Statistical Model Checking for Finite-Set Model Predictive Control Converters

A Tutorial on Modeling and Performance Verification

Novak, Mateja; Nyman, Ulrik; Dragicevic, Tomislav; Blaabjerg, Frede

Published in:
I E E E Industrial Electronics Magazine

DOI (link to publication from Publisher):
10.1109/MIE.2019.2916232

Publication date:
2019

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Novak, M., Nyman, U., Dragicevic, T., & Blaabjerg, F. (2019). Statistical Model Checking for Finite-Set Model Predictive Control Converters: A Tutorial on Modeling and Performance Verification. *I E E E Industrial Electronics Magazine*, *13*(3), 6-15. Article 8845791. https://doi.org/10.1109/MIE.2019.2916232

Application of Statistical Model Checking Methods to Finite-Set-MPC Controlled Converters: Modeling and Performance Verification

Abstract

Statistical model checking (SMC) is used as a method for checking the system behavior in all possible scenarios and it has been successfully implemented in embedded automotive systems, sensor networks, aeronautics and communication systems for solving the problems that were beyond the abilities of the traditional formal techniques. Using this method the system is simulated for a finite number of times and afterwards the samples are used for hypothesis testing. We will demonstrate how the area of application can also be extended to power electronics systems, particular how this method can be used to check the performance of Finite-Set controlled power electronic converters. The performance of Finite-Set controlled power electronic converters is usually assessed using multiple simulations or experiments. Because only an executable system model is needed to apply the method, SMC is most similar to multiple simulations approach. However, it can also provide the reliability of the procedure and modeling of the stochastic system components. For example: the load or grid conditions have a stochastic behavior and in order to check the transient system behavior they can't be modeled as deterministic components. The verification needs to provide the evidence that the control algorithm can quickly adjust to the new conditions in the system, provide a fast response and stay within the limits prescribed by the standards. Using SMC the probability of the system staying within these limits can be assessed. A detailed modeling procedure of the converter system components and design considerations are given in the paper. The results interpretation is explained using the UPPAAL SMC toolbox. To evaluate the correctness of created model in UPPAAL SMC the behavior of the system during steady state conditions is also compared to the equivalent Simulink model and the experimental results, before the SMC is performed.

Index Terms

Controller performance, DC-AC power conversion, finite control set, predictive control, power electronics, statistical model checking.

I. INTRODUCTION

Model checking is a very powerful technique that is used to debug complex system structures like embedded controllers, network protocols or hardware components. Any system that has states and transitions between states can be analyzed using a model checker. With a model checker a series of events that would lead to an error state can quickly be identified. Therefore, the controller code can be redesigned before the implementation and a possible damage can be prevented. UPPAAL is one of the integrated tool environments used to model and analyze the behavior of the systems. The tool is available for free for non-commercial applications in academia [1], [2]. UPPAAL tools are all based on variants of Timed Automata (TA). With a TA, a timed behavior of various systems can be modeled and analyzed. However, when traditional numerical approaches are used, the size of the problem is exponentially increasing with the number of inputs i.e we are facing the problem of state space explosion. In UPPAAL SMC this was solved with introduction of the Statistical Model Checking (SMC) and Priced Timed Automata (PTAs). SMC is well-known in the field of sensor networks, communication protocols, aeronautics, embedded automotive systems and biology systems [3]–[12]. Power electronics converter systems are a new application area for SMC but nevertheless very promising, as also in this area there are some problems that are not solvable using the well-known traditional methods [13], [14].

In this tutorial we will demonstrate how SMC can be used to analyze the performance of a Finite-Set MPC controlled converter. MPC is still in the need of finding the analytical tools and methods to improve the design process, verify the performance and the stability [15]–[18]. Linear controllers can rely on well-known matured methods for optimal design, performance and stability verification and this is the reason why many power electronics engineers still prefer these controllers over the MPC based control algorithms. Performance verification methods must prove that the control algorithm can handle the disturbances in the system and keep the deviations of the controlled variables from the references at a minimum. In recent publications the performance of the Finite-Set MPC was mostly assessed by running multiple simulations or experiments [19]–[21]. The simulations can be performed in parallel and are in theory less time consuming then experiments, however they are missing the reliability of the procedure. Errors could remain undetected as there is no guarantee that a finite number of experiments can cover all possible scenarios. On the other hand, if we look into the Finite-Set MPC stability assessment, most relevant progress was presented by the authors in [22], [23] where Lyapunov stability criteria is used to asses the asymptotically stability under specific constraints. This method still needs some further research in order to be applicable to all applications and classes of converters [17]. In our work we are focusing on the performance verification and through the sections of this paper we will analyze the case of a stand-alone voltage source converter during transient load changes. This paper is a part of the most recent research in applying SMA approach to the power electronics systems. The hardest task of the approach is the modeling procedure of the power electronics systems so the system will be suitable for use in a SMC software. Because the timed automata structures are not usually used in power electronics applications, it is important for the reader to understand how he can transfer the behavior of a specific component like a power electronics converter to a timed automaton behavior. Therefore, a lot of attention will be put on the detailed analysis of the modeling procedure and design considerations.

The structure of the paper is as follows. In Section II we will give an introduction to SMC and the tools used to perform the analysis. A simple timed automata structure which is used to model the system components in the SMC software will be explained. Next section will illustrate how using the timed automata structures, the power electronic system can be modeled in UPPAAL. In Section V it is explained how performance verification using the created UPPAAL model is performed. In the end conclusions and future research aspects are given.

II. STATISTICAL MODEL CHECKING

Within computer science, formal methods is a term generally used for analysis methods with a solid mathematical underpinning. Formal methods can provide guarantees that a system has certain properties. A group of formal methods that are often used to analyze software systems is model checking. Model checking suffers from a problem known as state space explosion, namely that the size of the state space that needs to be checked grows exponentially with the size of the model that is used. In many cases this has prevented classical model checking from being used on industrial sized systems.

Statistical model checking [24] is a technique that tries to solve this problem by applying statistical instead of exact analysis of the models in question. Statistical model checking is also known as Monte Carlo simulation and is especially well suited for probabilistic and stochastic models [25]. Statistical model checking is used in aeronautics and embedded automotive systems, sensor networks, and communication systems to provide solutions to problems that are beyond the abilities of classical formal techniques [4], [5], [26].

