

MODELLING AND EVALUATION OF DESYNCHRONIZATION STRATEGIES FOR CONTROLLABLE COOLING DEVICES

C. Hinrichs¹, U. Vogel¹, M. Sonnenschein¹
¹University of Oldenburg, Germany

Corresponding author: C. Hinrichs, University of Oldenburg, Dept. of Computing Science
D-26111 Oldenburg, Germany; christian.hinrichs@uni-oldenburg.de

Abstract. Power grid stability is currently maintained by the grid operators through the use of stand-by generators. Another approach to this task is demand side management. There, devices with load shedding capabilities are used to support balancing the grid. This paper addresses the use of refrigerators in private households as controllable thermal storage. Recent simulations have demonstrated that controlling a great number of cooling devices can be used for short time reserves. However, the control signals lead to a synchronization of the simulated devices, which in turn was followed by a periodic oscillation of the overall load of the population. In this paper we discuss and compare two strategies to desynchronise a population of cooling devices in order to damp the oscillation. For this objective, we extend the simulation model by algorithms to restore a particular distribution of fridge states after the impacts of a control signal. We will show that both signals lead to a complete desynchronisation of the devices, but differ in their implementation and operation complexity.

1 Introduction

Due to the time-varying availability of energy from renewable resources like sunlight or wind and their increasing use, the development of new load balancing mechanisms becomes a major task of grid operators. The storage of large quantities of electrical energy is very difficult. So "spinning reserve" generators, which are inefficient and expensive, are used to ensure stability of the grid frequency. An approach to reduce the use of such generators can be found in demand side management [3], where load balancing occurs on customer side by the use of intelligent devices. Possible types of such devices differ in their complexity, requirements for operation and location of control:

1. Stress in the grid can be sensed by monitoring the AC frequency. So a device may react on frequency changes by adapting its power consumption to reduce the stress. Such devices act autonomously and no further communication infrastructure is required. Neither the grid operator nor the consumer have the ability to control these devices.
2. A more complex approach is the use of dynamic price information from the utility in order to schedule devices. Here the grid operator will have to predict the working load in the grid over a period of time. According to this prediction, dynamic price information will be generated and announced. Consumers may now schedule their devices to operate in time intervals of low cost and low stress in the grid. This approach requires a communication infrastructure between utilities and households, and some kind of a set-top box plus communication interface between the box and the devices for automating the scheduling. The ability to control devices is shared between the grid operator (who may manipulate the price information to his needs) and the consumers (who can decide how to react to the price information).
3. A third approach moves the ability to control devices completely to the grid operator by using control signals. Such signals require a communication infrastructure between utilities and devices. Here the grid operators may send control signals in order to influence the devices' behaviours.

Here, we follow the third approach. We analyze the collective behaviour of a great number of devices after processing a control signal and investigate some of the emerging problems.

Related work and arising problems. Hammerstrom et al. [4] explored a controller which deactivates appliances, mainly water heaters and clothes dryers, whenever the grid frequency falls below 59.95 Hz. The authors recognized that a broad variety of geographically spreaded appliances is needful to achieve the desired effects, and that mechanisms have to be included which "re-create load diversity in the populations of appliances" after an occurred event. Short et al. [7] investigated the effects of dynamically adjusting the allowed temperature range of refrigerators. The adjustment was performed autonomously by the devices as a response to a frequency-fall caused by "sudden increase in demand (or loss of generation)". Eto et al. [2] demonstrated the use of customer's air-conditioning units as spinning reserve by demand response mechanisms. Finally, Stadler et al. [8] addressed the use of refrigerators in private households as controllable thermal storage. The authors explored different control signals for a simulation model of a great number of cooling devices. They concluded, that control signals to these devices can be used for short time reserves. However, like in some of the above publications, the signals lead to a synchronization of the simulated devices, which in turn was followed by a periodic oscillation of the overall load of the population.

