SystemC-AMS Simulation of Energy Management of Electric Vehicles

SystemC-AMS Simulation of Energy Management of Electric Vehicles

Donkyu Baek[1], Yukai Chen[1], Jaemin Kim[2], Naehyuck Chang[3], Enrico Macii[4], Sara Vinco[1], M. Poncino[1]
1 Department of Control and Computer Engineering, Politecnico di Torino, Italy
2 Seoul National University, Seoul, Korea
3 Korea Advanced Institute of Science and Engineering, Daejeon, Korea
4 Interuniversity Department of Regional and Urban Studies and Planning, Politecnico di Torino, Italy

***Abstract*—Electric vehicles (EV) are rapidly invading the market, since they are clean, quiet and energy efficient. However, there are many factors that discourage EVs for current and potential customers. Among them, driving range is one of the most critical issues: running out of battery charge while driving results in serious inconvenience even comparable to vehicle breakdown, as an effect of long fuel recharging times and lack of charging facilities.**

As a result, the dimensioning of the energy subsystem of an EV is a crucial activity. The choice of the power components and of the adopted policies should thus be validated at design time through simulations, that estimate the vehicle driving range under reference driving profiles. It is thus necessary to build a simulation framework that takes into account an EV power consumption model, dependent on the characteristics of the vehicle and of the driving route, plus accurate models for all power components, including batteries and green power sources.

The goal of this paper is to achieve early EV simulation, so that the designer can estimate at design time the driving range of the vehicle, validate the adopted components and policies and evaluate alternative configurations.

I. Introduction

Electric vehicles (EV) are more and more widespread, but they have not conquered the vehicle market yet. One of the main limitations of EVs is indeed the limited driving range, due both to the limited efficiency per cost, to the limited presence of charging facilities and to the much longer charging time [1]. Running out of battery charge while driving results indeed in serious inconvenience, that can be considered as severe as vehicle breakdown.

Extending fully charged vehicle range is thus critical to penetrate the vehicle market. On one hand, engineers work on making vehicles lighter and less power consuming [1]. On the other, the energy sub-system must be carefully designed, to ensure that components are well dimensioned and that energy conversion inefficiencies are reduced to the bare minimum.

To this extent, the simulation of the energy sub-system of EVs is crucial to improve the effectiveness of EV design. Many models have been proposed in the literature to simulate the power consumption of an EV. Hardware-in-the-loop approaches mix one or more real devices with software simulated models, to achieve high accuracy [2]. However, such works focus on the sole power consumption, thus not taking into account the energy-subsystem. Matlab/Simulink and Modelica models typically target the mechanics of the EV, thus resulting in long and complex simulations that focus on the internal components of the motor [3]–[5]. Thus, both tools incur in long simulation time with very detailed models of the dynamics of the systems. This makes unfeasible day-long simulations, that are on the other hand necessary to estimate the EV driving range and to validate the dimensioning of the energy components.

This work faces this challenge by simulating the energy subsystem of an EV through an extension of the framework proposed in [6], [7], that proved to efficiently support the modeling of energy systems ranging from smart embedded systems to smart grid [6], [8]. Such a framework formalizes the energy and information flows in the system, by defining classes of components and the corresponding interfaces. The components are then implemented in SystemC-AMS at different levels of abstraction, thus trading off accuracy and simulation speed. However, the framework supports only the direct current (DC) domain, and it does not support mechanical components. As a result, it does not support the modeling of EVs.

This work extends the framework in [6], [7] to allow the modeling and simulation of EVs. First of all, the framework is extended to allow the modeling also of alternate current (AC) components, by achieving a compromise between the complex sinusoidal nature of AC and the need for simulation speed of the framework. Then, the paper identifies a model of EV power consumption that is suitable for simulation inside of the framework, due to its system-level view of the EV [1]. An analysis of the model allowed to identify a solution to include mechanical models in the framework, thus accurately simulating EV power consumption given any driving range.

The whole simulation has been applied to a custom EV, powered by an electric motor and provided with photovoltaic (PV) modules to prolong its driving range [1]. The experimental results prove the effectiveness of the proposed framework at modeling the EV and its power consumption on actual traces of environmental information and driving conditions.

The paper is organized as follows. Section II provides the necessary background. Section III shows how the framework in [6], [7] is extended to support the modeling of EVs, and Section IV exemplifies the adoption of the extended framework on the custom EV. Section V shows how to generate the input environmental and driving traces. Finally, Section VI shows the experimental results and Section VII draws our conclusions.

II. Background

A. Simulation of EVs

The modeling of electrical energy systems has been widely investigated in the literature, addressing different application contexts, vehicle power consumption modeling [1], [9].

Hardware-in-the-loop approaches mix one or more real devices with software simulated models through sensors and actuators [2]. The resulting accuracy is higher w.r.t. software simulation, but application is restricted to small- and mid-scale EESs, and thus they are very complex to apply to EVs. Matlab/Simulink and Modelica models typically target the mechanics of the EV, thus resulting in long and complex simulations that take into account also the internal components of the motor [3]–[5]. Thus, both tools incur in long simulations with very detailed models of the dynamics of the systems. This makes on the other hand unfeasible day-long simulations, that are on the other hand necessary to estimate the EV driving range.

Some attempts have been made to adopt the standard SystemC framework also in the context of electrical energy systems [6], [7], [10], [11]. However, none of these approaches target the modeling of EVs. The only exception is [12], that on the other hand restricts the focus to the battery management sub-system.

B. SystemC and its AMS extension

SystemC extends C/C++ with libraries to describe HW constructs [13], and it is widely deployed in digital design for early-stage analyses and design-space explorations. Its AMS extension for modelling and simulating interacting analog/mixed-signal subsystems [14] proved over time to allow the adoption of a SystemC-based environment to simulate also non-functional, continuous time domains [15], [16].

