inertia service. In this example, we develop a block *VISBase* and then assemble it with *RTED* as the scheduling routine *RTED-VIS*. Along with the necessary input data, the developed *RTED-VIS* can be used for economic analysis study. Further details on the development can be found in the documentation provided in the Code Availability section.

IV. ADAPTABILITY OF THE MODELING FRAMEWORK

An adaptable modeling framework is essential in power systems to meet the evolving demands of scheduling modeling. First, such a framework should be extendable, allowing for easy integration of new grid technologies and energy products. Second, scalability is crucial to ensure reliable performance from rapid prototyping to large-scale applications. Next, a data structure compatible with transient engines is also a prerequisite for integration with a transient simulator. Lastly, interoperation with transient simulators is the key to the development of scheduling algorithms that integrate stability constraints.

This section elaborates on the adaptability of the proposed framework from the following four aspects, namely extensibility, scalability, compatibility, and interoperability.

```

1 class ESD1Base:
2     def __init__(self):
3         self.cpe = Constraint(
4             name='cpe', is_eq=True,
5             info='Select_pce_from_pg',
6             e_str='ce@pg-zce-zde',)
7         SOCb = 'mul(En, (SOC-SOCinit))'
8         SOCb += '-t_dot_mul(EtaC, zce)'
9         SOCb += '+t_dot_mul(REtaD, zde)'
10        self.SOCb = Constraint(
11            name='SOCb', is_eq=True,
12            info='ESD1_SOC_balance',
13            e_str=SOCb,)
14
15 class RTEDES(RTED, ESD1Base):
16     def __init__(self, system, config):
17         RTED.__init__(self, system, config)
18         ESD1Base.__init__(self)

```

Listing 3. Integration of energy storage into RTED.

A. Extensible Scheduling Formulation

The modular design in AMS ensures an extensible modeling framework to easily model new dispatchable elements. As illustrated in Figure 2, modeling efforts are organized into two parts, device-level and scheduling-level. Within this structure, new devices can be developed either by using *Parameter* and *Variable* or by reusing device-level modeling blocks. For more details on the device-level modeling blocks, see [2]. Similarly, new scheduling models can be derived from existing scheduling models by adding new elements. In this way, developers can reuse scheduling modeling blocks, and thus save modeling and maintenance efforts.

For instance, the energy storage model can be integrated into RTED by developing a modeling block. Eqns (9) - (10) describe the two common constraints in an energy storage model. In the equations, $z_{c,E}$ and $z_{d,E}$ are decision variables of the charging and discharging power of energy storage model. *SOC* is the decision variable of state of charge (SOC). SOC_{init} is the parameter of initial SOC, and E_n is the capacity. t is the RTED time interval, while η_c and η_d are parameters of charging and discharging efficiency. C_E is a coefficient matrix that identifies which static generator corresponds to the energy storage model.

$$C_E P_g - z_{c,E} - z_{d,E} = 0 \quad (9)$$

$$E_n (SOC - SOC_{init}) - t \eta_c z_{c,E} + t \eta_d z_{d,E} - SOC = 0 \quad (10)$$

Listing 3 illustrates the integration of energy storage into RTED, where *ESD1Base* is simplified to show only two constraints. Following the declaration of *ESD1Base*, the *RTEDES* model, which includes the energy storage model, can be easily derived from *RTED*. The above code snippet demonstrates the extensibility of AMS. Additionally, the streamlined prototyping scheme relieves the researcher's manual efforts in algorithm development and maintenance, benefiting from the descriptive modeling feature and existing modeling blocks.

B. Scalable Performance

As discussed in Subsection III-F, the modeling framework itself is scalable because it relies on the generated code and

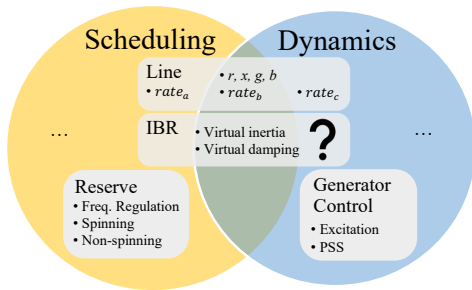


Fig. 3. Data overlap between scheduling study and dynamic simulation.

thus is case-independent. However, two concerns can still arise when the system size is large. First, the construction of the optimization model is highly dependent on the problem scale. Second, the computation performance is capped by the used solver, because the solver library is the one that solves the optimization problem.

Regarding the first concern, sparse techniques can be applied to ensure performance at large-scale. For instance, the connection matrices in power systems are usually highly sparse, thus enabling the effective use of sparse matrix storage and operations [27]. Additionally, the scheduling problem can be carefully formulated to reduce redundant variables and constraints.

As for the second concern, besides the continuous advancement of optimization theories and implementations, power system researchers can select appropriate solvers and methods given the problem's characteristics. Furthermore, they are skilled in developing dedicated scheduling algorithms to further simplify the problem with domain knowledge.

C. Compatible Data Structure

Figure 3 illustrates a demonstrative data structure of scheduling and transient simulators and their overlap. Reserve requirement data are used mainly in scheduling studies, while generator control, such as excitation systems and power system stabilizers, is used in transient simulations. Additionally, certain topology data is shared between scheduling and transient simulators. For instance, line parameters, including resistance r , reactance x , conductance g , susceptance b , and short-term line limit $rate_b$ are involved in both optimal power flow study and transient simulation. In addition, the long-term line limit $rate_a$ is used in scheduling analysis, whereas the emergency line limit $rate_c$ accounts for transient contingencies.

