Proceedings of 2016 Universal Technology Management Conference (UTMC), Minnesota, United States of America, 2016

An Approach to Runtime Synthesis of Energy Management Policies

Mark Allison* Michael E. Farmer†, and Matthew Spradling ‡
The Department of Computer Science, Engineering
and Physics
The University of Michigan - Flint
Flint, Michigan USA
Email: *markalli@umflint.edu, †farmerme@umflint.edu,†mjspra@umflint.edu

***Abstract*—Interpreting software models to control cyber-physical systems facilitate efficient behavior changes at runtime. Energy management within the Microgrid is one such domain in which this approach has been successfully applied. Energy management models are augmented with policies which are able to describe behavior of plant elements based on responses to external events. The current solution requires policies to be hand crafted with the help of domain experts. This limits granular behavioral specification especially in the context of future weather related events.**

In this paper we present our approach which utilizes demand and production energy forecasts based on cloud services to automate the generation of low level Event-Condition-Action (ECA) policies. These policies describe economically optimized behavior over a short-term forecast horizon. Our approach also addresses the volatile nature of demand-side load forecasting by utilizing kalman filters for profile correction.

***Keywords*—Model-Driven Software Engineering, Domain-Specific Modeling Languages**

.

I. Introduction

Model-Driven Engineering (MDE) has emerged as a promising paradigm for realizing high assurance complex software systems [1]. One approach within MDE, termed interpreted Domain-Specific Modeling languages (i-DSMLs) allows for the direct interpretation of domain-specific models to control some cyber-physical systems; engineered amalgamations of computational and physical processes [2]. One challenge within the i-DSMLs paradigm is the practice of manually creating models to describe complex and often mission critical behavior. Users of the language may need some level of sophistication within the domain to accurately describe preferred behavior. It becomes difficult to predict or describe how the system should react to future environmental events during runtime. It is the authors' contention that the human effort and expertise required contradicts the spirit of concealing complexity from the users and moreover exposes the system to human error. The desired approach should allow for the software to absorb much of the complexity relegated to the end user. We address this concern by automating the generation of model constructs(via policies) at runtime to describe environmentally reactive behavior. Our work is applied within the context of the *smart microgrid* domain for proof of principle.

The smart microgrid is a critical component to restructure the legacy electric grid, plagued by an inefficient centralized energy distribution model, to one enhanced with distributed energy resources(DES) and a data transfer overlay. This transition requires a rethinking of grid operation to ensure stability [3]. Our approach seeks to optimally schedule the charging and discharging of storage elements using resilience and economy as principal concerns. We use load and source profiles to render low level ECA policies which are continuously referenced by a model interpreter in control of the microgrid elements or *plant*. The microgrid domain is exceptionally suitable for MDE exploration as distributed energy generation via wind and solar means is highly susceptible to dynamic, non deterministic environmental factors. Within the microgrid there may exist any combinations of energy sources, storage cells and loads. The optimal state configuration of these components are dependent on a myriad of variables such as utility cost per KW, amount of sunshine and/or wind, and the amount of energy reserves in storage cells or even Electric Vehicles (EVs). Each variable may change several times daily, and each day is not necessarily a reflection of the previous or indicative of the next. Given the very nature of this problem we posit that this is a non trivial challenge well beyond the scope of human capability and therefore ripe for automation.

This research path is motivated by the need for autonomous software systems capable of controlling cyber-physical systems with minimal human intervention. For our application domain we seek to ensure high microgrid stability essential for high penetration of DERs with an eye towards generalizing the concept to other CPSs. We present an approach to microgrid behavior specification at runtime using generated policy constructs. Specifically the contributions entail:

1) i-DSML Policy synthesis from energy source/demand profiles utilizing a greedy algorithm
2) Addressing load deviation using concurrent Kalman Filters for profile classification

The remainder of this paper is organized as follows: In section II we present background on the problem domain and concepts upon which this work relies. Section III provides an analysis of the optimization problem and assumptions made to allow for tractability. Sections IV and V addresses source and load forecasting respectively, leading into Section VI which discusses the generation of policies. We take a look at closely related works in Section VII. We conclude and present future directions for this work in Section VIII.

II. BACKGROUND

In this section we describe the problem domain within which this work is applicable and discuss the core enabling technology, iDSMLs.

