On Evaluation of Alternative Switching Strategies for Energy-Efficient Operation of Modular Factory Automation Systems

Sebastian Mechs[1], Steffen Lamparter[2]
Siemens AG Corporate Technology
Intelligent Systems and Control
Otto-Hahn-Ring 6
Munich, Germany
[1]sebastian.mechs.ext@siemens.com
[2]steffen.lamparter@siemens.com

Jörg P. Müller
Department of Informatics
Clausthal University of Technology
Julius-Albert-Str. 4
Clausthal-Zellerfeld, Germany
joerg.mueller@tu-clausthal.de

Abstract

In this paper, we propose a state-based, modular model for automation systems enabling the calculation and optimization of switching strategies for energy-efficient operation. This concept is based on the model of networked (priced) timed automata that we call energy system model. It comprises structural, temporal and energetic information of automation subsystems' behavior providing a basis for automated analysis and calculation of feasible, energy-optimal switching strategies for the complete automation system.

The presented real-world automation system enables proof-of-concept and allows discussing the arising, domain-specific complexity of the temporal and energetic analysis. By means of the energy system model, we provide a basis for calculating energy-optimal operation of modular automation systems and support the comparison of alternative, energetic switching strategies of subsystems. For this purpose, a symbolic reachability analysis complemented by solving a constrained optimization problem is applied in this approach.

1 Introduction

Energy efficiency of industry's production has to be increased because of environmental reasons. Automation technology can make considerable contribution to the reduction of its current energy demand. As interface to production systems, it may serve as executing and supervising infrastructure for switching production systems and its subsystems to energy-efficient operating modes during idle times. Up to now, idle times in production attract no big attention with regard to energy savings and hence systems are left in operating modes that are not energy-efficient.

Assuming two-shift operation with one hour lunch time per day (Fig. 1), holiday weekends and 55 per cent power consumption during idling, about 22 per cent energy can be saved by powering off the plant completely during breaks. These observations are based on energetic measurements in the testing plant (Fig. 2) documenting the potential for energy savings during production breaks. In the context of bottling and storing objects, the power consumption of subsystem's operating modes was recorded. Fig. 2 shows individual measurement results for nine subsystems as part of the testing plant. The power consumption P [W] is shown for each subsystem's operating mode. The 55 per cent power consumption during idling and production breaks is due to the fact that modules like frequency converters that are left idling still run at one forth of their maximum power input.

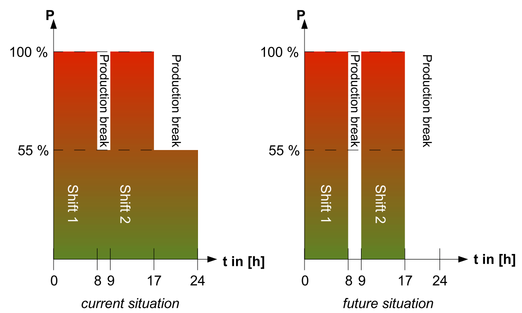

Figure 1. Current intra day power consumption (left) and future intra day power consumption (right) of the testing plant

Considering idling intervals and the presented saving potential, automation systems should be automatically switched to energy saving operating modes in order to reduce the carbon footprint and the operating costs of automated production effectively. In state-of-the-art automa-

978-1-4673-4737-2/12/$31.00 ©2012 IEEE

tion systems, this potential is currently not sufficiently exploited because of the complexity arising from guiding automation systems - that are generally unitized - to specific operating modes. Basically, this is due to the fact of complex, temporal-interdependent behavior of the automation system's subsystems. The lack of missing energy system models is the major obstacle for a software-supported, detailed analysis of the energy-optimal operation of automation systems. In this context, there is an urgent need for a model-driven description of temporal and energetic behavior with special regard to a modular component-view respecting process-related dependencies for instance.

Based on networked priced timed automata, we propose a mapping of the above requirements to a mathematical well founded formal model. Analyzing the temporal and energetic behavior of a modular automation system, the energy-optimal operation respecting high availability requirements of automation systems can be met. In this perspective, we compare alternatives of switching sequences in automation subsystems by applying a symbolic reachability analysis complemented by solving a constrained optimization problem in this approach. A key point in this context is the automatic calculation of feasible switching operations to reach energy-efficient operating modes. The formal energy system model serves as basis for the evaluation and analysis of alternative, energetic switching strategies.
The outline of this paper is as follows. The prerequisite of our approach is an energy system model for automation systems that enables the temporal and energetic behavior to be described and analyzed. This model uses networked (priced) timed automata for modeling temporal properties of modules and energetic aspects of operating modes (Section 2). In the remainder, the analytical aspects based on this energy system model are presented with special focus on temporal reachability. The way alternative energetic sequences for automation systems are calculated is presented in Section 3. The approach attenuates the state space explosion that is an inherent problem of such analysis by a two-step calculation: a symbolic reachability check which reduces the scope of switching strategies being analyzed (Section 3.1) as well as a constrained optimization step for finding energy-optimal switching and operating schedules (Section 3.2). Subsequently, the results of the reachability and optimization steps are presented (Section 3.3). For clarification, the theoretical considerations are exemplified referring to subsystems of an industrial automation system (use case references). Related work is discussed in Section 4, before concluding and giving an outlook in Section 5.

