DOI: 10.15439/2021F125
ISSN 2300-5963 ACSIS, Vol. 26

Towards Energy-aware Cyber-Physical Systems Verification and Optimization

Reza Soltani, Eun-Young Kang, Juan Esteban Heredia Mena
University of Southern Denmark
The Mærsk Mc-Kinney Møller Institute
SDU Software Engineering, Odense, Denmark
Email: {resol, eyk, jehm}@mmmi.sdu.dk

***Abstract*—Optimizing CPS behavior in terms of energy consumption can have a significant impact on system reliability. The environment influences the system's behavior, and neglecting the environmental behavior has an indirect negative impact on optimizing the system's behavior. In this work, to increase the system's flexibility, the behavior of the environment is modeled dynamically to apply the disorderliness of its behavior. The resulting models are formally verified. By examining the past environmental behavior and predicting its future behavior, energy optimization is done more dynamically. The verification results acquired using a UPPAAL-SMC show that the optimization of system behavior by predicting the environmental behavior has been successful. Our approach is demonstrated using a case study within an I4 setting.**

I. Introduction

CYBER-PHYSICAL SYSTEMS (CPS) are continuously developing and have become an integral part of Industry 4.0 (I4). The I4 revolution relies on the interconnectivity of machines and automation of processes to improve factory productivity. Modern industry uses assets designed to do different tasks; for example, a robotic arm can assemble, palletize, and solder. Consequently, many production cells can perform the same activity, but each cell's production time differs. For instance, total automation, sustainable production, and efficient scheduling are challenges that need to be addressed to achieve the I4 objective.

According to the United Nations Goals, production sustainability is one of the goals for industry [1]. One of the key factors in CPS is energy consumption, and neglecting energy consumption limits the reliability and safety of such systems. We differentiate between two types of techniques for reducing the energy consumption of a system: whether a CPS has to reconstruct its behavior or make structural modifications such as using light materials, different types of batteries, and efficient motors. Structural modifications can reduce energy consumption but are costly; instead, behavioral changes modify the device software and are cheap.

One of the challenges in CPS analysis is predicting the environmental behavior and modeling the environment since its behavior may change depending on various factors. Although CPS is highly affected by the environment, it has no control over it. For example, in an autonomous vehicle system, the system's performance is based entirely on the analysis of environmental behavior. Knowing information such as how often we are likely to see each sign or which sign is less likely to be seen based on the current situation allows the system to make optimal decisions. For example, the speed can increase with a gentler slope if it is likely to see a stop sign. Sometimes no strong logic can be found for modeling the environment. In this regard, the challenge is run-time data collection, and the system's ability to dynamically model the environment based on these data increases the system's flexibility. One way to do this is to use learning algorithms to study the past and predict future environmental behavior.

In this paper, energy-aware timed behaviors of CPS are specified in Stochastic Hybrid Automata (SHA) [2]. By considering the cost of each behavior, the amount of energy consumption of the system for different modes is obtained. In this way, The minimum and maximum energy consumption of the system can be calculated. Due to the dependence of energy consumption on environmental behavior and the disorderliness of this behavior, the environment with which CPS is associated is dynamically modeled.

Our earlier works [3]–[6] verified the safety of CPS (including the controller and physical parts) and analyzed performance in terms of time and resource constraints: In order to model the unpredictable behaviors of environment, fixed probabilities were considered and allocated to the possible transitions. Whereas our current work utilizes dynamic probabilities instead of the predetermined probability. The main contributions of this paper are:

- Formal modeling of energy-aware timed behaviors of CPS in SHA that captures discrete and continuous behaviors of both the controller and its environment and verifying the system using a statistical model checker, UPPAAL-SMC.
- Dynamic modeling of the environment to increase the flexibility of the system against the disorderliness behavior of the environment.
- Optimize the behavior of the system in terms of energy consumption by predicting the future environmental behavior.

In this work, we use the dynamic probability for each environmental incident automatically updated after a certain number of iterations. According to the past behavior of the environment, the system can predict its future behavior and make decisions based on it. Thus, the environmental behavior of the system is modeled more dynamically.