Objective. Based on the simulation model by Stadler et al. [8] and the findings from Hinrichs in the Master's thesis [6], we discuss and compare in this paper two strategies to desynchronise a population of cooling devices in order to damp an oscillation resulting from a control signal to the device. For this objective, we extend the simulation model by algorithms to restore a particular distribution of fridge states after the impacts of a control signal. We will show that both algorithms lead to a complete desynchronisation of the devices, but differ in their implementation and operation complexity.

2 Prerequisites

First, we will describe the underlying model of Stadler et al. This model is based on an equation originally assembled by Constantopoulos et al. in [1]. It calculates the inner temperature of a cooling device at a particular point in time t_i , using the temperature T_{i-1} of the previous time step:

$$T_i = \varepsilon \cdot T_{i-1} + (1 - \varepsilon) \cdot \left(T^O - \eta \cdot \frac{q_{i-1}}{A} \right) \quad \text{where} \quad \varepsilon = e^{-\frac{\tau}{60} \cdot \frac{A}{m_c}} \quad (1)$$

In this equation, T^O describes the (constant) surrounding temperature. ε defines the system inertia, which depends on the insulation A , thermal mass m_c and the parameter τ , which specifies the amount of time between t_{i-1} and t_i in minutes. The coefficient η describes the efficiency of the device, and q_{i-1} depicts the electrical load (in watts) in the timespan $[t_{i-1}, t_i]$. The electrical load of the cooling aggregate has the value $q_{cooling}$ if it is active, and else $q_{warming}$. The other parameters are assumed to be constant during the simulation. Therefore, the state of a device can be defined as the tuple (temperature, phase). Phase defines whether the device currently operates in cooling ($q_i = q_{cooling}$) or warming ($q_i = q_{warming}$) mode. In the simulation model from Stadler et al., a device's temperature oscillates in the temperature range $[T_{min}, T_{max}]$. The corresponding load curve will be a rectangular wave with lower and upper bounds $q_{cooling}$ and $q_{warming}$. In a large population of devices with uniformly distributed states, the sum of all load curves will produce a nearly constant overall energy demand q_{all} over time.

By evaluating formula (1), the duration of the warming phase (with $q_i = q_{warming} = 0$) and of the cooling phase can be calculated as $\tau_{warming}$ and $\tau_{cooling}$. So, an undisturbed device repeats its behaviour in a cycle of length $\tau_{cycle} = \tau_{warming} + \tau_{cooling}$. As the temperature progress in the temperature range is nearly linear with an coefficient of determination greater than 0.9998 and a standard error less than 0.007, we assume initially the linear model for the temperature development in order to keep the necessary device controller simple:

$$T(t) = T_i + a \cdot t \quad \text{with} \quad \begin{cases} a = a_{cooling} = \frac{(T_{min} - T_{max})}{\tau_{cooling}} < 0, & \text{if the device has mode cooling,} \\ a = a_{warming} = \frac{(T_{max} - T_{min})}{\tau_{warming}} > 0, & \text{if the device has mode warming.} \end{cases} \quad (2)$$

3 Desynchronization

Stadler et al. explored two types of control signals, *Direct Storage Control (DSC)* and *Timed Load Reduction (TLR)*. The DSC signal leads to an immediate warmup or alternatively cooldown of all devices. The TLR signal allows more detailed control over the device by parameters t_{activ} and τ_{reduce} . Let t_{notify} be the point in time at which the signal was sent. Then the parameter t_{activ} defines the point in time, at which the load reduction should occur. Hence, each device can use the lead time $\tau_{preload} = t_{activ} - t_{notify}$ to fill its thermal storage. A second parameter τ_{reduce} specifies the requested length of the load reduction, which can be used to influence the resulting load curve further on. Stadler shows that such control signals can be used to affect the overall energy demand. Both of these signal types are disadvantageous by inducing a synchronization of the population, because they generate synchronous phase shifts in the devices. If we assume a uniform distribution of the devices' states in advance to a control signal, each control signal leads to an accumulation of the devices' states in a smaller state space: So, if for example a TLR signal is sent at time t_{notify} ordering the devices to reduce their energy demand at time t_{activ} for a duration of length τ_{reduce} , all devices react by filling their thermal storage in the time span $[t_{notify}, t_{activ}]$. So, nearly all devices will be in mode cooling. At time t_{activ} , the mode of all devices is set to "off", that means all devices are in mode warming. Hence, the devices' temperatures are not uniformly distributed in the temperature range anymore but accumulate at lower temperatures. Although the heterogeneity of the single devices' parameters lead to individual lengths of cooling and warming phases, and therefore to a reduction of the resulting oscillation, this oscillation tends to be long lasting: devices with the same parameters which are in the same state after a control signal are diversified if and only if some individual random influences effect their parameters. So, the distribution of the devices' states would be different, when a second control signal is sent, and hence, the system would react differently.