A. Microgrid Energy Management

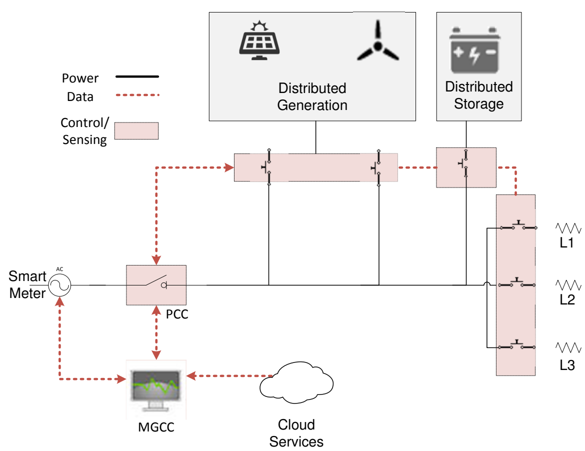


Fig. 1: The CERTS Microgrid Concept with Cloud Services Augmentation

The current electrical power grid, while being the single largest man made machine on the planet, is scheduled to be retooled with a data overlay enabling bidirectional flow of electricity and information to become the *Smart Grid* [4]. One important aspect of the Smart Grid is decentralized energy production which will rely on small quasi-autonomous, self reliant energy entities termed the *smart microgrid*. We are in accord with the Consortium for Electric Reliability Technology Solutions (CERTS) view of the microgrid concept as *an aggregation of sources and loads operating as a single entity and presenting itself to the macrogrid as a singular controllable unit* [5]. An important aspect of the microgrid is its ability to isolate itself from the larger grid in the event of a disturbance or for economic purposes. This operation is termed *islanding*.

Micrograms are poised to become more pervasive within the next decade primarily due to its higher reliability, cost reductions, and increased support from regulatory entities and energy consumers [6]. With the advent of the microgrid it is anticipated that distributed storage will become more prevalent. This is a departure from the current practice of varying the production of energy based on demand. The legacy electric grid has minimal storage capability due to its centralized model. The influx of distributed storage will increase overall reliability as: (1) loads can be supported for some time during power interruptions, and; (2) the microgrid may chose to isolate itself in the event of instability or power fluctuation. The overall impact of an increased proliferation of micro-sources within the smart grid has the potential to lower transmission losses as the sources are closer to the demand points [7].

Microgrids afford decentralized coordination of components based on consumer preferences. There are several controllers which regulate the flow of electricity at the physical layer. These controllers may be under the control of a software system. It is within this context that this research seeks to explore the efficacy and utility of a high assurance software overlay. It is important to now overview the components of a microgrid.

Figure 1 shows a typical microgrid based on the CERTS concept. The principal components are:

- A Microgrid central controller (MGCC) which coordinates the activities of the other essential plant elements. In our illustration we extend the CERTS concept to show a MGCC connection to cloud services necessary in forecasting ;
- Distributed Energy Resources (DERs) which is comprised of Distributed Generators (DG) such as wind turbines and photo-voltaic cell arrays, and Distributed Storage (DS);
- A point of common coupling (PCC) which allows for islanding;
- A smart-meter which monitors usage and acts as the point of information exchange between the macrogrid and the microgrid.
- Energy consumption elements or loads. Loads (L1, L2, L3) in the MGridML language are classified as scheduleable, critical, and noncritical. It is important that we have this distinction as it allows us to reshape our demand footprint. In our approach we can reschedule scheduleable loads and may choose to drop noncritical loads should our supply become suboptimal or unstable.

B. i-DSMLs

One key construct of the model driven engineering is the application of domain specific modeling languages(DSMLs). There exist a taxonomy of DSMLs which derive their semantics from model changes at runtime. These are termed i-DSMLs. i-DSMLs have a unique syntax which delineates the language in terms of its control and data instances. In other words these languages allow for two types of models; one which specifies the control and configuration and another which communicates data such as the settings for specific model elements found in the former. The execution semantics of the language is contained within its interpreter or Domain Specific Virtual Machine (DSVM). Figure 2 shows the layered architecture of the microgrid interpreter. At the topmost layer the user is able to describe models of the microgrid behavior preferences. It is at this layer our work applies. By automating the generation of policies we reduce human intervention in behavior constructs; an expert will still be required to setup and define the initial plant configuration and element specifications.

iDSMLs have been successfully applied in the domains of Cloud Computing [8], Communications [9] and in the domain of our discourse, microgrid energy management [10], [11]. The

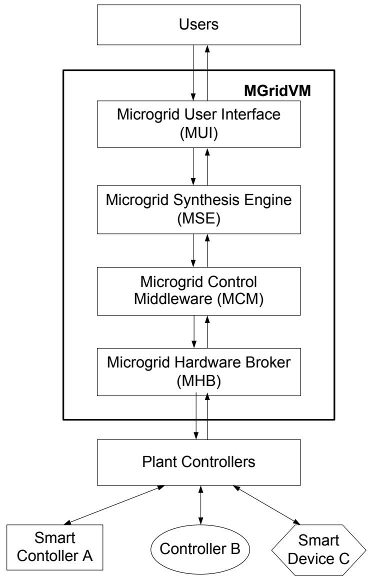


Fig. 2: The Microgrid I-DSML Interpreter

next section will describe policies and their place within the language.