2. Prerequisites

The basis for analytical, energy-focusing considerations of this paper is a weighted, discrete-time model providing a graphical representation [8]. It allows intuitive descriptions of the temporal and energetic switching behavior of networked automation subsystems to enable the analysis of switching behavior in automated production systems in Section 3. Switching behavior is executed in form of switching operations that cause a subsystem taking action from an initial or current operating mode to a specified target operating mode. Operating modes of the automation system are modeled with special respect to transition times between these modes and denoted as temporal switching behavior (Subsection 2.1) applying the formal notion of timed automata [3] for individual automation subsystems. Energetic information (Subsection 2.2) of a subsystem's operating modes is complemented by the notion of priced/weighted timed automata [5], [4]. This description is extended by a modular view on the automation system incorporating process-related dependencies that leads to a network of priced timed automata (Subsection 2.3).

2.1. Temporal switching behavior

In general, automated production systems have modular structures with system to subsystem subordination. In terms of modularity, automation subsystems are self-contained components of a system that fulfill dedicated automation tasks. Depending on the hardware and software of the subsystem, there are several operating modes available. The switching between operating modes is time-dependent and needs to be considered when making predictions about its temporal switching behavior. Therefore, each subsystem's temporal switching behavior is represented by its timed automaton that we call timed system in the following:

Definition
The temporal switching behavior of a single automation subsystem is represented by a timed automaton called timed system $\mathrm{TS} = (\mathrm{M}, \mathrm{Act}, \mathrm{m}_0, \mathrm{C}, \mathrm{Inv}, \mathrm{E})$:

- M is a finite set of modes, consisting of a set of operating modes OM and a set of transitional modes TM, $\mathrm{M} \subseteq \mathrm{OM} \cup \mathrm{TM}$
- Act is a set of action labels
- $\mathrm{m}_0 \in M$ is the initial operating mode
- C is a finite set of real-valued variables called clocks
- Inv: $\mathrm{M} \rightarrow \mathrm{F(C)}$ are invariants of operating modes allowing to define global, temporal constraints on operating modes in the form of $\mathrm{f} ::= \mathrm{x} \sim \mathrm{c}$ and $\sim \in \{<, \leq\}$
- A finite set of edges is denoted by $\mathrm{E} \subseteq \mathrm{M} \times \mathrm{Act} \times \mathrm{G(C)} \times 2^{\mathrm{C}} \times \mathrm{U(C)} \times \mathrm{M}$
with G(C): $\mathrm{g} ::= \mathrm{x} \sim \mathrm{c}$ and $\sim \in \{<, \leq, =, \geq, >\}$ denote clock guards and U(C): $\mathrm{u} ::= \mathrm{x} := \mathrm{c}$ denote clock updates

A state of a timed system is a tuple (m, p), where $\mathrm{m} \in \mathrm{M}$ and $\mathrm{p} \in \mathbb{R}_{\geq 0}$ is a clock valuation. The semantics

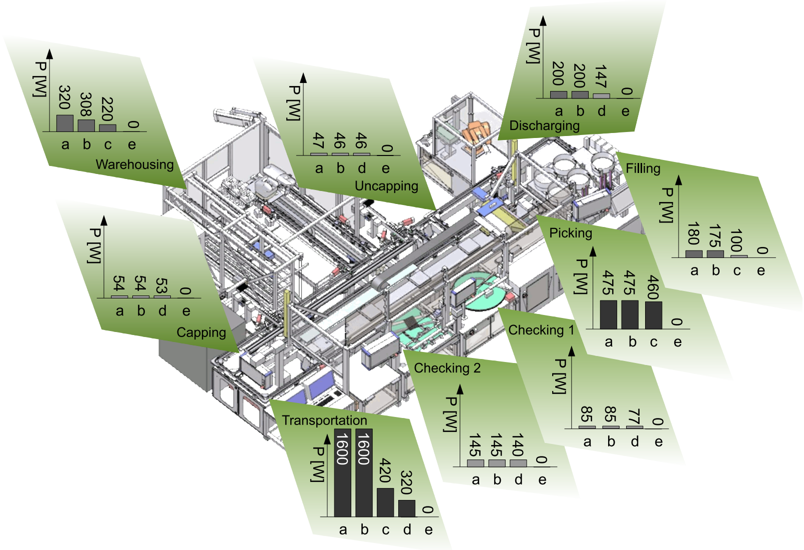

Figure 2. Plant model and power consumption of station's operating modes, testing plant (a = production mode, b = ready for production, c = standby mode, d = synchronization mode, e = off mode)

of a timed system is analog to that of a timed automaton where two different transition types occur: discrete transitions and delay transitions. A discrete transition is a transition $(m, p) \xrightarrow{a} (m', p')$, with an edge (m, g, a, r, m′) so that the guard *g* evaluates to true in (m, p), p′ is p where all clocks in the set r are reset. A delay transition is represented by $(m, p) \xrightarrow{d} (m, p')$ where all clocks are incremented by $d := p' = p + d$ while the invariant Inv(m) of the mode m is satisfied.