The rest of the paper is organized as follows. Section II presents our methodology. The industrial case study we used to demonstrate the applicability of our framework is presented in Section III. Section IV shows the modeling and verification results. Section V provides related works, and Section VI describes research challenges and intended future works.

II. Methodology

A. Energy Optimization

The behavior of CPS can be optimized by improving the behavior of its components without structural changes, such as energy modes. We conduct optimization by addressing energy consumption; We model the continuous behavior of the system through SHA, which enables us to find the best path of having a specific behavior with minimum energy consumption. By using statistical analysis, the probability of energy optimization has been investigated up to a certain confidence degree. The behavior of the environment, which directly impacts the energy consumption process, has also been studied.

B. Modeling environmental behavior

To model and predict the environmental behavior of CPS, it is essential to learn from its previous behavior. As discussed in Section I, the dynamic probability is used because the behavior of the environment can change, and modeling it with fixed probabilities will significantly simplify and limit the system model. Therefore, the current work uses variable probabilities for each environmental incident by considering its behavioral disorderliness. These probabilities are updated in each period using closed-loop control and by analyzing the output of the system. To do this, the occurrence time of each event is stored in an array of integers. After a certain number of iterations, this array is sorted using the ascending sort algorithm, and the number of occurrences of each event is calculated. Based on that, the probability of its recurrence in the future is predicted. This possibility will be updated automatically on each iteration. Thus, instead of using fixed and predetermined probabilities for each event, by considering the system's past behavior (environment) and predicting future behavior based on it, dynamic probabilities updated in each iteration are used.

III. Case Study

In this section, we present a case of study that is part of a production plant. Fig. 1 shows a reduced system architecture of a factory. In non-automatized factories, operators manufacture and package the final products. In parallel, the distribution schedule and routes of distribution are defined. In the next stage, operators sort elements for distribution via a truck.

The Industry 4.0 Laboratory at the University of Southern Denmark (SDU) [1] is working in a production line for drone assembling [7]. The current state of the project does not consider the distribution of the drones. We aim to add the capability of distribution to the production plant. The distribution system selects the elements from a storing warehouse, classifies and orders them. The expected final result is a pile of drone boxes ready to be picked by a truck.

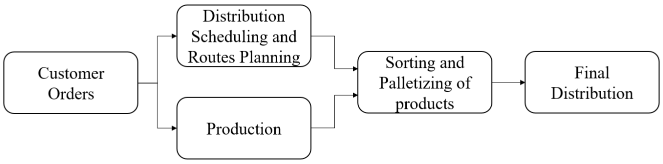
Customer Orders
Distribution Scheduling and Routes Planning
Production
Sorting and Palletizing of products
Final Distribution

Fig. 1. Reduced System Architecture of a Factory

In the proposed distribution system, the drones are obtained from the automated warehouse Effimat [8]. An ER mobile manipulator [9] picks the products from the warehouse and places them in the B&R's magnetic distribution conveyor. A robotic manipulator UR3e [10] sortes and palletizes the packages using distribution schedule and route plan. Fig. 2 shows the elements that compose the distribution system.

We particularly focus on the sorting capability, which is done by the manipulator UR3e. The manipulator UR3e is an anthropomorphic robotic arm made by Universal Robots [10]. The robot UR3e possesses six axes of rotation, as is shown in Fig. 3.

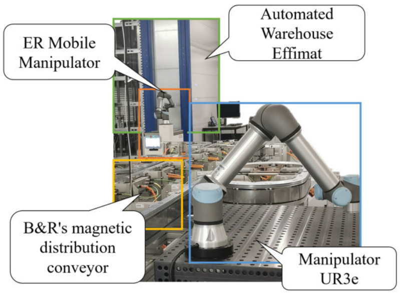
ER Mobile Manipulator
Automated Warehouse Effimat
B&R's magnetic distribution conveyor
Manipulator UR3e