Statistical model checking consists of performing enough independent simulations of the behavior of the model in order to gain a statistically valid result that predicts the behavior. Statistical model checking does not provide 100% guarantees for the properties of the system. On the other hand, the confidence of the results can be increased arbitrarily. If a larger degree of certainty is necessary, more experiments can be performed. The number of simulations increases linearly with the certainty and sub-linearly in the size of the model. Statistical model checking also naturally incorporates discrete and non-linear phenomena in the same model. The fact that the certainty of the technique can be scaled also means that it can be used to explore new concepts and approaches fast by setting a lower confidence level.

III. SMC TOOLS

In this paper we use the tool UPPAAL SMC, but there are other tools which can perform Statistical Model Checking. The paper *Statistical Model Checking: An Overview* [27] from 2010 provides a good introduction to the area. *Landscape of probabilistic tools* [25] provides an overview of the different types of probabilistic models, but focuses on other numerical methods of analyzing such models.

A relatively new tool, STORM, from 2016 aims at both high performance and versatility [28], [29]. It aims at being a toolbox of reusable modules such that new functionality can be implemented quickly. The paper *Comparative Analysis of Statistical Model Checking Tools* [30] provides a thorough comparison of five different SMC tools with a starting point in their application to systems biology. The tools surveyed in this paper are PLASMA-Lab [31], PRISM [32], MC2 [33], MRMC [34] and Ymer [35]. These tools support different probabilistic models. For instance PRISM can analyze the following probabilistic models: discrete-time Markov chains (DTMCs), continuous-time Markov chains (CTMCs), Markov decision processes (MDPs), probabilistic automata (PAs) and probabilistic timed automata (PTAs). The modeling formalism of UPPAAL SMC, Stochastic Timed Automata, can express as much as both DTMC and CTMC [36].

The expressive power of the modeling language of UPPAAL SMC is sufficient to describe the types of models needed for this paper, but the main reason for choosing the UPPAAL SMC tool suite is the direct collaboration with the research group behind the tool. This enables a deep understanding of how to use the tool and the potential future optimizations of the tool and models.

A. UPPAAL SMC

The family of UPPAAL tools are all based on variants of Timed Automata [36]. Timed Automata is a mathematical rigorously defined formalism for specifying the timed behavior of systems, which is described by example in the following section. A more recent development in the analysis of Timed Automata models is the introduction of SMC. These methods allow a precise modeling of dynamic aspects of models in the form of a Network of Stochastic Timed Automata. For a formal definition of Networks of Stochastic Timed Automata we refer the reader to [36]. UPPAAL SMC is an integrated modeling and verification tool which can be used to create models consisting of several Stochastic Timed Automata. It contains a graphical editor as well as a simulator. It also contains a verification part in which one can test statistical hypotheses about the current model. Apart from the hypothesis testing it is also possible to obtain many different types of graphs describing the behavior of the system. Such graphs are extremely important in practice when constructing the system models.

More information about SMC and UPPAAL SMC can be found in [37].

B. *Timed Automaton Model*

Timed automata were introduced in early 1990s by Raajeev Alur and David Dill [38], and were very quickly accepted as a very convenient formalism for modeling and analyzing real-time systems. In Fig. 1 a simple TA model in UPPAAL SMC is presented. Three locations can be identified: Init, State_1 and State_2. Init is the initial and committed location of the TA, which means the transition to the location State_1 has the highest priority and the TA will immediately make the transition. x is a special type of a variable called $clock$, whose domain consists of non-negative real numbers. When an automaton is waiting in one of the locations and the time is passing, then the value of its clocks are increasing. Each time unit models 1 μs in the real world. All locations are assigned with a value of the Output variable and an invariant $x <= 10$, which defines the maximum time the TA can stay in this location. The transition edges between the locations also have their guards $x >= 5$ defining the minimum time that has to pass before the TA can make the transition to the next location. When the transition will occur is not predefined, it is random and all clock values between $5 <= x <= 10$ have an equal probability to trigger the

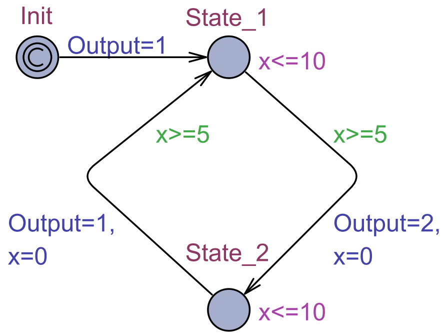


Fig. 1: Simple Timed Automaton model in UPPAAL SMC.

transition i.e. we have an uniform distribution for all clock values. UPPAAL SMC also offers the ability to use a custom probability function. If not defined differently, the distribution is always uniform. Because of the ability to randomly change the location, the simulation runs will have a different clock value triggering the transition. In this way we can observe how the system behaves under different conditions.

The processes in UPPAAL SMC can be synchronized through broadcast channels. The edge that has the synchronization label e.g. $Start!$ will emit the broadcast on channel $Start$ if the guard on the edge is satisfied. Afterwards all the edges with the receiving synchronization label $Start?$ will synchronize with the emitting process.

IV. MODELING THE SYSTEM IN UPPAAL SMC

The modeling approach will be presented on a stand-alone standard voltage source converter for a UPS application. First the system model is presented to identify the components that afterwards need to be modeled in UPPAAL SMC. Afterwards the components are explained in more detail. The whole modeling approach in UPPAAL SMC is very modular and can easily be adapted to different converter topologies or load types. Every model component can simply be exported and imported to other system models as a template.

