An Approach for Synthesizing Energy-Efficient Controllers for Production Systems from Scenario-Based Specifications*

Joel Greenyer[1] and Daniel Gritzner[1]

Leibniz Universität Hannover, Fachgebiet Software Engineering, Welfengarten 1,
D-30167 Hannover, Germany,
{greenyer|daniel.gritzner}@inf.uni-hannover.de

Abstract. Rising costs of energy production have made saving energy an important topic in many areas, including modern production systems. One idea is taking advantage of the braking energy becoming available when decelerating moving components of a machine. However, using the braking energy is a difficult task, as it requires the synchronization of the movement of different parts of a production system. Storing this energy often is not an economically viable solution. Optimizing the entire production process, in particular recognizing all optimization opportunities while still fulfilling all requirements, is a difficult and error-prone task. To support engineers in this task, we propose behavior modeling through scenario-based specifications, to be able to automatically generate all valid behavior strategies for the system, combined with energy usage modeling through state machines, to automatically identify the most energy-efficient strategy.

Keywords: Energy-Efficiency, Optimization, Scenarios, Controller Synthesis

1 Introduction

Rising costs of energy production have made saving energy an important topic in many areas, including modern production systems. However, not all ideas for saving energy are easily implementable. One increasingly popular idea is taking advantage of the braking energy becoming available when decelerating moving parts of a machine [3].

Using the braking energy is a difficult task, though, as it requires the synchronization of the movement of various parts of a production system. A simplified example of this is shown in Fig. 1 which shows the power consumed by two robot arms over time. When the two arms move after each other, the braking energy is lost entirely. If the second arm starts accelerating at the same time as the first arm starts decelerating, the total energy cost for moving both arms decreases significantly as the cost for moving the second arm is reduced by taking advantage of the braking energy of the first arm. Storing braking energy instead of using it directly is often not an economically viable solution.

* This research is funded by the DFG project EffiSynth.

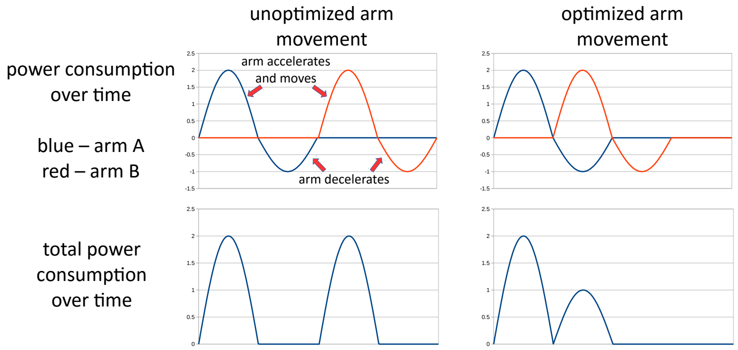

Fig. 1. Simplified curves of the power consumption over time of moving robot arms in a production system

While the basic idea of using braking energy is simple enough, applying this to an entire production process is a difficult task. The behavior of the system cannot be modified arbitrarily without violating requirements. This may make finding opportunities for optimization difficult and this difficulty is further increased if multiple mutually exclusive opportunities are present. Choosing the best combination of these opportunities is a non-trivial task, and manually implementing the optimizations may even introduce new defects.

As a solution to these challenges we propose an automated approach to finding the optimal behavior strategy for a production system. Our approach combines scenario-based behavior modeling and state machines for modeling energy usage. From the scenarios we can automatically generate all valid behavior strategies, i.e., strategies which do not violate the specification. The energy models can then be used to identify the most energy-efficient strategy.

In this paper we expand on our work in [3] by adding explicit energy models to improve readability of the models and to separate the modeling of an object's attributes such as energy usage from the modeling of the object's class' behavior. Additionally, we extend our approach to include iterative improvements of the generated strategies to add the ability to account for complex electrical phenomena not captured by our models. In ongoing research we also work towards generating executable code for the generated strategies.

This paper is structured as follows: Sect. 2 introduces an example of a production system used to explain our approach. Sect. 3 explains our approach for finding the most energy-efficient controller and Sect. 4 describes preliminary results of this approach. The paper concludes with an outlook for future research in Sect. 6 and related work in Sect. 5.

