In power systems, the traditional, non-interactive, and manually-controlled power grid has been transformed to cyber-dominated smart grid. This cyber-physical integration has provided the smart grid with communication, monitoring, computation, and controlling capabilities to improve its reliability, energy efficiency, and flexibility. A microgrid is a localized and semi-autonomous group of smart energy systems which utilizes the above-mentioned capabilities to drive modern technologies such as electric vehicle charging, home energy management, and smart appliances. Design, upgrading, test, and verification of these microgrids can get too complicated to handle manually. The complexity is due to the wide range of solutions and components that are intended to address the microgrid problems. This paper presents a novel Model-Based Design (MBD) methodology to model, co-simulate, design, and optimize microgrid and its multi-level controllers. This methodology helps in the design, optimization, and validation of a microgrid for a specific application. The application rules, requirements, and design-time constraints are met in the designed/optimized microgrid while the implementation cost is minimized. Based on our novel methodology, a design automation, co-simulation, and analysis tool, called GridMAT, is implemented. Our experiments have illustrated that implementing a hierarchical controller reduces the average power consumption by 8% and shifts the peak load for cost saving. Moreover, optimizing the microgrid design using our MBD methodology considering smart controllers has decreased the total implementation cost. Compared to the conventional methodology, the cost decreases by 14% and compared to the MBD methodology where smart controllers are not considered, it decreases by 5%.
over, renewable energy sources and different types of energy storage called Distributed Energy Resources (DER) have been aggregated to the power grid in order to meet the regular power demand. However, the introduction of the DER and new types of electricity loads (e.g., Electric Vehicles (EV)) add the challenge of multi-level supply and demand management to an already complex power grid [Al Faruque and Vatanparvar 2016; Vatanparvar and Al Faruque 2015a; Vatanparvar et al. 2015a].

The power grid consists of various electrical components in multiple levels (e.g., transmission and distribution). In the distribution grid, a microgrid is a localized group of distributed electrical energy resources (storage and generation) and loads (appliances) that may operate independently (island mode) or connect to the main power grid (macrogrid) [Berkeley 2015; Al Faruque 2014; Vatanparvar and Al Faruque 2015b]. The primary purpose of a microgrid is to ensure local, reliable, and affordable energy security for communities. For instance, microgrids have been deployed significantly at residential areas besides military installations, critical infrastructure areas (e.g., hospitals), commercial, and university locations [Shao et al. 2009; Shao et al. 2010]. Consumer EV, roof-top photovoltaic systems, residential-scale energy storage, and smart flexible appliances constitute the driving technologies in forming a residential microgrid [Jafari 2012]. The power demand and supply of these components might be under the monitor and control of multiple energy management techniques such as: Home Energy Management (HEM), microgrid-level energy management, Time-Of-Use (TOU) rate scheme, and Demand Response (DR) [Siano 2014; Aghaei and Alizadeh 2013; Al Faruque et al. 2012; Pipattanasomporn et al. 2012]; home and microgrid-level energy managements monitor the power demand and control the energy consumption of multiple components at high level for meeting a specific goal (e.g., reducing energy consumption or cost). DR in power grids is a dynamic demand mechanism to manage power consumption in response to supply conditions for balancing the supply and demand of power. In a TOU rate scheme, the electricity price increases during the time of day when the power demand is high giving incentives to the customers for shifting their power demand to other time of a day when more power supply is available and energy cost is lower. As the case study of this paper, we have mainly focused on the residential microgrid.

Rapid addition of new DER and types of electricity loads makes the power grid design more complex, time-consuming, and more challenging to detect, preempt, and address microgrid problems [Nassif et al. 2014]. This phenomenon has been seen before in Integrated Circuits (IC) design. To keep up with the Moore’s law in the semiconductor industry, various Electronic Design Automation (EDA) methods and tools have been successfully deployed to tackle complexity, cost, time-to-market, and heterogeneity. [Al Faruque and Canedo 2012; Vatanparvar et al. 2015b; Schaller 1997; Lavagno et al. 2006; Wang et al. 2009; Blume et al. 2002; Gajski et al. 1994]. Among various methods, EDA has facilitated us with improved modeling, simulation, validation, and optimization methods for IC design. The authors believe and also demonstrated in this paper that in the aspect of smart grids, automating the microgrid design process may help the designers such as utilities, aggregators, and energy cooperatives to handle the complexity and increase the reliability of their designed microgrid. Hence, Model-Based Design (MBD) which has been proposed in the literature [Ilic et al. 2010; Jensen et al. 2011] enables modeling both the physical and cyber components concurrently. Moreover, it enables cyber-physical co-simulation in order to explore various design alternatives and implement a correct-by-construction design [Saxena et al. 2003]. In other words, microgrids may be virtually analyzed and smart controllers may be developed without the need for physical prototypes because software would be able to estimate and ensure the dynamic behavior of the system under a large variety of conditions.
By modeling, simulating, and analyzing the microgrids, designers may implement more efficient and optimized designs more easily. However, a general-purpose microgrid might not be optimal to be used instead of a customized application-specific microgrid. In other words, knowing the dynamic power supply and demand behavior of the microgrid and its constraints may help the designers come up with a more optimal microgrid in terms of cost, efficiency, and reliability (e.g., using a uni-directional grid design for a bi-directional microgrid containing DERs). There are multiple parameters in a microgrid which should be optimized for a specific application and constraint.

As a case study, we analyzed the EV penetration influence on microgrid peak load for three different scenarios [Shao et al. 2010; Ahourai and Al Faruque 2013]:

1. **No Demand Response (No DR):** in this scenario, multiple EVs (156 or 312) are being charged without being controlled using DR.

2. **Baltimore Gas & Electricity (BG&E):** in this scenario, a TOU rate scheme has been implemented in which the electricity price increases during the time of day when the power demand is high. The scheme is used by Baltimore Gas & Electricity - a utility in Washington DC area.