A. System model

In Fig. 2 the following system components can be identified:

- 2-level voltage source converter
- LC output filter
- Load

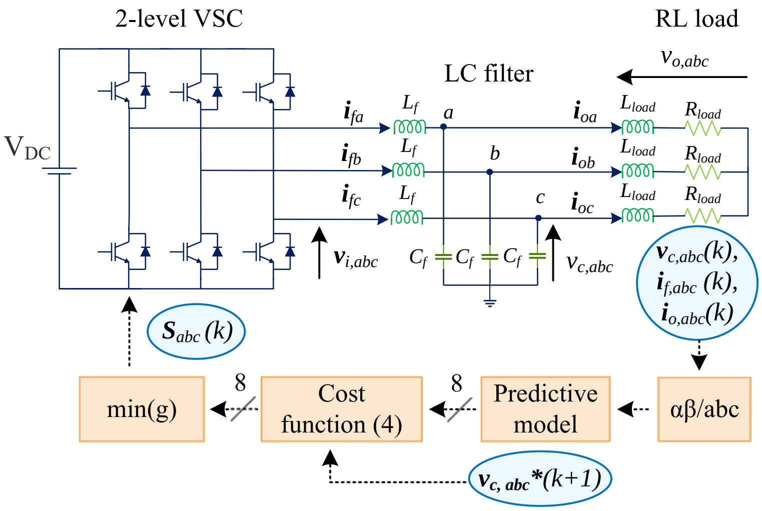


Fig. 2: Simplified scheme of the system model.

- Measurement system
- Control algorithm

As mentioned in the previous section; to model the system in UPPAAL SMC we need to describe the behavior of this model using Timed Automata. To do that, we first need to analyze what drives each of the model components. Let's start first from the converter model. It is very easy to imagine this component as a TA. The states of this TA are the switching states of the converter. For the presented topology this means we have eight switching states, e.g. for a 3L-level Neutral Point Clamped converter this will be 27. The next question that needs to be answered is what triggers the location changes in this TA? This is of course the control algorithm and in this model Finite-Set MPC algorithm is used. The core of the Finite-Set MPC algorithm is a discretized system model which is used to predict the future behavior of the system [18], [39]. In this case the model is defined through differential equations of the filter current and the capacitor voltage:

$$i_{f\,\alpha\beta}(t) = C_f \frac{dv_{c\,\alpha\beta}(t)}{dt} + i_{o\,\alpha\beta}(t) \tag{1}$$

$$v_{i\,\alpha\beta}(t) = L_f \frac{di_{f\,\alpha\beta}(t)}{dt} + v_{c\,\alpha\beta}(t) \tag{2}$$

$$i_{o\,\alpha\beta}(t) = \frac{-1}{R_{load}}\left(L_{load}\frac{di_{o\,\alpha\beta}(t)}{dt} - v_{c\,\alpha\beta}(t)\right) \tag{3}$$

where $i_{f\,\alpha\beta}$ and $i_{o\,\alpha\beta}$ are the inductor and load currents, $v_{c\,\alpha\beta}$ and $v_{i\,\alpha\beta}$ are filter capacitor and inverter output voltages, L_f and C_f are filter inductance and capacitance. Euler forward method is used to discretize (1) and (2) as presented in [40]. For every time sample new measurements of the $v_{c\alpha\beta}$, $i_{f\alpha\beta}$ and $i_{o\alpha\beta}$ are acquired by the measurement system. Using the discretized equations predicted values are calculated for all 8 possible converter switching states. These values and the reference values are then used in the cost function. The cost function usually has the form of the square of Euclidean distance between the reference signal and the prediction of the controlled variable. A good overview of the cost function design depending on the application is presented in [15]. For this system, the cost function suggested for the UPS applications in [40] will be used:

$$g = (v_{c\alpha}^{*} - v_{c\alpha}^{P})^2 + (v_{c\beta}^{*} - v_{c\beta}^{P})^2 + \lambda_d \cdot g_d + \lambda_{sw} \cdot g_{sw}^2 \tag{4}$$

$$g_d = (i_{f\alpha}^{P} - i_{o\alpha} + C_f \omega v_{c\beta}^{*})^2 + (i_{f\beta}^{P} - i_{o\beta} - C_f \omega v_{c\alpha}^{*})^2 \tag{5}$$

$$g_{sw} = \sum_{x=a,b,c} |S_x(k) - S_x(k-1)| \tag{6}$$

where $v_{c\alpha}^{*}$ and $v_{c\beta}^{*}$ represent the real and imaginary parts of the reference voltage vector, $v_{c\alpha}^{P}$ and $v_{c\beta}^{P}$ the real and imaginary parts of the predicted voltage vector, $i_{f\alpha}^{P}$ and $i_{f\alpha}^{P}$ the real and imaginary parts of the predicted filter current vector, $i_{o\alpha}$ and $i_{o\beta}$ the real and imaginary parts of the measured load current. Weighting factors λ_d and λ_{sw} define the secondary objectives. For the design of the weighting factors an approach based on artificial neural networks can be used as presented in [41]. $S_x(k)$ is the new potential switching state and $S_x(k-1)$ is the applied switching state from the previous switching instant. From the eight converter output vectors the one which produced the minimal cost function value is then selected and applied.

The load used in the system is an inductive load and will have two possible values of the resistance i.e. the TA will have two locations: high load and low load. Parameter values, which match the experimental UPS set-up are shown in Table I. In the next subsections components of the UPPAAL model from Fig. 3 will be explained in detail.

B. Physical system

In Section III-B a simple TA model was introduced with a clock `x`, which was triggering the location transitions. If we want to model the converter system it doesn't make much sense to use the time to trigger the transitions, especially if we know that in UPPAAL SMC we can

TABLE I: System parameters.

Parameter	Value
DC-link voltage (V_{dc})	700 V
Converter power rating (P)	18 kVA
Reference voltage and frequency (V_{ref},f_{ref})	400 V, 50 Hz
Output filter inductance and capacitance (L_f,C_f)	2.4mH, 25μF
Load (R_{nom},L_{nom})	48 Ω, 40 mH
Sampling time (T_s)	25 μs
Weighting factors (λ_d,λ_{sw})	0.4, 0.5

