

Validation, Synthesis and Optimization for Cyber-Physical Systems

Kim Guldstrand Larsen^(✉)

Department of Computer Science, Aalborg University,
Selma Lagerlöfs Vej 300, 9220 Aalborg East, Denmark
kgl@cs.aau.dk

Abstract. The growing complexity of Cyber-Physical Systems increasingly challenges existing methods and techniques. What is needed is a new generation of scalable tools for model-based learning, analysis, synthesis and optimization based on a mathematical sound foundation, that enables trade-offs between functional safety and quantitative performance. In paper we illustrate how recent branches of the UPPAAL tool suit are making an effort in this direction.

Cyber-Physical Systems

The term Cyber-Physical Systems (CPS) describes systems that combine computing elements with dedicated hardware and software having to monitor and control a particular physical environment. This combination of the physical with a virtual world provides the digital foundation for smart solutions throughout society and within all sectors. The constant demand for increased functionality and performance that needs to be produced with tight time schedules and cost budgets without compromising dependability of the final products constitutes a significant software engineering challenge.

What is needed are mathematically well-founded, scalable methods, tools and techniques that support the development of CPS. For this we have over more than 20 years pursued a model-based approach for the design of dependable and optimal CPS, supported by tools that are based on efficient algorithms and datastructures for analysis of semantically well-founded models. This has been the guiding principle behind the UPPAAL suite (www.uppaal.org) [54] which by now have been applied to a wide range of industrial applications from the domains of Embedded Systems and Cyber-Physical Systems.

The first version of the UPPAAL tool was presented at the very first TACAS conference in 1995 in Aarhus, Denmark. During the first several years the tool was developed in tight collaboration between Uppsala University, Sweden and Aalborg University, Denmark. Over the years a number branches has been developed, some of which will be described in the following sections.

This work is partly funded by the ERC Advanced Grant LASSO: Learning, Analysis, SynthesiS and Optimization of Cyber-Physical Systems as well as the Innovation Center DiCyPS: Data-Intensive Cyber Physical Systems.

1 The UPPAAL Tool Suite

UPPAAL. The underlying formalism of UPPAAL is that of timed automata with the tool providing support for model checking of hard real-time properties. Since the introduction of the tool in 1995, significant effort have been put into development and implementation of improved datastructures and algorithms for the analysis of timed automata. This includes guided search algorithms using heuristics from AI [6, 40, 46, 47], fully symbolic datastructures [9], minimal constraint normal forms [50], as well as a new symblistic DART datastructure [43, 45] making usefull tradeoffs between the effectiveness of discrete and symbolic semantics. Also, this research has included the development of a series of exact abstractions (or extrapolation) that not only ensures finiteness of the symbolic semantics, but also provide significant performance improvements [3, 4, 42]. Besides these advance with respect to the verification engine, significant effort has over the years been put on the graphical interface of the tool (e.g. [7]), and on the modelling side the introduction of user-defined, structured datatypes and procedures has undoubtedly made the tool significantly more usable in modeling real control programs and communication protocols [6].

UPPAAL CORA. Motivated by the need for addressing (optimal) usage of resource, extension of priced timed automata was introduced in 2001 with [2, 8] (independently) demonstrating decidability of cost-optimal reachability. Soon after efficient priced extension of the symbolic zone datastructures was implemented in the branch UPPAAL CORA, which combined with a symbolic A* algorithm providing a new generic tool for cost-optimal planning competetive to traditional OR methods such as Mixed-Integer Linear Programming [49]. Most recently new efficient extrapolation methods for priced timed automata has been introduced [17] and UPPAAL CORA has been used for the optimal planning of missions for battery-powered nano-satelittes [12].

UPPAAL TRON. In 2004 the branch UPPAAL TRON was introduced offering the possibility of performing on-line conformance testing of *real* real-time systems with respect to timed input-output automata [51, 56]. UPPAAL TRON implements a sound and (theoretically) complete randomized testing algorithm, and uses a formally defined notion of correctness to assign verdicts: i.e. relativized timed input/output conformance providing a timed extension of Jan Tretmans ioco [58]. Using online testing, events are generated and simultaneously executed on the system under test. UPPAAL TRON has been succesfully applied to a number of industrial case studies including an advanced electronic thermostat regulator sold world-wide in high volume by the Danish company Danfoss [52].

UPPAAL TIGA. In 2005 - encouraged by suggestions from Tom Henzinger – the branch UPPAAL TIGA was released, allowing for control strategies to be synthesized from timed games, i.e. two-player games played on a timed automata [5, 24]. The branch implements an efficient symbolic on-the-fly algorithm for synthesizing winning strategies for reachability, safety as well as Büchi objectives and taking possible partial observability into account [25]. The branch marks a disruptive direction with respect to development of control programs for embedded systems: rather than manually developing the control program with subsequent model

checking (and correction), UPPAAL TIGA provides a fully automatic method for deriving a correct-by-construction control program. In particular, this method allows for easy personalization of control program simply by modification of the objective. The branch has so far been industrially applied to the automatic synthesis of control strategies for zone-base climate control in pigsties [44] and safe and optimal operation of hydraulic pumps [26].