Thus, a strategy for reconstructing the previous distribution of states seems to be essential for an improved load management. In this paper, we will present an approach to desynchronizing a population of cooling devices based on the distribution of the devices' states. The following sections will first describe the used simulation environment and then present two strategies to achieve a state distribution equivalent to the situation before an occurred control signal.

3.1 Simulation environment

In a first step, we implemented an individual-based, discrete-event simulation environment which is able to simulate the linear model described above. The extendable design of this environment allows the development of any modification to the control signals proposed by Stadler et al. To be able to evaluate the simulation results, we performed a statistical analysis of preliminary simulation output. We identified two degrees of freedom: (1) population size, and (2) number of simulation runs. These characteristics were investigated by confidence interval estimations with the goal to allow a maximum deviation of 1% of the estimated mean load throughout the simulation. The confidence level was set to 99%. Detailed calculations can be found in [5], the results were as follows:

Population size. Stadler et al. used a population size of 5000 devices. Our investigation showed that this value is sufficient: calculating a 99% confidence interval for the mean load resulted in a half-width $h_0 \approx 0.016W$, which describes a maximal deviation of 0.105% from the mean load given by the simulation.

Number of simulation runs. Because the simulations start with random device parameters, each simulation run produces slightly different results. To obtain reliable simulation output, each simulation has to be performed several times. Afterwards, a mean of the results of the different simulation runs has to be used for further examination. Stadler et al. used 100 repetitions for each experiment. Our investigation showed that with 5000 devices, five already repetitions satisfy the aimed maximum deviation of 1%. The resulting half-width after five repetitions is $h_0 \approx 0.042W$, which describes a maximal deviation of 0.273%.

Therefore, the following simulations have been performed with a population size of 5000 devices and five repetitions for each experiment.

In a second step, we converted the developed simulation models (see the following sections) to statecharts according to the definition of D. Harel (1987, [5]). These statecharts allowed us to survey the models on a hardware abstraction level which would allow a straightforward realization in controller hardware.

The base model by Constantopoulos et al. (see section 2) requires knowledge about specific device parameters. The linear model in contrast needs only T_{min} , T_{max} , $\tau_{cooling}$ and $\tau_{warming}$ for operation, which can easily be captured by a temperature sensor and a real time clock. To keep the controller logic simple and to avoid complex additional sensors, the following damping strategies use the linear model for their calculations. The simulations in step one use the linear model as well to analyze the essential behaviour of the strategies. The statecharts (step two) were then simulated using the base model in order to validate the outcome of the linear calculations in a non-linear context and to predict the controller's behaviour in a hardware environment.

3.2 State recovery

As mentioned before, the temperature trajectory of a modelled cooling device oscillates between temperature borders T_{min} and T_{max} . The device nearly linearly warms up to the maximum temperature, before cooling down to the minimum temperature and so on. It approximately describes an irregular periodic triangle wave with an amplitude of $T_{range} = (T_{max} - T_{min})/2$.

Let now $t_{restore}$ be a point in time at which the former state distribution should be restored after a control signal has been sent. Given a device's state s_{notify} previous to the modification in t_{notify} , the virtual, i.e. unmodified temperature progress, as well as the actual, modified temperature progress of the device can be calculated. Let t_{cross} be the point in time at which the actual temperature trajectory crosses the virtual progress for the first time. At this time, two device states exist: the actual device state s_{cross} and for the virtual device state s'_{cross} . These states only differ in their phases. By programming the actual device controller to simply switch the phase in t_{cross} , it will subsequently follow the original temperature progress. This procedure is illustrated in figure (1).