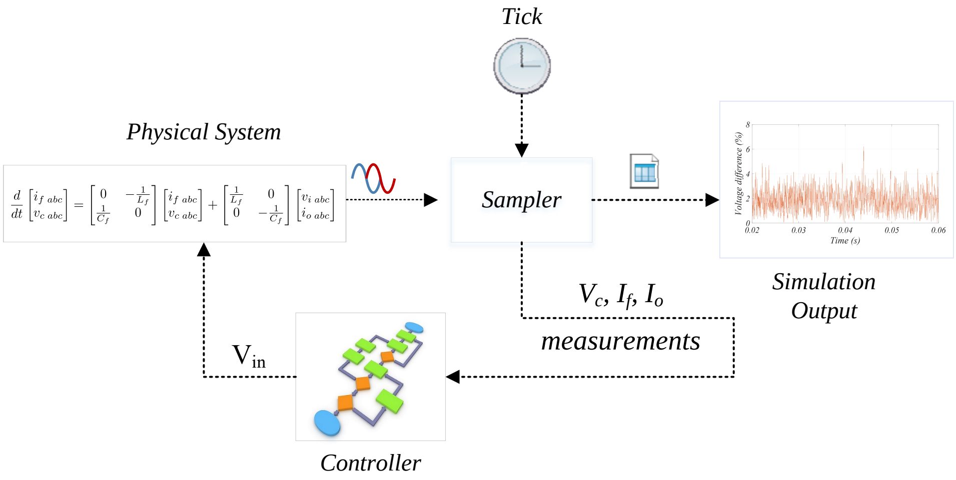


Fig. 3: Block scheme of the system model in UPPAAL.

define the behavior of the clocks trough differential equations. Then why not declare the system voltages and currents to be the global clocks in our TA network using the differential equations? Fig. 4a shows how this is done in UPPAAL SMC. It is easy to notice that these are the same equations as in (1), (2) and (3). The equations are multiplied by the global factor (GF) = 10^{-6} so the system will have the same dynamics like an equivalent simulation model with sample time of T_S = 1 μs.

C. Controller and Tick model

The *Controller* model can be observed in Fig. 4d and it is a combination of converter and control algorithm component from Fig. 2. We can observe that the values of variables `V_in_alpha` and V_in_beta, which represent the imaginary and real part of the converter voltage, are changing depending on the TA location (v0-v7). The TA has three more locations which are committed: Init, right and left. For the TA to transition from the (v0-v7) to right a synchronization signal $tick$ needs to be received from the $Tick$ model Fig. 4e. $Tick$ model is a very simple TA with only one location and it generates the synchronization signal $tick$ every 25 μs. When the TA reaches the location right it will call two functions. The functions are written in C-like imperative language. Function calc_cost() calculates the cost function values for all converter output voltages like it was explained in Section IV-A. The values are forwarded to the minControl which minimizes the cost function and selects the next TA location.

D. Sampler model

The task of the $Sampler$ model in Fig. 4c is to take measurements of the voltages and currents needed for calculating the predictions in the calc_cost() function of the $Controller$ model. Every 25 μs the task is repeated except for the initial time step which is shorter. This modification was made to fit the behavior of the TA model to the simulation model in Simulink. By comparing the waveforms it was noticed that the Simulink model added an approximately 1 μs delay in the model.

E. Load model

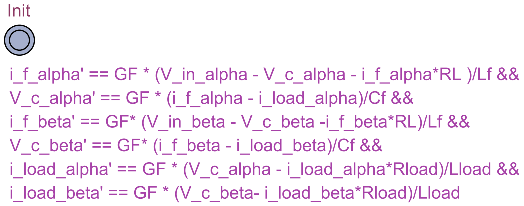
As it can be seen in the Fig. 4b the $Load$ model TA has one local clock variable y and three locations: Init, LowLoad and HighLoad. The invariants $y < 3000$ and $y > 2000$ transition guards have been assigned to the the locations. The transition trigger has a uniform distribution for the clock values between $2000 < y < 3000$ therefore in each simulation run a different clock value will trigger the change of the Rload value.

F. Simulation output

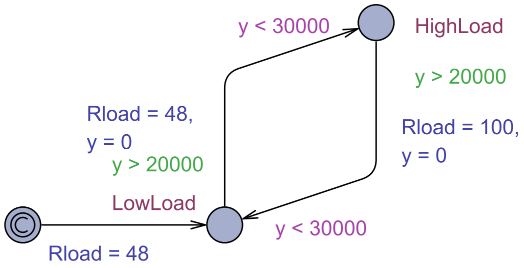
The last component of the model is $Simulation\ output$ in Fig. 4f and it is used to calculate the diffrence between the reference and measured voltage and the simple moving average of the diffrence. The following are calculated in the functions calc_diff() and rolling_average():

$$diff = (v_{c\alpha}^{*} - v_{c\alpha}^{m})^2 + (v_{c\beta}^{*} - v_{c\beta}^{m})^2 \tag{7}$$

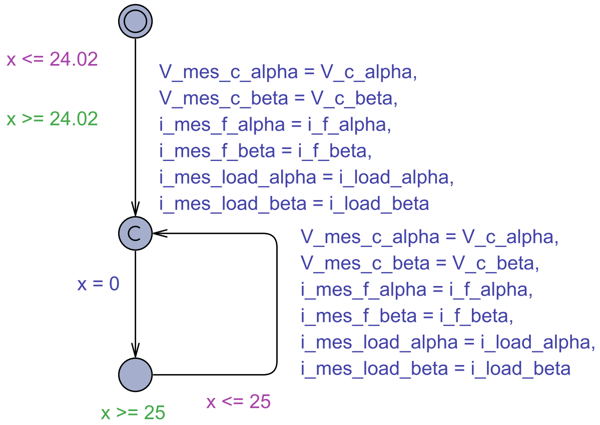
$$SMA = SMA_{prev} + \frac{diff_M}{n} - \frac{diff_M - n}{n} \tag{8}$$



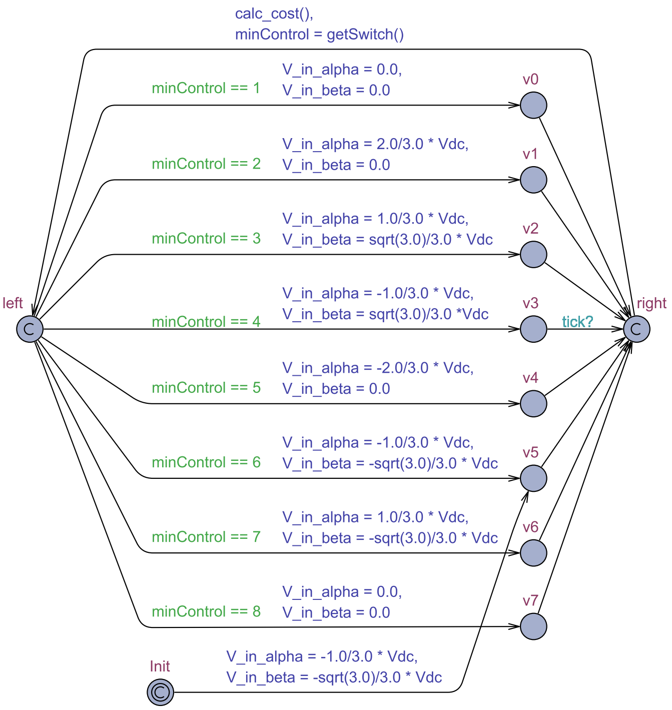
(a) Physical system.