The model of the timed system permits the real-time description of the temporal switching behavior of subsystems between operating modes M. Note, that transitional modes TM $\in$ M are introduced to describe the time necessary to switch from one operating mode om_i to another om_{i+1} with $i \in \mathbb{N}$.

Use case reference

In Fig. 3, a detail of the switching behavior of the subsystem *Filling* is shown using a timed system description:

- M = {off, off_stand-by, stand-by, stand-by_ready-for-production, ready-for-production, ready-for-production_off}
- m_0 = {ready-for-production}
- C = {g, l} where l (p and q is the valuation of l respectively g) is used to describe transitional times of the subsystem and g is employed for measuring global evolution of time
- Example temporal trace (bold arrows) with $p_0, q_0 = 0$ and $p, q \in \mathbb{R}_{\geq 0}$:

(ready-for-production, p_0, q_0) $\rightarrow$ (stand-by, p_0, q_0) $\xrightarrow[u_1:=0]{}$ (stand-by_ready-for-production, p_0, q_0) $\xrightarrow{p_1=p_0+60}$ (stand-by_ready-for-production, $p_1 = 60$, $q_1 = 60$) $\xrightarrow[u_1:=0]{}$ (ready-for-production, p_0, $q_1 = 60$)

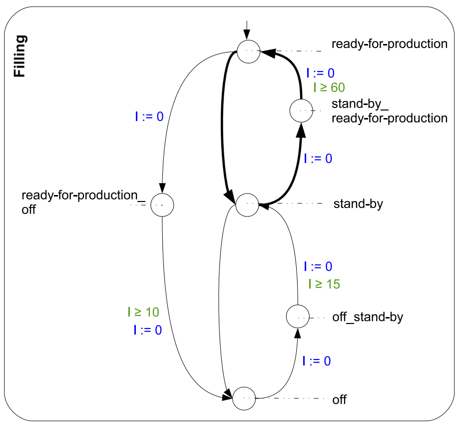

Figure 3. Temporal switching behavior of subsystem *Filling*

Note, that we show only parts of the temporal and energetic behavior in the use case references for simplification reasons.

2.2. Energetic information

Energetic information about modes is necessary in order to evaluate the system with regard to energy efficiency. Therefore this information is added to the existing model in form of cost rate annotations like in [5], [4].

Definition
An energy system is a tuple ES (TS, Ω) with Ω as cost function Ω: $M \cup E \rightarrow \mathbb{N}_0$ associating operating modes M and transitions E with costs. TS is a timed system. Discrete and delay transitions may be provided with cost information [6]. Discrete transitions are defined as (m, u) $\underset{\mathrm{pc}}{\longrightarrow}$ (m′, u′) with pc $\in \Omega$(e) and $e \in E$. Delay transitions are defined as (m, u) $\xrightarrow[\mathrm{pc}]{\mathrm{d}}$ (m, u′) with pc = d $\cdot$ Ω(m) and $m \in M$.

Use case reference
The energetic aspect is added to the automaton model in form of cost rate annotations to operating modes and transitional modes that represent the power consumption $pc\ [W] = \frac{[J]}{[s]}$ (Fig. 4):

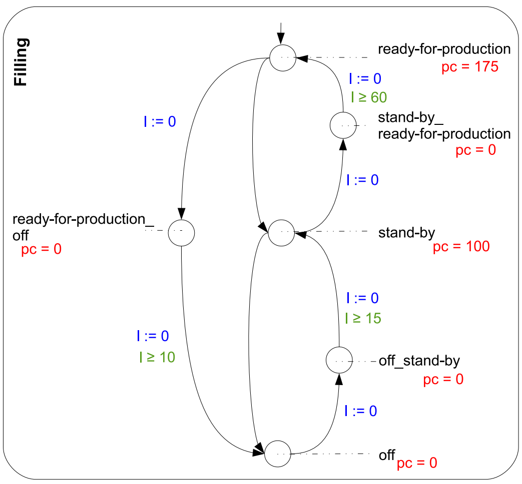

Figure 4. Energy system of subsystem *Filling*

Note, that for completeness reasons, insignificant power consumption rates are set to $pc := 0$ in this example.