Fig. 2. Distribution system and its components

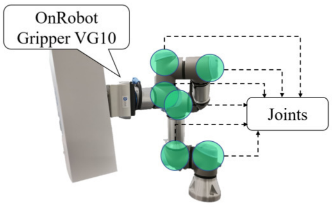

Fig. 3. Robot UR3e joints and griper VG10

In the sorting activity, the robot picks the objects from position A and then places the package in position B or C according to the distribution plan. The frequency of packages depends on the customer orders and the distribution plan. The system includes a position idle. When the robot is in

[1] https://www.sdu.dk/en/forskning/i40lab

the idle position, it consumes less energy than in the other positions. One of the objectives of this study is to design an energy optimal robot behavior based on environmental behavior. Therefore, we measure the energy consumption of each of the robot movements and the rate of consumption of the robot in a static position, which is presented in Table I.

TABLE I
ENERGY CONSUMPTION OF EACH OF THE ROBOT MOVEMENTS

Movement / position	Energy Consumption
A to B	374.0 [J]
A to C	293.3 [J]
B to A	196.1 [J]
C to A	125.7 [J]
B to Idle	198.6 [J]
C to Idle	145.0 [J]
Idle to A	112.3 [J]
A	40.20 [J/s]
B	39.87 [J/s]
C	40.45 [J/s]
Idle	37.45 [J/s]

IV. Modeling and Verification Results

The described case study in Section III, is modeled in SHA, including the energy-aware timed transitions and the amount of energy consumed in each state. The operational semantics of both the system and its environment are formally specified in SHA, and the models are verified against given requirements through UPPAAL-SMC [11] The verification results are displayed in Table II and discussed in Section IV-B

A. Modeling energy-aware behavior and environmental behavior prediction

Fig. 4 shows the robotic arm automaton. Initially, the robotic arm is in idle mode, and with the arrival of the package, it receives a message through the synchronization channel *move?*. In this case, each of the six joints of the robotic arm has to change the angle of their position to put the arm in the ideal position for state A; then, it will be ready to grab the package. Depending on where the package goes, it will reset its joints' angles again and move the package to the desired location. In this model, the amount of energy consumption for each transmission is calculated discretely according to Table I. The robotic arm consumes energy per time unit when it is in one of the three different modes (Idle, StandbyB, StandbyC). The amount of energy consumption for each of the three modes is calculated continuously. For example, the energy consumption per time unit when the robotic arm is in idle mode is calculated using formula *energy'==Idle*.

In this SHA model, the variant *FB* is the functional behavior of the system, which can have an integer value between 0 to 3, representing moving up, moving down, going idle, and coming back from idle behaviors, respectively. The variant *GoR* is the grabber's behavior which can have an integer value of 0 or 1, representing grabbing the box or releasing it, respectively. After moving box from position *A* to *B* or *C*, it is decided whether to stay in the same state (StandbyB or StandbyC in Fig. 4) or to go into the idle mode to save energy.