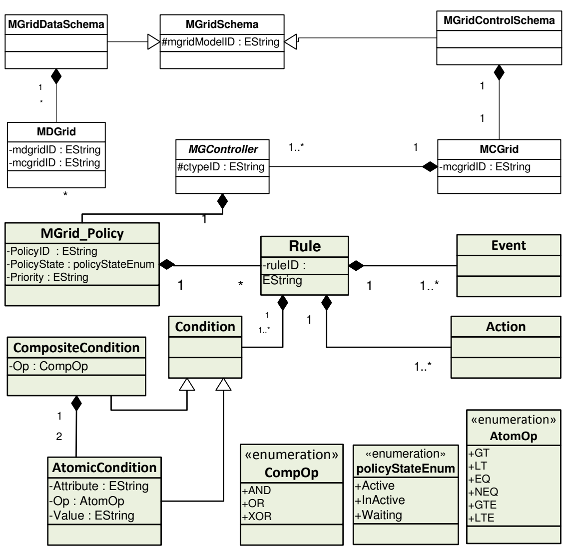


Fig. 3: Abstract Syntax for i-DSML Policies

1) Policy Augmented Models: Allison et al [12], augmented I-DSML models with policy constructs. Policies with this context are composed of a finite set of rules each being of the form:

$$if \quad Cond_i : EVT_i \rightarrow \langle Act_1, Act_2, ..., Act_n \rangle \tag{1}$$

These Event-Condition-Action rules state that upon the occurrence of an event (Evt), and while a Condition ($Cond$) holds true, then perform a set of actions (Act_n). Figure 3 shows the abstract syntax used to describe properly formed i-DSML policies. A proper policy is composed of `rules` which are themselves composed of at least one `event`, at least one `condition` which may be composite or atomic , and at least one `action`. In our earlier approach, rules would be constructed by hand in opposing pairs which typically sets and unsets a particular system state. This work extends this concept by automating the generation of rules. The interpreter is responsible for addressing conflicts through a policy prioritization framework. We will defer to [12] for a more detailed treatment of policy interpretation as it falls outside the scope of this discourse.

III. Problem Formulation

In order to adequately address the research problem we are required to constrain the variables impacting our model to assure tractability. Based on the CERTS microgrid concept presented in section II-A, our methodology assumes a microgrid wherein: (1) the storage is adequate to sustain all loads;(2) renewable sources are from wind and solar energy; (3) there exists a PCC capable of connecting/disconnecting from the macrogrid; and (4) there exists loads within the microgrid which are scheduleable and that their earliest and latest start times along with duration are known a priori. At this point in the research we have chosen not to include controllable sources such as generators as our aim is to optimize renewable sources by strategically charging and discharging our storage. Cost associated with the maintenance of equipment are assumed as negligible.
schedule storage operations
Figure 4 presents a simple source versus load scenario with

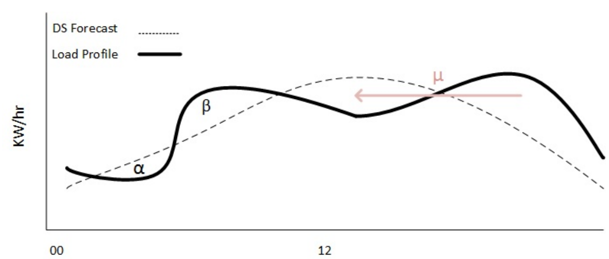


Fig. 4: Sample Analysis of Daily Source Requirements

a one day duration. The load is illustrated by a solid line and the renewable source is represented in dashes. Whenever production exceeds consumption as in α we may choose to direct the excess energy for storage. In the event that our storage capacity is full (condition) then we may sell the surplus to the macrogrid by enabling the PCC. When the reverse occurs and our demand exceeds the local production as in β then we have four main options: (1) reduce the amount of demand on the system by shedding loads designated `non-critical`; (2) reschedule `scheduleable` load operations such as laundry to be accommodated during a period of excess production, and; (3) discharge storage, and; (4) connect to the macrogrid and buy energy.

The latter course of action, (4), raises another interesting challenge as utilities may vary the tariff charged for electricity based on the time of day and even seasonally. Figure 5 shows

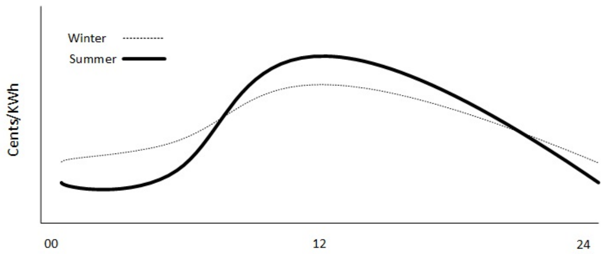


Fig. 5: Sample Daily Tariff Variation based on Season

a sample tariff variance for the Summer and Winter seasons. While the daily difference from peak to trough for Winter is slight, we see a stronger variance within Summer tariffs. This becomes an additional variable that has to be considered when generating policies. We may not want our storage strategy to force us to connect at times with the highest rate. Ideally the microgrid should island itself during peak tariff hours. The problem becomes:

Given S: The set of renewable Sources;
I: The set of Loads;
i: Reschedulable load where $i \in I$;
h: Forecast Horizon;
$\hat{L}_t$: Predicted Microgrid Total Load at a timestep t (based on profile);
$\hat{R}_{S_t}$: Forecasted Microgrid Total Renewable Sources at timestep t;
ρ_t: Energy Tariff at timestep t;
γ_t:Energy Resale at timestep t;
η_i:Earliest start time of load i;
ω_i:Latest start time of load i;
δ_i:Duration of load i;
$\hat{B}_t$: Calculated energy potential of storage at t;
B_{ut}: Upper threshold of storage;
B_{lt}: Lower threshold of storage;
$\hat{B}_p$: Energy produced from storage during discharge;
$\hat{B}_c$: Energy consumed by storage during charge;
UI: Energy bought from utility over forecast horizon.
UO: Energy sold to utility over forecast horizon.
Q: Cost of Energy over forecast horizon.

and our cost function f_0 becomes:

$$\underset{Q}{\text{minimize}} \quad f_0 \text{ where } Q = \sum_{t=0}^{h}((UI_t.\rho) - (UO_t.\gamma)) \tag{2}$$

subject to constraints:

Storage may never exceed its thresholds.

$$B_{lt} \leq \hat{B}_t < B_{ut} \tag{3}$$

Storage should start and end at its lower threshold. This is needed to support the dynamic programming outlined in section VI.

$$\sum_{t=0}^{h}(B_p + B_c)_t = 0 \qquad \forall B \mid B_{p_0} = B_{p_h} = B_{p_l t} \tag{4}$$

A task should not start after the latest starting time nor should it exceed the horizon

$$(i_{start} \leq \omega_i) \wedge ((i_{start} + \delta_i) < h) \tag{5}$$

accordingly, a task should not begin before the earliest starting time nor should it begin before time 0

$$(i_{start} \geq \eta_i) \wedge (i_{start} < 0) \tag{6}$$

The authors would like to point out at this juncture that while most constraints are directly translated to ECA policy rule conditions of high priority there are those which are hard coded into the interpreter's architecture for safety concerns. An example is:

$$\sum(L + \hat{B}_p)_t \leq \sum(S + \hat{B}_p)_t \tag{7}$$

A temporally global failsafe wherein total load may never exceed sources.

IV. Near Future Energy Production Forecasting

Since our primary focus is the generation of temporal ECA policy rules, we chose to rely heavily on short term cloud based services for our framework. We view these services as black boxes whereby we input geographical and instrumentation details and receive a short term estimate of the site specific energy yield. This short term prediction are on a timescale order of days which suits our purposes; our load forecast horizon (profile) assumes human power usage behavior is daily repetitive. We will next overview how solar and wind energy is converted within the model.

A. PV Energy Forecasting

Photo-voltaic(PV) cell arrays convert solar energy directly to electricity. In forecasting the energy produced may be seen as proportional to the solar radiance and operating temperature. Output is based on maximum achievable at Standard Test Conditions (STC) at solar intensity of $1000W/m^2$ and $25\,°\mathrm{C}$. In accordance with Gavanidouand Bakirtzis [13], the PV output P_t may then be described as:

$$P_t = A.I_t.\kappa \tag{8}$$

Where I_t is solar radiation at time t, A is the array surface and κ is array efficiency. There are several factors that influence the efficiency and radiation such as atmospheric conditions, age of the panels, tilt and even amount of debris or dust. However there are blackbox cloud services such as Solarcast [14] which may only require historical data and geographic location to provide a forecast.

B. Wind Energy Forecasting

Unlike the model for PV forecasting where energy generation is continuous in the presence of solar radiation, wind turbine energy production is governed by *cut-in* and *cut-out* wind velocities. The cut-in is the speed at which the turbine blade begins to rotate and generate electricity. The cut-out is that speed which exceeds a rated output limit and the turbine self protects by employing some braking or retardation mechanism. We use the model presented in Mohamed and Koivo [15] to use a quadratic to fit the generation curve to the manufacturer's specification. The power produced P_t becomes:

$$P_t = \begin{cases} 0 & \text{if } W_t < W_{ci} \\ a.W_t^2 + b.W_t + c & \text{if } W_{ci} \leq W_t < W_{ro} \\ E_{max} & \text{if } W_{ro} \leq W_t > W_{co} \\ 0 & \text{if } W_t \geq W_{co} \end{cases}$$

W_{ci}, W_{co}, W_{ro}, and W_t are the cut-in, cut-out, rated output and actual wind velocities respectively. The quadratic representation for speeds which fall between the cut-in and rated output relies on constants a, b and c. The model assumes that the turbine will brake should the actual wind-speed exceed the cut-out velocity.

We next address the representation of microgrid energy demand.

V. Load Profile Forecasting

A load forecasting strategy on a microscale presents a more interesting challenge from that done for the larger grid simply because the number of variations needed for the system to attenuate itself is not present in the microgrid. At this small scale the load profile is much more volatile so predictability becomes much more difficult [16].

The initial load profile estimation is accomplished by sensor readings over the forecast horizon which we have set at h. Since h is currently set for 24 hrs, we needed at least two profiles; one for weekdays, the other for weekends. Prior to classification profiles peaks are flattened by shifting scheduleable loads to times where there is an excess of renewable energy. Load profiles are continuously created and stored should the observed profile deviate significantly from the estimated profile signifying a change in behavior of the microgrid. We define an acceptable deviation using a threshold τ. It follows that by lowering τ we produce a greater number of profiles decreasing deviation for the next day, $h+1$, however this comes at the cost of performance degradation from classification overhead which we will discuss shortly. Further experimentation will be required to map τ to the number of microgrid elements and state changes.