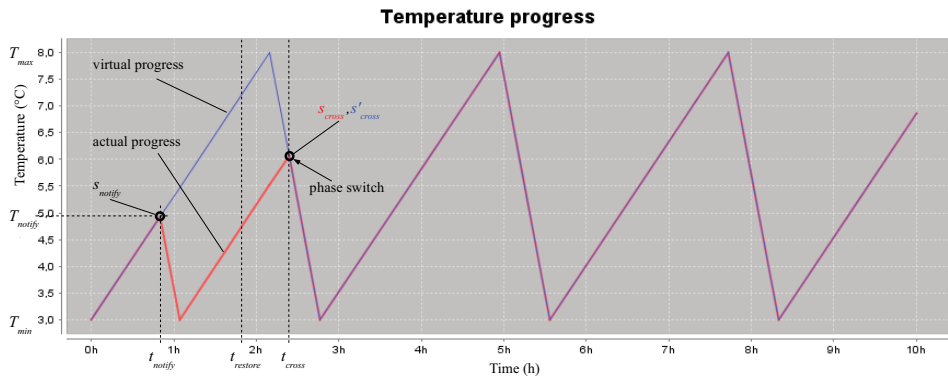


Figure 1: Temperature progress with state recovery – the device controller calculates the point in time t_{cross} , at which the actual temperature trajectory crosses the virtual progress for the first time, and then switches the phase.

So, given that a device controller knows its specific time spans $\tau_{warming}$ and $\tau_{cooling}$, it computes the first time of intersection between its original and its new trajectory simply by $t_{cross} = (T_{max} - T_{notify}) / a_{warming}$, when switching its mode at time t_{notify} . For a timed load reduction signal, this time might be during the requested reduce-intervall. As another intersection occurs at $t_{notify} + \tau_{cycle}$, and generally at $(T_{max} - T_{notify}) / a_{warming} + k \cdot \tau_{cycle}$ and at $t_{notify} + k \cdot \tau_{cycle}$, the controller simply chooses the minimal point in time which is larger as time $t_{restore}$. So, it can be guaranteed, that after time $t_{restore} + \tau_{cycle}$ with τ_{cycle} the maximum cycle time of all devices, all devices have reached their former trajectory, and hence rebuilt the former distribution of states. Figure (2) shows the results of a simulation of 5000 devices with TLR control signal with and without state recovery.