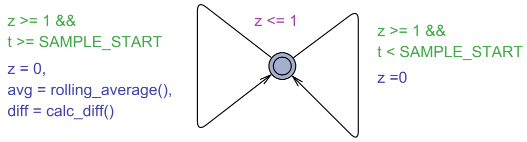
(b) Load.



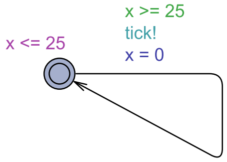
(c) Sampler.



(d) Controller.



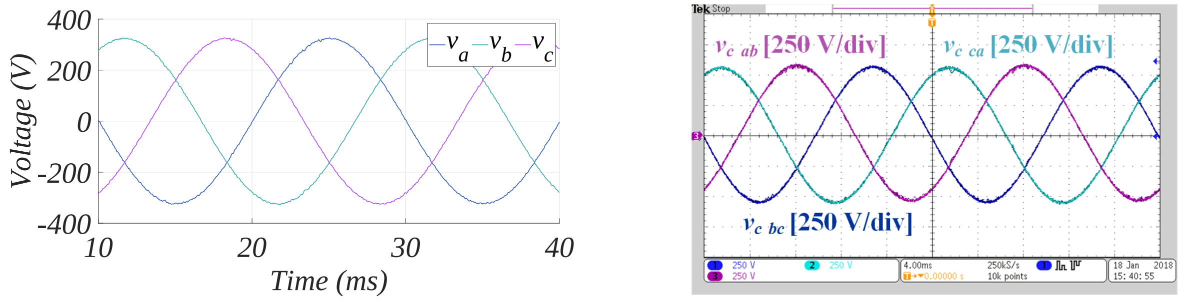
(e) Simulation output.



(f) Tick.

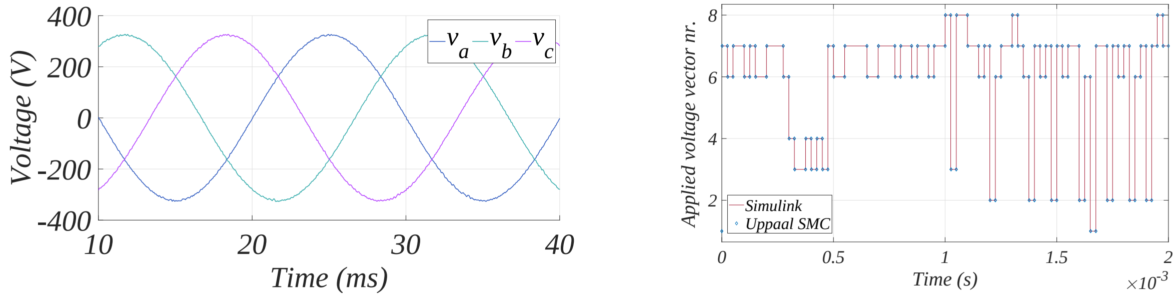
Fig. 4: System components in UPPAAL model.

where $v^*_{c\alpha}$ and $v^*_{c\beta}$ are the real and imaginary parts of reference voltage vector, $v^m_{c\alpha}$ and $v^m_{c\beta}$ are the values of measured output voltage, and $n = 10$ is the number of sample subsets. The TA has one local clock variable z which is ensuring that the $diff$ and SMA are calculated every 1 μs. Two transition edges with different guards can be noticed. Only the left edge is calculating the $diff$ and SMA from the $z >= SAMPLE\_START$, this modification was made to exclude the initial transient from the calculation.



(a) Capacitor voltage in UPPAAL SMC model.

(c) Capacitor voltage in experimental set-up.



(b) Capacitor voltage in Simulink model.

(d) Selected voltage vectors in Simulink and UPPAAL model.

Fig. 5: Design verification.

G. *Wind power and PV applications*

The application of the SMA approach can also be extended to other applications which feature stochastic components. Energy production from the renewable energy sources like wind power plants and photovoltaics (PV) are a typical example of this behavior. For example: the PV plant can be an automaton with states that have different solar irradiation values. What in the case of UPS application has not been implemented, but mentioned in the Section III-B is that there is a possibility to also set the probabilities of the transitions. Therefore, we can put higher probabilities for some irradiation values. In a similar way, an automaton for the wind power plant can be designed, where different wind speed values can be assigned to the automaton states. When we perform the statistical model checking approach with the automaton's that have the stochastic behavior, the SMC software will need to perform several simulation runs and in each run we will have a different irradiation or wind profile. Therefore, we would obtain information about our system performance also during this different energy production situations.

V. PERFORMANCE VERIFICATION

UPPAAL SMC allows the user to visualize the values of variables and clocks in the simulation runs and thus give insight to the user as which properties would be interesting to check using the model checker. Using this feature we can easily check the behavior of the system like in other available simulation software for power electronics. We used this feature to check the design by comparing the simulation output in the steady state to the equivalent Simulink model and also to the measurements obtained from the experimental set-up. For the complete guide on how to use UPPAAL SMC the reader is refereed to [36]. The procedure of modeling the system and performing the verification has also been illustrated in the flowchart presented in the Fig.6. In needs to be noticed that SMC is not a replacement of the traditional simulation methods performed in simulation tools like Simulink, moreover it is a specific tool that is used to check the performance of the controllers in the stochastic environments. The modeling procedure in Simulink is of course simpler as models of components are available in the Power Sim libraries, while when the user first starts using the SMC software he needs to model the components from zero.