UPPAAL ECDAR. In 2010 the branch UPPAAL ECDAR was introduced supporting a scalable methodology for compositional development and stepwise refinement of real-time systems [36,38]. The underlying specification theory is that of timed I/O automata being essentially timed games (with inputs being controllable, and outputs being uncontrollable) equipped with suitable methods for refinement checking (in terms of an alternating simulation between two timed game specifications), consistency checking, logical as well as structural composition. The UPPAAL ECDAR branch uses heavily the UPPAAL TIGA engine to solve various games that arise in the computing the various composition operators and refinements. For a full account of UPPAAL ECDAR we refer the reader to the tutorial [35].

2 UPPAAL SMC

One of most recent branches of the UPPAAL tool suite – UPPAAL SMC introduced in 2011 – allows for performance evaluation the much richer formalisms of stochastic hybrid automata and games [33,34] and has by now been widely applied to analysis of a variety of case studies ranging from biological examples [32], schedulability for mixed-critical systems [13,37], evaluation of controllers for energy-aware buildings [28], social-technical attacks in security [39] as well as performance evaluation of a variety of wireless communication protocols [59]. Also the statistical model checking engine of UPPAAL SMC is supported by a distributed implementation [23], and allows for the statistical model checking of a large subset of MITL [21,22]. For a full account of UPPAAL SMC we refer the reader to the recent tutorial [31].

The modeling formalism of UPPAAL SMC is based on a stochastic interpretation and extension of the timed automata formalism used in the classical model checking version of UPPAAL. For individual components the stochastic interpretation replaces the nondeterministic choices between multiple enabled transitions by probabilistic choices (that may or may not be user-defined). Similarly, the non-deterministic choices of time-delays are refined by probability distributions, which at the component level are given either uniform distributions in cases with time-bounded delays or exponential distributions (with user-defined rates) in cases of unbounded delays.

To illustrate the features of UPPAAL SMC let us consider the example in Fig. 1, providing an “extended” timed automata based model of a car, that needs to make it from its initial position **Start** to the final position **End**. In the model the driver of the car twice needs to make a choice between using a high road (**H1** and **H2**) or a low road (**L1** and **L2**). The four roads differ in their travel-times

between 0 and 100 min (respective 0 and 50 min) as reflected by the invariants on the clock x). Also the roads differ in fuel-consumption reflected by the difference in the rate of the continuous variable fc (representing the total amount of fuel consumed). The model is in fact a priced timed automaton (as supported by the branch UPPAAL CORA) with the total time that it will take to make it to **End** ranging between 0 and 200, and total fuel-consumption ranging between 0 and 900. However, interpreted as a *stochastic* priced timed automaton, the discrete choice between the high and the low roads are made based on a (uniform) random choice. Similarly, the travel times of the 4 roads are resolved using uniform distributions of the respect travel-time intervals.

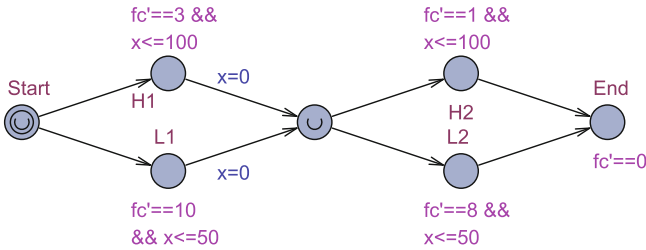


Fig. 1. The stochastic route model for a car

Now assume that we are interested in the expected fuel-consumption before reaching the goal **End**. Given the described stochastic semantics of the priced timed automaton in Fig. 1 this is easily seen to be the value of the following expression:

$$\left(0.5 \cdot \int_{t=0}^{100} 3t \cdot 0.01dt + 0.5 \cdot \int_{t=0}^{50} 10t \cdot 0.02dt \right) + \left(0.5 \cdot \int_{t=0}^{100} 1t \cdot 0.01dt + 0.5 \cdot \int_{t=0}^{50} 8t \cdot 0.02dt \right) = 325$$

For this model the above expression giving the desired expectation was particularly easy as the clock x is reset. In general – and formally – the stochastic semantics of a stochastic (priced) timed automata is given by a probability measure assigning probabilities to (certain) sets of runs, being countable unions or complements of so-called cylinder-sets, i.e. sets of runs that follow the same prefix of edges in the automaton. In general, the probability of such a cylinder will be a nested integral (the nesting depth being the length of the path in the automaton). When considering *networks* of stochastic timed automata, the probability measure will moreover reflect a repeated race between components (for who is to perform the next discrete action) of the networks. Decidability (and undecidability) results for the stochastic interpretation of timed automata

have so far – despite significant research – only given few conclusive results, e.g. that qualitative reachability (i.e. probability of reachability is 0 or 1) is decidable for one-clock stochastic timed automata [11, 14], or for acyclic models [55]. Instead, the statistical model checking engine of UPPAAL SMC resorts to simulation in order to settle a large range of quantitative questions, e.g. reachability probability or expectations. Being based on simulation, the results are however approximate (e.g. confidence intervals) but come with a statistically assured level of confidence. As an example, the UPPAAL SMC query

$$\text{Pr}[\leq 100] (\langle \rangle \text{Car.End})$$

will after some 7382 random runs of the model (made according to the stochastic semantic described) return the 95% confidence interval $[0.735636, 0.755635]$ as the probability that the location **End** is reached within 100 minutes. Addressing our original problem the query

$$E[\leq 200; 5000] (\text{max: fc})$$

will return the value 322.565 ± 4.70747 as an estimate of the expected fuel-consumption based on 5000 random runs of the model. In Fig. 2 we see the additional plots offered by UPPAAL SMC for the cumulative probability of the time for reaching **End** and the frequency count of the fuel-consumption over 5000 random runs.