A. Load Profile Classification using parallel Kalman Filters

One concern which arose as we evaluate the feasibility of our methodology was the adverse effect of utilizing a distinctly erroneous load profile. Load profiles can be highly susceptible to changes based on the behavioral characteristic of the individuals and of the household at large. It would be irresponsible to assume that the number and behavior of the household members would be rigidly static over the course of a year.

Changes to a particular profile ranges from *short-term* (noise) such as a household guest for the weekend, *medium-term* such as seasonal transitions, to the longterm such as a permanent addition to the household. It soon became apparent that one profile being modified may not sufficiently address this issue. Whenever the gap between observed and predicted for horizon h is sufficiently large the system may have to abandon the profile and seek one that is a closer match for generating rules for the next horizon, $h+1$. To accomplish this we employ the predictive power of the Kalman filter.

The Kalman filter, also known as a linear quadratic estimator, allows for state estimation based on current observation and the prior predicted state [17]. A thorough derivation of the concept is presented in [18]. For our application we present the concept as:

$$\hat{X}_k = Y_k.Z_k + (1 - Y_k).\hat{X}_{k-1} \tag{9}$$

Where $\hat{X}_k$ is the current estimation, Y_k is the Kalman gain,Z_k is our real time data, and $\hat{X}_{k-1}$ is the prior estimate. We next explore its usage within a case study relating to the impact of seasonal changes on demand.

1) Seasonal change case study: The detection of production excesses and shortages demonstrated as α and β in Figure 4, is a nontrivial concern as short-term temporal variations in demand may be propagated as policies which derive suboptimal behaviors. We explore the efficacy of Kalman filtering approach using a seasonal change scenario. Our intent is for the system to choose a best fit profile (Winter or Summer) given any date.

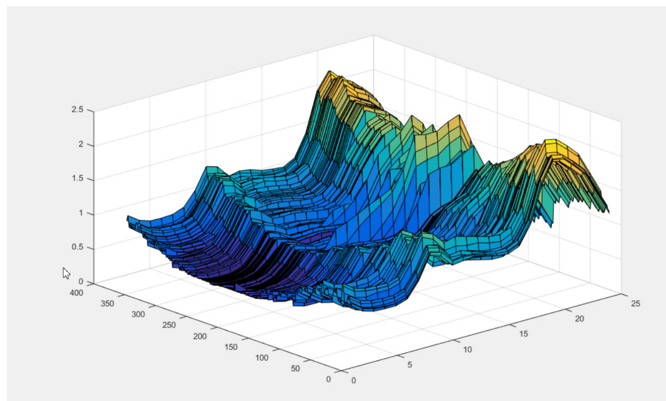


Fig. 6: A One Year View of Daily Demand Profiles

To better estimate change, a set of mathematical models of demand were developed using historical usage data for Flint MI from the US Department of Energy's OPENEI [19]. Figure 6 show a 3 dimensional profile of daily electricity requirements for a one year period. We can see that while each day shows a consistent pattern of more usage in the afternoons, there are striking differences over the years progression. Figure **??** shows our initial Winter and Summer demand profiles based on a small sampling from the months of January and July 2013. The actual reading from January 2nd (a typical Winter day for the region) is included to illustrate the close mapping to the winter profile.

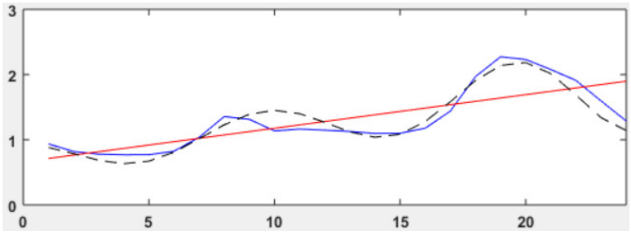
The consistency of the data was encouraging for developing an estimator to predict which of the profiles were most appropriate for a particular date in the year. We model these curves using a combination of three factors, 1) linearly increasing usage over the day, 2) a sinusoidal component across the day, and 3) a linearly increasing amplitude of the sine wave. The fitting of this model can be seen in Figure 7a for Winter usage and Figure 7b for Summer usage. A summary of the fit parameters is provided in Table I.