A. Design verification

The syntax used in UPPAAL SMC to perform the simulation of the model is following: `simulate N [<=bound] {E1,...Ek}` where $N \in \mathbb{N}$ indicates the number of simulation runs, `bound` is the time bound on the simulations and `E1,...Ek` are expressions that are monitored during the simulation run and afterwards visualized. For our system the most interesting expressions to monitor are following:

```
simulate [<=60000] {V_mes_alpha, V_mes_beta, V_ref_alpha(),
V_ref_beta()}
simulate [<=60000] {i_mes_load_alpha, i_mes_load_beta}
simulate [<=60000] {diff, avg}
```

The first expression was used to compare the UPPAAL model to the Simulink model and experimental set-up. The results can be observed in Fig. 5 As it can be seen the UPPAAL model features the same performance as the Simulink model and it is also matching the experimental results. Now that the created model has been verified we can proceed to the query formulations.

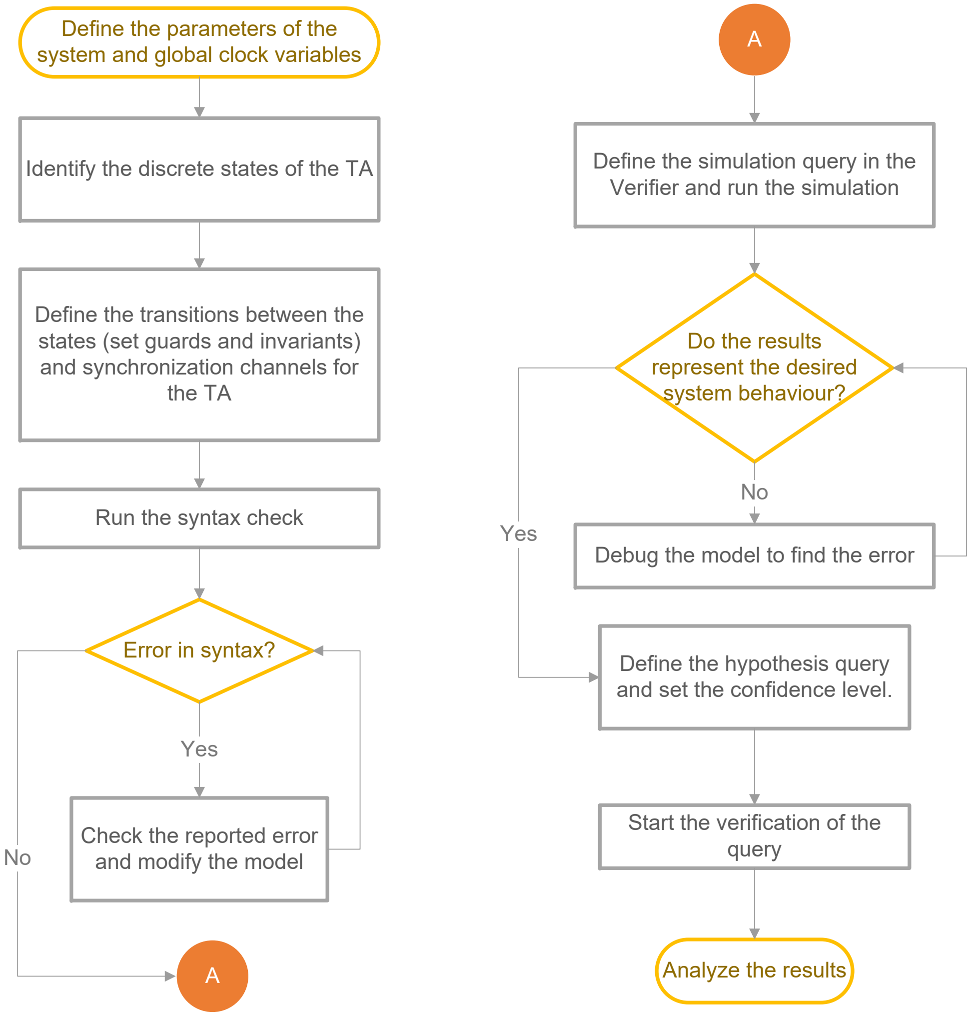


Fig. 6: Modeling and verification procedure in UPPAAL SMC.

B. *Query formulation*

Using the Verifier tool in UPPAAL SMC we can get an answer to three types of questions [42]:

- Probability Estimation - it will give a probability confidence interval for which the property in question is true (e.g 0.81-0.91)
- Hypothesis Testing - it will give a 0 or 1 depending if the probability of the property is lower or larger than the defined threshold
- Probability Comparison - it will give a 0 or 1 if asked question is false or true

The concept is the following: each time the system is simulated, it is encoded as a Bernoulli random variable that is true if the property in the asked question is satisfied or false otherwise.

Afterwards, a statistical algorithm will analyze the observations. For the first question an estimation algorithm that resembles the Monte Carlo simulation is used to give an answer and for the second and third question sequential hypothesis testing is used. For our model it is most interesting to use the first question. However, before starting with the testing we must define probability uncertainty ϵ. The lower the value, the higher is the reliability of the procedure but on the other hand more simulation runs will be necessary. The number of necessary runs is automatically calculated and updated with each simulation run by UPPAAL SMC. The probability confidence interval of the diff staying below 5% can be estimated using the following query:

`Pr[<= 44000]([] (t<SAMPLE_START || diff<0.05*325)`

The same principle is used to check the probability of SMA staying below the defined threshold.

FS-MPC performance relies on the correct system model so it would also be important to test the robustness of the algorithm to parameter uncertainties i.e. overestimations and underestimations of the model parameters like filter values.