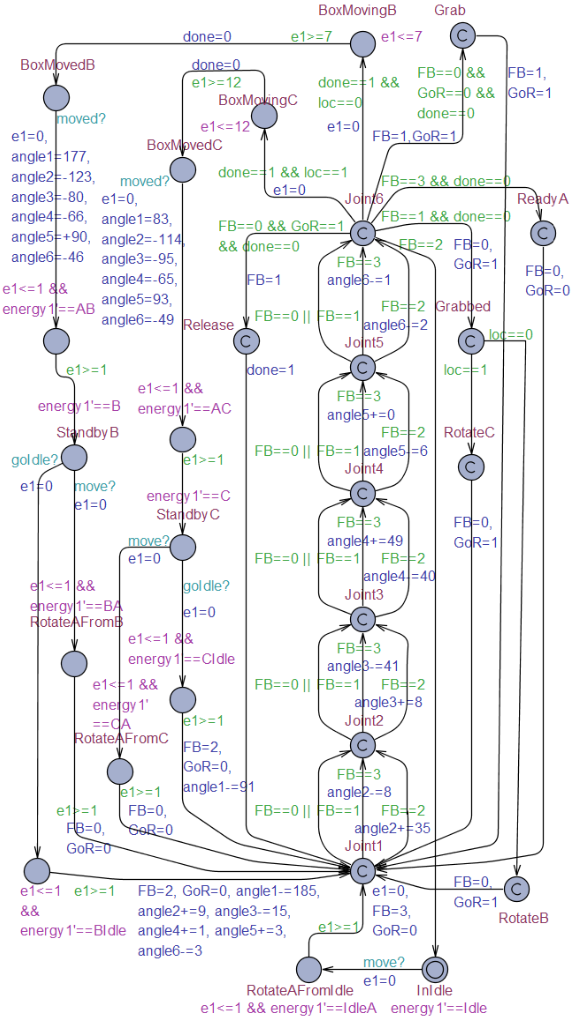

Fig. 4. Robotic arm automaton

Such a decision is made with the help of the environment model. Fig. 5 shows the SHA of the environment: 5 different periods are defined, which indicate the arrival time of the packages. The probability of occurrence of each is a variable of type integer, whose value is updated every 20 times the packages have arrived, and henceforth we call this period iteration.

The general goal of the environment SHA is: (1) to model the behavior of the environment dynamically to include its disorderliness behavior; (2) to change the behavior of the system based on the analysis of the past environmental behavior and to decide on the future behavior of the system based on a prediction. In the first case, five variables *TI1*, *TI2*, *TI3*, *TI4*, and *TI5* are used for the probability of occurrence of each time interval as shown in Fig. 5. These variables are changing in each iteration, and this increases the possibility of including critical factors such as environmental disorderliness

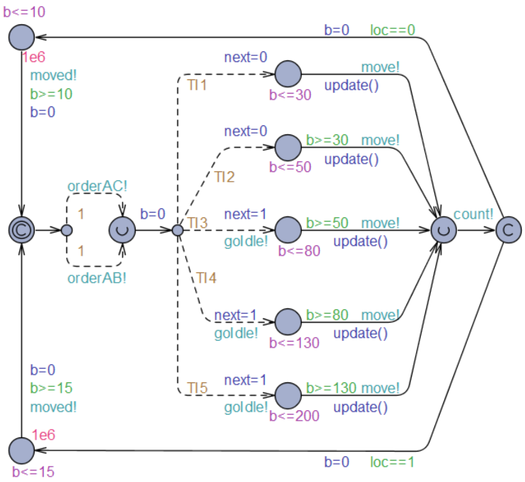

Fig. 5. Environment automaton

Algorithm 1 Decide about system behavior based on a prediction of environmental behavior

```
void update()
if $counter >= 19$ then
    env[19]=b;
    TI1=0; TI2=0; TI3=0; TI4=0; TI5=0;
    counter=0; i= 0; j=1;
    while $i < 20$ do
        while $j < 20$ do
            if $env[i] >= env[j]$ then
                q1 = sort[i]; sort[i] = sort[j]; sort[j] = q1;
            end
            j++;
        end
        i++;
    end
    i= 0;
    while $i < 20$ do
        if $env[i] <= threshold1$ then
            TI1++;
        end
        if $env[i] <= threshold2$ and $env[i] > threshold1$ then
            TI2++;
        end
        if $env[i] <= threshold3$ and $env[i] > threshold2$ then
            TI3++;
        end
        if $env[i] <= threshold4$ and $env[i] > threshold3$ then
            TI4++;
        end
        if $env[i] > threshold4$ then
            TI5++;
        end
        i++;
    end
    q2= TI1+TI2;
    q3= TI3+TI4+TI5;
    if $q2 >= q3$ then
        goIdle=0;
    else
        goIdle=1;
    end
else
    env[counter]=b;
    counter++;
end
```

in modeling. In the second case, using the *Update()* function, each time the boxes arrive is saved in the array, and every 20 times, at the end of the iteration, this SHA uses this function to decide on the behavior of the system for the next iteration. Algorithm 1 shows how the *Update()* function works.