The model U_t is defined as:

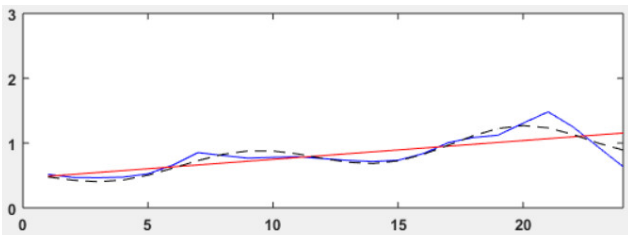
$$U_t = A_0 + r.t + A_t.\cos(2\pi f.t + \Phi) \tag{10}$$

Where A_0 is the initial usage amplitude at midnight, A_t is the time varying amplitude, r is the rate of increase of usage over

ISBN: 978-1-941968-29-1 ©2016 SDIWC



(a)



(b)

Fig. 7: (a) Model fitting for the Winter profile . (b) Model fitting for the Summer profile. Sinusoidal fitting shown as a broken line; Linear fitting shown as a red line

TABLE I: Model Parameters

	Winter	**Summer**
Frequency	2.45	2.25
$Amplitude_0$	0.25	0.15
Amplitude Growth	0.01	0.005
Phase	$\pi/16$	$\pi/3$

the day for the linear portion of the model, f is the frequency of the sinusoid and Φ is its phase. While not perfect fits to the data, the model is attractive because it is relatively simple which proves helpful for implementation and performance considerations.

Since there is noise due to the temporal variations and models may change at any time of day and rarely at midnight when atypical usage occurs, we propose employing a multiple model Kalman filter for detecting transitions between models. This model has been successfully used by one of the authors for detecting changes in things as diverse as automobile driving profiles to tracking occupants moving in a vehicle [20].

Algorithm 1 presents the rudimentary steps to select an existing profile or to create a new profile should none meet the acceptable criteria of $thresh$. In lines 2-4 multiple filters are run concurrently to generate a set of predictions for the next horizon. Lines 5-9 determines the which prediction was closest and therefore which profile would be the best choice. Lines 10-13 will create a new profile based on the best predictor should the deviation be unacceptable.

Now that we have a forecast of for power generation and demand we next delve into the generation of behavior policies.

Algorithm 1 Profile Selection

Input:
A finite set $G = \{a_1, a_2, \ldots, a_n\}$ of profiles
s_t, s_{t-1} sensor data array of usage for time horizons t and t-1
$thresh$ the acceptable deviation of a profile
Output: p the best-fit profile
1: **function** PROF–SEL(G, s_t)
2: **parfor** each profile $a \in G$ **do** ▷ run in parallel
3: Pred [index(G,a)] $\leftarrow$ Kalman_Prediction (a,s_{t-1})
4: **end parfor**
5: **for all** i such that $0 \leq i \leq$count(Pred[]) **do**
6: err[i] $\leftarrow$ diff (Pred[i],s_t)
7: **end for**
8: $r \leftarrow$ index_of(min(err[]))
9: $p \leftarrow$ G[r]
10: **if** sum (p) $> thresh$ **then** ▷ Is the deviation unacceptable?
11: $p \leftarrow$ Pred[r] ▷ Use best prediction as forecast
12: $G \leftarrow G + p$ ▷ Append to G as a new profile
13: **end if**
14: **return** p
15: **end function**

VI. A Greedy Approach to Policy Synthesis

Internally, load and source profiles are represented using simple arrays which consists of the forecasted production/usage for each interval t for 24 hrs . Once both profiles are available the policy generator creates policies based of the form: @t *while* (condition) *perform* (action). Conditions are used to ensure that the constraints outlined in section III are considered. Policies are sent to the interpreter where they are transformed to behavior. Let us delve a little deeper into policy construction, specifically how do we determine what action to perform at each timestep.

The approach is similar to a fractional knapsack with two sweeps. The role of the first is to procure energy to fill the gap between production and consumption. The role of the second pass is to sell the leftovers. Each time instance represents a commodity, while the battery and its maximum capacity represents the knapsack. The problem is interesting in that our knapsack has a "hole" in the bag with energy potentially being drained by usage, and in that there is a helper (renewables) frequently filling the bag with additional commodities. As such, we need to reconsider the maximum capacity available as we move forward in time.

First we determine what would happen if we had infinite battery storage capacity. We then determine how to purchase energy so as to evade deficits with minimal cost. We then determine how to sell excess energy so as to evade surplus with maximum profit. One special case is that the final time step must have a fully charged battery as stated in section III. The steps are as follows:

1) Given load and source profiles and starting storage level, first calculate the resulting storage level each time stept_i; this may be -ve or + ve signifying a surplus or demand. In this procedure we allow the battery to exceed the maximum and minimum

ISBN: 978-1-941968-29-1 ©2016 SDIWC

thresholds

2) Find the first time instance t_i during which there is a storage deficit. If there are none, skip to step 4. Otherwise
3) Find the time instance t_j where $j < i$, the battery is not full, and the cost of energy is minimal. If we can purchase enough energy to fill the deficit at t_i without overfilling the battery at t_j, purchase that exact amount. Otherwise, purchase as much as possible at t_j and then repeat
4) Find the first time instance t_i during which there is a battery surplus. If there are none, return. Otherwise find the time instance t_j where $j < i$, the battery is not below the minimal threshold, and the selling price of energy is maximal. If we can sell enough energy to empty the surplus without depleting the battery at $t_(j+1)$, sell that exact amount. Otherwise, sell as much as possible at t_j and then repeat

Time complexity: $O(n^2)$.