3. **Dominion Virginia Power (DOM):** in this scenario, another TOU rate scheme has been implemented in which the electricity price increases during the time of day when the power demand is high. However, the rates and the timing are different from the second scenario (BG&E). The scheme is used by Dominion Virginia Power - another utility in Washington DC area.

(a) EV penetration into the microgrid and influence on the peak load.

(b) Multiple solutions to the existing problems in the microgrid.

Fig. 1. Motivational case study on a residential microgrid design.
As shown in Figure 1(a), by increasing the number of EVs in the microgrid, the peak load increases significantly. Therefore, the current grid components may not handle the demanding power and may collapse. This problem may be addressed using different methodologies; e.g., by implementing the TOU rate scheme (Figure 1(a)) or upgrading/adding grid components in different parts of the grid [Jiang et al. 2014] (Figure 1(b)). These solutions might have overlapping effects where they need to be considered thoroughly, and may get too complex to be handled by traditional methodologies of design [Nassif et al. 2014]. According to [Jiang et al. 2014], the cost of upgrading the components (e.g., transformers, wires, and regulators) is significantly high. On the other hand, this issue can also be handled by cyber solutions and implementing smart controllers on EV charging stations to manage their demanding power. Hence, all the possible solutions to the existing problems in the power grid need to be taken into account during the design.

**Summary and conclusion:** Utilities, aggregators, and energy cooperatives attempt to design a new microgrid or upgrade an existing microgrid to solve its problems. However, the above analysis illustrates that utilizing traditional methodologies may not be possible or give the optimal solutions. Moreover, knowing the microgrid application such as dynamic power demand and supply behavior and constraints may help in further optimization of the design in terms of the implementation cost, reliability, and efficiency. Furthermore, new paradigm of solutions like demand-side power management and appliance controllers, need to be considered while designing a microgrid.

## 2. RELATED WORK

Smart grid covers various scales and levels including residential microgrid. Modeling a complex, large-scale, heterogeneous, and multi-physics system like a microgrid requires heterogeneous composition of physical, computational, and communication sub-systems. Various design challenges of modeling and simulation of a microgrid have been discussed in [Meliopoulos 2002; Cheung 2012], e.g. OpenDSS [Dugan 2012], InterPSS [InterPSS 2015], and MATPOWER [Zimmerman et al. 2011]. However, these tools lack the framework to simulate an energy distribution model like a residential microgrid where a power system modeling at a very detailed level (e.g., appliance level) is required. To solve this problem, GridLAB-D [Chassin et al. 2008], an agent-based power system modeling and simulation tool has been used. GridLAB-D is a power distribution system modeling, simulation, and analysis tool which incorporates detailed modeling libraries of the electrical systems in different levels and retail markets.

Developing an optimal microgrid is a complex task, due in part to the need for modeling and analyzing the system at both different scales and across different information domains. While various domain-specific power system simulation tools currently exist, there is a critical gap of advanced system-level design methodology and tools in modeling and simulating of the cyber and physical portions of smart grid concurrently. Authors in [Ilic et al. 2010] demonstrate a potential model of future energy systems. In [Jensen et al. 2011], a generic MBD methodology for a cyber-physical system design has been discussed. Power system modeling tools lack the capability to capture the cyber components during modeling and simulation. On the other hand, tools capable of describing the discrete-event dynamics of the cyber components are not well equipped with the models needed to represent the physical dynamics of power systems. Therefore, cyber-physical co-simulation of different domain-specific tools has been seen as the possible enabling technology [Buck et al. 1992; Wetter 2011; Al Faruque and Ahourai 2014a; 2014b].

Various MBD frameworks have been proposed for integrating the cyber and physical domains. For instance, SystemC is used for designing embedded systems in energy management [Molina et al. 2013]; MATLAB & EnergyPlus [Crawley et al. 2000] are
used for building energy automation solution (MLE+); and GridLAB-D [Chassin et al. 2008] and MATLAB are used for modeling integrated renewable energy and DR. In this co-simulation, the control developed in MATLAB is executed as a co-process of GridLAB-D and cannot support a debugging function for the control engineers. With the same spirit of connecting the cyber and physical worlds, we have developed a detailed cyber-physical microgrid co-simulation platform between GridLAB-D [Chassin et al. 2008] and MATLAB/Simulink [MathWorks 2015] using MATLAB toolbox GridMat. GridMat design automation tool may be used to facilitate the electrical engineers and control engineers to develop advanced and hierarchical smart controllers for the residential microgrid [Kleissl and Agarwal 2010; Weng and Agarwal 2012].

Multiple traditional and state-of-the-art methodologies of design and upgrading the power grid exist especially in residential microgrid considering DER [Nunna and Doolla 2013; Sousa et al. 2012; Atzeni et al. 2013]. These methodologies consider the general requirements for the power demand and supply while considering the constraints of the components, e.g., thermal capacity and voltage drop [Bahramirad et al. 2012; Wang et al. 2013]. They utilize various optimization techniques in the design in order to minimize the construction or upgrading time and cost. However, in these state-of-the-art methodologies, the requirements and constraints are too general for a microgrid that is going to be used for a specific application thereby resulting in a too general and non-efficient solution for microgrid design. Moreover, these methodologies lack the knowledge of cyber control domain and its influence on the power demand and supply behavior of the microgrid. Hence, the solution is not adjusted to the new power requirements of the microgrid in the existence of the cyber controllers.

2.1. Problem and Research Challenges
In summary, the problem of microgrid design poses the following research challenges:

1. Traditional methodologies for designing microgrid are not capable of considering all complex solutions including smart controllers especially in the cyber domain.
2. The microgrid design process is done for a general-purpose microgrid, while considering the specific application of a microgrid in the design optimization process may result in better (more optimal) solution.
3. A suitable tool has not been developed for modeling, simulating, design, and optimizing application-specific microgrids.