C. Result interpretation

An example of query testing can be seen in Table II. The simulation runs were run with a zero parameter uncertainty, with 30% overestimated filter values and with 30% underestimated filter values. The probability uncertainty ϵ was set to 0.05. During the simulations the value of the load was stochastically changing from R_{load} = 48 Ω to 100 Ω, the value of L_{load} = 40 mH remained constant in this example. Thus, R_{load} = 48 Ω is considerer as a nominal load and R_{load} = 100 Ω corresponds to low loading conditions of the converter. The values in the last column of the table represent the number of simulation runs that were performed. A low number of runs indicates a high probability of the query being true or false. From the obtained results we can see that the control algorithm has good tracking performance with overestimated filter values, however the probability for the diff and SMA to stay under the the 5% when underestimated values are used is extremely low. Even for 12% it is only around 50%. From this observations we can conclude how much the overestimated system values in the predictive model will negatively effect the performance during the transient load change. The performance was also evaluated for lower voltage reference frequencies. No significant degradation of the reference tracking performance was noticed for parameter uncertainty 0% and 30%. We can repeat these tests also for different values of weighting factors to see how the performance will be effected. For a different topology

TABLE II: Query probability results obtained from UPPAAL SMC toolbox.

Query	Parameter uncertainty	Probability	Number of runs
$\mathit{diff} < 5\%$	0	0.901 - 0.999	54
$\mathit{diff} < 5\%$	30%	0.902- 1	36
$\mathit{diff} < 5\%$	-30%	0 - 0.097	36
$\mathit{diff} < 12\%$	-30%	0.347 - 0.447	386
$SMA < 5\%$	0	0.902 - 1	36
$SMA < 5\%$	30%	0.902 - 1	36
$SMA < 5\%$	-30%	0 - 0.097	36
$SMA < 12\%$	-30%	0.377 - 0.476	394

or application other performance metrics would be interesting to investigate e.g. the neutral point balancing in NPC topologies.

VI. CONCLUSIONS

In the paper, it was presented how statistical model checking can be used to test the performance of Finite-Set MPC controlled power electronics converter in the UPPAAL SMC toolbox. Due to the nature of the FS-MPC algorithm it is easy to visualize the whole system as a network of timed automata, which are used in UPPAAL SMC to model the converter system. The system modeling using the time automata is the most difficult part of the approach but due to very modular approach of modeling in UPPAAL, it is very easy to adapt the model to different converter topologies or load types. The model can be used to test the system performance under transients by modeling the components e.g. electric load, as stochastic automaton. By defining an expression that will track the difference between the reference and the measured values in a system with a stochastic load automaton, the control algorithm response to many different system conditions can easily be tested. The whole procedure of testing the hypothesis in UPPAAL SMC is straightforward and automatic as the program uses built-in algorithms to calculate the number of necessary simulations runs to calculate the probability of the hypothesis being true for a user-defined uncertainty.

SMC can not provide a 100% guarantee for the properties of the system, as the higher the confidence is set the more simulation samples need to be collected. This is of course time consuming as the number of simulations increases linearly with the level of the confidence and sub-linearly with the size of the model. Therefore, it is evident that multi-level topologies or modular multi-cell converters, which have a high number of possible switching states, i.e a larger system size will also have an increased simulation time. An interesting research challenge would be to also model a grid-connected model of the converter with a stochastic grid automaton. Microgrid systems are also a potential application area, where also the communication links between the converters could be modeled. In general, every system that has states and transitions between the states is suitable for the SMC approach. One must notice that the procedure in this form cannot be used to verify the stability of the control algorithm, but with a further development of the procedure this could also be accomplished.

REFERENCES

[1] F. Vaandrager, *Chapter 1 A First Introduction to Uppaal*. Quasimodo Handbook, 2014.

[2] Statistical model-checker - new SMC extension of UPPAAL. [Online]. Available: http://people.cs.aau.dk/~adavid/smc/

[3] P. Zuliani, A. Platzer, and E. M. Clarke, “Bayesian statistical model checking with application to Simulink/Stateflow verification,” *Formal Methods in System Design*, vol. 43, no. 2, pp. 338–367, 2013.

[4] A. Boudjadar, A. David, J. Kim, K. Larsen, M. Mikucionis, U. Nyman, and A. Skou, “Statistical and exact schedulability analysis of hierarchical scheduling systems,” *Science of Computer Programming*, vol. 127, pp. 103–130, 5 2016.

[5] K. G. Larsen, M. Mikučionis, M. Muñiz, J. Srba, and J. H. Taankvist, *Online and Compositional Learning of Controllers with Application to Floor Heating*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 244–259.

[6] A. Basu, S. Bensalem, M. Bozga, B. Delahaye, and A. Legay, “Statistical abstraction and model-checking of large heterogeneous systems,” *International Journal on Software Tools for Technology Transfer*, vol. 14, no. 1, pp. 53–72, Feb 2012.

[7] E. Clarke, A. Donzé, and A. Legay, “Statistical model checking of mixed-analog circuits with an application to a third order δ σ modulator,” in *Hardware and Software: Verification and Testing*, H. Chockler and A. J. Hu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 149–163.

[8] E. M. Clarke and P. Zuliani, “Statistical model checking for cyber-physical systems,” in *Automated Technology for Verification and Analysis*, T. Bultan and P.-A. Hsiung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 1–12.

[9] J. Martins, A. Platzer, and J. Leite, “Statistical model checking for distributed probabilistic-control hybrid automata with smart grid applications,” in *Formal Methods and Software Engineering*, S. Qin and Z. Qiu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 131–146.

[10] Q. Cappart, C. Limbrée, P. Schaus, J. Quilbeuf, L. Traonouez, and A. Legay, “Verification of interlocking systems using statistical model checking,” in *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, Jan 2017, pp. 61–68.

[11] J. Strnadel, “Predictability analysis of interruptible systems by statistical model checking,” *IEEE Design Test*, vol. 35, no. 2, pp. 57–63, April 2018.

[12] T. Mancini, F. Mari, I. Melatti, I. Salvo, E. Tronci, J. K. Gruber, B. Hayes, M. Prodanovic, and L. Elmegaard, “Parallel statistical model checking for safety verification in smart grids,” in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Oct 2018, pp. 1–6.

[13] M. Novak, U. M. Nyman, T. Dragicevic, and F. Blaabjerg, “Analytical design and performance validation of finite set MPC regulated power converters,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 3, pp. 2004–2014, March 2019.

[14] ——, “Statistical performance verification of FCS-MPC applied to three level neutral point clamped converter,” in *2018 20th European Conference on Power Electronics and Applications (EPE’18 ECCE Europe)*, Sep. 2018, pp. 1–10.