2.3. Process-related dependencies

As mentioned in the previous subsection, the modularity of today's automation systems needs to be appropriately supported by a descriptive model. However, modularity leads to interconnected and dependent modules (subsystems). In modular automation systems, subsystems are interconnected via process-related dependencies and therefore the feasible switching possibilities are constrained. Process-related dependencies influence the allowed precedence order of switching operations within switching strategies. For the ease of modeling, automation subsystems are described as networks of timed energetic systems providing an appropriate graphical representation. The hierarchical structure of automation subsystems is projected onto a flattened network of communicating timed automata with energy annotations. Process-related dependencies arising from modularity of the automation system are considered as guards (logical constraints) on transitions referring to shared variables. Shared variables can be set (assigned) by one automaton and can be read by other automata. This detail enables to formulate hardware-specific and process-related dependencies using logical constraints.

Definition
The timed system network is a tuple *TSN* (TS, SV) that can optionally use shared variables *SV* to model conditional (process-related) switching behavior:

- TS is a set of timed systems
  $\mathrm{TS} = (\mathrm{M}, \mathrm{Act}, \mathrm{m}_0, \mathrm{C}, \mathrm{Inv}, \mathrm{E})$, where
  $\mathrm{E} \subseteq \mathrm{M} \times \mathrm{Act} \times \mathrm{G(C)} \times 2^{\mathrm{C}} \times \mathrm{U(C)} \times 2^{A(SV)} \times 2^{G(SV)} \times \mathrm{M}$
- SV $\in \mathbb{N}$ is a set of shared and finite integer variables
- A(SV) is a set of assignments to shared variables:
  A: SV $\rightarrow \mathbb{N}$
- G(SV) is a set of guards on shared variables in the form of g := v $\sim$ c with $\sim \in \{=, \neq\}$

Use case reference
The plant consists of nine subsystems each controlled by a process controller (PLC) that communicate via Ethernet (ring topology) as shown in Fig. 5. The overall automation task is executed cooperatively by the subsystems. Because of this modularity, process-related dependencies arise and affect the switching behavior of single subsystems which is illustrated as bidirectional arrows between subsystems in Fig. 5. Process-related dependencies limit the degree of freedom relating to its switching behavior.

In Fig. 6 two shared integer variables are introduced: $\mathrm{sv_f}$ and sv_{t}. At each transition, these variables are updated and can be referenced by other subsystems. In this way, transitions of a timed subsystem can be guarded referring to a shared variable. Referencing shared variables ensures a specific switching order to be followed because of safety, hardware-specific or process-related reasons. $\mathrm{sv_t}$ and $\mathrm{sv_f}$ represent the current mode of the timed subsystems *Filling* and *Transportation* respectively and can be referenced in logical constraints. For instance, the guard $\mathrm{sv_t} == 2$ at transition from m_4 to m_5 in subsystem *Filling* expresses the following fact. This transition is enabled if and only if the shared variable $\mathrm{sv_t}$ is equal to value 2 indicating the subsystem *Transportation* being in mode m_2.

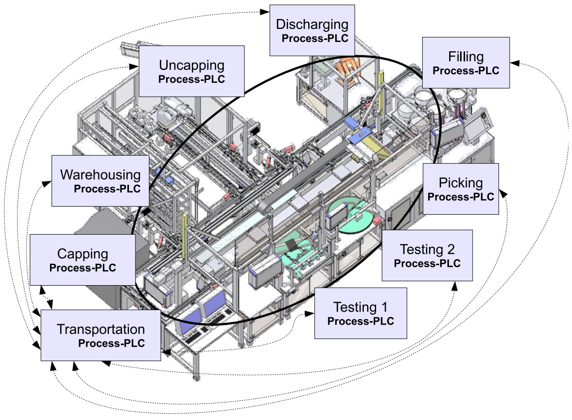

Figure 5. Process-related dependencies of the plant

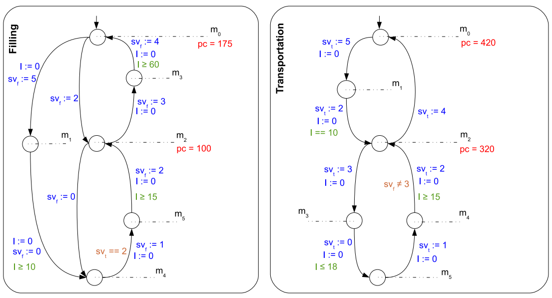

Figure 6. Networked subsystems *Filling* **and** *Transportation*

3. Calculating alternative energetic strategies

The approach discussed in this section is motivated by following requirement and objective. Assuming a temporal interval (pause interval) in which the automation system is idling, we are interested in feasible switching strategies for each subsystem that minimizes energy input of the complete system. In this setting, the pause interval is regarded as deadline which has to be met. The complexity of the problem is increased by interrelated (process-related dependencies) subsystems so that switching can not be calculated and executed independently. A specific precedence order has to be respected in order to execute safe switching operations avoiding failure situations. The temporal aspect of time consuming transitional modes has to be taken into account while calculating switching strategies.
For this problem, the energy system model of Section 2 provides means to describe all necessary information and constraints. This section focuses on the calculation of subsystems' feasible, energetic strategies meeting temporal and process-related constraints. The approach is split up into two steps. The first step (Subsection 3.1) covers the calculation of feasible strategies for each subsystem independently. This results in a set of sequences basically executable and constricts the state space effectively. The second step (Subsection 3.2) deals with the optimization problem of calculating energy minimum switching strategies for the automation system globally. With this two-step approach, only feasible switching sequences avoiding unnecessary switching are part of the optimization step and are considered further.