Recent emerging techniques introduce new dispatchable elements that also merit transient simulation to ensure stability. For example, virtual inertia scheduling has been discussed to mitigate low-inertia issues by using IBR to provide inertia and damping control. Virtual inertia and damping are not only decision variables in the scheduling study, but also dynamics parameters in transient simulation. Thus, a compatible data structure is a prerequisite for interoperability between transient and scheduling simulators.

In AMS, the input data is defined in different devices, including power flow data and scheduling data. Power flow data are the basic input data including bus, line, generator, and load. Scheduling data, such as generator cost and reserve

requirement, can be included as needed. In transient simulation, the input file includes power flow data and dynamics data. PSSE defines power flow data as RAW file and dynamics data as DYR file. ANDES defines power flow data and dynamics data in different devices and stores them in an XLSX file.

In conclusion, AMS employs a device-based input data format to ensure a compatible data structure, where power flow data bridges the input files of scheduling and transient by defining the system topology.

D. Interoperable Dynamics Interface

Effective interoperation between the scheduling and transient components is essential to integrate transient stability constraints into scheduling algorithms. There are two scenarios of interoperation, depending on the direction of data flow: 1) Scheduling to transient (*Send*): In this scenario, the scheduling simulator sends its results to the transient simulator for TDS; 2) Dynamic to scheduling (*Receive*): In this scenario, the scheduling simulator collects dynamics results to solve the optimization problem.

For example, when developing a new frequency regulation product, high-fidelity validation through TDS is required. In this case, setting points and reserve values are obtained from the scheduling simulation, while the next-round scheduling also takes the system states as input. A compatible interface must accommodate both types of data exchange.

Additionally, there is a gap between DC-based scheduling formulations and AC-based TDS. In scheduling, it is assumed that there are no losses in the transmission network, considering the relatively large X/R ratio. This reduces modeling complexity and ensures that the solving time complies with industry requirements. In TDS, however, AC-based power flow equations are a superset of system differential-algebraic equations and must be satisfied at each time step. In practice, two intuitive DC-AC conversion approaches are introduced: 1) Incorporate loss factors that account for active power losses [28]; 2) Preserve the regulating power from the generator output range and re-solve the ACOPF to get the generator output power [16].

As a result, AMS develops a dynamic interface to store the mapping information between scheduling and transient. With vectorized design, data is exchanged efficiently in parallel.

E. Full Timescale Digital Twin for Power Grid

The compatibility and interoperability with transient simulators allow for the further advancement of a "virtual power grid". The state estimation techniques provide both accuracy and resolution, and the dynamic interface enables smooth integration of transient stability assessment. Consequently, achieving a full timescale digital twin for power grids is possible through the scheduling-centric virtual power grid with dynamics-integration. This approach is advantageous for systems with high penetration of IBRs, as it ensures that their unique characteristics and dynamics are modeled and integrated into the scheduling.

V. CASE STUDIES

This section presents extensive case studies that validate and demonstrate the capabilities of the developed modeling framework. First, the accuracy of the framework is validated across various cases using OPF. Second, the computation time for three distinct types of problems is analyzed in various cases. Next, the extensible scheduling modeling scheme is demonstrated through the rapid prototype of RTED considering virtual inertia scheduling of IBRs. Finally, the framework's interoperability with the transient simulator is illustrated through the developed RTED with virtual inertia scheduling and its scheduling-dynamics co-simulation. More detailed usage instructions and tutorials for the implemented AMS can be found in the online documentation referenced in Section VI.

All case studies in this section are performed on a laptop with Apple M3 Pro processors and 36 GB RAM. The environment for AMS is deployed in Python-3.10.0 with the following packages: ANDES-1.9.1 [2], CVXPY-1.5.3 [29], [30], pandapower-2.14.11, GUROBI-11.0.3, MOSEK-10.2.6, and PIQP-0.4.2 [31]. Furthermore, MATPWOER-7.1 is installed in Matlab-R2024b.

A. Benchmark of Optimization Results

TABLE I
BENCHMARK OF OPF COSTS

Cost [\$]	AMS	pandapower	MATPOWER
IEEE 14-Bus	7,642.59	7,642.59	7,642.59
IEEE 39-Bus	41,263.94	41,263.94	41,263.94
PEGASE 89-Bus	5,733.37	5,733.37	5,733.37
IEEE 118-Bus	125,947.88	125,947.88	125,947.88
NPCC 140-Bus	810,033.37	810,016.06	810,033.37
WECC 179-Bus	411,706.13	411,706.13	411,706.13
IEEE 300-Bus	706,292.32	706,292.32	706,292.32
PEGASE 1354-Bus	1,218,096.86	1,218,096.86	1,218,096.86
PEGASE 2869-Bus	2,386,235.33	2,386,235.33	2,386,235.33
GOC 4020-Bus	793,634.11	793,634.11	793,634.11
EPIGRIDS 5658-Bus	1,195,466.12	1,195,466.12	1,195,466.12
EPIGRIDS 7336-Bus	1,855,870.94	1,855,870.94	1,855,870.94

