Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Parallelization Techniques for Behavioral Microgrid Simulation

ETHAN T. ALMQUIST[1], (Graduate Student Member, IEEE), AARON W. LANGHAM[1], (Graduate Student Member, IEEE), STEPHEN J. BRUNO[1], and STEVEN B. LEEB[1], (Fellow, IEEE)

[1]Massachusetts Institute of Technology, Cambridge, MA 02139 USA

Corresponding author: Aaron W. Langham (e-mail: alangham@mit.edu).

ABSTRACT Traditional microgrid modeling approaches based on differential equations are computationally expensive. These physics-based simulations from first principles sometimes provide more fidelity than is required for analysis and design tasks. Behavioral simulation and modeling move the computational burden from numerical solving algorithms to time-domain bookkeeping. This paper presents techniques to make the behavioral simulation of a microgrid "embarrassingly parallel," allowing modern multicore CPUs to throw their entire weight at the problem. Techniques are developed to remove state dependencies which inhibit parallelization. Benchmarks and a case study on real shipboard microgrid data serve as validation.

INDEX TERMS Behavioral simulation, microgrids, parallel computing, power flow

NOMENCLATURE

CPU	Central processing unit.
FSM	Finite state machine.
HVAC	Heating, ventilation, and air conditioning.
I/O	Input and output.
NILM	Nonintrusive load monitor.
RAM	Random access memory.
SPARCS	Shipboard Parallelized Analytics with Rapid Configuration Simulator.
USCGC	US Coast Guard cutter.

I. INTRODUCTION

Actionable electrical data provides value for the design, maintenance, and operation of microgrid power systems. A wealth of power simulation techniques generate this data across a diverse set of grids [1]–[3]. However, simulation speed is often an important component of what makes data actionable. Especially in early design activities, mapping out "what if" scenarios may not require the full fidelity provided by computationally intensive methods based on difference or differential equations. In these situations, the power of modern computation is wasted in simulating minutiae that may be unnecessary to compute the summary statistics of interest [4], [5].

As a compromise, designers may use tabular methods, summing nominal power ratings when considering aggregate loading with high uncertainty [4]. This approach provides only a low level of detail on how the electrical plant operates. Metrics such as peak loading and time spent light-loaded remain elusive without time-domain simulation. Designers requiring higher fidelity turn to general-purpose multiphysics numerical modeling software packages such as MathWorks Simulink and COMSOL Multiphysics [6], [7]. Numerical circuit simulators such as SPICE are ubiquitous across nearly all electrical domains. Utilities and grid engineers often use power flow modeling techniques to determine the complex power and voltages across large networks [8]–[10]. These techniques' runtimes can easily exceed the duration of time being simulated, delaying answers to grid designers' high-level statistical questions [11].

Behavioral techniques, by contrast, trade unnecessary modeling accuracy for speedy results. Numerical techniques for simulating loads and grid components focus on "what they are" in the form of constitutive differential equations. On the other hand, behavioral techniques model these components with "what they do" in the form of repeatable, pre-computed (or pre-recorded) electrical signatures. By avoiding numerical integration, a behavioral solver turns a power flow simulation into the simpler task of stitching together a series of signatures.

Plateauing CPU clock speeds and the proliferation of multicore CPU offerings have changed the nature of how efficient software (including simulators) is written [12], [13]. Rather than simply executing instructions *in seriatim* as fast as possible, modern applications often perform best when they make productive use of multicore hardware. Simulators in particular are bottlenecked by CPU performance, rather than I/O devices or network latency. Accordingly, they stand

to benefit from dividing the simulation task across multiple CPU cores. Parallelization through multithreading comes with challenges such as coordination across threads, sharing resources in memory, and preventing deadlocks and race conditions. However, programs that can be broken down into completely independent threads with no inter-thread state dependencies are naturally ripe for parallelization, sidestepping these issues. Common examples include Monte Carlo simulations and graphical processing such as ray tracing [14]. With some assumptions about the power grid, behavioral power simulation becomes one of these "embarrassingly parallel" problems that modern computing is especially equipped to handle [15].

Section II of this paper presents a review of both numerical and behavioral simulation techniques for dynamical systems. Section III gives a demonstration using a shipboard microgrid and a case study in using behavioral modeling to detect faults. Section IV shows experimental results of simulation time for an implementation of a parallel microgrid behavioral simulator. Section V concludes.

II. SIMULATING DYNAMICAL SYSTEMS

Dynamical physical systems are often described by continuous-time differential equations or discrete-time difference equations. These are often presented in state-space form as

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}(t), \mathbf{u}(t)), \tag{1}$$

and

$$\mathbf{x}_{k+1} = F(\mathbf{x}_k, \mathbf{u}_k), \tag{2}$$

respectively, where $\mathbf{x}$ is the vector of all system states and $\mathbf{u}$ is the vector of all system inputs. System states may include, for example, temperatures, battery states of charge, and rotor speeds. When combined with a numerical integrator and initial or boundary conditions, complex systems can be simulated with arbitrarily high precision. Although a miracle of modern computing, this approach to system simulation becomes computationally burdensome when dealing with large systems. In addition, systems with behavior on both short and long timescales require extra computation for both fine-grained detail and for long simulated timeframes. For example, electrical grids contain power electronic switching periods on the order of microseconds and cyclic demand fluctuations on the order of months. This tension between needing to simulate both the "short" and the "long" delays answers to higher-level statistical questions. This creates a particular pain point in design problems by restricting how much of the design space can practically be explored in a limited time.

A. SPEEDUP TECHNIQUES

Model order reduction seeks to speed up a simulation by making the model smaller. Techniques such as matrix projection and eigenvalue truncation reduce the number of system states at the cost of accuracy, either overall or at certain timescales

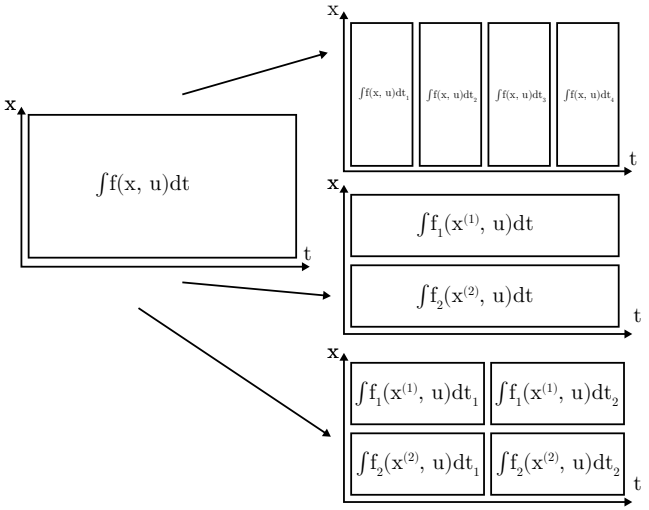


FIGURE 1: Simulation parallelization can be accomplished by distributing chunks of time (top), parts of the system (middle), or both (bottom) across multiple processors.

[16]. However, even a reduced-order model must be simulated serially through time, since each state vector depends on the previous computed state vector.