In Algorithm 1, *env* is an array of length 20 that saves the box arrival time. When the 20th box has arrived, it counts the number of times a box arrived in each time interval and updated the probabilities accordingly. Suppose the system goes into the idle mode and a box enters position *A* earlier than a specific time. In that case, the system consumes more energy because the transition cost to enter to and exit from the idle mode is more than staying in standby mode. For this reason, this function makes a decision using the calculated probabilities for the next iteration. It should be noted that in some transitions, energy consumption may increase. Still, the purpose of this prediction is to determine whether, on average, a particular transition (going idle) will decrease or increase energy consumption. Verification results show that this algorithm has been successful in reducing energy consumption.

B. Verification results

In this section, we discuss the verification results that are given in Table II: *energy* indicates the system's energy consumption in the presence of idle mode and *energy2* indicates the system's energy consumption in the absence of idle mode. The first five queries are related to energy consumption. As presented in the verification results, by predicting the behavior of the environment and changing the system's behavior accordingly, the amount of energy consumption is reduced by a probability of close to 100 (query 3). For example, in the table, in 3 hours (10800 seconds) and with an average of 20 runs, 60 kJ of energy consumption has decreased. In some transitions, there is a possibility that energy consumption in the case of having idle mode increases (queries 4 and 5).

As illustrated in Table I, the cost of going idle from position *C*, and coming back to position *A* is more than the cost of staying in standby mode for some time and then going to position *A*. Therefore, it is beneficial to move to the idle mode when the duration of staying in that mode is longer than a certain amount of time. The proposed method examines whether such behavior is beneficial or detrimental to the system. Since the proposed algorithm makes decisions for each iteration (for every 20 boxes), and the decision is based on an average reduction in energy consumption, in some cases, the variable *energy* may be more than *energy2* (especially at the beginning due to small differences between energy and energy2), because by predicting the behavior of the environment, decisions are made for the entire next iteration. This is why query 5 (which indicates that energy consumption is always lower by having idle mode) is invalid. However, energy consumption generally decreases over time.

The modeling and verification results (first five queries) show that, in the case of being aware of the environmental behavior, the system can optimize the energy consumption by improving the behavior of its components. The second group of verification results (queries 6 to 16) is related to the safety properties of the robotic arm. For example, in each of the different states, such as *Idle*, *A*, *B*, and *C*, the joints of the arm must be at a certain angle in order to achieve the ideal

TABLE II
VERIFICATION RESULTS

	Query	Result
1	**E[<=10800; 20] (max:energy)**	2899.54
2	**E[<=10800; 20] (max:energy2)**	2959.38
3	**Pr[t<=10800](<>energy<=energy2)**	[0.901855,1]
4	**E<> Behavior.StandbyC and Behavior2.StandbyC and energy >= energy2**	Valid
5	**A[] Behavior.StandbyC and Behavior2.StandbyC and energy <= energy2**	Invalid
6	**A[](Behavior.InIdle imply Behavior.angle1==-8 and Behavior.angle2==-79 and Behavior.angle3==-87 and Behavior.angle4==-105 and Behavior.angle5==87 and Behavior.angle6==-51)**	Valid
7	**A[] (Behavior.ReadyA imply Behavior.angle1==-8 and Behavior.angle2==-87 and Behavior.angle3==-128 and Behavior.angle4==-56 and Behavior.angle5==87 and Behavior.angle6==-52)**	Valid
8	**A[] (Behavior.StandbyC imply Behavior.angle1==83 and Behavior.angle2==-114 and Behavior.angle3==-95 and Behavior.angle4==-65 and Behavior.angle5==93 and Behavior.angle6==-49)**	Valid
9	**A[] (Behavior.StandbyB imply Behavior.angle1==177 and Behavior.angle2==-123 and Behavior.angle3==-80 and Behavior.angle4==-66 and Behavior.angle5==90 and Behavior.angle6==-46)**	Valid
10	**A[] (Behavior.BoxMovedC imply Behavior.e1<=15)**	Valid
11	**A[] (Behavior.BoxMovedB imply Behavior.e1<=10)**	Valid
12	**Behavior.BoxMovedC –> Behavior.Grab**	Valid
13	**Behavior.BoxMovedB –> Behavior.Grab**	Valid
14	**Behavior.InIdle –> Behavior.ReadyA**	Valid
15	**Behavior.InIdle –> Behavior.Grab**	Valid
16	**A[] not deadlock**	Valid

position.