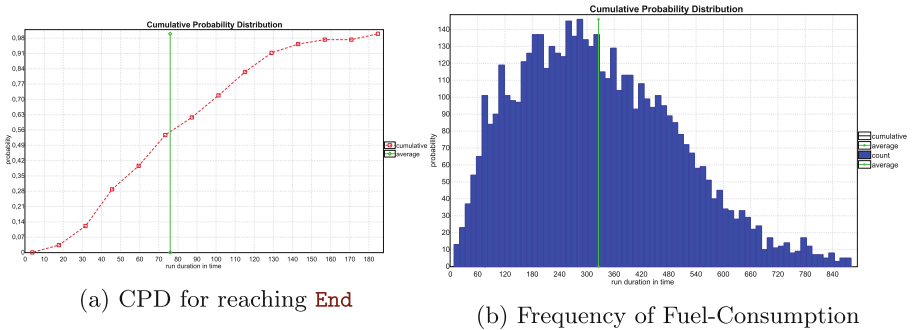


Fig. 2. Performance evaluation of stochastic car model.

3 UPPAAL Stratego

UPPAAL STRATEGO from 2014 [29, 30] is the most recent branch of the UPPAAL tool suite that allows to generate, optimize, compare and explore consequences and performance of strategies synthesized for stochastic priced timed games (SPTG) in a user-friendly manner. In particular, UPPAAL STRATEGO comes with an extended query language (see Table 1), where strategies are first class objects

Table 1. Various types of UPPAAL STRATEGO queries: “strategy S =” means strategy assignment and “under S” is strategy usage via strategy identifier S. Here the variables NS, DS and SS correspond to non-deterministic, deterministic and stochastic strategies respectively; bound is a bound expression on time or cost like $x \leq 100$ and n is the number of simulations.

| | |
|---|--|
| Strategy generators using [29]: | |
| Minimize objective: | strategy DS = minE (expr) [bound]: $\langle \rangle$ prop |
| Maximize objective: | strategy DS = maxE (expr) [bound]: $\langle \rangle$ prop under NS |
| Strategy generators using UPPAAL TIGA: | |
| Guarantee objective: | strategy NS = control: A $\langle \rangle$ prop |
| Guarantee objective: | strategy NS = control: A[] prop |
| Statistical model checking queries: | |
| Hypothesis testing: | Pr[bound]($\langle \rangle$ prop) ≥ 0.1 under SS |
| Evaluation: | Pr[bound]($\langle \rangle$ prop) under SS |
| Comparison: | Pr[bound]($\langle \rangle$ prop1) under SS1 \geq Pr[≤ 20]($\langle \rangle$ prop2) under SS2 |
| Expected value: | value E[bound;n](min: prop) under SS |
| Simulations | simulate n [bound] { expr1, expr2 } under SS |
| Symbolic model checking queries: | |
| Safety: | A[] prop under NS |
| Liveness: | A $\langle \rangle$ prop under NS |
| Infimum of value: | inf { condition } : expression |
| Supremum of value: | sup { condition } : expression |

that may be constructed, compared, optimized and used when performing (statistical) model checking of a game under the constraints of a given synthesized strategy. As such UPPAAL STRATEGO may be seen as a superset of UPPAAL TIGA and UPPAAL SMC.

To illustrate the features of UPPAAL STRATEGO, let us revise our running example of the car-route-problem as illustrated in in Fig. 3. Again there are four different roads with their individual required travel-times.

However, whereas the choice of road is up to the driver of the car to control (indicated by the solid transitions), the actual travel-time of the road is uncontrollable (indicated by the dashed transitions) reflecting the uncertainty of the amount of traffic on the particular day. In one scenario, the objective of the car it to choose the combination of roads that will ensure the shortest overall travel-time even in the most hostile traffic situation on the four roads. Under this interpretation, Fig. 3 represents a timed game. Clearly the strategy that would ensure the smallest worst-case travel-time is to take the two low roads, giving a guaranteed arrival time in 100 min. Taking the rates for the cost variable fc into account, makes Fig. 3 describe a priced time game, where the problem is to determine the best strategy in terms of minimizing the worst-case fuel-consumption. For our model this best strategy clearly consists in consistently choosing the

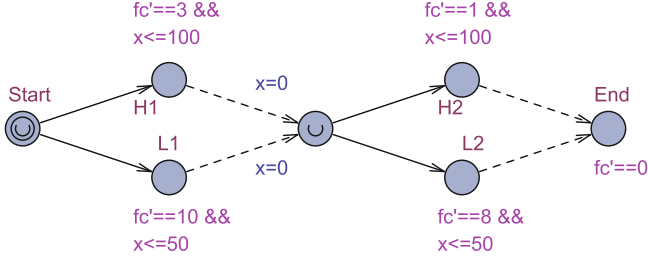


Fig. 3. The stochastic decision model for a car.

high roads. Unfortunately, cost-optimal winning strategies for priced games is undecidable in general when the underlying timed automata has three or more clocks [20]. Decidability results have been provided for one-clock priced timed games [19] and for so-called strongly cost-non-zero priced timed games [15, 16]; also approximate algorithms have been proposed [18].