Alternatively, surrogate machine learning-based models can mimic higher-order model behavior for inputs sufficiently similar to the training data [17]. These can either model a direct relationship between inputs and outputs, or an intermediate state-space representation similar to traditional techniques. However, this requires ample computation and training data obtained either from real-world data collection or from evaluations of an already-developed model.

Decoupling models in both time and space allows a simulator to "divide and conquer" the simulation into independent jobs, as shown in Figure 1. On multicore CPUs or distributed computing systems, these can be run in parallel. Ideally, this divides the total computation time by the number of processors.

If collections of model states can be decoupled, the simulation can be parallelized spatially. This strategy appears in a diverse set of problem domains such as fluid, thermal, and climate modeling [18]–[20]. Decoupling a system into N independent subsystems results in the following N independent simulations:

$$\begin{aligned} \frac{d\mathbf{x}^{(1)}}{dt} &= f_1(\mathbf{x}^{(1)}(t), \mathbf{u}(t)), \\ &\vdots \\ \frac{d\mathbf{x}^{(N)}}{dt} &= f_N(\mathbf{x}^{(N)}(t), \mathbf{u}(t)), \end{aligned} \tag{3}$$

and for discrete time:

$$\begin{aligned} \mathbf{x}_{k+1}^{(1)} &= F_1(\mathbf{x}_k^{(1)}, \mathbf{u}_k), \\ &\vdots \\ \mathbf{x}_{k+1}^{(N)} &= F_N(\mathbf{x}_k^{(N)}, \mathbf{u}_k). \end{aligned} \tag{4}$$

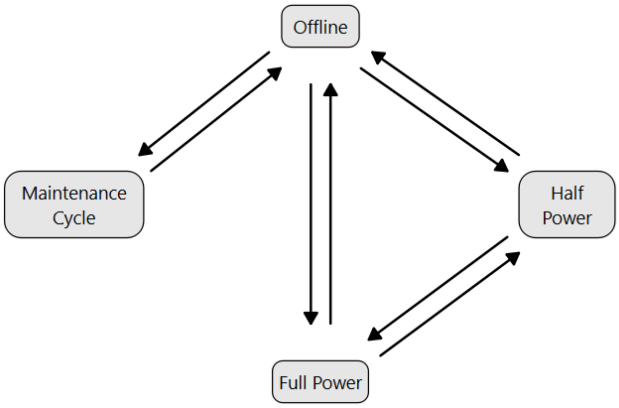


FIGURE 2: Example finite state machine for a load with four states of operation.

In pathological cases, one load's operation may affect another load's operation. For example, harmonics drawn by one load may introduce harmonics in the grid voltage that destabilize constant power loads or LEDs. Many fault condition studies require this "tight" load coupling to analyze how an equipment fault affects other loads, and as such, cannot decouple the power system into independent subsystems. However, in normal grid operation, the network voltage can often be usefully approximated to not change due to individual load operation. With this assumption, each load's operation can be decoupled from all other loads, and the system can be solved in parallel.

By contrast, some systems can be decoupled in time. For example, if state vector values at certain time points are known, the intervals between these known states can all be simulated independently. For example, decoupling the total time to be simulated into N independent intervals yields the following independent simulations:

$$\begin{aligned} \frac{d\mathbf{x}}{dt_1} &= f(\mathbf{x}(t_1), \mathbf{u}(t_1)), \\ &\vdots \\ \frac{d\mathbf{x}}{dt_N} &= f(\mathbf{x}(t_N), \mathbf{u}(t_N)), \end{aligned} \tag{5}$$

and for discrete time:

$$\begin{aligned} \mathbf{x}_{k_1+1} &= F(\mathbf{x}_{k_1}, \mathbf{u}_{k_1}), \\ &\vdots \\ \mathbf{x}_{k_N+1} &= F(\mathbf{x}_{k_N}, \mathbf{u}_{k_N}). \end{aligned} \tag{6}$$

For example, multigrid shooting algorithms such as Parareal produce a "soft" decoupling across time by first using a low-fidelity solver to find approximate state vector values, and then simulating the time intervals between these times in parallel with a high-fidelity solver [21].

B. BEHAVIORAL SIMULATION

For many physical systems, there is another way that avoids numerical integration entirely. Often, system operation can be characterized as existing within a finite set of states. For example, a microgrid power system containing a finite number of loads can only be running in a finite number of ways, as long as each load has a finite set of possible states. If each of the L loads has two states (e.g. OFF and ON), the total system must always be in one of the 2^L distinct states or transitioning between two states. Furthermore, the total system often can be separated into subsystems whose state transitions operate independently. For a power system with L loads, this creates L finite state machines (FSMs), each representing the current state of a load. Figure 2 shows an example FSM for a load with four states: Offline, Full Power, Half Power, and Maintenance Cycle. A behavioral simulator associates every FSM state and transition with a power signature that is either pre-simulated or pre-computed. Finally, at any given time, each load has a *behavior* that specifies how it traverses its finite state machine. A behavioral power simulator uses each load's behavior to produce the resulting power signature for each load. Although it never performs numerical integration, a behavioral simulator can still produce accurate and useful statistics about grid performance and operation [22].

Behaviors may be deterministic or stochastic in nature. The simplest behavior is to remain at one state, applicable to loads that remain on or off. A useful behavior for loads with hysteretic or "bang-bang" controllers is cycling between two states with some deterministic period and duty cycle. These loads often involve heating or cooling a space or component to a setpoint. Behaviors may also include randomness, such as state transition times sampled from a Gaussian or Poisson distribution. For example, loads that actuate in response to human activity can often be modeled with random processes [23], [24]. These three behaviors describe many industrial loads, but *any* repeatable path through the FSM can constitute a behavior.

To behaviorally simulate a load's power trace over a time period, the simulator infers all state transitions in that time period based on the load's behavior. For example, cycling behavior creates a series of state transitions separated by the cycling period. Simulating the load's power signature at any timestep only requires looking up the power trace corresponding to the state or transient the load takes at that time. Once each load's power trace is found, the aggregate power signature for a radial power network can be easily determined by summing up downstream power signatures.

C. BEHAVIORAL PARALLELIZATION

In general, loads take different behaviors at different times. For example, an occupancy-based room lighting circuit may turn on as a Poisson arrival process during the day, but remain off during the night when a facility is unoccupied. Here, a *mission* refers to a period of time in which all load behaviors do not change. Figure 3 illustrates a timeline to be simulated with five missions, with each colored bar corresponding to

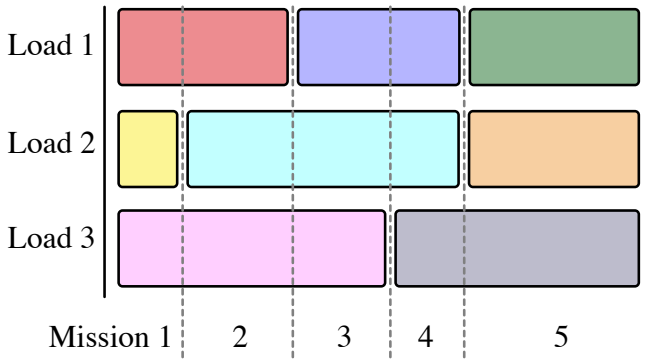


FIGURE 3: Continuous periods of time where load behaviors do not change are referred to as missions.

a different behavior. A load's power trace in a mission is entirely determined by the series of state changes, which themselves are entirely determined by the load's behavior during that mission. As a result, load power traces in a given mission are completely independent of the load behavior in other missions (other than the transients required to connect the ending and starting states across missions). Therefore, missions can form the building blocks of temporal parallelization in a behavioral simulator. Every mission can be simulated in parallel across several processors.