3.1. Symbolic reachability analysis

The reachability calculation is based on a zone abstraction for representing temporal constraints of switching behavior. Symbolic reachability is carried out by the use of a data structure based on difference bound matrices as well as operations on zones. This is motivated by the fact that temporal constraints need to be considered while calculating valid switching alternatives. In this step, we are interested in all executable switching strategies that meet a specific deadline having an initial as well as a target mode. In this context, all undesirable switching alternatives, especially switching cycles (constantly looping between operating modes), that are basically part of the solution, are eliminated, so that the symbolic state space is significantly reduced for calculating energy-optimal strategies in the next subsection. Calculating switching strategies can be based on a depth-first search in the state space of symbolic states resulting in all feasible switching sequences. At this point, we are just interested in calculating the sequence of modes for switching alternatives respecting temporal constraints. The exact temporal characteristic of such switching sequences is not important in this step as long as the modeled temporal constraints are met. So, the use case reference presents four basic switching alternatives.

Use case reference
For all further considerations, we assume ready-for-production for subsystem *Filling* and stand-by for subsystem *Transportation* as initial modes and require to return to these modes after 100 time units. There exist at most four different switching alternatives (in terms of mode changes) for subsystem *Filling* with adaptable temporal switching. Four example strategies are presented in Fig. 7 for subsystem *Filling*.

- $\mathrm{strategy}_1 := \mathrm{m}_0 \xrightarrow[175]{100} \mathrm{m}_0$ (energy input: 17.500)
- $\mathrm{strategy}_2 := \mathrm{m}_0 \rightarrow \mathrm{m}_2 \xrightarrow[100]{40} \mathrm{m}_2 \rightarrow \mathrm{m}_3 \xrightarrow[0]{60} \mathrm{m}_3 \rightarrow \mathrm{m}_0$ (energy input: 4.000)
- $\mathrm{strategy}_3 := \mathrm{m}_0 \rightarrow \mathrm{m}_2 \rightarrow \mathrm{m}_4 \xrightarrow[0]{18} \mathrm{m}_4 \rightarrow \mathrm{m}_5 \xrightarrow[0]{15} \mathrm{m}_5 \rightarrow \mathrm{m}_2 \xrightarrow[100]{5} \mathrm{m}_2 \rightarrow \mathrm{m}_3 \xrightarrow[0]{60} \mathrm{m}_3 \rightarrow \mathrm{m}_0 \xrightarrow[175]{2} \mathrm{m}_0$ (energy input: 850)
- $\mathrm{strategy}_4 := \mathrm{m}_0 \xrightarrow[175]{3} \mathrm{m}_0 \rightarrow \mathrm{m}_1 \xrightarrow[0]{10} \mathrm{m}_1 \rightarrow \mathrm{m}_4 \xrightarrow[0]{5}$

$m_4 \rightarrow m_5 \xrightarrow[0]{15} m_5 \rightarrow m_2 \xrightarrow[100]{5} m_2 \rightarrow m_3 \xrightarrow[0]{62} m_3 \rightarrow m_0$ (energy input: 1.025)

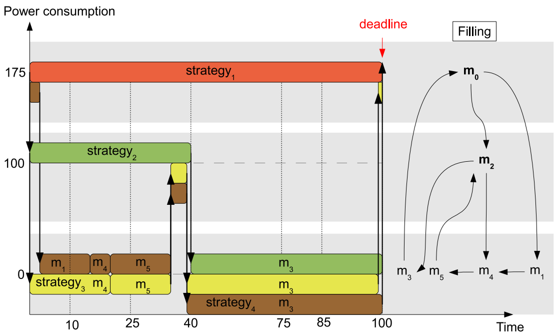

Figure 7. Alternative strategies of subsystem *Filling*

Three example strategies for subsystem *Transportation* are presented in Fig. 8.

- $\text{strategy}_1 := m_0 \xrightarrow[420]{100} m_0$ (energy input: 42.000)
- $\text{strategy}_2 := m_0 \rightarrow m_1 \xrightarrow[0]{10} m_1 \rightarrow m_2 \xrightarrow[320]{90} m_2 \rightarrow m_0$ (energy input: 28.800)
- $\text{strategy}_3 := m_0 \rightarrow m_1 \xrightarrow[0]{10} m_1 \rightarrow m_2 \xrightarrow[0]{5} m_2 \rightarrow m_3 \xrightarrow[0]{5} m_3 \rightarrow m_5 \xrightarrow[0]{65} m_5 \rightarrow m_4 \xrightarrow[0]{15} m_4 \rightarrow m_2 \rightarrow m_0$ (energy input: 0)