This subsection compares the performance of linearized OPF across various case sizes, given its fundamental role in production cost analysis and scheduling modeling. Table I lists the objective values obtained from this benchmark

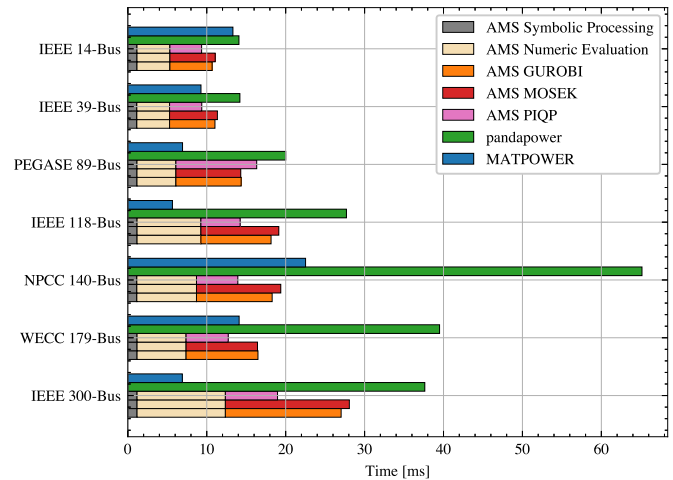


Fig. 4. Computation time of OPF on small-scale cases.

study. It is important to note that for the GOC 4020-Bus, EPRIGRIDS 5658-Bus, and EPIGRIDS 7336-Bus cases, the line rate was set to a high value to alleviate line flow constraints; otherwise, the OPF would fail for both pandapower and MATPOWER. Additionally, for the NPCC 140-bus case, the value from pandapower is slightly lower than those from AMS and MATPOWER. The cases come from MATPOWER [5], NPCC and WECC economic data [32], and IEEE PES Power Grid Benchmarks [33]. To ensure input consistency across the different tools, MATPOWER case files in m-file format are used as inputs for both AMS and MATPOWER. Then, the built-in file format converter converts the AMS grid to a PYPOWER format dictionary. This conversion facilitates a direct comparison by aligning the data structures used by each tool. Finally, pandapower can perform the OPF analysis using the converted case. Overall, the consistent results achieved by AMS, MATPOWER, and pandapower across these varied test cases not only validate the credibility of the developed scheduling modeling framework, but also demonstrate its reliability and accuracy, affirming its effectiveness for use in production cost analysis and power system optimization.

B. Benchmark of Computation Performance

Figure 4 shows the computation time of OPF using small scale cases. In the bar chart, the gray bar labeled "AMS Symbolic Processing" represents the time spent on symbolic processing, while the wheat-colored bar "AMS Numeric Evaluation" represents the time spent on system matrices calculation and optimization model construction. The orange bar labeled "AMS GUROBI" represents the optimization-solving time using the GUROBI solver. Similarly, the red bar labeled "AMS MOSEK" and the pink bar labeled "AMS PIQP" represent the time used by the solvers MOSEK and PIQP, respectively. Regarding the baselines, the blue and green bars represent the running time of MATPOWER using solver MIPS and pandapower using solver PIPS, respectively. The results for AMS, pandapower, and matpower are the average time consumed over ten repeat tests. When comparing the AMS results of different solvers, it shows that: 1) the symbolic

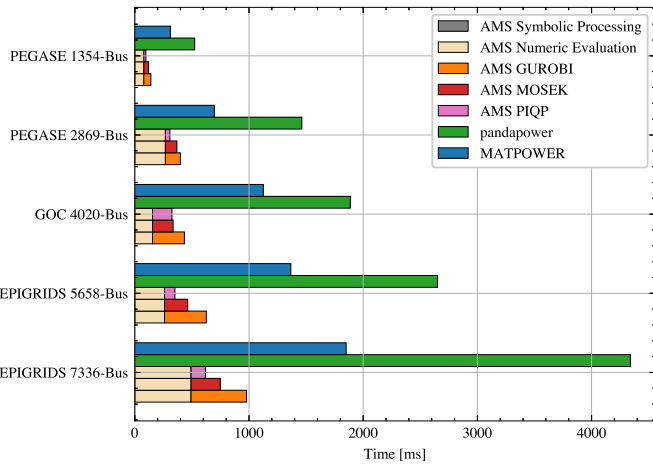


Fig. 5. Computation time of OPF on large-scale cases.

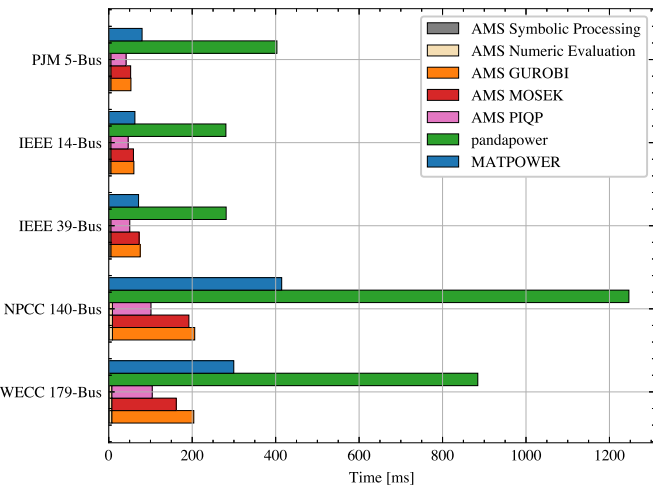


Fig. 6. Computation time of 24-hour load level scanning using OPF.

processing time is independent of the case sizes and remain to be very limited portion, and 2) the computation time for AMS mainly comes numerical evaluation and optimization solving. The comparison between AMS and other tools shows that the adapted hybrid symbolic-numeric approach is efficient and scalable.