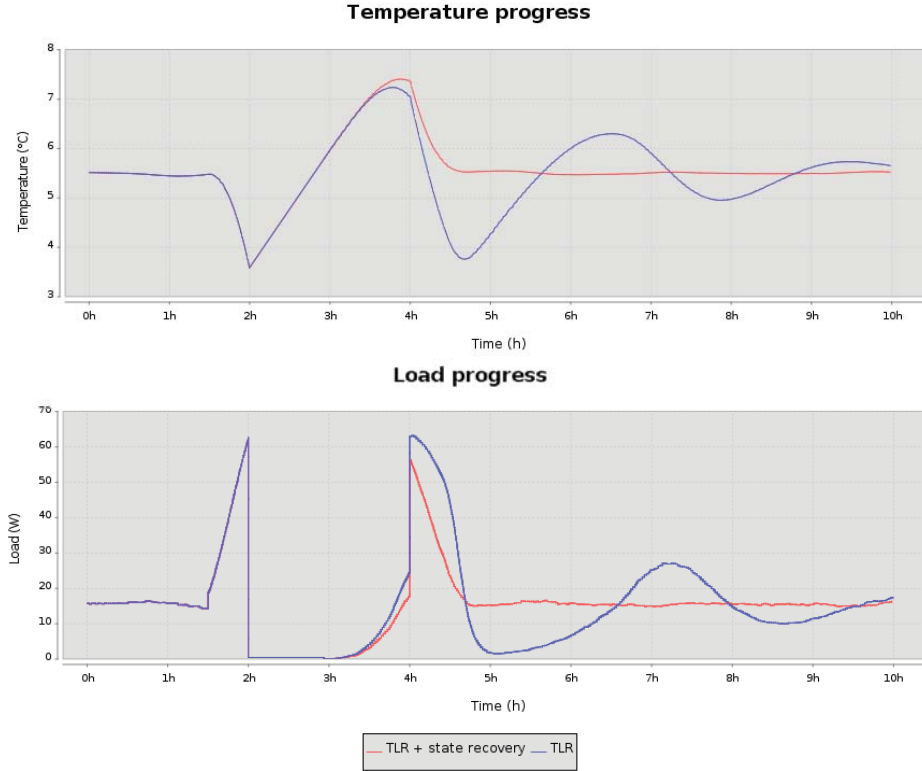


Figure 2: Simulation results which demonstrate the state recovery strategy – the TLR parameters were $t_{notify} = 1\text{h}30\text{m}$, $t_{activ} = 2\text{h}00\text{m}$ and $\tau_{reduce} = 2\text{h}00\text{m}$.

The blue trajectory (TLR) shows the mean progress of devices without state recovery while the red trajectory (TLR + state recovery) shows the progress with state recovery enabled. The figure includes both the temperature and load progress. The simulation has been run with heterogeneous devices (independently distributed device parameters). It is clearly to see that the state recovery performs well, and that the oscillation could easily be damped. The differing behaviour of the devices with state recovery in the interval $[3\text{h}15\text{m}, 4\text{h}00\text{m}[$ (thus *inside of* the reduction interval) is a side effect of the strategy implementation: normally, the devices should start with the state recovery immediately after the reduction interval. The current implementation, however, lets devices start the recovery as soon as they have to restart cooling due to T_{max} , even if they are at that point still inside of the reduction interval. As a result, this behaviour enhances the overall load curve by reducing the rise of the load just before the end of the reduction interval.

However, a disadvantage of this strategy is its sensitivity to deviations in the computed time spans, due to the linearization of the underlying model. By simulating this strategy on the statechart abstraction level (see section 3.1), we find that the strategy underestimates the duration for warming up and produces significant errors in calculating t_{cross} . In a great number of devices, this error results in a slight remaining oscillation of the overall load with a starting amplitude of $q_{osc} = 3\text{W}$. Therefore, we investigated another strategy that doesn't rely as heavily on the computed time spans.

3.3 Randomization

Another possibility to spread device states after a sent control signal randomizes the states explicitly after the control signal. By analyzing the state distribution which holds before a control signal, it is possible to produce a similar distribution afterwards by spreading phase shift temperatures. In a large population of heterogeneous devices, the temperatures of devices can be assumed to be uniformly distributed in the interval $[T_{min}, T_{max}]$, before the first control signal is received. In the randomization strategy, each device chooses a uniformly distributed random

temperature $T_{rnd} \in [T_{min}, T_{max}]$, when receiving a control signal. When the device reaches this temperature after the reaction to a control signal has been completed, its controller immediately switches its operating phase. This action is illustrated in figure (3).

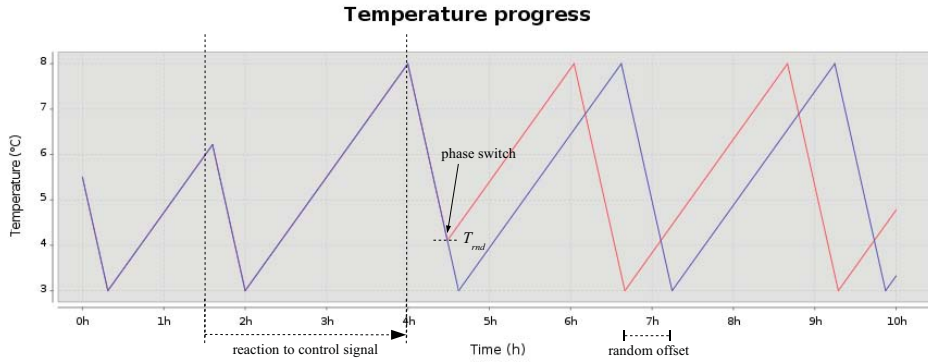


Figure 3: Temperature progress with randomization – the device controller chooses a random temperature T_{rnd} and switches the phase when the device reaches this temperature after the reaction to the TLR signal, resulting in a random offset.