The time required to process each mission in parallel is approximately proportional to the its duration. Assuming that each processor is identically performant, the total time required to simulate all missions depends on how evenly each mission job can be scheduled across processors. For example, simulating three hour-long and one ten hour-long mission on a four-core CPU will require at least as much time as it takes to simulate the ten hour-long mission. Stated another way, the total runtime is lower bounded by the time to simulate the longest mission. When there are more missions than there are processors, the total runtime depends on which processors they are assigned to, and in what order. In optimal job scheduling, minimizing the total runtime across identical machines with no further constraints on job execution is denoted as optimization of the $P||C_{max}$ problem. In general, this problem is NP-hard. However, simply assigning missions to processors as they become available results in a total simulation runtime upper bounded by $2-1/M$ times the optimal solution's runtime, where M is the number of processors [25]. It is generally useful to keep missions around the same length in order to maximize throughput.

D. IMPLEMENTATION DETAILS

SPARCS (Shipboard Parallelized Analytics with Rapid Configuration Simulator) is a custom simulation package designed to implement temporally parallel behavioral simulation for shipboard microgrids [15], [26]. Figure 4 shows SPARCS's home screen. From here, a user can select components, design the power system network, allocate components to systems, define behaviors, and define missions. SPARCS models a microgrid as a collection of systems, where each load and generator is assigned to a system. Each system is defined with a set of statuses. For example, an HVAC system may take the statuses *Off*, *Heating*, and *Cooling*. A user divides the time to be simulated into several missions. Each mission is associated with statuses for all systems and global variables that describe environment conditions, such as temperature and humidity. For each mission, a series of conditional propositional logic statements determines each load's behavior during that mission. Behaviors implemented in SPARCS include constant steady state, cycling between two states with a parameterized duty cycle and period, and random progression between two states according to a parameterized distribution. For example, *IF* the outside temperature is less than 20° C, *THEN* the space heater cycles between the OFF and ON states with a 40% duty cycle and a period of 60 minutes. *ELSE IF* the outside temperature is above 20° C, the space heater remains in the OFF state.

Each load in the network is defined with a finite state machine and corresponding steady-state and transient power spectral envelope traces. An electrically simple load such as a heater can be defined with an ideal step power trace based on its nominal power rating. Traces for motors and loads with inrush currents can be defined similarly with user-parameterized exponential decay functions. Traces for these and more complicated loads can be defined with previously collected or numerically simulated data. For example, Figure 5 depicts a power spectral envelope trace associated with two shipboard vacuum pumps that energize together [27].

Upon running the simulator, a pre-processing algorithm computes each load's sequence of states and transients across all missions. Next, it breaks inter-mission dependencies by inserting any transients required to connect load states at mission transition points and notes ending indices in power traces to preserve trace continuity. Finally, the simulator iterates over all missions in parallel, and simulates them behaviorally. The relevant power traces for each load are stored in random access memory (RAM) for quick access, and are stitched together as the simulator progresses to build the full power trace for each load over time. As load power traces are obtained, Kirchoff's current law and conservation of power are used to propagate load power traces through radial subnetworks. When a ring is encountered, a reduced model of the ring is simulated using numerical methods [28]. The solution to the network is saved periodically to the file system as the simulation progresses.

III. DEMONSTRATION

Shipboard and marine power networks are among the most common examples of electric microgrids. Ranging from the kilowatt to multi-megawatt scale, they provide power to critical on-board systems from navigational equipment to potable water systems. They typically employ several generators powered by diesel or gas turbine engines, or less commonly, nuclear reactors. The critical nature of their electrical equipment makes them highly studied and modeled throughout

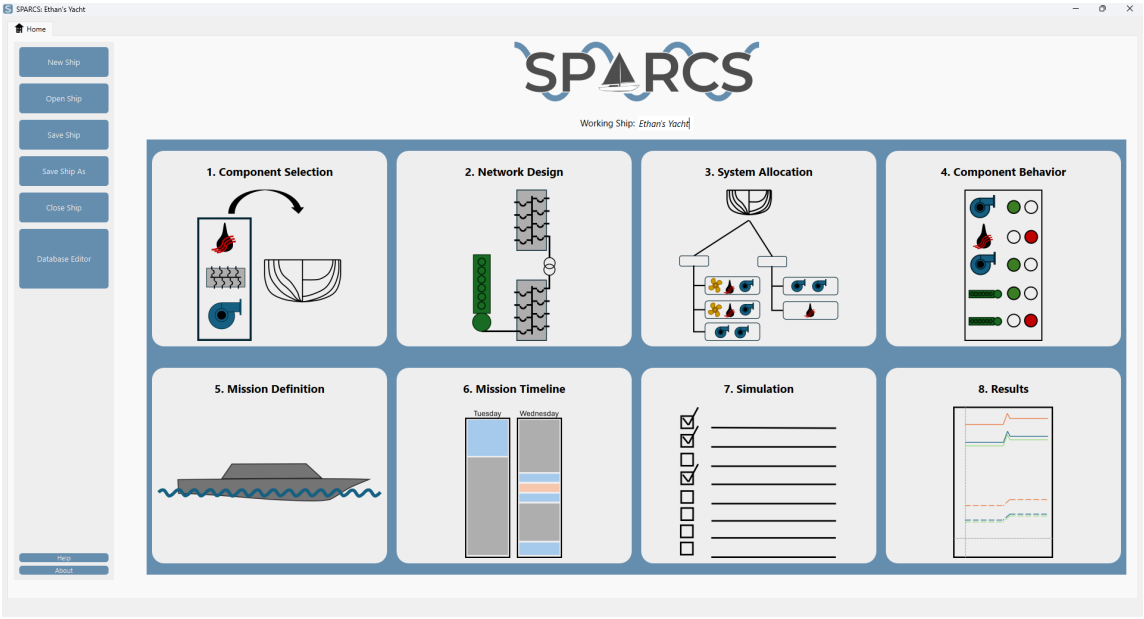


FIGURE 4: SPARCS home screen. Adapted from [15].