2.2. Our Novel Contributions and Concept Overview
To address the above-mentioned challenges, a novel MBD methodology for application-specific residential microgrid is proposed that employs:

1. **Defining the Microgrid Components and Application (Section 3)** helps in specifying the design-time constraints and rules, and predicting the voltage and power requirements for different parts of the microgrid, which may have smart controllers implemented.
2. **Model-Based Design and Optimization Methodology for an Application-Specific Microgrid (Section 4)** that designs an optimized microgrid given the application, rules, and design-time constraints, which requires:
3. **Modeling, Simulation, Design, and Validation GridMat Tool (Section 5)** that models the microgrid components and smart controllers for simulation and analysis purposes. An optimization algorithm (see Section 4.4) is applied in order to optimize, design, and implement an application-specific microgrid while meeting the design-time constraints and rules. Eventually, validating the microgrid by simulating and analyzing the design.
Figure 2 describes our novel methodology implemented in GridMat tool for model-based design of an application-specific microgrid. Designers such as utilities, aggregators, and energy cooperatives can utilize this tool to handle the complexity of the microgrid design and optimization. They can plan, design, and upgrade residential microgrids in order to meet the customer power demand or address the existing microgrid problems (e.g., thermal violation and voltage drop). A residential microgrid should be designed according to the power supply and demand requirements, connectivity between different components, design-time constraints, and rules defined by the designer. Hence, a microgrid application may represent these parameters of the microgrid design for a specific application of the microgrid. The planners and designers define the microgrid application, provide a library of the solutions to the power grid, and consider the rules and design-time constraints for the design. This data is fed to the GridMat design and optimization tool and the best design is selected and optimized for the application by modeling, co-simulation, and validation of all the possibilities.

Fig. 2. Our novel MBD methodology for application-specific microgrid.

3. SYSTEM MODELING AND ESTIMATION

As stated in Section 1, in this paper, we consider a residential microgrid as an example of a smart grid and demonstrate the capability of our MBD methodology.

**Residential Microgrid:** In traditional power grids, electricity typically flows only from large-scale generation sites (e.g., fossil-fuel and nuclear power plants) to a distribution grid through the transmission lines. However, it has been transformed to a more (bi-/multi-directional electricity flow system due to integration of various DERs including renewable electricity generating sources (see Figure 3). This shift in electricity flow, rising costs of electricity production, and a better understanding of environmental impacts, require balancing mechanisms between the electricity supply and demand. A smart microgrid has been seen as a solution which can operate in two modes: (1) grid-connected mode where the microgrid is connected directly to the power grid and (2) island mode where the microgrid can disconnect itself from the power grid so as to operate autonomously in various physical and economical conditions [Berkeley 2015]. This capability allows a microgrid to provide additional reliability to the demand side. In high-level, a residential microgrid may consist of residential loads as well as DERs (e.g., renewables and energy storage). Demand-side energy management, power quality, imbalance/asymmetry, plug-and-play operation of DER systems,
distributed voltage/frequency profile control, and non-autonomous/autonomous operation are some of the challenges in designing and operating a microgrid in the residential area [Meliopoulos 2002; Berkeley 2015]. Before starting to model a residential microgrid behavior, the requirements, components/subsystems, structure, location, and problems must be defined.

3.1. Microgrid Components

A microgrid consists of multiple connected nodes. These nodes encapsulate various grid components (e.g., generators, storage, and appliances) and generate or consume power. They may be the end-use appliances or complex buildings. Hence, they may represent hierarchical grid components and their dynamic behavior can be modeled as described in [Al Faruque and Ahourai 2014b].

The microgrid nodes are connected using edges. The edges represent the grid components transferring electricity (e.g., wires, switches, and transformers). Their characteristics are modeled to define the relationships between dynamic behavior of two/more connected nodes in terms of voltage, power, etc.

To develop and analyze a microgrid, the model of the microgrid consisting of both the structural and behavioral models must be developed according to the requirements. The structural model of a residential microgrid includes the building-architectural model of houses, end-use appliances, distributed energy resources, transformers, and distribution power grid. On the other hand, the dynamic parts of the residential microgrid such as the appliance tasks, energy demand, and weather are captured by the behavioral modeling. Hence, the microgrid model could be obtained by structural and behavioral modeling of the nodes and edges hierarchically.

3.2. Microgrid Application

During the design or upgrading a microgrid, multiple requirements are typically specified. For instance, as the power grid is growing, more possible nodes are added to the grid. These nodes will probably demand more power. The microgrid should be able to provide the nodes with the demanded power. The edges that represent the connection between nodes, should be able to meet the requirements, for instance, a wire needs to be selected such that it handles the current flow. Therefore, a microgrid application
represents these parameters specified by the planner/designer for the newly designed or upgraded microgrid:

(1) **Microgrid Topology:** illustrates how the nodes are placed within the microgrid (e.g., distance between nodes). This is used to find the length of the edges or wires.

(2) **Node Profile:** represents the design-time constraints and requirements for each microgrid node:
   - Nominal voltage: defines the working voltage, in which the node (appliance) may only operate properly.
   - Power consumption: of a node (appliance as a load). Although it is a time-varying parameter, we are considering the maximum value which is the worst case and it is sufficient as a design-time constraint.
   - Power generation: by a node (solar panel). Although, it is a time-varying parameter, we are considering the maximum value which is the worst-case and it is sufficient as a design-time constraint.
   - Power factor: is a metric defining the reactive power consumed by the node (e.g., electric motor).

(3) **Edge Profile:** represents the design-time constraints and requirements for each microgrid edge:
   - Current flow: going through an edge (wire, transformer) influences the edge characteristics (voltage drop).
   - Voltage conversion: defines the relationship between the voltages of two nodes where an edge connects two different voltage domains (transformer).