Figure 5 further validates the computation time of OPF on large-scale cases. When comparing the results of AMS across difference-size cases, one can see that the time spent on symbolic processing is almost negligible. Additionally, the comparison between AMS and other tools further confirms the advantages of the adapted hybrid symbolic-numeric approach. The results in Figures 4 and 5 show that the consumed time increases more slowly than the case sizes, which indicates the good scalability of the developed framework.

Note that given the modular design, the symbolic processing and numeric evaluation process runs only once. After successful initialization, the optimization problem can be revised directly by updating the parameters. Figure 6 shows the benchmark computation time for 24-hour load level scanning using OPF, where the 24 load level factors are synthesized from the corresponding system operators' hourly load data.

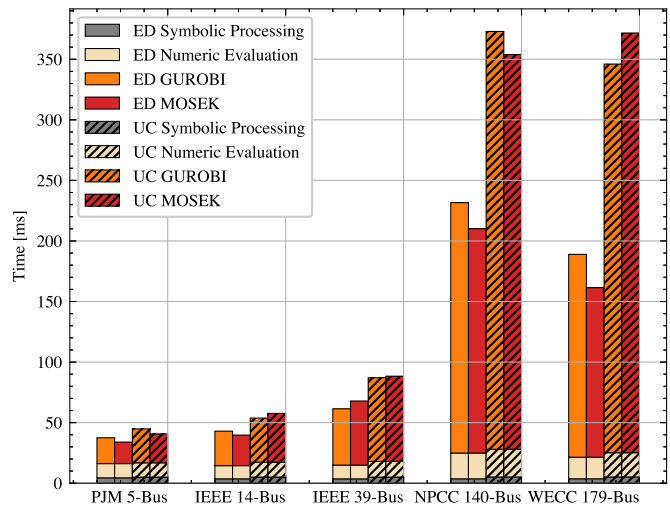


Fig. 7. Computation time of multi-period scheduling.

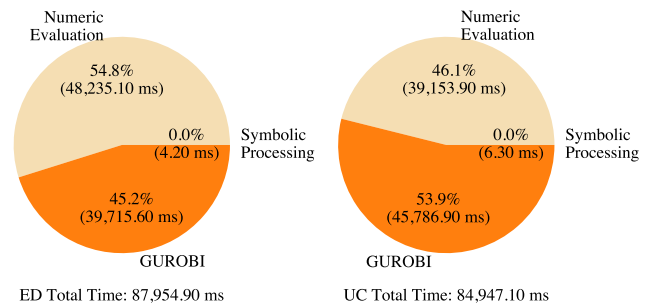


Fig. 8. Computation time distribution for multi-period economic dispatch and unit commitment using a 31,777-bus case.

In this graph, the symbolic processing is still negligible and the numeric evaluation process takes much less time than the solver running time. Furthermore, the comparison between AMS and other tools confirms the improved performance resulting from the modular design.

Concerning different problem types, Figure 7 presents the computation time of multi-period economic scheduling and unit commitment [34], [35], where the time is calculated from the average time of ten repeat tests. The cases considered here include a 24-hour scheduling with 1-hour time intervals. Economic dispatch remains a quadratic programming problem, while unit commitment becomes a mixed-integer programming problem.

Next, an ultra-large case is used to further validate the multi-period economic dispatch and unit commitment. The case, sourced from [36], includes 31,777 buses, 4,664 generators, 41,573 transmission lines, and 5 intervals. Figure 8 illustrates the time distribution of solving the economic dispatch and unit commitment of this ultra-large case. As shown in the figure, the time spent on symbolic processing remains negligible.

Comparison of AMS across different cases and types of problems further confirms the scalability of the proposed modeling framework. Besides, the high-performance solver library gives AMS a competitive capability for solving large-scale multi-period scheduling.

This subsection benchmarks the computational performance

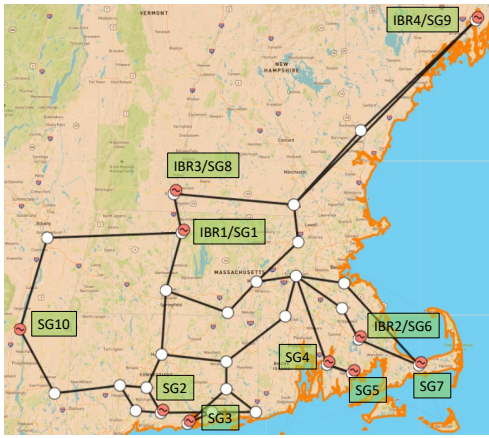


Fig. 9. AGVis visualization of revised IEEE 39-bus system with IBRs

```

1 # --- Scheduling Simulation ---
2 sp = ams.load('ieeee39_uced_vis.xlsx')
3 sp.RTEDVIS.run()
4 # --- Interoperation ---
5 sa = sp.to_andes(addfile='ieeee39_vis.xlsx')
6 sp.RTEDVIS.dc2ac()
7 sp.dyn.send(adsys=sa, routine='RTEDVIS')
8 # --- Dynamic Simulation ---
9 sa.PFlow.run()
10 sa.TDS.run()
11 sp.dyn.receive(adsys=sa, routine='RTEDVIS')

```

Listing 4. Example code of RTEDVIS cosimulation. Imports are omitted for simplicity.

of the developed framework, using different size cases and problem types. The benchmarks clearly confirm the framework's capability for complex problem types and good scalability for large-scale cases.