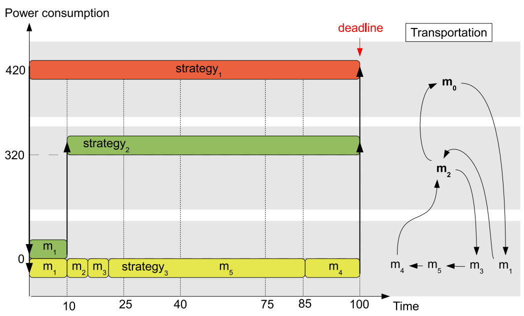

Figure 8. Alternative strategies of subsystem *Transportation*

The component-wise (per subsystem *comp*) search on which this approach is based has a complexity of $\mathcal{O}(\text{comp} \cdot (\text{tm}_{\text{average}})^{\frac{\text{depth}_{\text{average}}}{2}})$ with $\text{tm}_{\text{average}}$ as average number of transitional modes successive to operating modes and $\text{depth}_{\text{average}}$ as average number of modes in a switching strategy. This complexity can be compared to the search in a generated single transition system (composition of the subsystems' transition systems) describing the switching behavior of the automation system: $\mathcal{O}((\text{comp} \cdot \text{tm}_{\text{average}})^{\frac{\text{comp} \cdot \text{depth}_{\text{average}}}{2}})$. By using the component-wise calculation of this approach the state space explosion problem can reduced for this kind of problems. In a component-by-component calculation, we can benefit from the aspect of independent subsystems enabling independent reachability checks. This analysis can be seen as precalculating step.

3.2. Constrained optimization

For optimization purposes, we transform the switching strategies of individual subsystems (in automaton notion) into a constrained optimization problem (COP) of the interrelated subsystems. The decision variables in the COP are interval variables derived from the switching strategies of the subsystems. For each mode (operating mode respectively transitional mode) in the switching strategies of a subsystem, we define an interval variable $V_p^{S_i}$. Each interval variable has a starting time *start*($V_p^{S_i}$), an ending time *end*($V_p^{S_i}$) and a length *length*($V_p^{S_i}$) = *end*($V_p^{S_i}$) - *start*($V_p^{S_i}$), so that for its domain holds: *dom*($V_p^{S_i}$) = $\{[\text{start}(V_p^{S_i}), \text{end}(V_p^{S_i}))|\text{start}(V_p^{S_i}), \text{end}(V_p^{S_i}) \in \mathbb{Z}, \text{start}(V_p^{S_i}) \leq \text{end}(V_p^{S_i})\}$. The set of modes $M_p^{S_i}$ of a subsystem S_i is mapped to the set of interval variables $V_p^{S_i}$ by function I with $i, p \in \mathbb{N}$ denoting the number of subsystems respectively the p-th interval variable in the subsystem.

$$I : M_p^{S_i} \rightarrow V_p^{S_i} \tag{1}$$

There exist two kinds of constraints on the domain of interval variables. On the one hand, the domain is constrained because of guards on clocks (automaton). Guards *G(C)* on edges *E* in the automaton with G(C): $g ::= x \sim c$ and *M* as modes are transformed to a constraint by:

$$E \subseteq M_p \times G(C) \times M_{p+1} \rightarrow length(V_p^{S_i}) \sim c \tag{2}$$

On the other hand, the domain of interval variables is constrained due to the predefined global deadline *D* of the complete system. This is formulated as equality constraints h_i on the sum of interval variables for each sequence of a subsystem S_i, $i, p \in \mathbb{N}$:

$$h_i = \sum_p length(V_p^{S_i}) = D \tag{3}$$

The objective function is stated as follows with *length*($V_p^{S_i}$) denoting the interval length of an interval variable $V_p^{S_i}$ and $pc_p^{S_i}$ denoting the power consumption of the p-th mode in a subsystem:

$$min! \sum_{S_i} \sum_p pc_p^{S_i} \cdot length(V_p^{S_i}) \tag{4}$$

The switching behavior within a subsystem is expressed by precedence constraints *C*. The precedence of modes in automaton semantics is transformed into a precedence of interval variables within a subsystem. The description that $M_p^{S_i}$ precedes $M_{p+1}^{S_i}$ is transformed into a precedence constraint of interval variables:

$$\mathrm{end}(V_p^{S_i}) \rightarrow \mathrm{start}(V_{p+1}^{S_i}) \quad (5)$$

Furthermore, process-related dependencies (defined as logical conditions on transitions in automaton semantics) affect the order of switching operations. Guards g_{eq}, g_{neq} (Fig. 9, part (2a)) on shared variables sv^{S_i} defined in subsystem S_k are transformed into precedence constraints ($i \neq k$, p, q $\in \mathbb{N}$).