The figure shows a situation where a TLR control signal has been sent at $t_{notify} = 1h30m$ with $t_{activ} = 2h00m$ and $\tau_{reduce} = 2h00m$. The reaction to the signal is completed in $t = 4h00m$. The controller chooses $T_{rnd} \approx 4.1^\circ C$ and switches the phase as soon as the device reaches this temperature. Thus the following phase shift points are moved by an offset which depends on the chosen random temperature. Applied to a great number of devices, this action leads to a uniformly distributed set of phase shift points. Figure (4) shows the results of a simulation of 5000 devices with TLR control signal with randomization.

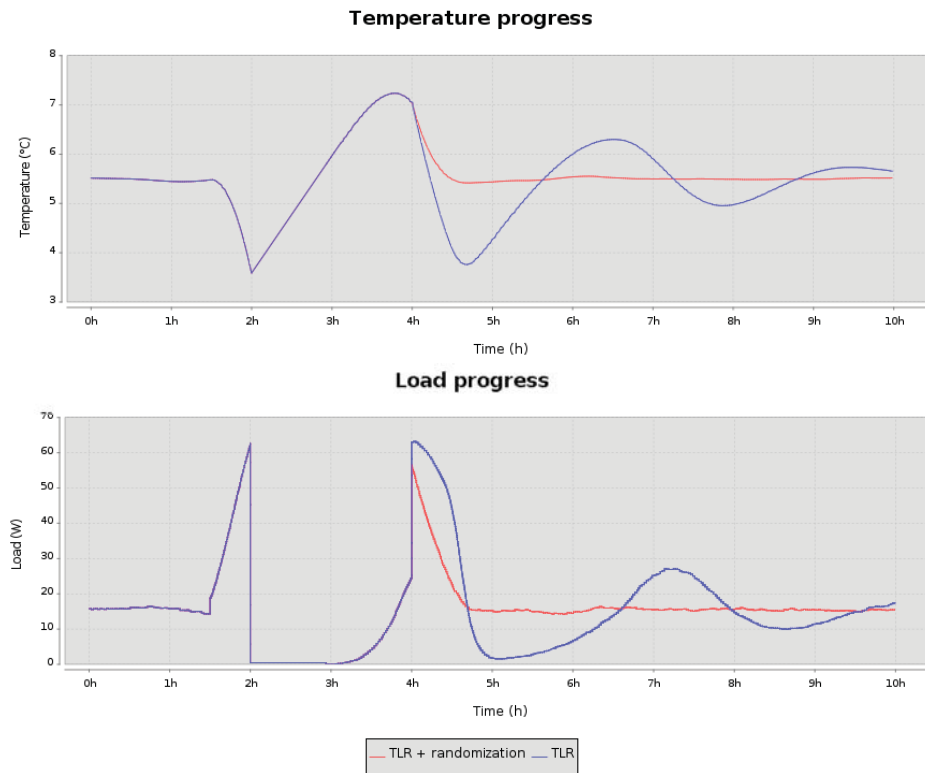


Figure 4: Simulation results which demonstrate the randomization strategy – the TLR parameters were again $t_{notify} = 1h30m$, $t_{activ} = 2h00m$ and $\tau_{reduce} = 2h00m$.

The blue trajectory shows the mean progress of devices without damping while the red trajectory shows the progress with randomization enabled. It is again clearly to see that the strategy performs well with the linear model. However, simulations on the statechart abstraction level show that there are still some remaining oscillations of the overall load. Figure (5) contrasts a simulation using the linear model (blue) with a simulation on the statechart abstraction level (red, labeled with "base model"). The remaining oscillation after the desynchronization in the base model simulation is still visible. But it is far less distinctive ($\pm 3W$) than the oscillation of a