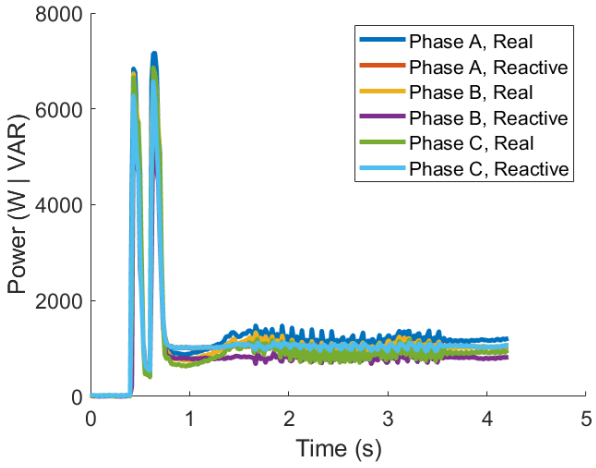


FIGURE 5: A power trace associated with two vacuum pumps turning on, as measured on a shipboard microgrid.

Mission	Start Time	End Time
Moored	Day 00 - 00:00	Day 00 - 14:00
Underway	Day 00 - 14:00	Day 01 - 12:00
Boarding Operations	Day 01 - 12:00	Day 01 - 13:00
Underway	Day 01 - 13:00	Day 02 - 10:00
Anchored	Day 02 - 10:00	Day 03 - 08:00
Underway	Day 03 - 08:00	Day 03 - 09:00
Moored	Day 03 - 09:00	Day 09 - 10:00
Underway	Day 09 - 10:00	Day 10 - 16:00
Anchored	Day 10 - 16:00	Day 11 - 09:00
Underway	Day 11 - 09:00	Day 11 - 09:00
Moored	Day 11 - 09:00	Day 12 - 12:00
Underway	Day 12 - 12:00	Day 12 - 15:00
Boarding Operations	Day 12 - 15:00	Day 12 - 16:00
Underway	Day 12 - 16:00	Day 13 - 14:00
Search and Rescue	Day 13 - 14:00	Day 14 - 09:00
Underway	Day 14 - 09:00	Day 14 - 11:00
Moored	Day 14 - 11:00	Day 15 - 00:00

TABLE 1: Nominal operational period of the USCGC *William Chadwick*. This period is observed with onboard sensors, and modeled using behavioral methods.

the design and operational stages of life. To demonstrate the behavioral modeling and parallelization techniques presented here, we model the electric power network of a U.S. Coast Guard (USCG) Sentinel-class fast response cutter (FRC). Results are validated against nonintrusive power monitoring data collected from the USCG cutter (USCGC) *William Chadwick*, an FRC based in Boston, Massachusetts. Although ships serve as a case study, these techniques generalize to any kind of microgrid.

A. SHIP MICROGRID OPERATION

The *William Chadwick*'s electrical grid supports more than 200 pieces of auxiliary and mission equipment with two 215 kW diesel generators and an emergency 78 kW diesel generator. Its electrical plant is configured radially, with either main generator able to supply power to the entire network. Figure 6 shows a one-line diagram of the ship's grid. As part of an ongoing monitoring effort, two nonintrusive load monitors (NILMs) are installed on the port and starboard machinery panels. These are highlighted in Figure 6. Each NILM records the current and voltage of each power system phase on its respective panel. The port and starboard machinery panels power auxiliary support systems, including machinery room heaters, an air conditioning compressor, sewage vacuum pumps, and water heaters.

The *William Chadwick* and other FRCs undergo missions such as policing waters for illegal fishing and search and

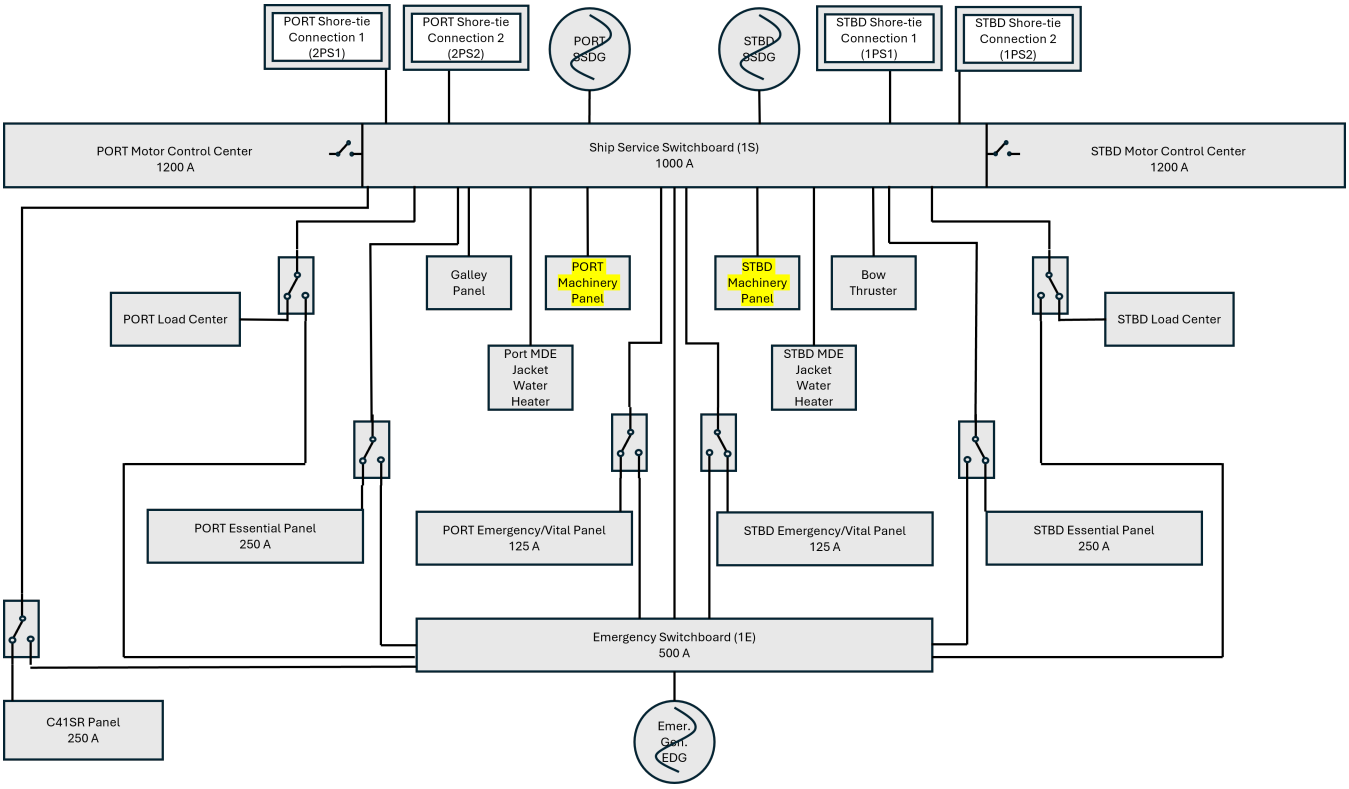


FIGURE 6: A high-level one-line diagram of the USCGC *William Chadwick*'s electric power network. The installed nonintrusive load monitors are highlighted yellow on the port and starboard machinery panels.

rescue. To benchmark the behavioral model developed in the following section, a 14-day ship-level operational profile was recorded from the NILMs installed on the ship. Ship operation was classified into five mission types:

1) "Underway," where the cutter is traveling at its efficiency speed
2) "Moored" to a pier
3) "Anchored" in a harbor
4) "Boarding Operations," where the cutter launches its intercept boat for fishery or narcotics inspection
5) "Search and Rescue" patrolling.