For $g_{eq} := sv^{S_i} = m_p^{S_i}$ holds (Fig. 9, part (1a, 1b)):

$$\mathrm{end}(V_q^{S_k}) \rightarrow \mathrm{end}(V_p^{S_i}) \wedge \mathrm{start}(V_p^{S_i}) \rightarrow \mathrm{end}(V_q^{S_k}) \quad (6)$$

For $g_{neq} := sv^{S_i} \neq m_p^{S_i}$ holds (Fig. 9, part (2a, 2b)):

$$\mathrm{end}(V_q^{S_k}) \rightarrow \mathrm{start}(V_p^{S_i}) \vee \mathrm{end}(V_p^{S_i}) \rightarrow \mathrm{start}(V_q^{S_k}) \quad (7)$$

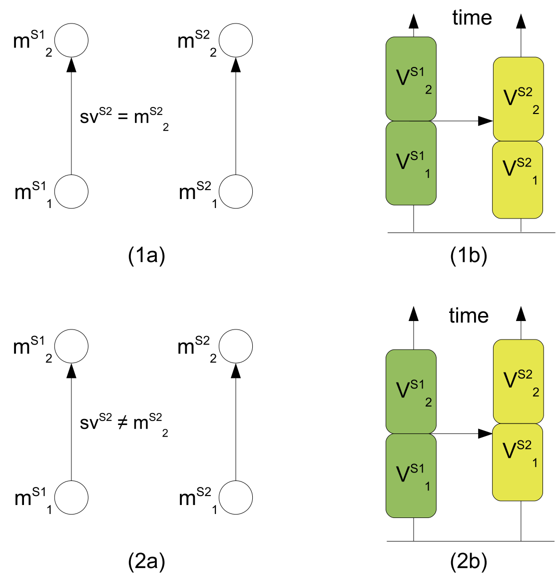

Figure 9. Guards on shared variables expressed in the semantics of intervals

3.3. Results and optimization complexity

In this section, the results based on the switching sequences of Fig. 7, Fig. 8 and of the requirement to return to the initial mode after 100 time units (deadline) are presented. Fig. 10 shows energy-optimal switching strategies meeting precedence, temporal and process-related constraints. The presented switching strategy for both subsystems is the best alternative in terms of energy demand and uses the commercially available ILOG CP Optimizer (version 12.4) for solving constrained optimization problems [1]. The time complexity for finding solutions for the constraint optimization problem formulated in ILOG CP Optimizer is given as $\mathcal{O}(n \cdot \log n)$ with n as number of activities modeled as interval variables [14].

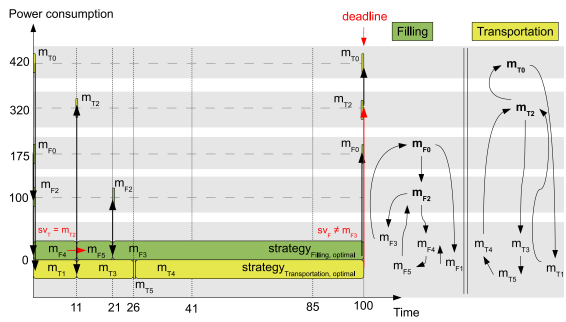

Figure 10. Energy-optimal schedule (operation) of subsystems *Filling* and *Transportation*

The energy-optimal switching strategies of subsystem *Filling* and subsystem *Transportation* are as follows:

- $\mathrm{strategy}_{\text{Filling, optimal}} := m_{F0} \rightarrow m_{F2} \rightarrow m_{F4} \xrightarrow[0]{11} m_{F4} \rightarrow m_{F5} \xrightarrow[0]{10} m_{F5} \rightarrow m_{F2} \rightarrow m_{F3} \xrightarrow[0]{79} m_{F3} \rightarrow m_{F0}$ (energy input: 0)
- $\mathrm{strategy}_{\text{Transportation, optimal}} := m_{T0} \rightarrow m_{T1} \xrightarrow[0]{11} m_{T1} \rightarrow m_{T2} \rightarrow m_{T3} \xrightarrow[0]{15} m_{T3} \rightarrow m_{T5} \rightarrow m_{T4} \xrightarrow[0]{74} m_{T4} \rightarrow m_{T0}$ (energy input: 0)

The overall energy input (sum of the energy input of optimal strategy of subsystem $Stra_{Fill}$ and energy input of optimal strategy of subsystem $Stra_{Trans}$) is equal to zero: $\mathrm{energy}(\mathrm{overall}) = \mathrm{energy}(\mathrm{Stra}_{\mathrm{Fill}}) + \mathrm{energy}(\mathrm{Stra}_{\mathrm{Trans}}) = 0$