C. Validating IBR providing virtual inertia services through the interoperation with other LTB packages

This subsection demonstrates the validation of IBR providing virtual inertia services through the interoperability of the proposed scheduling modeling framework, interfacing with the transient simulator ANDES and the geographical visualizer AGVis.

IEEE 39-bus system is used where four synchronous generators are replaced with IBRs:

- Case1: All generators are synchronous generators.
- Case2: Four synchronous generators are replaced with IBR under VSG control, where the penetration level is 40%.
- Case3: Replaced VSG-controlled IBR are under virtual inertia scheduling, where the penetration level is 40%.

The example code of scheduling-dynamics co-simulation for RTED-VIS is illustrated in the listing 4. In the co-simulation, there are three parts, namely scheduling simulation, interoperation, and transient simulation. In the Scheduling Simulation part, first, case system is loaded from the input file as shown by line 2. Second, the developed scheduling routine RTED-VIS can be solved as shown in line 3. Moving on to the Interoperation part, the corresponding dynamic case is converted from the scheduling case with a supplemented dynamic input file as

TABLE II
SCHEDULING RESULTS OF VIRTUAL INERTIA AND DAMPING

Generator	M [s]			D [MW/Hz]		
	Case1	Case2	Case3	Case1	Case2	Case3
IBR1/SG1	8.40	8.40	10.00	0.50	0.50	1.00
IBR2/SG6	6.96	6.96	10.00	0.50	0.50	1.00
IBR3/SG8	4.86	4.86	10.00	0.50	0.50	1.00
IBR4/SG9	6.90	6.90	8.34	0.50	0.50	0.83

shown in line 5. Subsequently, line 6 converts the DC-based results to AC as discussed in Section IV-D. Upon successful conversion, the dynamics interface can send the scheduling results, including generator output, virtual inertia, and damping to the corresponding dynamic devices. Regarding the Dynamic Simulation part, lines 9 and 10 solve the AC power flow and TDS, respectively. After the simulation, line 11 can retrieve necessary system states from TDS and feedback them into the scheduling simulator for next-round optimization.

This framework can facilitate the seamless integration of the steady state generation scheduling optimization and the dynamics simulation to ensure frequency stability under high-penetration IBR integration. This integration can be achieved in both open-loop stability checking after steady-state generation scheduling with dynamics simulation or through the closed-loop economic dispatch and automatic generation control simulation. In both cases, the generation scheduling set-points will be sent to the dynamic simulators to analyze the frequency performance of the specific operating point, and the dynamics simulation results will be fed back to the generation scheduling model as the initial status of generators. In real-time economic dispatch, parameters such as bus voltage amplitudes and angles are obtained from the state estimation of the power grid. These signals are then used in the subsequent round of real-time economic dispatch. When transient simulation serves as the grid simulator, the signals from the simulation can enhance the fidelity. For example, a detailed secondary frequency regulation study is demonstrated using the developed framework, showcasing both the economic results and dynamics performance [16].

Table II shows the scheduling results of virtual inertia and damping. The columns of Case1 list out the original synchronous generators' inertia and damping. The columns of Case2 list out assigned IBRs' virtual inertia and damping, while the columns of Case3 list out calculated IBRs' virtual inertia and damping from the RTED-VIS model. It shows that, when using IBR with virtual inertia scheduling in Case3, IBR is assigned to provide higher virtual inertia and damping compared to Case2, where M and D are fixed values.

Figure 10 illustrates the dynamics simulations under generator tripping at 1s. In Figures 10 (a) and (b), the curves are from IBR1 or SG1, where "aux power" refers to the difference between the assigned power and the actual output power. The comparison between Case1 and Case2 in both (a) and (b) indicates that introducing IBR results in a fast dynamics interaction. Conversely, the comparison between Case2 and Case3 in both (c) and (d) demonstrates the potential of IBR

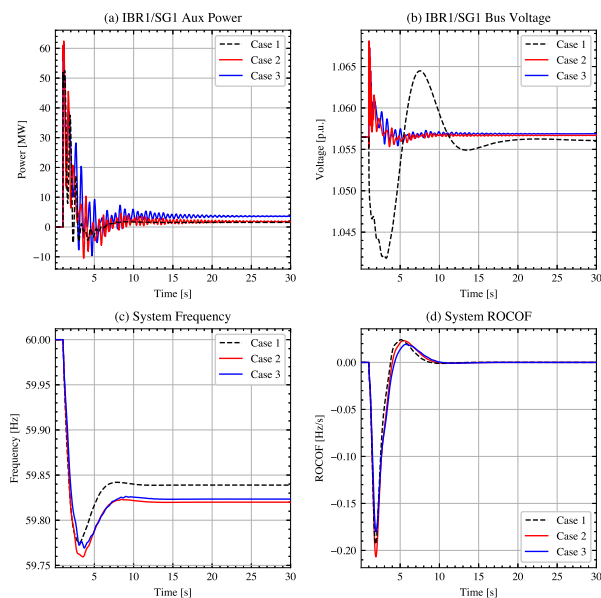


Fig. 10. Dynamics performance validation of VIS through TDS.

to provide frequency support services.