V. Related works

For optimizing the energy consumption, authors in [12]–[15] have focused on behavioral techniques, such as time scaling, task scheduling, and break-release scheduling. In the time scaling technique, which refers to modifying energy consumption by scaling the task execution time, The optimization technique is not straightforward because the relation between energy consumption and task execution time is not linear.

Authors in [16], presented a scheme for energy transfer between nodes. The authors used a solar panel with a certain efficiency and energy loss to charge the nodes. The solar panel was considered to be in clear-sky daily irradiation, which is almost ideal. Even issues such as energy loss can be increased or decreased under the influence of environmental conditions.

Authors in [4], [6], predict the energy consumption of the system and examine whether the energy consumption is in a certain range or not. In these works, the performance of the system is based on the analysis of environmental behavior. Both of these works could be improved by considering the behavior of the environment and its impact on energy consumption. Also, in [17], which examines the impact of security on system safety, the probability of a successful message is permanently fixed, something that may not always be true in reality.

VI. Discussion and future work

Among the important challenges that CPS face is environmental behavior. The behavior of CPS is highly dependent on how the environment proceeds. In our case study, reducing energy consumption by changing the system's behavior depends on whether the boxes arrive earlier or later than a specific time. As mentioned in Section IV, the amount of energy consumption for the case study is reduced when the idle mode is not used in the case that the boxes' arrival time to the position *A* is earlier than a specific time. Since the box arrival time is unpredictable, such environment behaviors cannot be decided by the system. This presents a challenge to the improvement of energy consumption. To solve this challenge, we dynamically modeled the environment's behavioral disorderliness and decided on the next iteration continuously.

In this work, we analyze the reduction of energy consumption by improving the behavior of the components without structural changes by using cost-optimal reachability analysis. Due to the dependence of system behavior on the environment, we examine the behavior of the environment from two aspects. First, the behavior of the environment may be quite irregular, so using fixed, predefined probabilities for incidents makes the existing system much simplified. For this purpose, to increase the system's flexibility against the disorderliness behavior of the environment, it is modeled dynamically. Second, the system can optimize its behavior by predicting the behavior that the environment may have in the future. So, by examining the past environmental behavior, the system can dynamically adjust its behavior to optimize its behavior in terms of energy consumption. We demonstrate the applicability of our framework on an industrial case study within the I4 Lab at SDU, which can also be extended to other assets in the CPS and IoT domains.

To increase system reliability, the system must be able to operate despite the behavioral disorder of the environment. This study demonstrates that it is possible to improve system performance by predicting its future behavior. Therefore, as future work, we intend to use machine learning (ML) in the decision-making process [18] to improve the way we predict environmental behavior and check the mathematical expectation of each incident. This helps the system make a decision, based on whether it will gain a profit or a loss by choosing a specific route. Moreover, by including the Entropy criterion [19], we will measure the severity of this disorderliness as well as the efficiency of our decision-making algorithm in different conditions.

Since such systems communicate with the main server through IoT communication protocols, we will also include security checks. As a continuation of our previous work [17], which showed how a breach in cybersecurity could endanger safety, we will improve both the safety and security aspects of CPS with IoT features. As ongoing work, we will: (1) Use ML algorithms in the decision-making process to increase the efficiency of the decision; (2) Measure the severity level of the disorderliness of the environment by including the Entropy criterion; (3) Improve and integrate safety and security in CPS & IoT verification; (4) Demonstrate the applicability with other verification/optimization tools supporting these techniques.

Acknowledgement: This research has been performed under the EF-CPS: Trustworthy CPS & IoT within I4 project, and “Energy-efficient Programming of Collaborative Robots” project funded by ELFORSK.