SystemC-AMS provides different abstraction levels to cover a wide variety of domains. *Timed Data-Flow* (TDF) features the modeling of discrete time processes, that are scheduled statically by considering their producer-consumer communication dependencies. *Linear Signal Flow* (LSF) supports the modeling of continuous time behaviors as mathematical relations between quantities through a library of pre-defined primitive modules (e.g., integration, or delay), each associated with a linear equation. *Electrical Linear Network* (ELN) models electrical networks through the instantiation of predefined linear network primitives, e.g., resistors or capacitors, where each primitive is associated with a corresponding electrical equation. The SystemC-AMS AD solver analyzes the ELN and LSF system to derive the equations modeling system behavior, that will be solved to determine system state at any simulation time. ELN is conservative, i.e., the AD solver guarantees that energy conservation laws are satisfied by the equation system. The same does not apply to LSF.

Despite of its nature, mainly focused on digital and AMS systems, SystemC-AMS has been applied in a number of extremely heterogeneous domains, ranging from fluidic systems [17] to chemical sensors [18] and power electronic modeling [6], [19]. However, there is no work in the literature targeting the modeling of complex and large scale mechanical systems like EVs. The focus of state-of-the-art research is indeed on the modeling of MEMS systems, that do require complex constructs and techniques such as Laplace transfer functions, model order reduction and state space equations to take into account their non-linearities, the presence of resistive effects and the inter-dependence of the mechanical and electrical domains [20], [21].

III. Proposed Framework

This work builds upon the framework presented in [6], [7], that supports the modeling of energy systems ranging from smart embedded systems to smart grid. The current framework does not support physical modeling. Vice versa, modeling an EV (electric vehicles) requires to trace its physical and mechanical evolution together with the energy flows and the environmental characteristics: EV power consumption (during movement) and production (during regenerative breaking) are indeed highly dependent on the mechanics of the vehicle (e.g., motor power rating and efficiency) and on its operating conditions (e.g., motor torque and angular speed). Additionally, the framework supports only DC components, while the power consumed by EVs is in AC. It is thus necessary to extend the framework to cover also AC components.

A. Framework for electrical energy systems simulation

The goal of the framework proposed in [6], [7] is the simulation of electrical energy systems, restricted to the DC domain. Components naturally have different roles w.r.t. the power flow, i.e., they either consume, generate, distribute, or store energy. Thus, the framework classifies the components, and formalizes the relevant signals for each class of components and the typical adopted models. The identified classes of components, together with the corresponding relevant signals, are reported in the first half of Table I. Each component has a V port and an I port for DC voltage and current. Additionally, energy storage devices (e.g., batteries) feature a SOC port to export their state of charge, and components may feature an enabling signal En, used to activate an energy storage device or a power source according to a certain charge allocation policy.

Table I. Classification and main characteristics of each class of components in the proposed framework.

Component	Power interface	Domain
Load	(V, I)	Electrical (DC)
ESD	(V, I, SOC, E, En)	
Power source	(V, I, En)	
Converter	(V, I, V, I)	
Bridge	(V, I, V, I)	
AC load	(P)	Electrical (AC) + Mechanical
Inverter	(P, V, I)	Electrical (AC and DC)

The framework has been implemented by using the HW description language SystemC [13], [14], that supports models at mixed levels of accuracy (ranging from circuit-equivalent models to higher level functional models). SystemC proved to guarantee fast yet accurate simulation w.r.t. state-of-the-art tools like Matlab/Simulink, with speedups of up to two orders of magnitude and a high level of accuracy (errors lower than a single simulation. This is enabled by the support for multiple models of computation provided by SystemC and by its solver, that handles different domains and levels of abstraction in a single simulation run. This proved to be a winning strategy w.r.t. state-of-the-art solutions, that either rely on a single tool (that restricts support to specific domains) [22], simulate specific properties independently by executing ad-hoc simulators sequentially [23], or build co-simulation frameworks [24], that may introduce time misalignments and data sharing overheads.

B. Extending support to AC components

Simulating the power perspective of a system requires to trace its power flows, that are represented in terms of voltage (V)

and current demand/production (I), or by power demand/-production (P) over time of its components. This holds for any energy system despite of its scale (e.g., smart embedded system or smart grid).

Under this point of view, the main distinction is between DC (direct current) components, featuring a unidirectional flow of electric charge, and AC components, whose current and voltage curves periodically reverse direction. The framework in [6], [7] supported only DC components, whose I signal represents the DC current produced/consumed by the component. However, this limitation is too tight to model EVs: any motor indeed produces and consumes power in AC.

C. Modeling AC power signals

Explicitly modeling the sinusoids of AC current and voltage would be very time consuming, and thus would slow down simulation time. Vice versa, simulating an EV requires day-long simulations, to validate the driving range of the EV and to explore the behavior of its constituting components. The modeling of AC power must thus abstract from the sinusoidal behavior of current.

First of all, we abstract the sinusoidal by representing the moving average of power over time, i.e., the *root mean square* (RMS) of the sinusoidal, where using a moving average allows to take into account variations in the amplitude of the sinusoidal over time. This implies that the involved quantities can be represented by a single value over time, as done for the DC domain.

Additionally, it is important to consider that AC power is made of two main components: active power, i.e., power that performs work (e.g., is used by the motor to move the EV), and reactive power, i.e., power dissipated by the presence of a phase ϕ between the I and V sinusoids, even if the load device consumes no energy itself. The sum of the two is called *apparent power*, that is the power that must be taken into account at simulation time: although the current associated with reactive power does no work at the load, it still must be supplied by the power source. From Figure 1 it is evident that reactive power can be simply derived from active power by multiplying it by a factor $\cos\phi$ (called also power factor). As a result, the power demand of an AC load can be represented by a single port P, representing the RMS of apparent power over time.