Table 1 shows the log of ship missions over this time period, as confirmed with the crew.

Most loads on the port and starboard machinery panels support auxiliary systems that depend on the ship's environmental conditions, rather than the ship's high-level task. To complement the operational profile in Table 1, weather data for the local region of operation was also recorded. Temperature, humidity, and precipitation rates were recorded hourly from a local weather station, in addition to the sunrise and sunset times. Cycling and random behavior parameters for each were derived from either observed statistics or mariner intuition about the load's operation.

An operational timeline consisting of a series of missions was constructed for simulation. Each mission was defined with a ship-level task (e.g., search and rescue) and a value for each global variable such as temperature. A new mission was defined each time the ship's task or a global variable changed. Weather data was sampled once an hour, resulting in approximately hour-long missions. Figure 7 shows three example missions, each defined with a ship-level task and different global variable values.

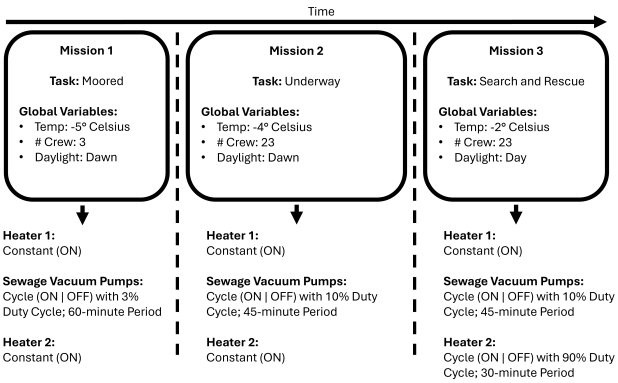


FIGURE 7: A progression of missions for the *William Chadwick*. Global variables and the ship's task remain constant throughout the mission duration. Each mission determines the behavior of each component, which then defines each component's sequence of states and transients.

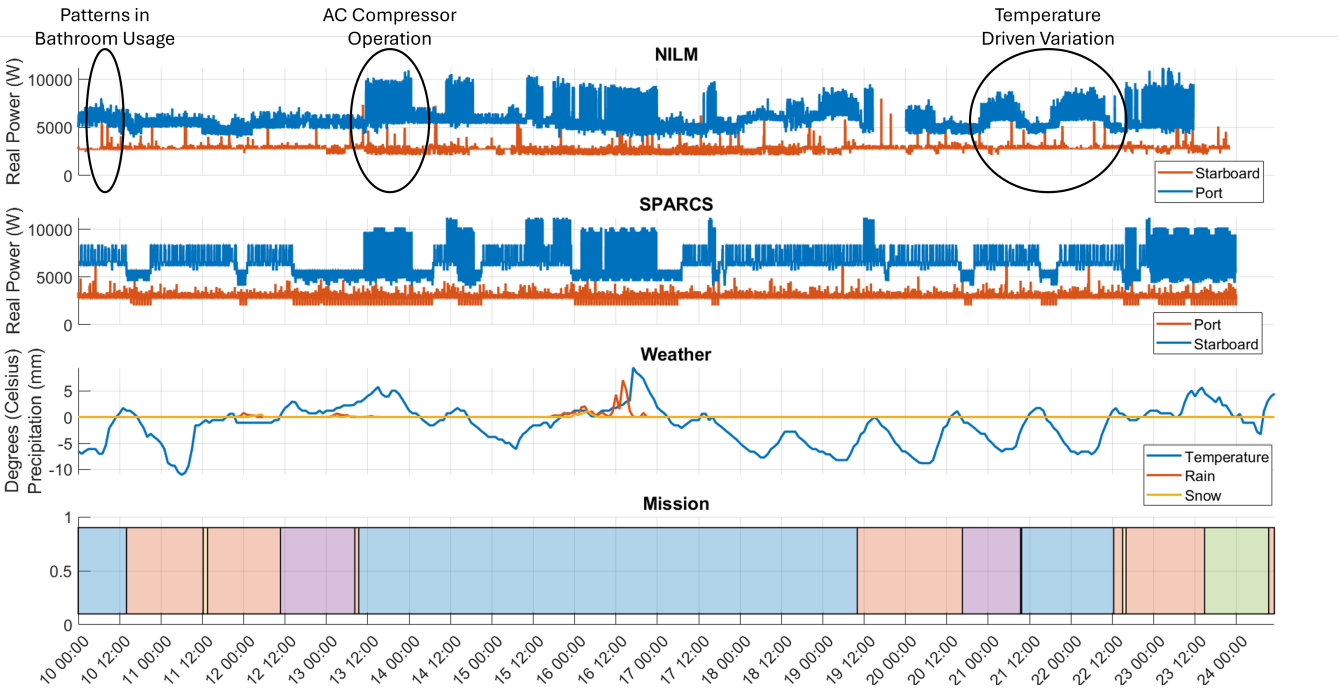


FIGURE 8: Phase A real power of the USCGC *William Chadwick* as sampled by onboard NILMs, and simulated by SPARCS. Plotted below are weather conditions and the ship's tasking during the operational period. The mission timeline at the bottom represents periods where the cutter is underway (orange), moored (blue), performing boarding operations (yellow), anchored (purple), and assisting search and rescue (green).

B. CASE STUDY: FAULT DETECTION

With behaviors for each component defined, SPARCS simulated the *William Chadwick*'s 14-day operational profile. Figure 8 shows a plot of the phase A real power of the port and starboard panels, both simulated and collected by the NILMs over the same period. Below the power profiles is a time history of the corresponding weather data and the ship's mission timeline. A qualitative inspection reveals similar loading behavior between the NILM and SPARCS-simulated data. Around day 21, both starboard power profiles are inversely correlated to the outside temperature, with lower temperatures resulting in increasing heating loads. During daylight hours the water heater and sewage system are in operation more than at night. The air compressor draws a large load during periods where the crew has the humidifier active. The behavioral model does not provide a perfect recreation of the power system, nor does it seek to. For example, while periods of time where the air compressor is online are captured, its exact turn-on and turn-off times within these periods do not precisely match the real compressor's behavior. Instead, behavioral simulation's performance is best judged with statistical metrics.

Unbalanced per-phase power is an important metric for operation that also serves as a useful metric for comparison between simulated and recorded data. Unbalanced loading on generators can indicate voltage imbalances, which can lead to increased generator vibration and maintenance requirements. Figure 9 presents a sample period from the starboard machinery panel as simulated by SPARCS and recorded by the panel's NILM. Several line-to-line single-phase loads cycle in both simulated and recorded data, leading to fluctuating levels of phase balance.