2 Example

Fig. 2 shows an example of a production system. In this example blanks arrive via a conveyor belt, are picked up by a robot arm and then put into a press. Once

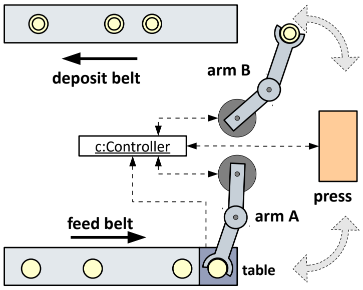

Fig. 2. Example of a modern production system with two conveyor belts, two robot arms and a press.

the press finishes, a different robot arm picks up the pressed item and places it onto a deposit conveyor belt.

The intended system behavior and requirements can easily be described in short, intuitive scenarios, e.g., (a) arm A must be back at the conveyor belt before the next blank arrives and (b) both arms must move away from the press before it starts pressing to avoid damage to the arms.

3 Approach

Our approach combines scenario-based modeling and state machines as models for energy usage. An overview of the approach is shown in Fig. 3. Scenarios model the intended behavior of the system and can be used to automatically derive *controllers* for the system, i.e., strategies for the system to behave that do not violate any requirements. The energy state machines can then be used to find the most energy-efficient candidates. Since the energy models and algorithms to identify these candidates use approximations, a more detailed, computationally more expensive, evaluation of each candidate is performed. These evaluations take into account the actual power over time curves of each component as well as the time it takes to perform the different actions. This evaluation step offers two benefits (a) the confidence in the ranking of the candidates is increased and (b) the approximate values in the energy model can be updated to reflect the actual energy usage more closely. Candidate selection and evaluation is performed iteratively until the most energy-efficient controller is found. For this controller, PLC code, to run on an actual controller used for production systems, is generated.

3.1 Scenario-Based Behavior Modeling

Scenarios are an intuitive way to describe how components of a system may, must or must not behave. They can be used to easily describe the intended behavior, as well as requirements, of a system from an inter-component point of view. The scenario in the top-left of Fig. 3 describes how the system must react when an arm successfully picks up a blank. The arm is instructed by the controller to

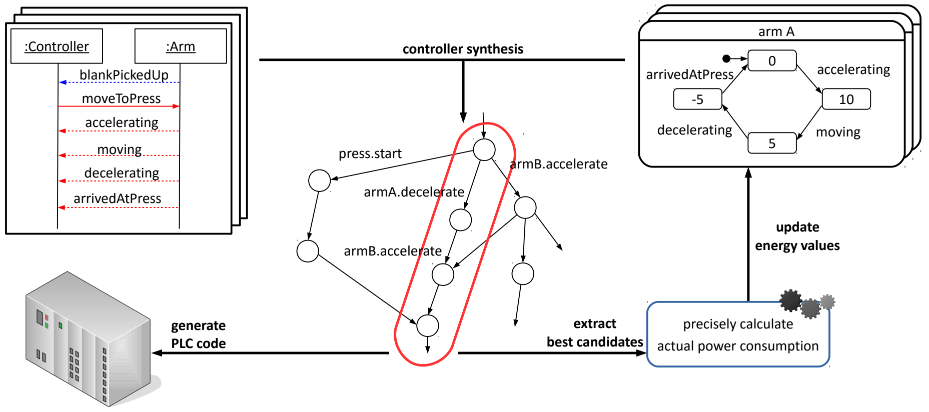

Fig. 3. Overview of our proposed approach.

move towards the press and, while doing so, it will continuously report its state to the controller.

A formal, scenario-based specification is a collection of scenarios. In our approach, we use a modeling language extending the concepts of Live Sequence Charts to specify scenarios and we use the Play-Out algorithm [4] to determine how a collection of scenarios gets interwoven into a valid behavior strategy.

3.2 Energy Consumption Modeling

We propose using state machines to model the approximate energy consumption of each component in a system. In such a state machine, the transitions are labeled with the same events which are also used in scenarios. The example shown at the top of Fig. 3 illustrates this. When arm A gets instructed to move to the press, it will start accelerating. This event will cause both the scenarios and the associated state machine to progress further. The arm is now in a state in which it will use approximately 10 units of energy. Eventually, a decelerating event will occur indicating that 5 units of braking energy are now available to other components.

3.3 Controller Synthesis

The aforementioned Play-Out algorithm, together with a modified game-solving algorithm used by UPPAAL TIGA [1], can be used to automatically generate behavior strategies for the specified system [2]. The scenario-based specification induces a game graph whose transitions correspond to actions and state changes of the components of the system. Controllers for the system can be found in this graph by searching for cycles which satisfy all scenarios. The existence of such cycles, i.e., being able to follow transitions indefinitely without running into a state that violates the specification, is equivalent to the specification being free of inconsistencies.