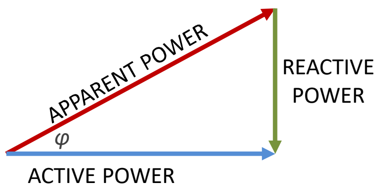

Figure 1. The power triangle: the AC load must be fed not only with active power (producing work) but also with reactive power, caused by the presence of a phase between the sinusoids. AC loadare thus modeled as the evolution of apparent power over time, that includes both.

D. Converting between AC and DC

The conversion to the DC domain is realized by *inverters*, that convert power between the DC and the AC domains. Once again, explicitly modeling the conversion circuitry would be a tight bottleneck for the simulation. Interestingly, even datasheets of inverters give a quite abstract measure of conversion losses: they characterize inverters in terms of efficiency, i.e., of ratio between the generated power (including conversion losses) w.r.t. the input power. This efficiency tends to be constant and almost independent on the amount of input power (when this is at least 15% of rated power) [25]. Thus, it is possible to approximate inverter behavior as its efficiency, that is function of the sole input power. The framework is thus extended with an additional class of components, i.e., *inverters*. Each inverter features a P port for the AC domain, and a couple of V and I ports for the DC domain.

E. Extending support to mechanical and physical models

An EV power model consists of physics equations and motor efficiency equations, that must be populated with data and coefficients derived from EV specification provided by vehicle manufacturers (e.g. rolling coefficient, drag coefficient, motor efficiency, etc.) [26], [27]. Such models are accurate, as they rely on the mechanics of the system. However, it might be difficult to populate the model unless the vehicle components have already been bought and configured. Additionally, mechanical simulations incur in very long simulation times. This prevents an effective evaluation of the energy sub-system, as the dynamics of the energy components (e.g., of batteries) have much longer times than the simple mechanics. Additionally, such models complex equations and the support for dynamic systems, e.g., bond graphs [5], [28]. SystemC-AMS does not provide any support for dynamic systems, and its extension to bond graphs remained in a very preliminary version [21], [29].

On the other hand, the goal of this work is not to accurately reproduce the mechanical behavior of an EV, including e.g. models of suspensions and of the steering subsystem. The goal is to rather have a quick estimation of the power consumption given a driving cycle, so that the designer can get a quick feedback on battery lifetime and on the autonomy of the EV. For this reason, our choice fell on the model proposed by [9], that is the current reference in the domain of EV energy simulation. The model derives EV power consumption based on road slope, vehicle speed and acceleration over time, and it results in a good trade off in terms of accuracy and simulation speed: the empirical polynomial equations ensure good simulation performance, and the model takes into account mechanical phenomena, such as loss on the motor and drivetrain that heavily impact on vehicle power consumption.

1) Instantaneous power consumption: The *instantaneous power consumption* of an EV is dominated by the propulsion power of the EV, and the dynamics equation P_{dyna} is a function of road slope, EV acceleration, EV mass and EV velocity, such that:

$$
\begin{gathered}
P_{dyna} = F\frac{ds}{dt} = Fv = (F_R + F_G + F_I + F_A)v \\
F_R \propto C_{rr}W,\ F_G \propto W sin\theta,\ F_I \propto ma,\ \text{and}\ F_A \propto \frac{1}{2}\rho C_d A v^2 \\
P_{dyna} \approx (\alpha + \beta sin\theta + \gamma a + \delta v^2)mv
\end{gathered}
\tag{1}
$$

where F_R, F_G, F_I, and F_A respectively are rolling resistance, gradient resistance, inertia resistance, and aerodynamic resistance. C_{rr}, W, θ, m, a, C_d, A in each resistance are rolling coefficient, weight, road slope, vehicle mass, acceleration, drag coefficient, vehicle facial area, respectively. The power consumption by dynamics is modelded by a function of coefficients α, β, γ and δ, derived as in [1].

2) Mechanical efficiency: The model is additionally enriched by considering the efficiency of the motor and of the drivetrain:

$$P_{EV} = \frac{P_{dyna}}{\eta} \tag{2}$$

$$\eta_{EV} = \frac{P_{dyna}}{P_{dyna} + C_0 + C_1 v + C_2 v^2 + C_3 T^2}$$

where C_0, C_1, C_2, and C_3 mean coefficients for constant loss, iron and friction losses, drivetrain loss, and copper loss, respectively. All coefficients in the hybrid model are derived experimentally [1].

3) Regenerative breaking: EVs exploit breaking periods to produce power, i.e., whenever the torque applied to the motor is in the opposite direction of the rotation, the kinetic energy of the EV can be converted to electric energy. This phenomenon, called regenerative breaking, is linear to the increase of EV velocity: regenerative energy comes from the electromagnetic induction between the rotor and the stator. When rotor speed increases, so does also the amount of electromagnetic induction of the stator. Simplifying, the relation is:

$$P_{regen} = \epsilon T v + \zeta$$

where ϵ is the coefficient corresponding to regenerative force and ζ is the minimum power to generate regenerative power. The coefficients for the motor loss and regenerative braking are obtained from driving experiment results with a custom EV [1], [9].

F. *Resulting extended framework*

The bottom half of Table I summarizes the extensions to the framework applied by this work. We defined two new classes of components, i.e., AC loads and inverters, so to allow the modeling of EV. AC loads are modeled with one output P port for AC power, and they span across mechanical and electrical domain. Inverters convert from AC power (P) to DC voltage and current (V and I). It is important to note that these extensions were possible thanks to the flexibility of SystemC-AMS, that allows to simultaneously simulate environmental evolution, physical and mechanical phenomena, energy distribution and the application of cyber aspects, e.g., the necessary charge allocation policies.