The microgrid topology is modeled using a two-dimensional matrix in which the element with \((i, j)\) index represents the required distance between \(i^{th}\) and \(j^{th}\) nodes. The node profiles are modeled as a matrix in which the element with \(i\) index represents the node profile of the \(i^{th}\) node, which contains the voltage profile and power consumption for that node. The edge profiles are modeled as a matrix in which the element with \(i\) index represents the edge profile of the \(i^{th}\) edge, which contains the current flow and voltage transfer for that edge.

### 3.3. Microgrid Design-Time Constraints and Rules

Besides the requirements defined from the consumer side, many regulating bodies and owners of the power systems set specific rules for reliability and efficiency purposes. Reliability is met by following the operational constraints set by the designers of the various elements that comprise the grid network (such as the transformers and voltage regulators), which would help maintain the longevity of these devices. Efficiency is met through both meeting reliability rules as well as examining where losses occur most significantly, either due to the system operating at a too-low voltage level, or due to an inability to maintain high power factor as desirable. **It would be considered as a microgrid problem, if one of these design constraints or rules is not met:**

---

1Steady-state grid analyses are done in the process of planning and upgrading a residential microgrid by simulating and validating the behavior of the microgrid components in terms of meeting the power and reliability requirements. Typically, the worst-case scenario is considered for the power requirements in order to prevent any mishaps in the power grid. However, it needs to be noted that in our paper the worst case might change depending on the controlling solutions applied to the nodes. Moreover, the tool is orthogonal to other cases (e.g., typical case) which can be considered in order to improve the optimality of the solution.
— **Voltage Drop:** is the voltage potential degradation across a series of grid components [Duffey and Stratford 1989], due to their inherent impedance. Devices have a limited voltage drop tolerance from the nominal voltage for proper operation. Therefore, the utility is responsible for voltage drop correction, in order to ensure a high standard reliability and power quality.

\[
\Delta V \% \leq 10\%
\]  

— **Thermal Overloading:** limits the maximum acceptable current flow (i.e., thermal limit) through devices and must be respected for longevity purposes. Although many utilities have standards to allow above-normal current flow during contingency conditions, the expected frequency of these contingencies is fairly low.

\[
I_{\text{Max}} \% = 100\%
\]

— **Power Factor:** is a metric to reflect the power efficiency. Grid components may have constraints on the power factor they are capable of operating at. Consequently, this constraint is a more useful guide towards improving power efficiency, rather than as a critical power operation criterion.

\[
PF_{\text{lagging}}^\text{Max} = PF_{\text{leading}}^\text{Max} = 0.95
\]

### 4. MODEL-BASED DESIGN METHODOLOGY

In this section, we present a MBD methodology in which a microgrid that implements run-time smart controllers is modeled, simulated, and analyzed (see Section 4.1). The library of components and smart controllers required for the microgrid design is defined (see Section 4.2). Our model-based application-specific microgrid design methodology is further explained in Section 4.3. An optimized component selection algorithm is implemented in order to minimize the implementation cost while meeting the design-time constraints and rules (see Section 4.4).

#### 4.1. Co-Simulation and Validation

The dynamic behavior of the components in different levels of the microgrid is modeled using multiple equations. After modeling both the continuous- and discrete-time dynamics of the residential microgrid, the integrated model needs to be simulated for analysis, verification, and validation according to the design requirements. By adjusting the model parameters, different behaviors may be captured through cyber-physical co-simulation allowing the designers to validate the design precisely. For a residential microgrid, we have utilized our GridMat tool (see Section 5) that supports major design automation functionality such as: 1) advanced, multi-level, and hierarchical control design using MATLAB/Simulink toolboxes; 2) user-friendly GUI for the state-of-the-art GridLAB-D - power system modeling and simulation tool; 3) structural and behavioral residential microgrid model creator using the model libraries; 4) data analysis for various scenarios; and 5) embedded C code generator in order to conduct Hardware-In-the-Loop Simulations (HILS) for the purpose of validating the fidelity of the design (see Section 5 for further details).

#### 4.2. Component and Smart Controller Library

Designing and planning a microgrid are conducted to either address the existing problems or prevent any possible problem in the future. The problems mentioned in Sec-
tion 3.3, are typical issues that utilities consider important to examine, however, the types of solutions that could be implemented are not quite as uniform. The solutions can be abstracted into adding/upgrading components or managing the load across the microgrid. Some utilities may consider just a subset of all the possible solutions, due to the limited available components or difficulty of component replacement. Besides replacement, different restricting policies may exist regarding load transfers or the addition of new circuit elements such as capacitors to resolve, for instance, power factor issues. Area restrictions may limit the possible maximum size of the capacitors. Moreover, the costs of purchasing, placing, and maintaining the grid components limit the range of the component alternatives.

Various grid components are necessary for applying a possible solution to the microgrid. The availability of the components limit the designer's option and our methodology towards applying the solutions and optimizing the microgrid. Component libraries are typically split based on their types. For instance, $\Psi_{WU}$ is a wire library used for upgrading the wires in order to solve voltage drop problem or thermal overload problem and $\Psi_{CA}$ is a capacitance library used for solving possible power factor problems. Replacing components is not the only solution to the possible microgrid problems. Implementing smart controllers within nodes may be a more appropriate solution by managing the load. Hence, a library of smart controllers ($\Psi_{SC}$) is also provided.

Besides the specifications of the components and smart controllers, their implementation cost is another criterion that should be considered during the microgrid design.

4.3. Application-Specific Microgrid Design

The microgrid application (see Section 3.2) consists of the design-time parameters for the microgrid and is an input to our design methodology. It describes the node profiles, edge profiles, and microgrid topology for the specific application of the microgrid.