[15] S. Kouro, M. A. Perez, J. Rodriguez, A. M. Llor, and H. A. Young, “Model predictive control: MPC’s role in the evolution of power electronics,” *IEEE Ind. Electron. Mag.*, vol. 9, no. 4, pp. 8–21, Dec 2015.

[16] S. Vazquez, J. I. Leon, L. G. Franquelo, J. Rodriguez, H. A. Young, A. Marquez, and P. Zanchetta, “Model predictive control: A review of its applications in power electronics,” *IEEE Ind. Electron. Mag.*, vol. 8, no. 1, pp. 16–31, March 2014.

[17] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, “Model predictive control for power converters and drives: Advances and trends,” *IEEE Tran. Ind. Electron.*, vol. 64, no. 2, pp. 935–947, Feb 2017.

[18] V. Yaramasu and B. Wu, *Model Predictive Control of Wind Energy Conversion Systems*, ser. IEEE Press Series on Power Engineering. Wiley, 2017.

[19] R. N. Fard, H. Nademi, and L. Norum, “Analysis of a modular multilevel inverter under the predicted current control based on finite-control-set strategy,” in *Proc. Int. Conf. Elect. Power Energy Convers. Syst*, Oct 2013, pp. 1–6.

[20] J. Rodriguez, J. Pontt, C. A. Silva, P. Correa, P. Lezana, P. Cortes, and U. Ammann, “Predictive current control of a voltage source inverter,” *IEEE Trans. Ind. Electron.*, vol. 54, no. 1, pp. 495–503, Feb 2007.

[21] H. A. Young, M. A. Perez, J. Rodriguez, and H. Abu-Rub, “Assessing finite-control-set model predictive control: A comparison with a linear current controller in two-level voltage source inverters,” *IEEE Ind. Electron. Mag.*, vol. 8, no. 1, pp. 44–52, March 2014.

[22] R. P. Aguilera and D. E. Quevedo, “On stability of finite control set MPC strategy for multicell converters,” in *Proc. IEEE Int. Conf. Ind. Technol.*, March 2010, pp. 1277–1282.

[23] R. P. Aguilera and D. E. Quevedo, “On stability and performance of finite control set MPC for power converters,” in *Proc. Workshop PRECEDE*, Oct 2011, pp. 55–62.

[24] K. G. Larsen and A. Legay, “Statistical model checking: Past, present, and future,” in *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part I*, 2016, pp. 3–15.

[25] J. Katoen, “The probabilistic model checking landscape,” in *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS ’16, New York, NY, USA, July 5-8, 2016*, 2016, pp. 31–45.

[26] A. David, K. G. Larsen, A. Legay, and M. Mikučionis, “Schedulability of herschel revisited using statistical model checking,” *Int. J. Softw. Tools Technol. Transf.*, vol. 17, no. 2, pp. 187–199, April 2015.

[27] A. Legay, B. Delahaye, and S. Bensalem, “Statistical model checking: An overview,” in *Runtime Verification - First International Conference, RV 2010, St. Julians, Malta, November 1-4, 2010. Proceedings*, 2010, pp. 122–135.

[28] C. Dehnert, S. Junges, J. Katoen, and M. Volk, “The probabilistic model checker storm (extended abstract),” *CoRR*, vol. abs/1610.08713, 2016.

[29] ——, “A storm is coming: A modern probabilistic model checker,” in *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II*, 2017, pp. 592–600.

[30] M. E. Bakir, M. Gheorghe, S. Konur, and M. Stannett, "Comparative analysis of statistical model checking tools," in *Membrane Computing - 17th International Conference, CMC 2016, Milan, Italy, July 25-29, 2016, Revised Selected Papers*, 2016, pp. 119–135.

[31] A. Legay, S. Sedwards, and L. Traonouez, "Plasma lab: A modular statistical model checking platform," in *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, ISoLA 2016, Imperial, Corfu, Greece, October 10-14, 2016, Proceedings, Part I*, 2016, pp. 77–93.

[32] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 4.0: Verification of probabilistic real-time systems," in *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, ser. LNCS, G. Gopalakrishnan and S. Qadeer, Eds., vol. 6806. Springer, 2011, pp. 585–591.

[33] R. Donaldson and D. Gilbert, "A monte carlo model checker for probabilistic LTL with numerical constraints," University of Glasgow, Department of Computing Science, Tech. Rep., 2008.

[34] J. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, and D. N. Jansen, "The ins and outs of the probabilistic model checker MRMC," *Perform. Eval.*, vol. 68, no. 2, pp. 90–104, 2011.

[35] H. L. S. Younes, "Ymer: A statistical model checker," in *Computer Aided Verification, 17th International Conference, CAV 2005, Edinburgh, Scotland, UK, July 6-10, 2005, Proceedings*, 2005, pp. 429–433.

[36] A. David, K. G. Larsen, A. Legay, M. Mikučionis, and D. B. Poulsen, "Uppaal SMC tutorial," *Int. J. Softw. Tools Technol. Transf.*, vol. 17, no. 4, pp. 397–415, Aug. 2015.

[37] K. G. Larsen and A. Legay, *On the Power of Statistical Model Checking*. Springer International Publishing, 2016, pp. 843–862.

[38] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, pp. 183–235, 1994.

[39] J. Rodriguez and P. Cortes, *Predictive Control of Power Converters and Electrical Drives*, ser. Wiley - IEEE. Wiley, 2012.

[40] T. Dragicevic, "Model predictive control of power converters for robust and fast operation of AC microgrids," *IEEE Trans. Power Electron.*, vol. 33, no. 7, pp. 6304–6317, July 2018.

[41] T. Dragicevic and M. Novak, "Weighting factor design in model predictive control of power electronic converters: An artificial neural network approach," *IEEE Trans. Ind. Electron.*, no. Early Access, pp. 1–12, 2018.

[42] P. Bulychev, A. David, K. Larsen, M. Mikučionis, D. Poulsen, A. Legay, and Z. Wang, "Uppaal-smc: Statistical model checking for priced timed automata," vol. 85. Open Publishing Association, 2012, pp. 1–16.