It is important to note that the model presented in Section III-E, together with the abstraction of AC signals in Section III-B, allow to ease the modeling of the mechanics of the EV as a C++ function. Thus, the EV mechanics and power consumption can be reconciled inside of the SystemC-AMS framework.

IV. Application to an EV

The proposed framework is validated on a custom EV, presented in [1] and shown in Figure 2. The EV is a pipe buggy, powered by an electric motor and provided with photovoltaic (PV) modules to prolong its driving range. The EV sub-system under simulation is depicted in Figure 3 and includes:

- A model of the EV mechanical power, estimated through an advanced dynamics model;
- A model converting EV mechanical power into EV electrical power demand, expressed in terms of voltage and current;

Figure 2. The EV case study.

- A 48Ah 3.7kWh battery pack, provided of a battery management system that performs protection from over charging/discharging and monitors the state of charge;
- A battery charger, managing battery charge when connected to a charging facility;
- A PV panel made of 10 PV modules, that cover the EV roof top and back panel;
- A DC-DC bus enriched with a cyber portion implementing a charge allocation policy.

Note that the schema in Figure 3 is an abstract representation of the system, and it represents the main blocks included in the simulation framework, together with their interfaces. Each block includes also the circuitry necessary to convert the voltage level of the component to that of the DC-DC bus.

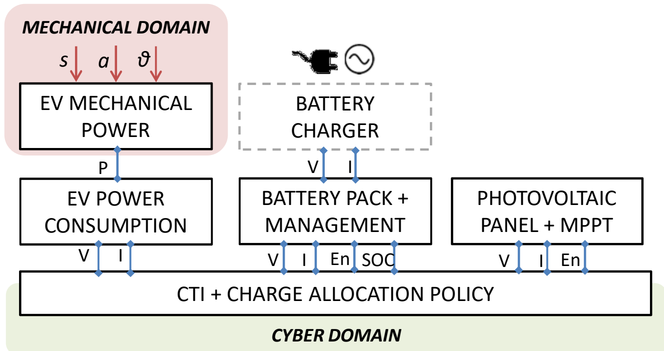

Figure 3. Simulated portion of the EV in Figure 2.

A. *Energy subsystem*

1) Battery pack: The EV is powered by a 76.8V nominal voltage 48Ah battery pack, including 96 3.2V 12Ah LiFePO4 cylindrical cells with 24 in series and 4 in parallel configuration. The power capacity is enough to drive the EV without significant discharge capacity effect.

The battery pack has been modeled with a circuit-equivalent model [30], that modelling the capacity dependency on current magnitude and dynamics. The model, depicted in Figure 4, consists of a left-hand part that models the battery lifetime, including a capacitor C (representing the storage capacity of the battery in Amp-hours), and a current generator representing the current requested by the load I_{batt}. The two voltage generators are used to model that battery available capacity is affected by the load current values distribution (e.g. constant and varying load currents with the same average value affect the capacity differently), as well as its frequency spectrum (e.g. periodic load currents with the same amplitude and shape, but different frequencies). The right-hand part models characterizes the transient behavior of battery voltage by using two pairs of R and C represent long-term and short-term transient. The additional variable resistor R denotes battery internal resistance, that mimics the fact that the battery voltage is adversely affected by larger currents. Such circuits are modeled thanks to the enhanced support of SystemC-AMS for Electrical Linear Network (ELN) of Model of Computation (MoC) : each network primitive can indeed be directly mapped onto an instance of the SystemC-AMS classes.

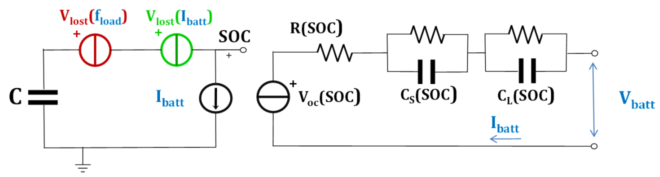

Figure 4. Adopted circuit-equivalent model for the battery pack.

The connection between the two parts of the circuits is then modeled by the voltage generator $V_{OC}(V_{OC})$, representing the fact that the open-circuit voltage (V_{OC}) of the battery depends on its state-of-charge (SOC), represented by the potential V_{SOC} of the capacitance. To model these dependence, the voltage generators are controlled by a SystemC-TDF module, that explicits the dependency as a linear function.

The battery block includes a battery management system (BMS), in charge of controlling the charge and discharge phases, and of avoiding over-charging and over-discharging to reduce the capacity aging degradation of battery. We assume there are battery balancing methods are implemented in the BMS and battery pack, and each cell in the battery pack is charged and discharged equally.

2) Battery Onboard Charger: The EV is normally charged during night by plugging the charger. Figure 5 describes a whole discharge and charge cycle of battery. In order to focus on the charge phase, we compress the period of discharge phase. Concerning the charging policy of Li-ion batteries, the policy is based on the Constant Current-Constant Voltage (CC-CV) protocol, characterized by a well-defined charge process that cannot be altered (red solid line in Figure 5). The CC-CV constraint is motivated by cost (simple hardware implementation) and by safety reasons. However, the CC-CV protocol does not take into account the capacity aging degradation of battery. Because the cost of battery in the EV normally occupies more than 50% of the total cost, the cost of using EV will increase dramatically if the battery replacement interval is too short due to fast aging degradation. There are four main aspects affect the capacity aging, i.e., discharge and charge currents; average SOC; deviation of SOC and temperature. In order to reduce the aging degradation, previous works proposed some alternative charge protocols: for reducing charge current, the alternative protocol uses smaller current to charge battery as indicated by green dotted line in Figure 5; to avoid high average SOC, the alternative protocol postpones the standard CC-CV starting time as illustrated by purple solid line in Figure 5. However, these kinds of charge protocol do not consider both aspects, thus for our EV we adopted the optimal aging reduction charge protocol introduced by [31].