However in UPPAAL STRATEGO, the model of Fig. 3 is interpreted as a *stochastic* priced timed game (SPTG), assuming that the travel-times of the four roads are chosen by uniform distributions, and the objective of the control strategy is to minimize the expected overall travel-time, or the expected overall fuel-consumption (e.g. the rate or fuel-consumption $fc'==3$ on the first high road **H1** indicates that the cost variable fc grows with rate 3 in this location).

We are interested in synthesizing strategies for various objectives. Being primarily concerned with fuel-consumption we may want to determine the strategy that will minimize the expected fuel-consumption. For our simple decision model Fig. 3 this is clearly given by the following expression:

$$\begin{aligned} & \min \left\{ \int_{t=0}^{100} 3t \cdot 0.01 dt, \int_{t=0}^{50} 10t \cdot 0.02 dt \right\} + \\ & \min \left\{ \int_{t=0}^{100} 1t \cdot 0.01 dt, \int_{t=0}^{50} 8t \cdot 0.02 dt \right\} = 200 \end{aligned}$$

However, posing the UPPAAL STRATEGO query

```
strategy Opt = minE (fc) [<=200] : <> Car.End
```

will provide (by reinforcement learning¹) the strategy **Opt**, that minimizes the expected total fuel-consumption, learning from runs which are maximally 200 time units long. The relativized query $E[<=200 ; 1000]$ ($\max: fc$) under **Opt**, generates 1000 runs of length 200 time units and then averages the maximum value of fc from each run. this is used to estimate the expected cost to be 200.39. Figure 4a summarizes 10 random runs according **Opt** illustrating fuel-consumption. None of the runs had a fuel consumption of 400 indicating that we

¹ The reinforcement learning uses machine learning techniques to learn strategies from sets of randomly generated runs. See [29] for more details.

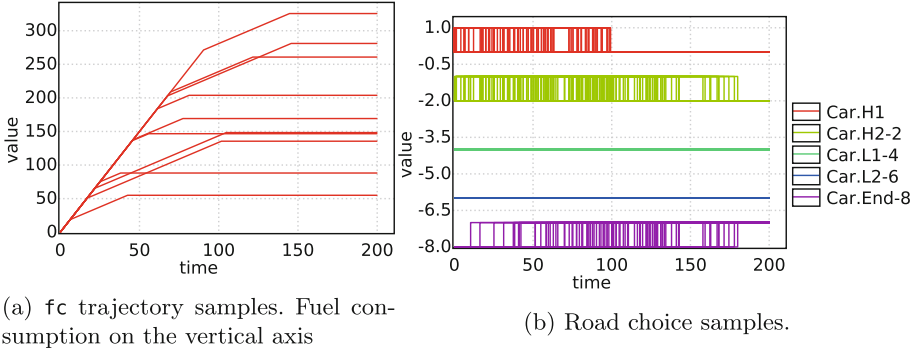


Fig. 4. Evaluation of strategy `Opt` via simulation.

always choose the energy-efficient roads. In Fig. 4b we see that this is actually the case as the simulations always choose to go to locations `H1` and `H2`, which models the energy-efficient roads.

Now, assume that the task *must* be completed before 150 time-units. From Fig. 4 it can be seen that the strategy `Opt` unfortunately does not guarantee this, as there are a few runs which exceeds 150 before reaching `End`. However, the query

```
strategy Safe = control: A<> Car.End and time<=150
```

will generate the most permissive (non-deterministic) strategy `Safe` that guarantees this bound but unfortunately with a high expected total fuel-consumption of 342.19. However, the relativized learning query

```
strategy OptSafe = minE (fc) [<=200] : <> Car.End under Safe
```

will provide a sub-strategy `OptSafe` that minimizes the expected total fuel-consumption – here found to be 279.87 – subject to the constraints of `Safe`. Figure 5 summarizes 10 random runs according to `SafeOpt`, indicating that only road `L1` is never chosen. Also, the failed model checking of `E<> Car.H2 and time>=51 and Car.x==0 under Safe` reveals that the high road `H2` may only be chosen in case the first phase is completed before 50 time-units, confirming the observations from the simulations.

For learning the strategy `OptSafe` the reinforcement learning method required 5 iterations each with 1000 runs. We illustrate in Fig. 6 the outcome of the runs in the last 3 iterations focusing on the choice, time and resulting fuel-consumption at the choice-point between `H2` and `L2`. As can be seen the method correctly learns to take the low road whenever the choice point is reached before a total time of 50 min (leaving enough time to guarantee that `End` will be reached within 150 min).

In general, as shown in the overview Fig. 7, UPPAAL STRATEGO will start from a SPTG \mathcal{P} . It can then abstract \mathcal{P} into a timed game (TGA) \mathcal{G} by simply ignoring

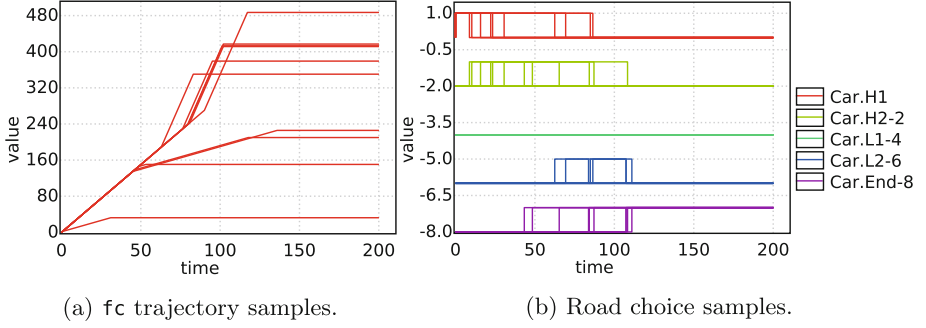


Fig. 5. Evaluation of strategy `OptSafe` via simulation.