The interoperation between scheduling and the transient simulators facilitates the development of the scheduling algorithm that involves stability constraints.

VI. CONCLUSION

In conclusion, this paper proposed a hybrid symbolic-numeric approach-based scheduling modeling framework to address the scheduling modeling challenges. The proposed framework has four important features for power grids with IBRs: extensible scheduling formulation, scalable performance, compatible data structure, and interoperable operation with transient simulators. The AMS framework reduces the development efforts required for both scheduling modeling and scheduling-dynamics co-simulation, with demonstration for IBR integrations. Furthermore, extensive case studies have shown its accuracy and competitive performance in various problem types. Case studies also demonstrate the modeling scheme of IBRs providing ancillary services, as evidenced by detailed transient simulations. The developed AMS completed CURENT LTB with the scheduling functionality. Thus, a full-timescale digital twin for the power grid is achievable using the scheduling-centric virtual power grid with dynamics integration.

Consequently, this framework opens up new research opportunities. First, the framework enables the development of scheduling algorithms that can leverage the flexibility of renewable energy sources effectively and reliably. Second, it facilitates the creation of a virtual energy market by developing a market layer, which is essential to devise and test new energy market mechanisms.

ACKNOWLEDGMENTS

The authors would like to acknowledge the financial support from CURENT, a National Science Foundation (NSF) Engi-

neering Research Center funded by NSF and the Department of Energy under NSF Award EEC-1041877.

CODE AVAILABILITY

The implementation of AMS is available on PyPI and conda-forge as the Python package "ltbams", and its source code can be found at <https://github.com/CURRENT/ams>, and the documentation is accessible at <https://ltb.readthedocs.io/projects/ams>. The source code and cases files used in Case Studies is accessible at https://github.com/CURRENT/demo/tree/master/demo/ams_benchmark.

REFERENCES

- [1] F. Li, K. Tomsovic, and H. Cui, "A large-scale testbed as a virtual power grid: For closed-loop controls in research and testing," *IEEE Power and Energy Magazine*, vol. 18, no. 2, pp. 60–68, Mar. 2020.
- [2] H. Cui, F. Li, and K. Tomsovic, "Hybrid symbolic-numeric framework for power system modeling and analysis," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1373–1384, Mar. 2021, Number: 2.
- [3] N. Parsly, J. Wang, N. West, Q. Zhang, H. Cui, and F. Li, "DiME and AGVis: A distributed messaging environment and geographical visualizer for large-scale power system simulation," in *2023 North American Power Symposium (NAPS)*, Asheville, NC, USA: IEEE, Oct. 15, 2023, pp. 1–5.
- [4] J. Bialek, J. MacDowell, T. Bowen, *et al.*, "System services and needs for systems with high IBR penetration," Global Power System Transformation Consortium, Oct. 2021.
- [5] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011.
- [6] R. Lincoln, *PYPPOWER*.
- [7] L. Thurner, A. Scheidler, F. Schafer, *et al.*, "Pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018.
- [8] T. Brown, J. Hörsch, and D. Schlachtberger, "PyPSA: Python for power system analysis," *Journal of Open Research Software*, vol. 6, no. 1, p. 4, Jan. 16, 2018.
- [9] J. D. Lara, C. Barrows, D. Thom, D. Krishnamurthy, and D. Callaway, "PowerSystems.jl — a power system data management package for large scale modeling," *SoftwareX*, vol. 15, p. 100747, Jul. 2021.
- [10] J. D. Lara, R. Henriquez-Auba, D. Ramasubramanian, S. Dhople, D. S. Callaway, and S. Sanders, "Revisiting power systems time-domain simulation methods and models," *IEEE Transactions on Power Systems*, pp. 1–16, 2023.
- [11] M. Tuo and X. Li, "Security-constrained unit commitment considering locational frequency stability in low-inertia power grids," *IEEE Transactions on Power Systems*, vol. 38, no. 5, pp. 4134–4147, Sep. 2023, Conference Name: IEEE Transactions on Power Systems.
- [12] M. Jiang, Q. Guo, H. Sun, and H. Ge, "Short-term voltage stability-constrained unit commitment for receiving-end grid with multi-infeed HVDCs," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 2603–2613, May 2021.
- [13] H. Jia, Q. Hou, P. Yong, *et al.*, "Voltage stability constrained operation optimization: An ensemble sparse oblique regression tree method," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 160–171, Jan. 2024, Conference Name: IEEE Transactions on Power Systems.
- [14] X. Fang, H. Yuan, and J. Tan, "Secondary frequency regulation from variable generation through uncertainty decomposition: An economic and reliability perspective," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 4, pp. 2019–2030, Oct. 2021.
- [15] N. Gao, D. W. Gao, and X. Fang, "Manage real-time power imbalance with renewable energy: Fast generation dispatch or adaptive frequency regulation?" *IEEE Transactions on Power Systems*, vol. 38, no. 6, pp. 5278–5289, Nov. 2023.