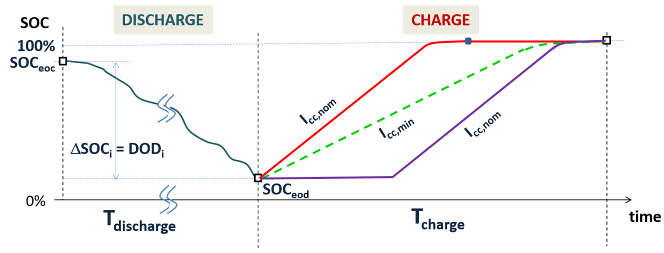

Figure 5. Different Charging Policies in Charge-Discharge Cycle.

3) PV Panel: PV modules are clean, light-weight and durable, and are thus an ideal onboard power source for EVs [32]. For this reason, 10 PV modules have been accommodated on the rooftop and the back panel of the EV. Given the small area covered by the PV modules, they won't be enough to fully operate the EV, but they can be exploited to charge the battery during movement or when the EV is parked and not plugged to any charging facility.

The PV modules generate 20.8V under open circuit conditions and 20W at standard conditions (i.e., 1000W/m^2 irradiance and 25C temperature) [33]. The PV modules are connected to an inverter, that applies a Maximum Power Point (MPP) tracking algorithm, i.e., that determines the operating voltage over time by trying to maximize the output power.

For the PV modules, we decided to adopt a simple functional model, that directly extracts the MPP and that is built directly from the sole datasheet information. The choice fell on the model proposed in [34], that ensures fast model setup and a low simulation impact. The input is the current vs. voltage graph available on the datasheet, that is converted to a power vs. voltage graph by multiplying voltage values per the corresponding current values. The identified maximum power points are then used as a mapping of irradiance values to output power, and fitted to a polynomial curve to be used as model for the PV module.

4) Inverters and DC-DC Converters: All conversion components have been implemented in terms of their efficiency. In particular, inverters have been implemented with fixed efficiency, by observing that whenever input power is at least 15% of the rated power efficiency is de facto constant [25].

B. Mechanical and physical subsystem

The EV power consumption model has been implemented as a SystemC-AMS TDF module, repeatedly estimating power consumption on the updated input values of EV speed and acceleration and of road slope (Figure 6). The environmental inputs are loaded from input trace files, and they are saved to arrays. The load estimation function, reproducing the model presented in Section III-E, is then encapsulated in the `processing()` function of TDF module. In this way, EV power consumption is estimated at each time step on fresh input data, and it is written in output to the inverter.

C. Cyber subsystem

The cyber subsystem consists of the charge allocation policy implemented by the arbiter of the CTI bus. The policy im-

```
SCA_TDF_MODULE(EV_load)
{
 sca_tdf::sca_out<double> P;
 ...
};

void EV_mechanical::processing()
{
 double Q, power;

 Q = (p_alpha + p_beta * sin(theta[t]*PI/180) + p_gamma * acc[t]) * mass;

 if(Q >= 0.0)
    power = Q * v[t] + C0 + C1 * v[t] + C2 * v[t] * v[t] + C3 * Q * Q;
 else if(Q > -1 * R1 / (R0 * v [t]))
    power = + C0 + C1 * v[t] + C2 * v[t] * v[t];
 else power = -1 * R0 * v[t] + R1;

 P.write(power);
 t++;
}
```

Figure 6. SystemC-AMS implementation of the EV power consumption presented in Section III-E.

plemented is as follows. Whenever the EV is moving and consuming power, power is provided by the PV panel (if produced power is higher than demanded power) or by the battery. The battery is charged its SOC is lower than 80% in two different conditions: by the PV panels when the EV is still (consumed power equal to 0), or during regenerative break periods (i.e., when the EV produces power). The policy implements also monitors battery status, and it stops simulation when the SOC is below 20%.

V. Generation of Input Driving Cycle

EV simulation must be fed with traces of the driving cycle, consisting of speed, acceleration and road slope. The traces have been generated by considering the typical driving schedule of the EV, as reported in Table II.

Table II. One-day driving cycle.

EV use	Time	Period
Go to work	7:00 to 7:15	15 min.
Parking at work	7:20 to 17:00	9 hours and 40 min.
Go to a market	17:00 to 17:30	30 min.
Parking at market	17:40 to 18:30	50 min.
Go to home	18:30 to 19:43	13 min.
Parking at home and battery charging	19:43 to 7:00	11 hours and 17 min.

Power consumption by EV is estimated based on input driving cycle consisting of speed, acceleration and road slope. We generate an one-day driving cycle for an EV driver in a city as a case study of an use of EV. Table II shows the departure time of each driving mission in a day.

We used Google Maps [35] to identify the route and traffic information. Figure 7(a) shows the traffic of an example route going to a market from work, where red is heavy traffic, orange is medium traffic and blue means no traffic. Given the color, we estimated the corresponding average speed of each part of the route. We extracted road elevation from the geography information with a series of GPS data of the route as shown in Figure 7(b) [36]. This allowed to derive road slope as variation of elevation along the route.

All traces have then been converted to the time domain and written to files, that show per each time slot the corresponding value of speed, acceleration and road slope.