References

[1] U. Nations, “The 17 goals for sustainable development,” 2015. [Online]. Available: https://sdgs.un.org/goals

[2] J. Lygeros and M. Prandini, “Stochastic hybrid systems: A powerful framework for complex, large scale applications,” *European Journal of Control*, vol. 16, p. 583–594, 11 2010.

[3] E.-Y. Kang, D. Mu, and L. Huang, “Probabilistic verification of timing constraints in automotive systems using uppaal-smc,” in *Integrated Formal Methods*, ser. EuroSys ’10. Cham: Springer International Publishing, 2018, pp. 236–254.

[4] E.-Y. Kang, D. Mu, L. Huang, and Q. Lan, “Model-based analysis of timing and energy constraints in an autonomous vehicle system,” in *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2017, pp. 525–532.

[5] ——, “Verification and validation of a cyber-physical system in the automotive domain,” in *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2017, pp. 326–333.

[6] E.-Y. Kang, L. Huang, and D. Mu, “Formal verification of energy and timed requirements for a cooperative automotive system,” in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, ser. SAC ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 1492–1499. [Online]. Available: https://doi.org/10.1145/3167132.3167291

[7] S. C. Jepsen, T. I. Mørk, J. Hviid, and T. Worm, “A pilot study of industry 4.0 asset interoperability challenges in an industry 4.0 laboratory,” in *2020 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2020, pp. 571–575.

[8] “Effimat automated warehouses.” [Online]. Available: https://effimat.com/

[9] “Enabled robotics.” [Online]. Available: https://www.enabled-robotics.com/

[10] “Collaborative robots universal robots.” [Online]. Available: https://www.universal-robots.com/

[11] A. David, K. Larsen, A. Legay, M. Mikučionis, and D. Poulsen, “Uppaal smc tutorial,” *International Journal on Software Tools for Technology Transfer*, vol. 17, 01 2015.

[12] D. Meike, M. Pellicciari, G. Berselli, A. Vergnano, and L. Ribickis, “Increasing the energy efficiency of multi-robot production lines in the automotive industry,” *IEEE International Conference on Automation Science and Engineering*, pp. 700–705, 2012.

[13] A. Vergnano, C. Thorstensson, B. Lennartson, P. Falkman, M. Pellicciari, F. Leali, and S. Biller, “Modeling and optimization of energy consumption in cooperative multi-robot systems,” *IEEE Transactions on Automation Science and Engineering*, vol. 9, no. 2, pp. 423–428, 2012.

[14] D. Meike, M. Pellicciari, and G. Berselli, “Energy efficient use of multirobot production lines in the automotive industry: Detailed system modeling and optimization,” *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 798–809, 2014.

[15] M. Pellicciari, G. Berselli, F. Leali, and A. Vergnano, “A method for reducing the energy consumption of pick-and-place industrial robots,” *Mechatronics*, vol. 23, no. 3, pp. 326–334, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.mechatronics.2013.01.013

[16] A. Gamatié, G. Sassatelli, and M. Mikučionis, *Modeling and Analysis for Energy-Driven Computing using Statistical Model-Checking; Design, Automation and Test in Europe Conference, Virtual, France.*, 02 2021.

[17] L. Huang and E.-Y. Kang, “Formal verification of safety & security related timing constraints for a cooperative automotive system,” in *Fundamental Approaches to Software Engineering*, R. Hähnle and W. van der Aalst, Eds. Cham: Springer International Publishing, 2019, pp. 210–227.

[18] A. Kuhnle and G. Lanza, “Application of reinforcement learning in production planning and control of cyber physical production systems,” in *Machine Learning for Cyber Physical Systems*, J. Beyerer, C. Kühnert, and O. Niggemann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 123–132.

[19] H. Li, “Entropy reduction via communications in cyber physical systems: How to feed maxwell’s demon?” in *2015 IEEE International Symposium on Information Theory (ISIT)*, 2015, pp. 2206–2210.