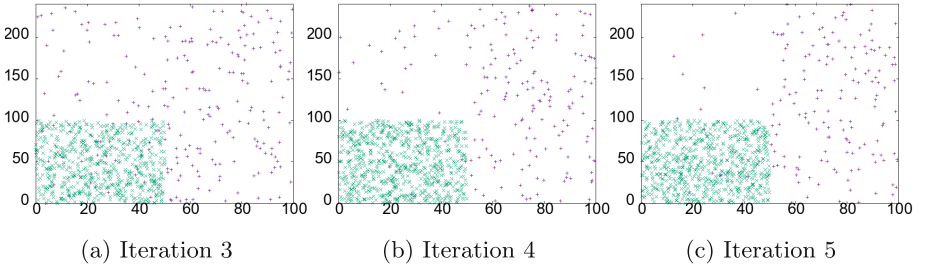


Fig. 6. Total fuel-consumption versus time at mid-location for runs 1000 runs choosing between `H2` and `L2` towards learning the strategy `OptSafe`.

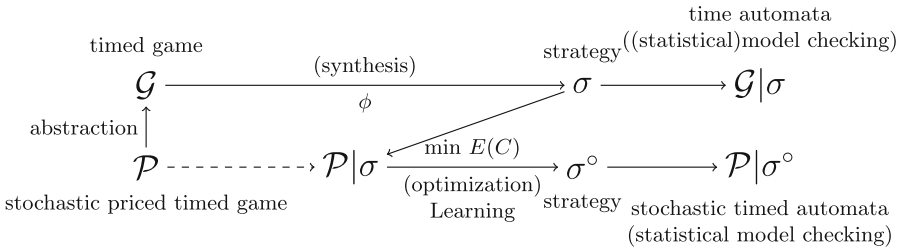


Fig. 7. Overview of UPPAAL STRATEGO

prices and stochasticity in the model. Using \mathcal{G} , UPPAAL TIGA [5] may now be used to (symbolically) synthesize a (most permissive) strategy σ meeting a required safety or (time-bounded) liveness constraint ϕ . The TGA \mathcal{G} under σ (denoted $\mathcal{G}|\sigma$) may now be subject to additional (statistical) model checking using classical UPPAAL [54] and UPPAAL SMC [31, 34]. Similarly, the original STGA \mathcal{P} under σ may be subject to statistical model checking. Now using reinforcement learning [29], we may synthesize near-optimal strategies that minimizes (maximizes) the expectation of a given cost-expression *cost*. In case the learning is performed from $\mathcal{P}|\sigma$, we obtain a sub-strategy σ^o of σ that optimizes the expected value of

cost subject to the hard constraints guaranteed by σ . Finally, given σ^o , one may perform additional statistical model checking of $\mathcal{P}|\sigma^o$.

4 Applications

The importance of CPS is clear within the domains of energy and transport with the emergence Smart Grid, Home Automation, Autonomous Driving, Advanced Driver Assistance and Intelligent Traffic Control where optimizing critical functionality is provided by intelligent and flexible software components. UPPAAL stratego has already been applied to a number of case studies including synthesis of a safe and optimal adaptive cruise control [53], synthesis of optimal floor heating system [48], and most recently synthesis of optimal control of traffic lights in intersections as described in the following sub-sections.

Adaptive Cruise Control. These days the Google Self-Driving car is about to become a reality: legislation has been passed in several U.S. states allowing driverless cars, in April 2014, Google announced that their vehicles had been logging nearly 1.1 million km, and it is forecast that Google’s self-driving cars will hit the roads this summer. Also, in Europe driverless cars have been actively pursued, both by the automotive industry itself and within a number of national and European research projects (e.g. FP7 and Horizon2020). With more and more traffic, European roads are becoming increasingly congested, polluted and unsafe. One potential solution to this growing problem is seen to be the use of small, automated, low-polluting vehicles for driverless transport in (and between) cities. Within the last decade, a number of European projects have been launched for making transport systems capable of fully automated driving, energy efficient and environmentally friendly while performing. In addition, many individual driving assistant systems based on suitable sensors have been developed for cars.

In [53], we have considered a small part of lane-change manoeuvres, namely the existence of a safe-distance controller (assumed in the above work of Olderog et al.). In particular, we demonstrated how UPPAAL STRATEGO may be applied to automatically obtain a safe yet optimal adaptive strategy *safe* for the cruise control. Modelling the cruise control as a game with a car in front a *safe* strategy was synthesized ensuring that the distance to the front care would never get below 5 meters. In fact utilizing the distinct feature of UPPAAL STRATEGO – allowing additional properties to be verified of a synthesized strategy – we may verify the smallest distance possible to the front care which will not violate the *safe* as shown in Fig. 8.