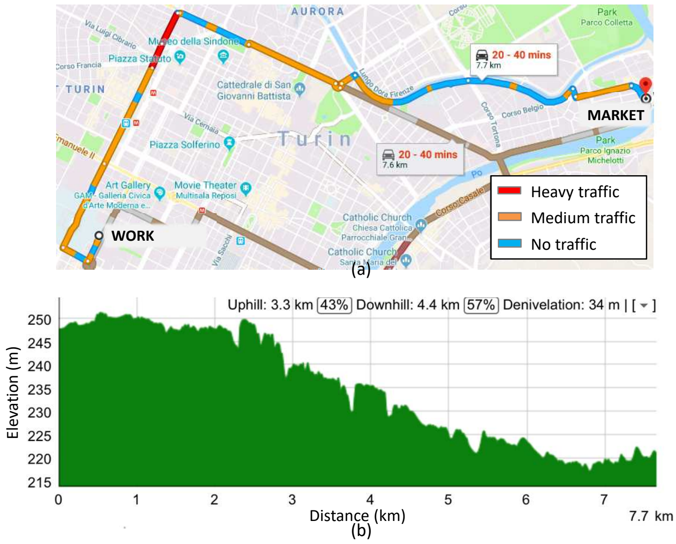

Figure 7. (a) Road traffic information and (b) road slope information.

VI. Simulation Results

We implemented the EV case study by using SystemC 2.3.1 and SystemC-AMS 2.1, and we run it on one-day-long traces generated as in Section V. Note that the one-day long driving cycle includes segmental driving and parking periods, as illustrated in Table II. Figure 8 shows how the different parts in the EV evolve over time: vehicle activity (top); battery usage and state of charge (a); PV power generation (b); EV power consumption (c); and some control signals that are used to control the charging mechanism of the EV (d-f).

The SOC of the battery (Figure 8.A) shows that the battery is charged plugged to a charging facility every night from 9:00pm to 5:00am, as highlighted by the enable charging signal of the charger reported in the Figure 8.e. Figure 8.a highlights also that the maximum value of SOC is 90%. This is due to the management policy, that sets 90% as the maximum SOC to reduce the average value of SOC of the battery, thus alleviating its capacity degradation.

As soon as the sun rises, the PV panel produces power, that is used to disconnect the vehicle from the grid and to charge the battery. Figure 8.d highlights that the PV panel provides power to the EV throughout the day (from 5:00am to 9:00pm), despite of when the EV is moving. The PV power generation after 7:00pm (i.e., sunset) is negligible because of low irradiance.

Figure 9 illustrates the details of SOC profile (blue solid line). The charging phase ends at 5:00am, when the battery reaches 90% SOC. From 5:00am to 7:00am the SOC of the battery is constant, even if the PV panels generate power (red solid line), as an effect of the SOC upper bound limitation imposed by the policy. The same circumstance happens from 2:20pm to 5pm: even if the PV panels generate power, the battery cannot absorb it once that the SOC reaches 90%. The driving periods discharge the battery, since the PV panel does not provide power while moving. It is interesting to note that after the first driving period (around 7am) the battery is charged by the PV

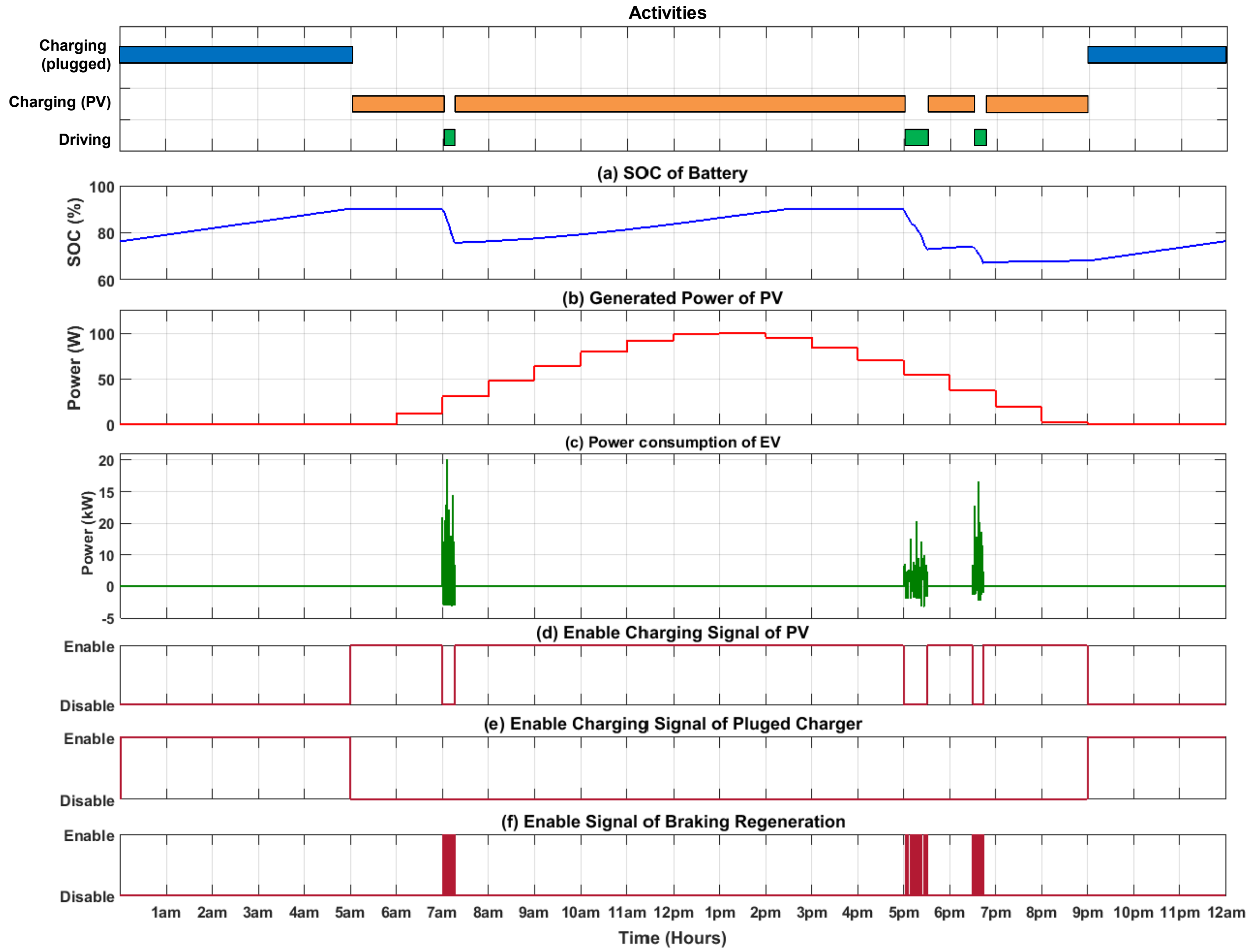

Figure 8. One-day long simulation results traces of different parts of EV and enable charging signals to indicate the power sources