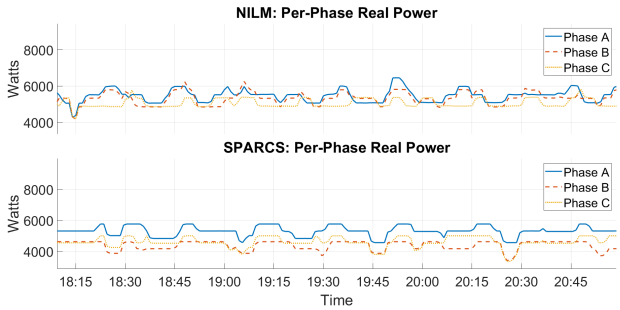


FIGURE 9: A sample period from day 1 of the operational period simulated on the *William Chadwick*, where all three phases on the starboard panel are shown to fluctuate through different magnitudes of imbalance.

TABLE 2: Imbalance mean and standard deviation for the port machinery panel, with each of the components modeled in their ideal state.

Statistic	NILM	SPARCS	% Error
Mean	17.79	0.27	98.48
Standard Deviation	1.37	0.45	67.15

Table 2 shows the mean and standard deviation of the

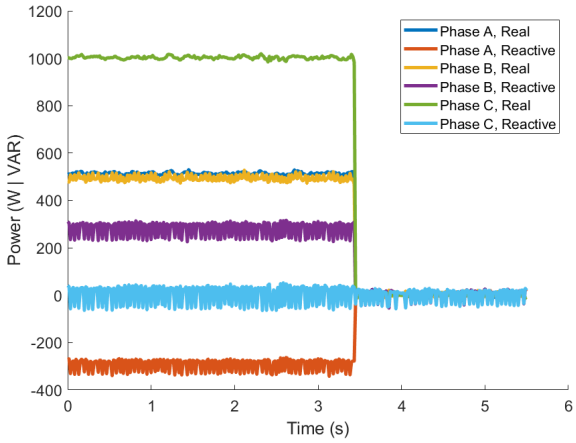


FIGURE 10: Sample power traces from a faulted heater during a turn-off event.

FIGURE 11: The faulted three-phase space heater found onboard the *William Chadwick*.

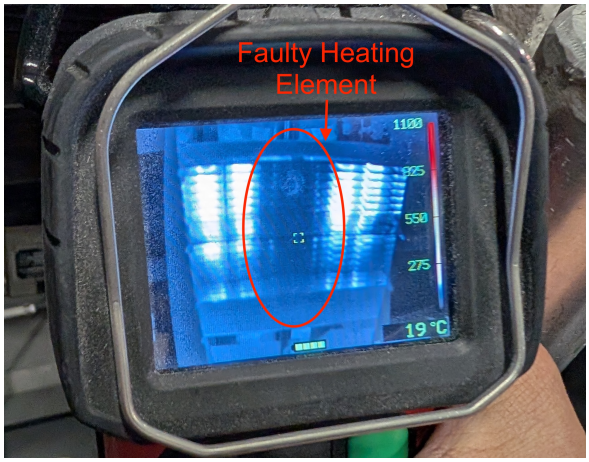


FIGURE 12: Thermal inspection of the faulted heater revealed the center two heating elements were not energized.

phase imbalance on the port panel, calculated over the 14-day operational period. The following equation computes phase imbalance for phase ϕ:

$$\text{Imbalance} = \frac{S_{\phi,max} - S_{\phi,min}}{\frac{S_A+S_B+S_C}{3}}, \tag{7}$$

where S_ϕ is the apparent power. Percent error is calculated with respect to the NILM measurements according to:

$$\text{Error} = \frac{|\text{NILM} - \text{SPARCS}|}{\text{NILM}}. \tag{8}$$

Interestingly, the 17.8% average imbalance measured by the NILM on the port panel is not reflected in the data simulated by SPARCS. According to the one-line diagram, each component on the port panel *should* be a balanced three-phase load, being predominantly pumps and heaters. The presence of a significant phase imbalance in the NILM data therefore suggests the possibility of a faulty load.

Visual inspection of the NILM data on the port panel revealed the unbalanced conditions were primarily due to a load whose turn-off event is shown in Figure 10. This load draws approximately 1000 W on phase C, and 500 W on phases A and B. It also appears to draw -300 Var of reactive power on phase A, and 300 Var on phase B. This profile is consistent with the per-phase power consumption from two single-phase resistive loads across phases B/C and C/A [29]. The profile is also consistent with a line-line open fault on a three-phase heater [27].

An in-person inspection revealed that the lower forward auxiliary space heater (pictured in Figure 11) had only two of its three heating elements energized. This heater is responsible for preventing freezing in pipes and maintaining crew comfort during the winter. The fault was confirmed with a thermal imaging camera, as shown in Figure 12. The behavioral model in SPARCS was then updated to use the faulted power traces of the heater in Figure 10. The imbalance statistics were recalculated, as shown in Table 3. After accounting for the fault, the behavioral model aligns much more closely with the observed data. Table 4 shows a comparison of the simulated and recorded mean, standard deviation, minimum, and maximum loading observed over the entire timeline.

C. CASE STUDY: ENVIRONMENTAL CONDITIONS

The behaviors of the loads on the port and starboard machinery panels are generally more sensitive to the ship's external environment than to the ship's task at hand. To evaluate the behavioral model's ability to capture these influences, the same summary statistics (mean, standard deviation, minimum, and maximum) are calculated for phase A real power for different ranges of two variables: outside temperature and daylight, as shown in Tables 5 and 6, respectively. Daylight is quantified between -1 to 1, where -1 corresponds to the sun at its lowest point below the horizon, and 1 corresponds to the sun at its highest point in the sky. Four bins of daylight conditions

TABLE 3: Imbalance mean and standard deviation for the port machinery panel, with the broken 3-phase heater modeled in its faulted state.

Statistic	NILM	SPARCS	% Error
Mean	17.79	17.31	2.68
Standard Deviation	1.37	1.51	10.00

TABLE 4: Loading statistics comparison of the NILM sensor data and SPARCS-simulated data from phase A. These statistics are calculated over the entire 14-day operational period.

Statistic	Panel	NILM *(kW)*	SPARCS *(kW)*	Percent Error *(%)*
Mean	Port	2.80	2.82	0.62
	Starboard	5.97	6.42	7.68
Standard Deviation	Port	0.30	0.30	0.68
	Starboard	1.19	1.21	2.04
Minimum	Port	2.08	2.02	2.67
	Starboard	3.52	3.64	3.36
Maximum	Port	8.00	7.58	5.26
	Starboard	11.22	12.84	14.42

were assessed: $[-1,-0.5)$, $[-0.5,0)$, $[0,0.5)$, and $[0.5,1]$, representing 29.2%, 20.8%, 12.5%, and 37.5% of the assessed operational profile. Periods around sunset and sunrise experienced higher variance in loading due to increased water heater and sewage system operation when the crew wakes up and goes to sleep. The 39% error in maximum loading during the $[0.5,1]$ daylight window is likely because the water heater was not used. A longer simulation period would likely reveal an instance of its usage, as its behavior was modeled as a random process.