Now asking for a sub-strategy *safeFast* of *safe* that will minimize the expected accumulated distance to the front care yields a substantial improvement as seen in Fig. 9.

Home Automation. Home automation includes the centralized control of a number of functionalities in a house such as lighting, HVAC (heating, ventilation and air conditioning), appliances, security locks of gates and doors as well as other

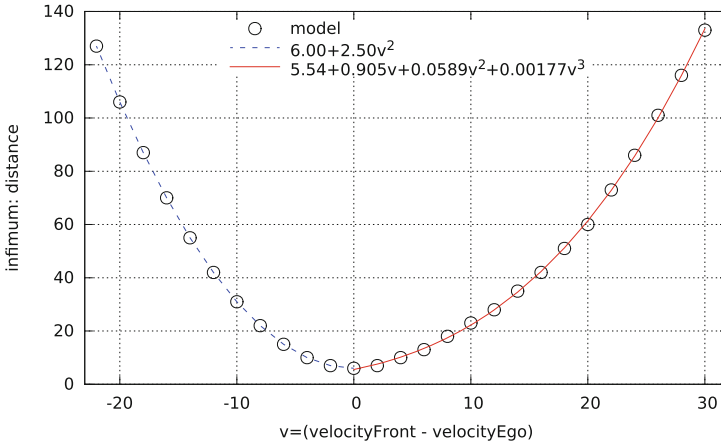


Fig. 8. Smallest distance possible under the *safe* strategy as a function of speed difference computed using $\inf\{\text{velocityFront}-\text{velocityEgo}=v\}$: distance under *safe* for each v value. Connecting lines are from linear regression analysis.

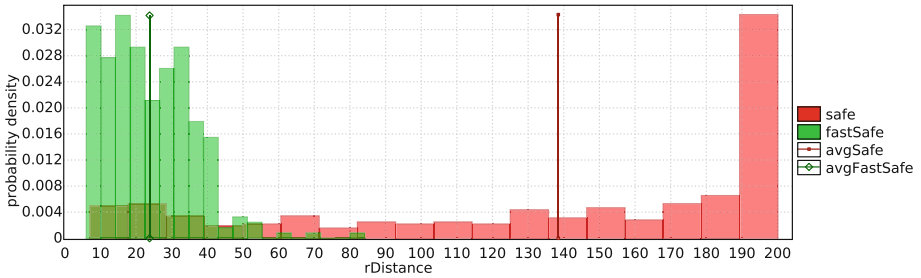


Fig. 9. The probability density distribution over $r\text{Distance}$ at time ≥ 100 thus after 100 time units under the strategies *safe* and *safeFast*. The (dark) red bars for *safe* and the (light) green bars for *safeFast*. (Color figure online)

systems. The overall goal is to achieve improved convenience, comfort, energy efficiency as well as security. The popularity of home automation has increased significantly in recent years through affordable smartphone and tablet connectivity. Also the emergence of “Internet of Things” has tied in closely with the popularization of home automation.

In [48] we collaborated with the Danish company Seluxit within the European project CASSTING². The focus was on the floorheating system of a family house, where each room of the house has its own hot-water pipe circuit. These are controlled through a number of valves based on information about room temperatures communicated wirelessly (periodically due to energy considerations) from a number of temperature sensors. In the existing system, a simple

² <http://www.cassting-project.eu/>.

“Bang-Bang”-like strategy is applied, however, there are though several problems with this strategy, as experienced by the house owner: it completely disregards the interaction between rooms in terms of heat-exchange, the impact of the outside temperature and weather forecast as well as information about movements in the house. Taking this knowledge into account should potentially enable the synthesis of significantly improved control strategies. Unfortunately, direct application of UPPAAL STRATEGO does not scale: due to the enormous number of control modes it is virtually impossible to learn optimal control. Instead, we proposed a novel on-line synthesis methodology, where we periodically—and on-line—learn the optimal controller for the near future based on the current sensor readings. For additional scalability, we proposed and applied a novel compositional synthesis approach.

In particular, the strategy provided by UPPAAL STRATEGO takes weather information into account, as illustrated by Fig. 10 showing the spring stability scenario. From points of time between 0 and 500 min, the outside temperature increases and exceeds the target temperature. We observe that since the controller synthesized by UPPAAL STRATEGO is able to look at the weather forecast for the next 45 min, it shuts down the valves much earlier than the other controllers. This results in energy savings and increased comfort.

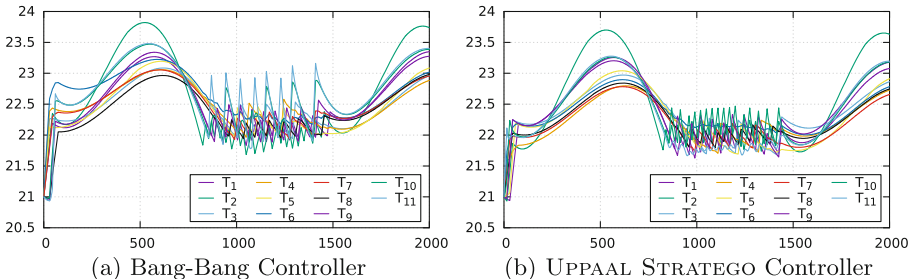


Fig. 10. Room temperatures in the spring stability scenario

Intelligent Control of Traffic Light. The Danish Congestion Commission calls in its recent report for improved traffic signal control in order to reduce congestion, travel time and energy consumption. This project has been formulated to contribute to a more efficient utilisation of the existing infrastructure by improving traffic signal control. However, modern traffic lights use information from induction loops and to some extent radar information. Recent developments in radar technology has made it possible to obtain more detailed information relevant to the control mechanism of the traffic light. Unfortunately much of the current controllers do not profit from this additional information. Using this information could minimize waiting times and energy waste.