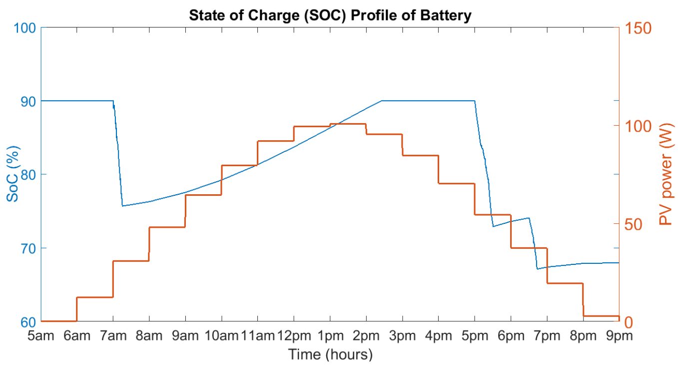

Figure 9. Battery SOC (blue) and PV panel production (red).

panel, and the SOC increases superlinearly as the angle of the sun to the ground increases.

Figure 8.c shows that regenerative breaking has been implemented and that it is exploited over time to charge the battery. This is highlighted by Figure 8.f, i.e., the control signal that is set to high whenever regenerative braking works. Figure 10 gives insight into a period where regenerative braking system works. It demonstrates that the regenerative braking system provides power to the EV in case of deceleration or driving downhill. The increase of SOC is very difficult to observe in Figure 8.a, due to the limited and instantaneous effect of regenerative breaking.

These experiments prove that the proposed SystemC-AMS based simulation allow to validate an one-day operation of EV, including driving, regenerative braking, and battery charging. This allows to analyze the energy flow and to gather information for the implementation of the energy management policy and on the correct configuration and dimensioning of the EV components.

VII. Conclusions

This work extended an existing framework for the simulation of energy systems, with the goal of supporting EVs. The proposed extension effectively covers both the mechanical domain and the AC domain by using the SystemC-AMS language. The framework has been tested on an example EV by running a

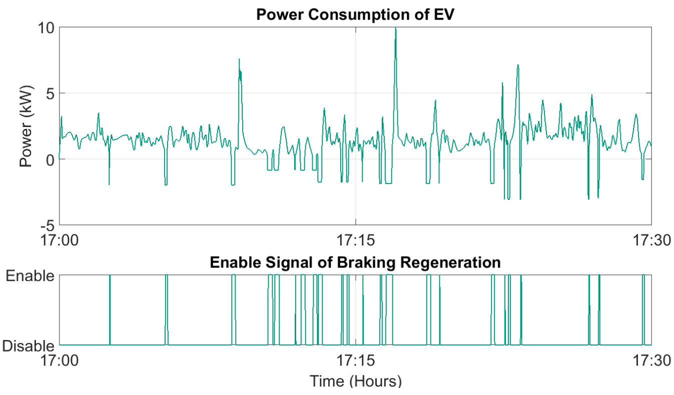

Figure 10. Regenerative braking system.

day-long simulation, thus validating different characteristics of the EV, including power consumption, regenerative breaking, power generation and storage. Future works will focus on more accurate models of the EV, thus taking into account additional phenomena that affect power consumption, and on the design and simulation of policies to extend EV autonomousness.

REFERENCES

[1] N. Chang, D. Baek, and J. Hong, "Power consumption characterization, modeling and estimation of electric vehicles," in *IEEE/ACM ICCAD*, 2014, pp. 175–182.

[2] L. Gauchia and J. Sanz, "Study of the effect of adverse external conditions on electric vehicles control: adaptive energy management as a solution," *IFAC Proceedings Volumes*, vol. 45, no. 21, pp. 460 – 465, 2012.

[3] I. The Mathworks, "Electric vehicle configured for HIL," 2005, mathworks.com/help/physmod/elec/examples/electric-vehicle-configured-for-hil.html.

[4] S. Alegre, J. V. Miguez, and J. Carpio, "Modelling of electric and parallel-hybrid electric vehicle using matlab/simulink environment and planning of charging stations through a geographic information system and genetic algorithms," *Renewable and Sustainable Energy Reviews*, vol. 74, pp. 1020 – 1027, 2017.

[5] Y. Hirano, S. Inoue, and J. Ota, "Model based performance development of a future small electric vehicle using modelica," in *Proc. of IEEE SICE*, 2015, pp. 1369–1374.

[6] S. Vinco, A. Sassone, F. Fummi, E. Macii, and M. Poncino, "An open-source framework for formal specification and simulation of electrical energy systems," in *IEEE/ACM ISLPED*, 2014, pp. 287–290.

[7] S. Vinco, A. Sassone, D. Lasorsa, E. Macii, and M. Poncino, "A framework for efficient evaluation and comparison of ees models," in *2014 24th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS)*, 2014, pp. 1–8.

[8] S. Vinco, A. Sassone, M. Poncino, E. Macii, G. Gangemi, and R. Canegallo, *Modeling and Simulation of the Power Flow in Smart Systems*. Springer, 2016, pp. 169–194.

[9] J. Hong, S. Park, and N. Chang, "Accurate remaining range estimation for electric vehicles," in *Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 781–786.