[16] J. Wang, F. Li, X. Fang, *et al.*, "Electric vehicles charging time constrained deliverable provision of secondary frequency regulation," *IEEE Transactions on Smart Grid*, vol. 15, no. 4, pp. 3892–3903, 2024.

[17] B. She, F. Li, H. Cui, J. Wang, Q. Zhang, and R. Bo, "Virtual inertia scheduling (VIS) for real-time economic dispatch of IBRs-penetrated power systems," *IEEE Transactions on Sustainable Energy*, pp. 1–14, 2023.

[18] B. She, F. Li, J. Wang, H. Cui, X. Wang, and R. Bo, "Virtual inertia scheduling (VIS) for microgrids with static and dynamic security constraints," *IEEE Transactions on Sustainable Energy*, pp. 1–12, 2024.

[19] X. Liu, X. Fang, N. Gao, *et al.*, "Frequency nadir constrained unit commitment for high renewable penetration island power systems," *IEEE Open Access Journal of Power and Energy*, pp. 1–1, 2024.

[20] Z. Chu and T. Teng, "Stability constrained optimization in high IBR-penetrated power systems-part II: Constraint validation and applications," 2023, Publisher: arXiv Version Number: 1.

[21] H. Sun, B. Zhao, S. Xu, *et al.* "PSD-BPA manual," GitHub. (Feb. 21, 2024), [Online]. Available: https://github.com/lbl-hub/CSEE-Benchmark/tree/main/PSD-BPA_manual (visited on 11/06/2024).

[22] W. Zhongxi and Z. Xiaoxin, "Power system analysis software package (PSASP)-an integrated power system analysis tool," in *POWERCON '98. 1998 International Conference on Power System Technology. Proceedings (Cat. No.98EX151)*, vol. 1, Aug. 1998, 7–11 vol.1.

[23] H. Cui, F. Li, and X. Fang, "Effective parallelism for equation and jacobian evaluation in large-scale power flow calculation," *IEEE Transactions on Power Systems*, vol. 36, no. 5, pp. 4872–4875, Sep. 2021.

[24] J. K. Ousterhout, *A philosophy of software design*, First edition. Palo Alto, CA: Yaknyam Press, 2018, 178 pp.

[25] J. E. F. Friedl, *Mastering regular expressions*, 3rd ed. Sebastopol, CA: O'Reilly, 2006, 515 pp., OCLC: ocm76945355.

[26] N. Hatziaargyriou, J. Milanovic, C. Rahmann, *et al.*, "Definition and classification of power system stability – revisited & extended," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3271–3281, Jul. 2021.

[27] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "SciPy 1.0: Fundamental algorithms for scientific computing in python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, Mar. 2, 2020.

[28] F. Li and R. Bo, "DCOPF-based LMP simulation: Algorithm, comparison with ACOFP, and sensitivity," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 1475–1485, Nov. 2007.

[29] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," 2016.

[30] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," 2017, Publisher: arXiv Version Number: 2.

[31] R. Schwan, Y. Jiang, D. Kuhn, and C. N. Jones, "PIQP: A proximal interior-point quadratic programming solver," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, Singapore, Singapore: IEEE, Dec. 13, 2023, pp. 1088–1093.

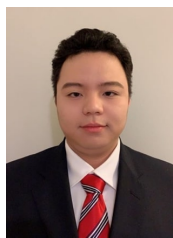
[32] Q. Zhang and F. Li, "A dataset for electricity market studies on western and northeastern power grids in the united states," *Scientific Data*, vol. 10, no. 1, p. 646, Sep. 22, 2023.

[33] S. Babaeinejadsarookolae, A. Birchfield, R. D. Christie, *et al.*, "The power grid library for benchmarking AC optimal power flow algorithms," 2019.

[34] Y. Huang, P. Pardalos, and Q. Zheng, *Electrical power unit commitment*. New York, NY: Springer Berlin Heidelberg, 2017.

[35] D. A. Tejada-Arango, S. Lumbreras, P. Sanchez-Martin, and A. Ramos, "Which unit-commitment formulation is best? a comparison framework," *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2926–2936, Jul. 2020.

[36] S. Elbert, J. Holzer, A. Veeramany, *et al.*, *ARPA-e grid optimization (GO) competition challenge 2*, Artwork Size: 29 files Pages: 29 files, 2024.



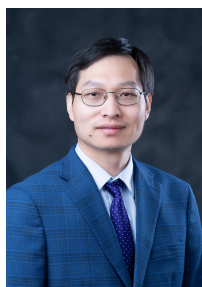
Jinning Wang (S'20, IEEE) received the B.S. and M.S. degrees in electrical engineering from the Taiyuan University of Technology, Taiyuan, China, in 2017 and 2020, respectively. He is currently pursuing a Ph.D. degree in electrical engineering with the University of Tennessee, Knoxville, TN, USA. His research interests include data mining, scientific computation, and power system simulation. He is the author of AMS, a power system scheduling simulator, which is a key component of the CURENT Large-scale Testbed. He was also an

outstanding reviewer for the *IEEE Open Access Journal of Power and Energy* in 2024.