Design flow for a microgrid based on a specific application is illustrated in Figure 4:

![Application-Specific Microgrid Design Flow](image)

The microgrid application is used to model and describe the functionality of the microgrid in abstract level (steps 1-2); in the first block, a microgrid is designed using the simplest components (zero-cost) just to describe the high-level functionality of the microgrid, e.g., using a zero-cost wire to represent the connection between two nodes.

The pseudocode in Algorithm 1 illustrates how a microgrid is planned in abstract level based on a specific application. The input to the algorithm is the microgrid application ($A$) described in Section 3.2 and the output is the planned microgrid model ($M$).
In lines 1-2, the nodes and edges specified by the application are imported to the designated vectors for microgrid nodes (N) and edges (E). The topology of the microgrid is imported as a matrix, in which an element defines the distance between two nodes (line 3). In lines 4-16, the new nodes that are not connected by any edge, will be connected to the best possible node based on the microgrid topology (T). The disconnected node (n) will be paired with another possible node (m), if they have the same voltage profile (line 8) and have the least distance from each other (lines 9-11). A new edge between the best node (target) and n is assigned to E (line 15). Eventually, the created abstract-level microgrid (the planned microgrid returned in line 19) is simulated (step 2) to verify that it is functioning properly as required, e.g., the nodes are connected properly, not necessarily according to the design-time constraints (see Section 3.3). If the validation fails, the microgrid should be corrected and simulated again (step 1-2).

ALGORITHM 1: Abstract-Level Microgrid Planning

Input: Microgrid Application A
Output: Planned Microgrid M

1. $E \leftarrow A[edges]$
2. $N \leftarrow A[nodes]$
3. $T \leftarrow A[topology]$
4. foreach $n$ in $N$ do
5.   if $n$ is not in $E$ then
6.     min = inf
7.     foreach $m \neq n$ in $N$ do
8.       if $m[voltage] = n[voltage]$ then
9.         if $T[m, n] \leq min$ then
10.            target = $m$
11.            min = $T[m, n]$
12.       end
13.     end
14.   end
15. $E \leftarrow edge(target, n)$
16. end
17. $M \leftarrow [N, E]$
18. return $M$

After validating the microgrid functionality, the microgrid should be analyzed and its logic should be validated according to the constraints; the microgrid needs to be optimized and its possible problems need to be solved using the existing libraries (steps 3-4). The possible solutions such as upgrading/adding microgrid components or implementing smart controller are applied to the microgrid in order to meet the design-time constraints and rules of the application, while minimizing the implementation cost (microgrid optimization). The design-time constraints and rules (see Section 3.3) will be validated at step 3, and the microgrid will be further optimized or revised if required at step 4 (see Section 4.4). Eventually, the optimized microgrid will be simulated and validated again at step 5. Validation failure in this step represents that the microgrid optimization could not solve all the problems in the microgrid using the current library. Therefore, the microgrid placement (planning) should be corrected. By re-planning the microgrid components, adjustments and optimization might be needed for the microgrid edges or even the microgrid nodes.
4.4. Microgrid Optimization

In the process of designing a cost-optimal application-specific microgrid, we need to decide which solutions to apply to the microgrid (e.g., upgrading components or smart controllers) while meeting design-time constraints and rules. This optimization is described as follows:

We define state variables in matrix $X_{WU}$ to represent the wire assignments to edges; $X_{WU}[i,j]$ equals to one if wire $j$ is selected for edge $i$ from the library of wires. State variables in matrix $X_{CA}$ are defined for capacitance assignments to nodes in the microgrid. The state variables in matrix $X_{SC}$ are also defined for assigning smart controllers to nodes.

Constant matrices $\Gamma_{WU}$, $\Gamma_{CA}$, and $\Gamma_{SC}$ are defined to specify the costs involved in upgrading/adding component or smart controller. $\Gamma_{WU}[i,j]$ ($\Gamma_{CA}[i,j]$) represents the cost of upgrading from wire (capacitance) $i$ to wire (capacitance) $j$. Element $\Gamma_{SC}[i]$ represents the cost of implementing the smart controller $i$. Hence, the total cost of applying solutions (i.e., implementation cost) is evaluated as:

$$Cost = \sum(\text{diag}(X_{WU}^0 \Gamma_{WU} X_{WU}^T)) + \sum(\text{diag}(X_{CA}^0 \Gamma_{CA} X_{CA}^T)) + \sum(\Gamma_{SC} X_{SC}^T)$$

where the matrices with 0 superscript represent the existing assignment. The matrices with $T$ superscript represent their transposed matrix. $\text{diag}$ function extracts the matrix’s diagonal elements as a vector. $\sum$ function adds all the elements in the vector. In this equation, the cost of changing the wires or capacitance (upgrading) is calculated by multiplying the assignment and cost matrices.

**Algorithm 2:** Function of Non-Linear Constraints in Optimization

```plaintext
Input: Microgrid $M$
Input: Assignments $X_{WU}, X_{CA}, X_{SC}$
Output: Constraints Satisfied bool
1 $M = \text{Apply}(M, X_{WU}, X_{CA}, X_{SC})$
2 $N \leftarrow M[\text{nodes}]$
3 $E \leftarrow M[\text{edges}]$
4 $[V_N, I_E] = \text{Simulate}(M)$
5 $[V_{\text{Drop}}, PF] = \text{Evaluate}(V_N, I_E, N, E)$
6 foreach $n \in N$ do
7     if $V_{\text{Drop}}[n] \geq V_{\text{Drop}}^{\text{Max}}$ then return -1 if $PF[n] \leq PF_{\text{lagging}}$ then return -1
8 end
9 foreach $e \in E$ do
10     if $I_e[n] \geq I_e^{\text{Max}}$ then return -1
11 end
12 return 0
```

The pseudocode in Algorithm 2 illustrates how the satisfaction of non-linear constraints is evaluated for the optimization problem. The function is defined by the design-time constraints and rules. To check the constraints, the microgrid $M$ and the

---

2In some situations, the nodes may not need a capacitance or smart controller, hence they are assigned to a null element in the library.
state variables \((X_{WU}, X_{CA}, X_{SC})\) are passed to the function. In order to check the microgrid, we need to first apply the selected solutions to the microgrid (line 1). Apply function replaces the components existing in the microgrid \(M\) (e.g., capacitance, wire, or smart controller) with the new components selected in the state variables. The new microgrid is simulated and grid states are evaluated (lines 2-5). Based on the new grid states, the constraints are checked for the nodes (lines 6-9) and edges (lines 10-12). Meanwhile, if any constraint is not satisfied, \(-1\) is returned, otherwise, if all the constraints are satisfied, \(0\) is returned at the end (line 13).