Correctness: At each time instance where there is a deficit, we always purchase at the lowest available cost from all prior time steps and purchase only the minimum requirement. As we could not have purchased for a lesser rate at any point, this ensures that the battery is not depleted and that we spend the least amount possible to ensure it. At each time instance where there is a surplus, we always sell at the highest available rate and only as much as we can safely sell. If there should be a surplus again in the future, we can still sell additional units at not only the previously considered rates but at all future time steps. As such the possible sale value only gets better as we move forward. The algorithm buys first to meet minimum requirements so as to avoid selling energy we might have to rely on later. The algorithm assumes that there is no situation where we might profit by purchasing energy from the utility at t_i and then selling that same energy back at some t_j, where j¿i. We are not expecting to turn a profit by speculating on the power company's rates for sale and purchase. Rather, we try to buy only the minimum amount and then sell only what we can after minimum needs are met.

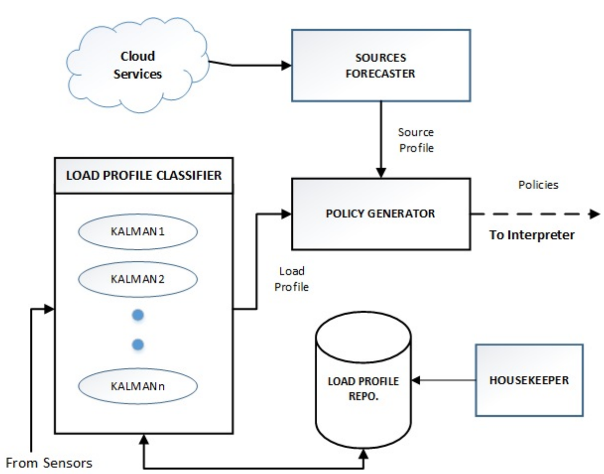


Fig. 8: High Level Architecture of Policy Synthesis

Figure 8 shows an overview of the entire policy synthesis process. A `source forecaster` queries cloud services to generate a profile of microsource generation based on equipment specification is sent to the `policy generator`. Load profiles are generated and classified by a load `profile classifier` from sensory support. The classifier is also responsible for new profile generation which are stored in the `profile repository`. To prevent profile explosion the system employs a `housekeeper` to remove profiles which have not been recently used.

VII. Related Approaches

Our survey of the literature has not produced other works which apply policy synthesis to model interpretation. However, there are several works which are comparable to the broader contributions of this paper, namely microgrid optimization and energy forecasting based on Kalman filtering.

Xie et al[23] proposes a model predictive control system for generating economically minimized dispatch. The availability of intermittent renewable resources is predicted then used within their optimization algorithm. Narayanaswamy et al. [24] presents an optimization algorithm for the microgrid based on intelligent online generation scheduling. While this approach is similar to our near future forecasting of renewables we deviate remarkably in our approach to load profiling and classification. Jiang and Geng [25], introduced a two layered approach to microgrid energy management.A scheduling layer develops an operational plan based on forecasting and a dispatch layer for realizing the plan. This approach is similar to ours not only by the use of forecasting but by separating concerns at the planning and implementation layers. While we use Kalman filtering to deal with errors between planning and real-time data, this approach used allocated reserves to compensate.

Al-Hamadi and Soliman [6], applies Kalman filtering to load forecasting within the energy management domain. The approach, like ours, applies to short-term forercasting with a moving window. The primary difference is in our use of the filter in profile selection instead of direct prediction.

VIII. Conclusion and Future Direction

In this paper we have presented an approach to microgrid energy management by policy synthesis at runtime using dynamic programming. This process is parameterized by source and load profiles which are created or selected (in the case of loads) on a on a 24 hr moving window basis. We additionally presented an approach to select existing user profiles using Kalman filters.

This work furthers our overarching goal of self-adaptive control of cyber-physical systems utilizing models as first class artifacts in development and operation. Although this is confined within the energy management domain at this time, we contend that this approach may be generalizable to other physical systems whereby there is the need to balance the demand and supply of resources.

While our model of the microgrid allows us to reduce the problem surface it curtails the applicability of the solution set. We have assumed a 24 hr forecast horizon with a time step of 15 minutes. We cannot unequivocally declare at this point that these are optimal given the environmental predictive algorithms being employed; further testing will be required to validate these assumptions. We intend exploration into cloud infrastructure as a service as these concepts mature. There is much room for further refinement through empirical studies, however we are very pleased and encouraged by the applicability of this approach.

Acknowledgment

This work is funded in part by the University of Michigan-Flint's Research Activity grant 2455 from the Office of Research and Sponsored programs.

References

[1] J. Bézivin, "Model driven engineering: An emerging technical space," in *Generative and transformational techniques in software engineering*. Springer, 2006, pp. 36–64.

[2] E. Lee *et al.*, "Cyber physical systems: Design challenges," in *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*. IEEE, 2008, pp. 363–369.

[3] P. P. Varaiya, F. F. Wu, and J. W. Bialek, "Smart operation of smart grid: Risk-limiting dispatch," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 40–57, 2011.