IV. SIMULATION SPEED PROFILING

Both the behavioral framework and parallelization offer significant speedups in power system simulation. This section presents an empirical characterization of a parallel behavioral simulator's time complexity, as implemented with SPARCS.

A. SIMULATION TIME ON THE WILLIAM CHADWICK MODEL

Table 7 presents the results of timing tests under several different parameters. First, two machines are considered. Machine A is a 13th Gen Intel i7-13700 2.10 GHz processor with 16 CPU cores. Machine B is a personal laptop, equipped with a 10-core 13th Gen Intel i7-1355U 1.70 GHz processor. The simulation is also processed in both a single-core serialized and multicore parallelized manner. Finally, two models are simulated: a full *William Chadwick* model with all 200+ loads, and a simplified version with only the port and starboard panels monitored by the installed NILMs. Simulations were timed on the machine while no other tasks were being performed, and averaged over three trials. Parallelizing the simulation on machine A resulted in a 7.35x speed increase, and a 3.3x increase on machine B. This difference can be attributed to the different number of cores in each machine. Machine A has 8 cores with a maximum frequency of 5.10 GHz and 8 cores with a maximum frequency of 4.10 GHz.

TABLE 5: Loading statistics on the starboard machinery panel phase A calculated for different temperature ranges.

Statistic	Temperature *(Celsius)*	NILM *(kW)*	SPARCS *(kW)*	Percent Error *(%)*
Mean	< 0	5.90	6.65	12.73
	≥ 0	6.07	6.09	0.36
Standard Deviation	< 0	0.97	0.97	0.70
	≥ 0	1.47	1.44	2.02
Minimum	< 0	3.89	4.09	5.08
	≥ 0	3.52	3.64	3.36
Maximum	< 0	10.45	12.84	22.81
	≥ 0	11.22	12.84	14.41

TABLE 6: Loading statistics on the port machinery panel phase A calculated for different daylight hours.

Statistic	Daylight	NILM *(kW)*	SPARCS *(kW)*	Percent Error *(%)*
Mean	$[-1,-0.5)$	2.78	2.82	1.17
	$[-0.5,0)$	2.83	2.83	0.06
	$[0,0.5)$	2.83	2.81	0.71
	$[0.5,1]$	2.80	2.83	0.98
Standard Deviation	$[-1,-0.5)$	0.24	0.23	7.47
	$[-0.5,0)$	0.39	0.43	9.92
	$[0,0.5)$	0.39	0.35	10.13
	$[0.5,1]$	0.24	0.23	2.48
Minimum	$[-1,-0.5)$	2.09	2.02	3.24
	$[-0.5,0)$	2.09	2.02	3.16
	$[0,0.5)$	2.08	2.02	2.67
	$[0.5,1]$	2.08	2.02	2.76
Maximum	$[-1,-0.5)$	5.92	5.07	14.38
	$[-0.5,0)$	6.75	7.58	12.30
	$[0,0.5)$	7.54	7.58	0.55
	$[0.5,1]$	8.00	4.87	39.18

Machine B has only two 5.00 GHz cores and eight 3.7 GHz cores.

B. EMPIRICAL TIME COMPLEXITY EVALUATION

A set of nominal benchmark power networks was constructed to evaluate how simulation time grows with an increasing number of system loads. These networks had 20, 40, 60, 80, or 100 nominal loads, each defined with the same cycling behavior. Each network had a ring topology with 0, 1, 2, or 3 rings. The networks were simulated over a nominal two week period, and the solution was computed at a 1 Hz sample rate. All networks were simulated on Machine A, with the zero-ring network simulated in both parallel and series. The results are presented in Figure 13.

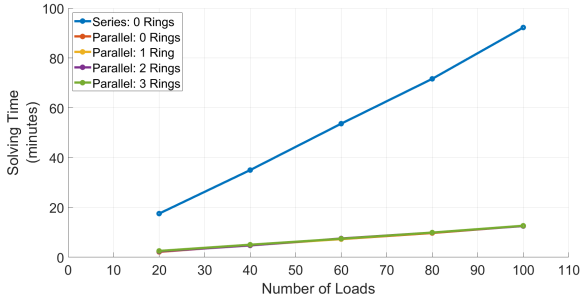
As made evident by Figure 13a, parallelization techniques dramatically decreased the simulation time. The same 100-load network required approximately 7.5x less time when processed in parallel. In addition, the solving time of all networks grows in an approximately linear fashion. Introducing a ring to the network minimally increased the solution time, and the solving time scales roughly linearly independent of the number of rings.

V. CONCLUSION

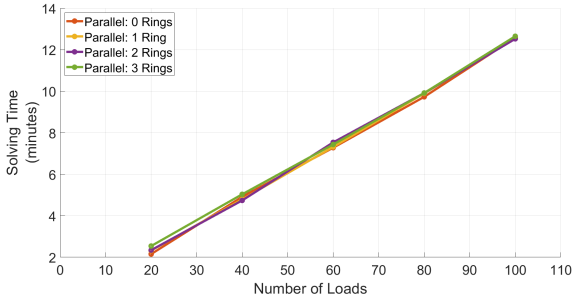
Behavioral methods offer fast simulation of microgrid operation. This paper demonstrates that with proper organization, behavioral simulation can be made naturally parallelizable,

TABLE 7: Results of the timing test matrix for SPARCS, evaluated across different machines, the number of processing cores used, and model size. Each simulation was performed over the same 2-week operating profile with a 1 Hz sample rate.

Machine	Cores	Model Size	Average Simulation Time (minutes)
A	Serial	Large	163.4
A	Serial	Small	27.2
A	Parallel	Large	22.1
A	Parallel	Small	3.4
B	Serial	Large	182.7
B	Serial	Small	30.2
B	Parallel	Large	54.3
B	Parallel	Small	8.94



(a) Simulation time with and without parallelization.



(b) Simulation time with the addition of a ring bus.

FIGURE 13: The simulation time of behavioral methods converges to a linear function of the number of loads in the network. The simulation time remains roughly linear in the number of loads with the introduction of a ring bus.

maximizing the throughput on modern CPUs. As an example application, rapid microgrid simulation enables Monte Carlo-style simulations to evaluate extreme loading conditions. Operating states can be modeled under designated fault conditions to evaluate which modes of operation are still viable under these conditions. As demonstrated in this work, faulted and healthy load states can be modeled to identify sources of a fault.

ACKNOWLEDGMENT

The authors gratefully acknowledge the US Coast Guard and in particular the crew of the USCGC *William Chadwick*. This work was made possible through the generous support of several groups and individuals, namely: the Department of Mechanical Engineering at MIT through the MathWorks and Chryssostomidis fellowships; the Office of Naval Research NEPTUNE program and the inspired leadership of Dr. Scott Higgins, Dr. Corey Love, and Maria Medeiros; and the Government of Portugal through the Portuguese Foundation for International Cooperation in Science, Technology and Higher Education. This work was undertaken in the MIT Portugal Program. This work is supported in part by the Office of Naval Research Grant No. N00014-21-1-2124.