The optimization problem is formulated as follows:

\[
\begin{align*}
\text{minimize} & \quad \text{Cost} \left( X_{WU}, X_{CA}, X_{SC}, X_{WU}^0, X_{CA}^0 \right) \\
\text{subject to} & \quad \left[ X_{WU}^0, X_{CA}^0 \right] = \text{Extract from} \ (M) \\
& \quad X_{WU}(i, j) \in \{0, 1\} \quad 1 \leq i \leq |\Psi_W| \quad 1 \leq j \leq |\Psi_E| \\
& \quad X_{CA}(i, j) \in \{0, 1\} \quad 1 \leq i \leq |\Psi_C| \quad 1 \leq j \leq |\Psi_N| \\
& \quad X_{SC}(i) \in \{0, 1\} \quad 1 \leq i \leq |\Psi_{SC}| \\
& \quad \text{Const.} \ (M, X_{WU}, X_{CA}, X_{SC}) = 0.
\end{align*}
\]

where \text{Extract from} function initializes the microgrid current state variables (existing wires and capacitance), as the input to the \text{Cost} function (see Eq. 4). The constraints should always be satisfied, hence, the \text{Const.} function (Algorithm 2) should return zero. And, the target is to minimize the implementation cost by changing optimization knobs, the state variables for component assignments while meeting the constraints.

5. CO-SIMULATION AND DESIGN AUTOMATION TOOL

5.1. GridMat Co-Simulation Feature

GridLAB-D is a promising state-of-the-art tool for modeling a distribution power grid, especially a residential microgrid. GridLAB-D is an open-source, agent-based, multi-domain (e.g., power, market, weather, and built-environment domain such as building-related structural parameters) modeling and simulation tool developed by PNNL [Chassin et al. 2008]. Discrete Event (DE) model of computation, and agent-based simulation enable GridLAB-D to model and simulate a large scale distribution power grid at different levels of granularity which allows for the modeling of the end-use appliances necessary for a residential microgrid. However, GridLAB-D is limited at modeling and designing the discrete dynamics required for the microgrid management (e.g., the smart controllers). GridMat features a multi-tool/multi-domain co-simulation platform to overcome the limitation of modeling and simulation of a cyber-physical distribution power grid. GridMat deploys the full advantages of GridLAB-D to model and simulate the physical plant (power system), and takes full advantage of MATLAB/Simulink to control the physical plant of a smart grid (e.g., residential microgrid). It leverages a novel GridLAB-D and MATLAB interfacing technique through a TCP/IP port by exploiting the existing HTTP server hosted in GridLAB-D. Hence, we have put MATLAB as the master of the simulation and it has full control of the simulation. Therefore, it supports fine-grained debugging of the developed control algorithms.

Figure 5 shows various sequences of the co-simulation. During the co-simulation, the GridMat core acts as the master of the simulation. It starts and pauses the simulation on GridLAB-D at each step time. In the beginning of each step, GridMat reads the sensor values from GridLAB-D. After that, GridMat executes the controller in Matlab with new sensor values. The controller then computes the new controlling signals ac-
5.2. GridMat Design Feature

For integrating the model-based application-specific design feature into our GridMat tool, the algorithms and equations specified in Sections 4.3 and 4.4 have been implemented using MATLAB. In our MBD methodology, the GridMat is used to model the behavior of the microgrid and its components with their smart controllers implemented. The optimizer in the design algorithm utilizes the co-simulation feature described in the previous subsection to apply different solutions, simulate, and analyze the microgrid. The simulation and analysis help the optimizer to adjust the state variables and validate the designed microgrid according to the design-time constraints while minimizing the implementation cost (see Section 4.4). The GridMat tool has been open-sourced for the research community and provided in [AICPS 2015].

6. EXPERIMENTAL RESULTS AND ANALYSIS

6.1. Microgrid Controller Implementation and Co-Simulation

We demonstrate the features of co-simulation and control implementation in our tool and methodology using various applications of a residential microgrid.

Experimental Setup: a detailed residential microgrid is modeled. In this model, residential single family houses are distributed in the IEEE 13 node test feeder [Subcommittee 2015]. The houses contain a variety of end-use appliances, such as dishwasher, lights, water heater, plug load (miscellaneous), refrigerator, clothes washer, dryer, and oven [Al Faruque and Ahourai 2014b]. Moreover, the weather in Newark, NJ, USA is simulated for both Summer and Winter seasons.

As we discussed in Section 1, peak load reduction, power quality, and voltage drop are some of the challenges for current residential microgrids. Design of an application-specific microgrid and implementing smart controllers are the ways to tackle such challenges. Therefore, we implement distributed, multi-level, and hierarchical smart controllers developed in GridMat by using different MATLAB/Simulink toolboxes such as Model Predictive Control (MPC) and Stateflow.

Figures 6 to 9 show experimental results obtained by implementing the smart controllers. The design automation tool has selected them efficiently to achieve a complex microgrid management capability.