4. Related work

Although energy-efficiency is an emerging topic in industrial factory automation, research in the context of control concepts and models for energy-efficiency is very few. A first approach which uses the communication infrastructure of an automation system to control the on and off operation of single components is proposed in [10]. However, the focus on individual components is its major drawback. It lacks of a descriptive model of a complete automation system and does not provide data structures for optimization purposes. Propositions regarding energy management systems (EMS) [11] in factory automation reside on the process control level (manufacturing execution level) and do not take into account all specific conditions and constraints on control level as well as on field level. EMS are relevant for energy data aggregation, visualization and user-dependent analysis. The presented

approach contributes to closing the gap between a details-abstracting energy management and the requirements on control level in factory automation.
Parallels between the calculation of the operating duration of modes as well as the switch between modes can be detected to the scheduling of tasks on resources. Durations of operating modes can not be arbitrarily assigned to subsystems like tasks can be assigned to (redundant) machines. [12] evaluates minimum-cost reachability analysis in the context of energy-optimal task graph scheduling. There, the precedence constraints are formulated as task graphs. Tasks in task automata must be expressed with predefined time bounds. The problem described here can not be adequately mapped to that formalism. Furthermore, the focus lies on energy-optimality for embedded systems and the specific requirements of automation systems are not addressed. An approach which addresses the power management in embedded devices is [13]. The so called dynamic power management focuses on optimization in single devices, but does not address the system view. In [7] the authors describe the energy consumption and generation using the semantics of priced/weighted timed automata. They ask the theoretical question if the accumulated weight (e. g. energy input) for any finite prefix satisfies certain constraints. In [2], the integration of timed automata models and scheduling is discussed. Although the integration is a parallel to this approach, the focus of that work is on job-shop problems and uncertainty in task durations. The scheduling of jobs respecting power consumption as well as timing of switching operations is discussed in [9]. In contrast to our approach, the focus lies on a single machine and no system view introducing process-related dependencies between machines is provided. We consider the system view on automation systems with their modular structures by proposing a formal, state-based framework for modeling temporal switching and energetic operating behavior as essential for optimization objectives.

5. Conclusion and future work

A two-step approach - symbolic reachability analysis followed by an optimization of switching strategies - is focus of this paper. The reachability analysis provides basically, feasible switching strategies and filters the space of solutions. Only feasible strategies are considered as relevant for the optimization step. The formulation in form of a constrained optimization problem allows finding the energy-optimal operation of modular automation systems. The approach provides a way to find the optimal switching and operating strategy minimizing the energy input of the automation system supported by a graphical representation of the switching behavior (energy system model).

Future work will focus on aspects exploiting the structural information between subsystems in an automation system to accelerate the optimization process. Even if an optimization, e. g. based on heuristics, does not find a global optimum, it is supposed to be much more efficient in terms of computational effort.

References

[1] ILOG CP Optimizer: http://www-01.ibm.com/software/integration/optimization/cplex-cp-optimizer/. 2012.

[2] Y. Abdeddaim, E. Asarin, and O. Maler. Scheduling with timed automata. *Tools and algorithms for the construction and analysis of systems*, pages 272–300, 2006.

[3] R. Alur and D. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[4] R. Alur, S. La Torre, and G. J. Pappas. Optimal paths in weighted timed automata. *Proc. 4th Int. Workshop Hybrid Systems: Computation and Control, LNCS 2034*, pages 49–62, 2001.

[5] G. Behrmann, A. Fehnker, T. Hune, K. Larsen, P. Pettersson, J. Romijn, and F. Vaandrager. Minimum-cost reachability for priced timed automata. *HSCC '01 Proceedings of the 4th International Workshop on Hybrid Systems: Computation and Control*, 2001.

[6] G. Behrmann, K. G. Larsen, and J. I. Rasmussen. Priced timed automata: Algorithms and applications. *International Symposium Formal Methods for Components and Objects (FMCO)*, pages 162–182, 2006.

[7] P. Bouyer, U. Fahrenberg, K. G. Larsen, N. Markey, and J. Srba. Infinite runs in weighted timed automata with energy constraints. *Proceedings of the 6th international conference on Formal Modeling and Analysis of Timed Systems*, pages 33–47, 2008.

[8] S. Mechs, J. P. Müller, S. Lamparter, and J. Peschke. Networked priced timed automata for energy-efficient factory automation. *IEEE Transactions on Automatic Control, accepted*, 2012.

[9] G. Mouzon and M. B. Yildirim. A framework to minimise total energy consumption and total tardiness on a single machine. *International Journal of Sustainable Engineering*, pages 105–116, 2008.

[10] N.N. The PROFIenergy PROFILE . *http://www.profibus.com/nc/downloads/downloads/pi-white-paper-the-profienergy-profile/display/, URL checked: 2012-04-20*, 2010.

[11] N.N. ISO 50001:2011, Energy Management Systems - Requirements with guidance for use. 2011.

[12] J. Rasmussen, K. G. Larsen, and K. Subramani. Resource-optimal scheduling using priced timed automata. *Proc. 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 220–235, 2004.

[13] D. Shorin and A. Zimmermann. Model-based development of energy-efficient automation systems. *The 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2011.

[14] P. Vilim. Max energy filtering algorithm for discrete cumulative resources. *Proceedings of the 6th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages vol. 5547 LNCS, 294–308, 2009.