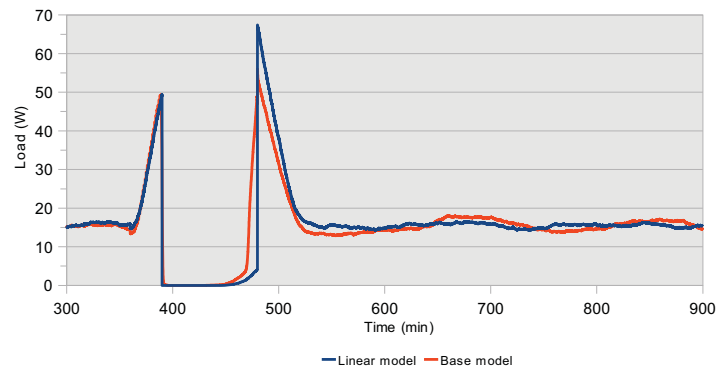


Figure 5: Comparison of simulations of the randomization strategy based on different models – the “base model” simulation represents the statechart abstraction level which still contains a slight oscillation after the desynchronization.

fully synchronized population ($\pm 13W$, e.g. figure (4)). So, the randomization strategy can still be classified as effective.

4 Conclusion

The presented strategies for desynchronizing an oscillating population of cooling devices differ in their method of operation and therefore in the requirements on an implementation. The state recovery strategy needs device controllers which are able to compute the intersection between the modified and the unmodified temperature trajectory. In order to do this, it has to know the basic device characteristics and it needs a certain amount of computing power plus some storage requirements. The randomization strategy in contrast needs a controller which possesses a random number generator and some storage as well. Compared to the requirements of the device controllers for the signals DSC and TLR introduced by Stadler et al., the two extensions proposed in this paper do not add any significant new requirements. Overall, the strategies provide an elegant way to damp an oscillating population of cooling devices. However, the deviations in the computed time spans due to the linearization of the underlying model give reason to optimize the strategies on a hardware abstraction level.

5 Current and future work

As mentioned above, the desynchronization strategies perform perfectly well in a context of a linear model. The performance on a hardware abstraction level is still acceptable but can surely be optimized. Current work covers the realization of the control signals DSC and TLR and the proposed strategies in this paper in a hardware controller which can then be examined in a field test.

Future work may include the development of additional, optimized control signals. Furthermore, grouping of cooling devices and sending cascades of control signals to these groups may be investigated. Thereby it should be possible to compose a desired overall load curve (e.g. to compensate small-scale oscillations or lows in the grid) without the drawbacks of unwanted peaks and other irregularities.

6 References

- [1] Constantopoulos, P. et al.: *ESTIA: a real-time consumer control scheme for space conditioning usage under spot electricity pricing*. Computers and Operations Research, 18(8), 1991.
- [2] Eto, J. et al.: *Demand Response Spinning Reserve Demonstration*. Energy Analysis Department, Ernest Orlando Lawrence Berkeley National Laboratory, LBNL-62761, 2007.
- [3] Gellings C., and Chamberlin, J.: *Demand-Side Management: Concepts and Methods*. The Fairmont Press, Inc., 1988.
- [4] Hammerstrom, D. et al.: *Pacific Northwest GridWise(tm) Testbed Demonstration Projects; Part II. Grid Friendly(tm) Appliance Project*. Pacific Northwest National Laboratory, PNNL-17079, 2007.
- [5] Harel, D.: *Statecharts: A visual formalism for complex systems*. Science of Computer Programming, 8(3), 1987, 231–274.
- [6] Hinrichs, C.: *Untersuchung und Entwicklung von Verfahren zur Desynchronisation adaptiver Kühlgeräte*. Master’s thesis, University of Oldenburg, Dept. of Computing Science, Germany, 2008.
- [7] Short, J. et al.: *Stabilization of Grid Frequency Through Dynamic Demand Control*. IEEE Transactions on Power Systems, 22(3), 2007.
- [8] Stadler, M. et al.: *The Adaptive Fridge – Comparing different control schemes for enhancing load shifting of electricity demand*. In: Proc. 21st Conference on Informatics for Environmental Protection – Enviroinfo Warsaw 2007, Shaker Publishing, Germany, 2007, 199–206.