[10] J. Molina, X. Pan, C. Grimm, and M. Damm, "A framework for model-based design of embedded systems for energy management," in *Proc. of IEEE MSCPES*, 2013, pp. 1–6.

[11] ——, "A Framework for Model-Based Design of Embedded Systems for Energy Management," in *Proc. of IEEE MSCPES*, 2013, pp. 1–6.

[12] C. Scavongelli, F. Francesco, S. Orcioni, and M. Conti, "Battery management system simulation using SystemC," in *Proc. of IEEE WISES*, 2015, pp. 151–156.

[13] IEEE Standard, "Standard SystemC language reference manual," *IEEE Std 1666-2011*, pp. 1–638, 2012.

[14] ——, "Standard SystemC analog/mixed-signal extensions language reference manual," *IEEE Std 1666.1-2016*, pp. 1–236, 2016.

[15] C. Unterrieder, M. Huemer, and S. Marsili, "SystemC-AMS-based design of a battery model for single and multi cell applications," in *Proc. of IEEE PRIME*, 2012, pp. 1–4.

[16] S. Vinco, Y. Chen, F. Fummi, E. Macii, and M. Poncino, "A layered methodology for the simulation of extra-functional properties in smart systems," *IEEE TCAD*, vol. 36, no. 10, pp. 1702–1715, 2017.

[17] V. Fernandez, E. Wilpert, H. Isidoro, C. B. Aoun, and F. Pêcheux, "SystemC-MDVP modelling of pressure driven microfluidic systems," in *IEEE MECO*, 2014, pp. 10–13.

[18] F. Cenni, E. Simeu, and S. Mir, "Macro-modeling of analog blocks for SystemC-AMS simulation: A chemical sensor case-study," in *Proc. of IEEE VLSI-SoC*, 2009, pp. 211–214.

[19] L. Gil and M. Radetzki, "SystemC AMS power electronic modeling with ideal instantaneous switches," in *Proc. of IEEE/ECSI FDL*, vol. 978-2-9530504-9-3, 2014, pp. 1–8.

[20] T. Machne, Z. Wang, B. Vernay, L. Andrade, C. B. Aoun, J. P. Chaput, M. M. Lourat, F. Płcheux, A. Krust, G. Schrpfer, M. Barnasconi, K. Einwich, F. Cenni, and O. Guillaume, "UVM-SystemC-AMS based framework for the correct by construction design of MEMS in their real heterogeneous application context," in *Proc. of IEEE ICECS*, 2014, pp. 862–865.

[21] F. Cenni, O. Guillaume, M. Diaz-Nava, and T. Maehne, "SystemC-AMS/MDVP-based modeling for the virtual prototyping of MEMS applications," in *IEEE DTIP*, 2015, pp. 1–6.

[22] A. Elsheikh, E. Widl, and P. Palensky, "Simulating complex energy systems with modelica: A primary evaluation," in *2012 6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, 2012, pp. 1–6.

[23] W. Damm, H. Hungar, B. Josko, T. Peikenkamp, and I. Stierand, "Using contract-based component specifications for virtual integration testing and architecture design," in *Design, Automation Test in Europe*, 2011, pp. 1–6.

[24] D. Broman, C. Brooks, L. Greenberg, E. A. Lee, M. Masin, S. Tripakis, and M. Wetter, "Determinate composition of fmus for co-simulation," in *2013 Proceedings of the International Conference on Embedded Software (EMSOFT)*, 2013, pp. 1–12.

[25] T. Khatib, "Optimization of a grid-connected renewable energy system for a case study in Nablus, Palestine," vol. 9, pp. 311–318, 11 2013.

[26] X. Lin, D. Gorges, and S. Liu, "Eco-driving assistance system for electric vehicles based on speed profile optimization," in *IEEE Conference on Control Applications (CCA)*, 2014.

[27] Q. Yan, B. Zhang, and M. Kezunovic, "Optimization of electric vehicle movement for efficient energy consumption," in *North American Power Symposium (NAPS)*, 2014.

[28] L. I. Silva, P. M. de la Barrera, and C. H. D. Angelo, "Multi-domain modeling of electric vehicles including lead-acid battery dynamics," *IFAC Proceedings Volumes - Power Plant and Power System Control Symposium*, vol. 45, no. 21, pp. 447 – 453, 2012.

[29] T. Maehne, A. Vachoux, and Y. Leblebici, "Development of a bond graph based model of computation for SystemC-AMS," in *IEEE PRIME*, 2008, pp. 77–80.

[30] Y. Chen, E. Macii, and M. Poncino, "A circuit-equivalent battery model accounting for the dependency on load frequency," in *Proceedings of the Conference on Design, Automation & Test in Europe (DATE)*, 2017, pp. 1177–1182.

[31] Y. Chen, A. Bocca, A. Macii, E. Macii, and M. Poncino, "A li-ion battery charge protocol with optimal aging-quality of service trade-off," in *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ser. ISLPED '16, 2016, pp. 40–45.

[32] I. Arsie, G. Rizzo, and M. Sorrentino, "Optimal design and dynamic simulation of a hybrid solar vehicle," *SAE TRANSACTIONS - JOURNAL OF ENGINES*, 2007.

[33] *420J 20W Photovoltaic module*, Solar Electric Supply Inc., 2018, www.solarelectricsupply.com/solar-electric-supply-ses-420j-solar-panels-540.

[34] S. Vinco, Y. Chen, E. Macii, and M. Poncino, "A unified model of power sources for the simulation of electrical energy systems," in *Proc. of International Great Lakes Symposium on VLSI (GLSVLSI)*, 2016, pp. 281–286.

[35] Google Maps, https://www.google.com/maps.

[36] Geocontext–Center for Geographic Analysis, http://www.geocontext.org.