Within the Innovation Center DiCyPS³ we have collaborated with researchers in traffic control to apply UPPAAL STRATEGO to the synthesis of an efficient

³ Center for Data-Intensive Cyber-Physical Systems, www.dicyps.dk.



Fig. 11. Intersection between Nylandsvej and Værkstedvej at Køge municipality. Layout of loops and radar area.

traffic signal control strategy that takes advantage of the continuous traffic monitoring made available by radar detectors. The purpose of the strategy is to optimize the total traffic flow in the junction, i.e. to reduce the total delay, queue length and the number of stops. The synthesis of UPPAAL STRATEGO is done on-line offering every 5 s a new updated optimal strategy for the next operation of a signalized intersection in the municipality of Køge, Denmark, Fig. 11. In doing so the UPPAAL STRATEGO model takes into account the random generation of traffic in the various directions. The on-line strategy generated is fed to a richer simulation engine in SUMO, an open source tool which allows to model and simulate traffic systems. SUMO also provides a number of supporting tools which allow for visualization, network transformation, waiting time calculations, traffic light performance, etc.

In the resulting evaluation shown in Fig. 2 we have compared the performance of a so-called Static controller, the Loop Controller and the UPPAAL STRATEGO controller. In the most demanding MAX scenario – with highest intensity of traffic – it is clear that the UPPAAL STRATEGO controller is performing significantly better than any of the others. For MID scenario the findings are similar and for the LOW scenario all the controllers perform quite similar, but the Loop controller is in general the best (Table 2).

Table 2. Results of the experiments. We show the mean and the 95 percentile for respectively the waiting time of the cars and the queue length. This is done for each controller in all scenarios.

| Scenario | Direction | Delay in Seconds (Waiting Time) | | | | | | Queue Length in Meters | | | | | |
|----------|-----------|---------------------------------|------|----------|--------|------|----------|------------------------|------|----------|--------|------|----------|
| | | Mean | | | 95p | | | Mean | | | 95p | | |
| | | Static | Loop | STRATEGO | Static | Loop | STRATEGO | Static | Loop | STRATEGO | Static | Loop | STRATEGO |
| MAX | A1 | 19 | 7 | 10 | 69 | 49 | 52 | 23 | 10 | 13 | 67 | 45 | 60 |
| | A2 | 25 | 8 | 9 | 87 | 50 | 47 | 31 | 11 | 12 | 105 | 45 | 54 |
| | B1 | 69 | 89 | 25 | 221 | 300 | 77 | 24 | 31 | 8 | 142 | 188 | 45 |
| | B2 | 108 | 169 | 28 | 263 | 389 | 88 | 44 | 68 | 11 | 188 | 286 | 53 |
| | ALL | 38 | 37 | 13 | 162 | 242 | 61 | 31 | 30 | 11 | 144 | 195 | 52 |
| MID | A1 | 13 | 8 | 8 | 40 | 36 | 32 | 17 | 11 | 11 | 52 | 38 | 39 |
| | A2 | 13 | 10 | 7 | 49 | 42 | 33 | 17 | 14 | 10 | 54 | 52 | 37 |
| | B1 | 15 | 25 | 21 | 43 | 63 | 57 | 5 | 8 | 7 | 22 | 30 | 30 |
| | B2 | 26 | 38 | 25 | 82 | 105 | 64 | 10 | 15 | 10 | 37 | 52 | 30 |
| | ALL | 15 | 14 | 11 | 48 | 61 | 44 | 12 | 12 | 10 | 45 | 45 | 37 |
| LOW | A1 | 7 | 6 | 5 | 22 | 25 | 23 | 6 | 5 | 4 | 23 | 22 | 22 |
| | A2 | 5 | 4 | 5 | 22 | 21 | 22 | 4 | 4 | 4 | 15 | 15 | 22 |
| | B1 | 11 | 11 | 16 | 33 | 38 | 45 | 2 | 2 | 2 | 7 | 15 | 15 |
| | B2 | 13 | 9 | 16 | 35 | 30 | 45 | 3 | 2 | 3 | 15 | 15 | 15 |
| | ALL | 7 | 6 | 8 | 29 | 26 | 30 | 4 | 3 | 4 | 15 | 15 | 15 |