[4] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart gridthe new and improved power grid: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 14, no. 4, pp. 944–980, 2012.

[5] R. Lasseter, A. Akhil, C. Marnay, J. Stephens, J. Dagle, R. Guttromson, A. Meliopoulous, R. Yinger, and J. Eto, "The certs microgrid concept," *White paper for Transmission Reliability Program, Office of Power Technologies, US Department of Energy*, 2002.

[6] N. Amjady, F. Keynia, and H. Zareipour, "Short-term load forecast of microgrids by a new bilevel prediction strategy," *Smart Grid, IEEE Transactions on*, vol. 1, no. 3, pp. 286–294, 2010.

[7] J. Mitra, S. B. Patra, and S. J. Ranade, "Reliability stipulated microgrid architecture using particle swarm optimization," in *Probabilistic Methods Applied to Power Systems, 2006. PMAPS 2006. International Conference on*. IEEE, 2006, pp. 1–7.

[8] M. Allison, S. Turner, and A. Allen, "Towards interpreting models to orchestrate iaas multi-cloud infrastructures," *IEEE International Conference on Computer Science & Education*, 2015.

[9] Y. Deng, S. M. Sadjadi, P. J. Clarke, V. Hristidis, R. Rangaswami, and Y. Wang, "Cvm–a communication virtual machine," *Journal of Systems and Software*, vol. 81, no. 10, pp. 1640–1662, 2008.

[10] M. Allison, K. Morris, Z. Yang, P. J. Clarke, F. M. Costa *et al.*, "Towards reliable smart microgrid behavior using runtime model synthesis," in *High-Assurance Systems Engineering (HASE), 2012 IEEE 14th International Symposium on*. IEEE, 2012, pp. 185–192.

[11] M. Allison, K. A. Morris, F. M. Costa, and P. J. Clarke, "Synthesizing interpreted domain-specific models to manage smart microgrids," *Journal of Systems and Software*, vol. 96, pp. 172–193, 2014.

[12] M. Allison, S.-J. L. Lee, and A. Kuvvarapu, "A policy based approach to models at runtime within cyber-physical systems: A case study in microgrids," in *The International Conference on Software and Information Systems*, 2015.

[13] E. Gavanidou and A. Bakirtzis, "Design of a stand alone system with renewable energy sources using trade off methods," *Energy Conversion, IEEE Transactions on*, vol. 7, no. 1, pp. 42–48, 1992.

[14] S. Iyengar, N. Sharma, D. Irwin, P. Shenoy, and K. Ramamritham, "Solarcast: a cloud-based black box solar predictor for smart homes," in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. ACM, 2014, pp. 40–49.

[15] F. A. Mohamed and H. N. Koivo, "Modelling and environmental/economic power dispatch of microgrid using multiobjective genetic algorithm optimization," *Fundamental and Advanced Topics in Wind Power*, pp. 361–378, 2011.

[16] G. E. Box, G. M. Jenkins, and G. C. Reinsel, *Time series analysis: forecasting and control*. John Wiley & Sons, 2011, vol. 734.

[17] D. Willner, C. Chang, and K. Dunn, "Kalman filter algorithms for a multi-sensor system," in *Decision and Control including the 15th Symposium on Adaptive Processes, 1976 IEEE Conference on*. IEEE, 1976, pp. 570–574.

[18] R. G. Brown and P. Y. Hwang, "Introduction to random signals and applied kalman filtering: with matlab exercises and solutions," *Introduction to random signals and applied Kalman filtering: with MATLAB exercises and solutions, by Brown, Robert Grover.; Hwang, Patrick YC New York: Wiley, c1997.*, vol. 1, 1997.

[19] D. Brodt-Giles, "Wref 2012: Openei–an open energy data and information exchange for international audiences," National Renewable Energy Laboratory (NREL), Golden, CO., Tech. Rep., 2012.

[20] M. E. Farmer and A. K. Jain, "Smart automotive airbags: Occupant classification and tracking," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 1, pp. 60–80, 2007.

[21] D. Maly and K. Kwan, "Optimal battery energy storage system (bess) charge scheduling with dynamic programming," *IEE Proceedings-Science, Measurement and Technology*, vol. 142, no. 6, pp. 453–458, 1995.

[22] R. E. Bellman and S. E. Dreyfus, "Applied dynamic programming," 1962.

[23] L. Xie and M. D. Ilić, "Model predictive dispatch in electric energy systems with intermittent resources," in *Systems, Man and Cybernetics, 2008. SMC 2008. IEEE International Conference on*. IEEE, 2008, pp. 42–47.

[24] B. Narayanaswamy, V. K. Garg, and T. Jayram, "Online optimization for the smart (micro) grid," in *Proceedings of the 3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet*. ACM, 2012, p. 19.

[25] Q. Jiang, M. Xue, and G. Geng, "Energy management of microgrid in grid-connected and stand-alone modes," *Power Systems, IEEE Transactions on*, vol. 28, no. 3, pp. 3380–3389, 2013.