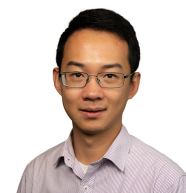
Fangxing Li (S'98-M'01-SM'05-F'17, IEEE) is also known as Fran Li. He received the B.S.E.E. and M.S.E.E. degrees from Southeast University, Nanjing, China, in 1994 and 1997, respectively, and the Ph.D. degree from Virginia Tech, Blacksburg, VA, USA, in 2001. Currently, he is the John W. Fisher Professor in electrical engineering at the University of Tennessee, Knoxville, TN, USA. He has been the Director of CURENT since 2024. His research interests include resilience, artificial intelligence in power, distributed energy resources (DERs) and microgrid, and electricity markets. From 2020 to 2021, he served as the Chair of IEEE PES Power System Operation, Planning and Economics (PSOPE) Committee. Currently, he is serving as the Chair of IEEE WG on Machine Learning for Power Systems and the Editor-In-Chief of *IEEE Open Access Journal of Power and Energy (OAJPE)*.

Prof. Li has received numerous awards and honors including R&D 100 Award in 2020, IEEE PES Technical Committee Prize Paper award twice in 2019 and 2024, 6 best or prize paper awards at international journals, and 7 best papers/posters at international conferences.



Xin Fang (S'12-M'16-SM'18, IEEE) received the B.S. degree from Huazhong University of Science and Technology, Wuhan, China, in 2009, the M.S. degree from China Electric Power Research Institute, Beijing, China, in 2012, and the Ph.D. degree at the University of Tennessee (UTK), Knoxville, TN, USA, in 2016.

He is currently an Assistant Professor at the Department of Electrical and Computer Engineering, Mississippi State University (MSU). Before joining MSU, he was a senior researcher at National Renewable Energy Laboratory (NREL) from 2017 to 2022. Dr. Fang is an associate editor of *IEEE Transactions on Power Systems* and *IEEE Transactions on Sustainable Energy*. His research interests include power system planning and operation, electricity market operation considering renewable energy integration, and cyber-physical transmission and distribution modeling and simulation.

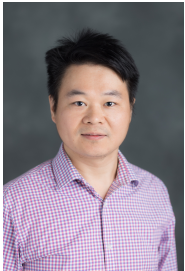


Hantao Cui (S'13-M'18-SM'20, IEEE) received the B.S. and M.S. degrees from Southeast University, China in 2011 and 2013, and the Ph.D. degree from the University of Tennessee, Knoxville in 2018, all in electrical engineering. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, North Carolina State University. His research interests include power system modeling, simulation, and high-performance computing.



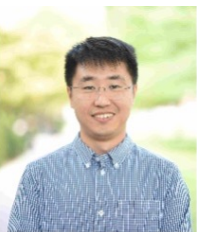
Buxin She (S'20-M'24, IEEE) received the B.S.E.E and M.S.E.E degrees from Tianjin University, China in 2017 and 2019, and the Ph.D. degree from the University of Tennessee, Knoxville in 2023, all in electrical engineering. He is currently a research engineer in Pacific Northwest National Laboratory (PNNL). He served as a student guest editor of IET-RPG. He was an outstanding reviewer of IEEE OAJPE (2020) and MPCE (2022 and 2023). His research interests include modeling and control of inverter-based resources, dynamics-constrained

power system planning and operation, and resilience analysis of power systems.



Hang Shuai (S'17-M'19, IEEE) received the B.Eng. degree from Wuhan Institute of Technology (WIT), Wuhan, China, in 2013, and the Ph.D. degree in Electrical Engineering from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2019. He was also a Visiting Student Researcher with the University of Rhode Island (URI), Kingston, RI, USA, from 2018 to 2019. He was a Postdoctoral Researcher with the URI and University of Tennessee, Knoxville (UTK) from 2019 to 2022. Currently, he is a Research Assistant Professor with

the UTK, Knoxville, TN, USA. His research interests include reinforcement learning for power system, microgrid operation and control, and bulk power system resilience.



Qiwei Zhang (S'17-M'22, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from The University of Tennessee, Knoxville (UTK), in 2018 and 2022, respectively. He is currently a Researcher at National Renewable Energy Laboratory (NREL). Prior to this, he was a Postdoc Research Fellow and Research Scientist with Johns Hopkins University and UTK. His research interests include electricity markets, power system optimizations, and power system plannings. He was an outstanding reviewer of IEEE Open Access Journal of Power and Energy

and is an associate editor of IET Generation, Transmission & Distribution.



Kevin L. Tomsovic (Member of the National Academy of Engineering, Fellow of IEEE) received the B.S. degree in electrical engineering from Michigan Technological University, Houghton, in 1982 and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, in 1984 and 1987, respectively. Currently, he is Director of the Clemson University – Charleston Innovation Campus and holds the Duke Energy Endowed Chair in Smart Grid Technology. He was previously the Director of the NSF/DOE Engineering Research

Center entitled CURENT, and the Chancellor's Professor in Electrical Engineering and Computer Science (EECS) at University of Tennessee, Knoxville (UTK). He was the Department Head of EECS at UTK from 2008-2013. His visiting university positions have included Boston University, Boston, MA; National Cheng Kung University, Tainan, Taiwan, R.O.C.; National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C.; and the Royal Institute of Technology, Stockholm, Sweden. He was on the faculty of Washington State University from 1992 to 2008. He held the Advanced Technology for Electrical Energy Chair at Kumamoto University, Kumamoto, Japan, from 1999 to 2000 and was an NSF program director in the ECS division from 2004 to 2006. He is a member of the National Academy of Engineering and a Fellow of the IEEE.