Figure 6 shows a Supervisory Control And Data Acquisition (SCADA) controller, where the power consumption of selected devices is decreased in response to DR signal from the utility; a residential-level microgrid control algorithm is implemented where 5 houses are connected to a 25 KVA transformer and only one house has EV charger.
with rate 15A @ 240 V. This residential microgrid has prior contract with the utility to respond to DR signals and therefore a microgrid-level controller may initiate direct load/appliance control from the transformer-level controller in each house of the microgrid based on a pre-defined priority-based schedule. As shown in Figure 6, the microgrid controller works during a DR signal from utility to save 3 KW of power between 5 PM and 9 PM. The microgrid controller first looks for reducing the EV charging as the EV has the highest priority to be controlled in the presence of such a DR signal in this algorithm. However, the controller finds no EV at the beginning of the DR signal at 5 PM and therefore starts reducing water heater set point temperature to 125°F for all the houses (5 houses in this experiment) to fulfill the commitment of reducing 3 KW during this DR period. This experiment also shows that when EV becomes available to be charged at 6 PM the controller is able to adapt to this changing scenario and therefore increases the water heater temperature to their previous set points and manages the EV charging to fulfill the DR commitment. This example shows the capability of our GridMat tool to develop control algorithms for a microgrid controller.

In Figure 7, a hybrid lighting control is implemented, where lighting control is based on occupancy and TOU energy rate; the hybrid lighting control algorithm involves both lighting control based on occupancy presence (device-level control for energy efficiency) and managing the lights from the HEM based on energy price (TOU rates). The HEM turns-off 400W (30%) of the installed power (this amount is part of the outdoor lighting, ambient interior lighting, and accent interior lighting) during the peak price.

In Figure 8, a control algorithm is implemented to manage HVAC energy consumption while maintaining thermal comfort. In this algorithm, ASHRAE-specified thermal comfort (room temperature between 72°F and 77°F) is maintained for the occupants while energy usages are minimized almost by 5% and energy cost by 12.5% for a single home.
6.2. Application-Specific Microgrid Design

We demonstrate the optimized microgrid design feature of our tool and methodology by experimenting the case studies explained below.

**Experimental Setup:** the "IEEE 123 Node Test Feeder" [Subcommittee 2015] is used as the baseline microgrid of our experiment. In the microgrid, multiple load nodes are added which they are able to be controlled remotely if required. Moreover, multiple loads have been increased on average 30%, to represent the load growth in the microgrid. Due to the load growth and newly added loads, the microgrid cannot efficiently handle the demanding power. Hence, the microgrid should be upgraded. Three methodologies are compared:

1. **Conventional Methodology [Duffey and Stratford 1989]**: addresses the problems individually and separately. The conventional methodology lacks the ability to model and optimize the microgrid. Therefore, the solution might not be optimal.

2. **MBD Methodology**: is our methodology presented in this paper. The microgrid is modeled, simulated, and optimized using our design automation tool. However, the smart controllers are ignored as possible solutions.

---

3The textbook is the IEEE Recommended Practice for Electric Power Distribution in Industrial Plants, also known as the "Red Book".
(3) **MBD w/ Smart Controllers**: is our MBD methodology while considering the existence smart controllers. We want to illustrate the influence of smart controllers as possible solutions while optimizing the microgrid. In this methodology, the controlled loads are supposed to consume at least 20% less during the peak time.

The solution libraries, e.g., wire library, capacitance library, and smart controller library, provided for these methodologies are the same. For instance, five elements of the wire library are illustrated in Table I with their detailed specifications.

<table>
<thead>
<tr>
<th>Code Word</th>
<th>Price ($/MFT)</th>
<th>Size AWG</th>
<th>Stranding</th>
<th>Diameter (inch)</th>
<th>Resistance (Ω/mile)</th>
<th>GMR (ft)</th>
<th>Ampacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWAN</td>
<td>184.12</td>
<td>4</td>
<td>6/1</td>
<td>0.25</td>
<td>2.72</td>
<td>0.008</td>
<td>140</td>
</tr>
<tr>
<td>RAVEN</td>
<td>342.09</td>
<td>1/0</td>
<td>6/1</td>
<td>0.398</td>
<td>1.15</td>
<td>0.008</td>
<td>242</td>
</tr>
<tr>
<td>PENGUIN</td>
<td>663.36</td>
<td>4/0</td>
<td>6/1</td>
<td>0.563</td>
<td>0.63</td>
<td>0.008</td>
<td>357</td>
</tr>
<tr>
<td>LINNET</td>
<td>1214.05</td>
<td>336.4</td>
<td>26/7</td>
<td>0.72</td>
<td>0.33</td>
<td>0.024</td>
<td>529</td>
</tr>
<tr>
<td>KIWI</td>
<td>6908.52</td>
<td>2167</td>
<td>72/7</td>
<td>1.735</td>
<td>0.06</td>
<td>0.057</td>
<td>1607</td>
</tr>
</tbody>
</table>

* (MFT) 1000 feet, (AWG) American Wire Gauge
  (GMR) Geometric Mean Radius, (Ampacity) Maximum Thermal Capacity

**Implementation cost analysis**: we have utilized the GridMat tool for design, simulating, and analyzing the microgrid, the results are shown in Figure 10.

Figure 10 illustrates that our proposed **MBD Methodology**, when not considering smart controllers, minimizes the total implementation cost by 9% compared to the **Conventional Methodology**. This minimization is due to the fact that, upgrading the microgrid using the **Conventional Methodology** may give unneeded solutions which overlap each other. In **MBD w/ Smart Controllers** methodology, by implementing low-cost (compared to upgrading component) smart controllers, the power consumption of nodes has decreased, hence less components need to be upgraded to meet the constraints. Therefore, using our MBD methodology with considering smart controllers has decreased the implementation cost about 5% compared to **MBD Methodology** (without smart controllers) and 14% compared to the **Conventional Methodology**.