REFERENCES

[1] S. Subedi, M. Rauniyar, S. Ishaq, T. M. Hansen, R. Tonkoski, M. Shirazi, R. Wies, and P. Cicilio, "Review of methods to accelerate electromagnetic transient simulation of power systems," *IEEE Access*, vol. 9, pp. 89 714–89 731, 2021.

[2] P. Maffezzoni and G. Gruosso, "Complex-array-operation newton solver for power grids simulations," *IEEE Access*, vol. 8, pp. 47 984–47 992, 2020.

[3] R. D. Zimmerman, C. E. Murillo-Sánchez, 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, 2011.

[4] U. Orji, B. Sievenpiper, K. Gerhard, S. B. Leeb, N. Doerry, J. L. Kirtley, and T. McCoy, "Load modeling for power system requirement and capability assessment," *IEEE Transactions on Power Systems*, vol. 30, no. 3, pp. 1415–1423, 2015.

[5] D. W. Van der Meer, J. Widén, and J. Munkhammar, "Review on probabilistic forecasting of photovoltaic power production and electricity consumption," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1484–1512, 2018.

[6] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE suite," *SIAM journal on scientific computing*, vol. 18, no. 1, pp. 1–22, 1997.

[7] COMSOL, *Introduction to COMSOL Multiphysics*, 2019, vol. 5.5.

[8] Steady State Power Flow. General Electric Vernova. 2025.

[9] CYME Power Flow Analysis Module. Eaton. 2025.

[10] PSSE Power Simulator. Siemens. 2025.

[11] R. AhmadiAhangar, A. Rosin, A. N. Niaki, I. Palu, and T. Korõtko, "A review on real-time simulation and analysis methods of microgrids," *International Transactions on Electrical Energy Systems*, vol. 29, no. 11, p. e12106, 2019.

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Fourth Edition.* The MIT Press, 2022, vol. Fourth edition.

[13] N. R. Council, *The Future of Computing Performance: Game Over or Next Level?*, S. H. Fuller and L. I. Millett, Eds. Washington, DC: The National Academies Press, 2011.

[14] T. Gunarathne, T.-L. Wu, J. Qiu, and G. Fox, "Cloud computing paradigms for pleasingly parallel biomedical applications," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*, ser. HPDC '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 460–469.

[15] E. T. Almquist, "Behavioral methods for next-generation shipboard power system simulation: Letting SPARCS fly," Master's thesis, Massachusetts Institute of Technology, May 2025.

[16] W. Schilders, *Introduction to Model Order Reduction*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 3–32.

[17] Y. Zhao, C. Jiang, M. A. Vega, M. D. Todd, and Z. Hu, "Surrogate modeling of nonlinear dynamic systems: A comparative study," *Journal of Computing and Information Science in Engineering*, vol. 23, no. 1, p. 011001, 05 2022.

[18] A. Afzal, Z. Ansari, A. R. Faizabadi, and M. Ramis, "Parallelization strategies for computational fluid dynamics software: state of the art review," *Archives of Computational Methods in Engineering*, vol. 24, no. 2, pp. 337–363, 2017.

[19] N. Tricard, G. C. Fraga, and X. Zhao, "Optimal parameters of monte carlo ray tracing solver with line-by-line spectral database for radiation modeling in fire," *Proceedings of the Combustion Institute*, vol. 40, no. 1, p. 105293, 2024.

[20] J. Drake, I. Foster, J. Michalakes, B. Toonen, and P. Worley, "Design and performance of a scalable parallel community climate model," *Parallel Computing*, vol. 21, no. 10, pp. 1571–1591, 1995.
[21] J.-L. Lions, Y. Maday, and G. Turinici, "A "parareal" in time discretization of PDE's," *Comptes Rendus de l'Académie des Sciences - Series I - Mathematics*, vol. 332, no. 7, pp. 661–668, 2001.
[22] T. Deeter, D. H. Green, S. Kidwell, T. J. Kane, J. S. Donnal, K. Vasquez, B. Sievenpiper, and S. B. Leeb, "Behavioral modeling for microgrid simulation," *IEEE Access*, vol. 9, pp. 35 633–35 645, 2021.
[23] D. Monagle, T. C. Krause, A. W. Langham, and S. B. Leeb, "Energy storage design for energy harvesting sensors," *IEEE Sensors Journal*, vol. 25, no. 13, pp. 24 614–24 625, 2025.
[24] J. Paris, J. S. Donnal, R. Cox, and S. Leeb, "Hunting cyclic energy wasters," *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2777–2786, 2014.
[25] M. L. Pinedo, *Parallel Machine Models (Deterministic)*. Cham: Springer International Publishing, 2022, pp. 115–152.
[26] E. T. Almquist, S. J. Bruno, A. W. Langham, M. C. Buchanan, T. Powell, S. B. Leeb, and D. H. Green, "Fast shipboard electric power systems simulation: Applications of behavioral modeling," in *ASNE Intelligent Ships Symposium*, 2025.
[27] J. D. Skimmons, "Distributed sensors, data analysis, and non-intrusive load monitoring: Foundations for reliability-centered maintenance on ships," Master's thesis, Massachusetts Institute of Technology, 2024.
[28] E. T. Almquist, A. W. Langham, A. I. Martínez, S. J. Bruno, T. C. Krause, M. C. Buchanan, and S. B. Leeb, "Behavioral simulation techniques for the next generation of shipboard microgrids," *Naval Engineers Journal (to appear)*, 2025.
[29] D. H. Green, P. A. Lindahl, and S. B. Leeb, "Three-phase electrical measurement representations for nonintrusive load diagnostics," *IEEE Open Journal of Instrumentation and Measurement*, vol. 1, pp. 1–14, 2022.

ETHAN T. ALMQUIST (Graduate Student Member, IEEE) received the M.S. degree at MIT in Mechanical Engineering in 2025. He received his B.S.E. degree from the University of Michigan in 2023, studying naval architecture and marine engineering with a minor in computer science.

AARON W. LANGHAM (Graduate Student Member, IEEE) received the B.E.E. degree in electrical engineering from Auburn University in 2018, and the M.S. and E.E. degrees in electrical engineering and computer science from MIT in 2022 and 2024, respectively. He is currently pursuing the Ph.D. degree in electrical engineering and computer science at MIT. His research interests include signal processing, machine learning, and IoT platforms for energy systems.

STEPHEN J. BRUNO is a Lieutenant in the U.S. Coast Guard. He is currently pursuing a M.S. in Naval Architecture and Marine Engineering at the Massachusetts Institute of Technology. He was previously stationed as Damage Control Assistant aboard USCGC POLAR STAR and as a Port Engineer for the National Security Cutter (NSC) fleet.

STEVEN B. LEEB (Fellow, IEEE) received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1993. He has served as a Commissioned Officer in the USAF reserves, and he has been a member of the MIT Faculty in the Department of Electrical Engineering and Computer Science, since 1993. He also holds a joint appointment in MIT's Department of Mechanical Engineering. He is the author or coauthor of over 200 publications and 20 U.S. Patents in the fields of electromechanics and power electronics.