Fig. 3 shows a state space generated via the Play-Out algorithm from a scenario-based specification with a controller candidate highlighted in red. Once a candidate has been identified, the energy model can be used to assign energy usage values to the states and transitions of the candidate. This information can then be propagated along paths to determine the approximate energy consumptions of entire paths and, ultimately, of the entire controller.

3.4 Precise Power Consumption Calculation

The controller synthesis generates candidates which are only optimal according to approximate energy values. In order to increase the confidence in our results, we generate multiple good candidates and evaluate each of those candidates using more precise models such as the actual power curves and time required for each activity. Additionally, these calculations also provide us with new approximate values to be used in our energy models to generate better candidates in subsequent iterations.

3.5 PLC Code Generation

To prevent defects, including wasting energy, as well as mitigating some of the cost caused by putting effort into writing a scenario-based specification, we work on automatically generating code for programmable logic controllers (PLCs) from controller we synthesized in previous steps.

4 Preliminary Results

We implemented a prototype which assigns energy values to the transitions of candidate controllers. Every time a transition causes an energy model to change its state, the transition is assigned a non-negative energy value. If, e.g., the arm modeled in the top-right of Fig. 3 changes its state from accelerating to moving, the corresponding transition in the controller is assigned a value of 10 minus any available braking energy.

To compare different candidates, we calculate the occurrence probability of each transition in a controller by treating controllers as Markov chains with each outgoing transition being equally probable. A controller's estimated mean energy consumption is the sum of each transition's probability times its energy value.

With this prototype and a model of the system shown in Fig. 2 with both arms consuming and producing equal amounts of energy when moving from or to the press, we were able to identify a strategy which could recoup more than 80% of the braking energy of the arms.

5 Related Work

Pellicciari et al. [6] try to utilize idle times of individual robot trajectories. Wigström et al. [7] use dynamic programming to determine an optimal task schedule for a multi robot cell. However, neither approach is able to automatically find optimization opportunities outside of isolated time windows or predefined energy exchange opportunities between components. Other approaches usually have similar issues, as well as often relying only on approximations.

To the best of our knowledge, synthesizing energy-efficient discrete controllers for production systems has barely been investigated outside of our own work in [3]. There are algorithms for synthesizing time-optimal strategies from timed game automata in UPPAAL TIGA [1] or from Markov decision processes in PRISM [5]. However, neither tool is able to optimize for energy-efficiency. Also, we strongly believe that our scenario-based approach is easier to use.

6 Future Work

In future research, we are cooperating with researchers from the field of mechatronics. Issues, we want to address cooperatively, are the precise calculation of how much power a controller consumes, PLC code generation, as well as the evaluation of the approach proposed in this paper on an actual hardware as it is found in real production systems.

Furthermore, we want to expand our models and synthesis algorithms to also include time information. Our goals are to (a) model real-time requirements such as "new blanks arrive at least every four seconds", as well as (b) to identify components which may move more slowly without slowing down the overall process to take advantage of the fact that slower movement requires less energy.

References

1. G. Behrmann, A. Cougnard, A. David, E. Fleury, K. G. Larsen, and D. Lime. UPPAAL-Tiga: Time for Playing Games! In *Proc. 19th Int. Conf. on Computer Aided Verification (CAV)*, pages 121–125, 2007.
2. J. Greenyer, C. Brenner, M. Cordy, P. Heymans, and E. Gressi. Incrementally synthesizing controllers from scenario-based product line specifications. In *Proc. 9th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering 2013*, 2013.
3. J. Greenyer, C. Hansen, J. Kotlarski, and T. Ortmaier. Towards synthesizing energy-efficient controllers for modern production systems from scenario-based specifications. *Procedia Technology (Proc. of the 2nd Int. Conf. on System-Integrated Intelligence, SysInt 2014)*, 15(0):388–397, 2014.
4. D. Harel and R. Marelly. Specifying and Executing Behavioral Requirements: The Play-In/Play-Out Approach. *SoSyM*, 2:82–107, 2003.
5. M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. 23rd Int. Conf. on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.
6. M. Pellicciari, G. Berselli, F. Leali, and A. Vergnano. A method for reducing the energy consumption of pick-and-place industrial robots. *Mechatronics*, 23(3), 2013.
7. O. Wigstrom, B. Lennartson, A. Vergnano, and C. Breitholtz. High-level scheduling of energy optimal trajectories. *IEEE Transactions on Automation Science and Engineering*, 10(1):57–64, 2013.