![Fig. 10. Implementation cost analysis for different methodologies.](image-url)
6.3. Microgrid Design Exploration

In the process of designing an optimized microgrid, various parameters and/or cost metrics may change the implementation cost. In a microgrid, multiple houses may utilize solar panels in order to generate power (we need to build an application-specific microgrid for this particular scenario). The power generated from solar panels might get too much for a house to consume. Hence, the utilities (who provide electricity to consumers) may buy the solar power from the owners with a predefined price. By putting larger solar panels more power can be generated and sold to the utility, however the implementation cost may get much higher. The trade-off between the profit achieved by having larger solar panels and putting more money for implementation cost is part of the microgrid optimization.

Experimental Setup: we consider a residential microgrid with multiple houses, where each house has a solar panel. The solar panels implemented for houses are owned by a single entity (popularly known as an aggregator [Vatanparvar and Al Faruque 2015b]). The aggregator may sell the power generated by the solar panels to the house owners with cheaper price compared to the electricity price offered by the utility. This policy may give the house owners the sufficient incentive to implement solar panels without any down payment. On the other hand, the aggregator may sell the excessive generated power to the utility. The variable Return On Investment (ROI) demonstrates the number of years it takes to return all the money invested for the implementation cost. The aggregator may decide on the solar panel sizes and number of houses in the microgrid, based on the ROI.

![Diagram of solar panel area vs ROI for different number of houses.](image)

Fig. 11. Effect of solar panel area on ROI for different number of houses.

Figure 11 illustrates the estimated ROI for different number of houses and solar panel sizes. Keeping the solar panel size constant, increasing the number of houses, increases the solar power generated, thereby increases the gained profit and decreases the ROI until it saturates (the implementation cost is also increasing). However, changing the solar panel size for different number of houses shows different influence. For smaller microgrids, the bigger solar panel is more profitable. While, for larger microgrids, smaller solar panel is more profitable. Therefore, the microgrid designer needs to decide on this parameter for optimizing the microgrid while designing.

7. CONCLUSIONS

Day-to-day increasing complexity of the microgrid makes it harder for the designers to apply the conventional methodology during the design and optimization. Moreover, existing microgrid problems, can be addressed not only by upgrading the components,
but also by implementing smart controllers on the loads. Therefore, a cost optimization may be done by choosing the best solution between upgrading/adding components or implementing smart controllers. In this paper, we have presented a novel MBD methodology and the corresponding tool (GridMat) to design, model, simulate, and optimize the microgrid. A model-based application-specific microgrid design methodology has been presented, which optimizes the implementation cost of a microgrid by specifying the microgrid application, the design-time constraints, and rules. Multiple case studies have been analyzed and our experiments have illustrated that implementing a hierarchical controller reduces the average power consumption by 8% and shifts the peak load for cost saving. Moreover, optimizing the microgrid design using our MBD methodology considering smart controllers has decreased the total implementation cost. Compared to the conventional methodology, the cost decreases by 14% and compared to the MBD methodology where smart controllers are not considered, it decreases by 5%.

**NOMENCLATURE**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Psi_{WU}$</td>
<td>Wire library used for adding/upgrading the wires</td>
</tr>
<tr>
<td>$\Psi_{CA}$</td>
<td>Capacitance library used for solving possible power factor problems</td>
</tr>
<tr>
<td>$\Psi_{SC}$</td>
<td>Library of smart controllers implemented on nodes for managing their load</td>
</tr>
<tr>
<td>$A$</td>
<td>Microgrid application</td>
</tr>
<tr>
<td>$M$</td>
<td>Model of the planned microgrid</td>
</tr>
<tr>
<td>$N$</td>
<td>Nodes in the microgrid; each element represents the node profile</td>
</tr>
<tr>
<td>$E$</td>
<td>Edges in the microgrid; each element represents the edge profile</td>
</tr>
<tr>
<td>$T$</td>
<td>Topology of the microgrid; a two-dimensional matrix where element $T[i,j]$ represents the required distance between $i^{th}$ and $j^{th}$ nodes</td>
</tr>
<tr>
<td>$X_{WU}$</td>
<td>Wire assignments to edges; element $X_{WU}[i,j]$ equals to one if wire $j$ is selected for edge $i$</td>
</tr>
<tr>
<td>$X_{CA}$</td>
<td>Capacitance assignments to nodes; element $X_{CA}[i,j]$ equals to one if capacitance $j$ is selected for node $i$</td>
</tr>
<tr>
<td>$X_{SC}$</td>
<td>Smart controller assignments to nodes; element $X_{SC}[i,j]$ equals to one if smart controller $j$ is selected for node $i$</td>
</tr>
<tr>
<td>$\Gamma_{WU}$</td>
<td>Cost of upgrading/adding wire; element $\Gamma_{WU}[i,j]$ represents the cost of upgrading from wire $i$ to wire $j$</td>
</tr>
<tr>
<td>$\Gamma_{CA}$</td>
<td>Cost of upgrading/adding capacitance; element $\Gamma_{CA}[i,j]$ represents the cost of upgrading from capacitance $i$ to capacitance $j$</td>
</tr>
<tr>
<td>$\Gamma_{SC}$</td>
<td>Cost of upgrading/adding smart controller; element $\Gamma_{SC}[i,j]$ represents the cost of upgrading from smart controller $i$ to smart controller $j$</td>
</tr>
<tr>
<td>$Cost$</td>
<td>Total implementation cost</td>
</tr>
</tbody>
</table>

**ACKNOWLEDGMENTS**

The authors would like to acknowledge graduate and undergraduate students including: Fereidoun Ahourai, Sani Fakhouri, and Irvin Huang, who contributed in developing the early version of the GridMat tool at the Advanced Integrated Cyber-Physical Systems (AICPS) lab.
REFERENCES


Received May 2016; revised September 2016; accepted xx