References

1. Third International Conference on the Quantitative Evaluation of Systems (QEST 2006), Riverside, California, USA, 11–14 September 2006. IEEE Computer Society (2006)
2. Alur, R., La Torre, S., Pappas, G.J.: Optimal paths in weighted timed automata. In: Benedetto, M.D., Sangiovanni-Vincentelli, A. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 49–62. Springer, Heidelberg (2001). doi:[10.1007/3-540-45351-2_8](https://doi.org/10.1007/3-540-45351-2_8). [10]
3. Behrmann, G., Bouyer, P., Fleury, E., Larsen, K.G.: Static guard analysis in timed automata verification. In: Garavel, H., Hatcliff, J. (eds.) TACAS 2003. LNCS, vol. 2619, pp. 254–270. Springer, Heidelberg (2003). doi:[10.1007/3-540-36577-X_18](https://doi.org/10.1007/3-540-36577-X_18)
4. Behrmann, G., Bouyer, P., Larsen, K.G., Pelánek, R.: Lower and upper bounds in zone-based abstractions of timed automata. STTT **8**(3), 204–215 (2006)
5. Behrmann, G., Cougnard, A., David, A., Fleury, E., Larsen, K.G., Lime, D.: UPPAAL-tiga: time for playing games! In: Damm, W., Hermanns, H. (eds.) CAV 2007. LNCS, vol. 4590, pp. 121–125. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-73368-3_14](https://doi.org/10.1007/978-3-540-73368-3_14). [29]
6. Behrmann, G., David, A., Larsen, K.G., Håkansson, J., Pettersson, P., Yi, W., Hendriks, M.: UPPAAL 4.0. In: Third International Conference on the Quantitative Evaluation of Systems (QEST 2006), Riverside, California, USA, 11–14 September 2006, pp. 125–126 (2006). [1]
7. Behrmann, G., David, A., Larsen, K.G., Pettersson, P., Yi, W.: Developing UPPAAL over 15 years. Softw. Pract. Exp. **41**(2), 133–142 (2011)
8. Behrmann, G., Fehnker, A., Hune, T., Larsen, K., Pettersson, P., Romijn, J., Vaandrager, F.: Minimum-cost reachability for priced time automata. In: Benedetto, M.D., Sangiovanni-Vincentelli, A. (eds.) HSCC 2001. LNCS, vol. 2034, pp. 147–161. Springer, Heidelberg (2001). doi:[10.1007/3-540-45351-2_15](https://doi.org/10.1007/3-540-45351-2_15). [10]
9. Behrmann, G., Larsen, K.G., Pearson, J., Weise, C., Yi, W.: Efficient timed reachability analysis using clock difference diagrams. In: Halbwachs, N., Peled, D. (eds.)

- CAV 1999. LNCS, vol. 1633, pp. 341–353. Springer, Heidelberg (1999). doi:[10.1007/3-540-48683-6_30](https://doi.org/10.1007/3-540-48683-6_30)
10. Benedetto, M.D., Sangiovanni-Vincentelli, A. (eds.): HSCC 2001. LNCS, vol. 2034. Springer, Heidelberg (2001)
 11. Bertrand, N., Bouyer, P., Brihaye, T., Markey, N.: Quantitative model-checking of one-clock timed automata under probabilistic semantics. In: Fifth International Conference on the Quantitative Evaluation of Systems (QEST 2008), Saint-Malo, France, 14–17 September 2008, pp. 55–64. IEEE Computer Society (2008)
 12. Bisgaard, M., Gerhardt, D., Hermanns, H., Krčál, J., Nies, G., Stenger, M.: Battery-aware scheduling in low orbit: the GOMX-3 case. In: Fitzgerald, J., Heitmeyer, C., Gnesi, S., Philippou, A. (eds.) FM 2016. LNCS, vol. 9995, pp. 559–576. Springer, Heidelberg (2016). doi:[10.1007/978-3-319-48989-6_34](https://doi.org/10.1007/978-3-319-48989-6_34)
 13. Boudjadar, A., David, A., Kim, J.H., Larsen, K.G., Mikucionis, M., Nyman, U., Skou, A.: Degree of schedulability of mixed-criticality real-time systems with probabilistic sporadic tasks. In: 2014 Theoretical Aspects of Software Engineering Conference, TASE 2014, Changsha, China, 1–3 September 2014, pp. 126–130. IEEE Computer Society (2014)
 14. Bouyer, P., Brihaye, T., Jurdzinski, M., Menet, Q.: Almost-sure model-checking of reactive timed automata. In: Ninth International Conference on Quantitative Evaluation of Systems, QEST 2012, London, United Kingdom, 17–20 September 2012, pp. 138–147. IEEE Computer Society (2012)
 15. Bouyer, P., Cassez, F., Fleury, E., Larsen, K.G.: Optimal strategies in priced timed game automata. In: Lodaya, K., Mahajan, M. (eds.) FSTTCS 2004. LNCS, vol. 3328, pp. 148–160. Springer, Heidelberg (2004). doi:[10.1007/978-3-540-30538-5_13](https://doi.org/10.1007/978-3-540-30538-5_13)
 16. Bouyer, P., Cassez, F., Fleury, E., Larsen, K.G.: Synthesis of optimal strategies using hytech. *Electr. Notes Theor. Comput. Sci.* **119**(1), 11–31 (2005)
 17. Bouyer, P., Colange, M., Markey, N.: Symbolic optimal reachability in weighted timed automata. In: Chaudhuri, S., Farzan, A. (eds.) CAV 2016. LNCS, vol. 9779, pp. 513–530. Springer, Heidelberg (2016). doi:[10.1007/978-3-319-41528-4_28](https://doi.org/10.1007/978-3-319-41528-4_28)
 18. Bouyer, P., Jaziri, S., Markey, N.: On the value problem in weighted timed games. In: Aceto, L., de Frutos-Escrig, D. (eds.) 26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, 1–4 September 2015. LIPIcs, vol. 42, pp. 311–324. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2015)
 19. Bouyer, P., Larsen, K.G., Markey, N., Rasmussen, J.I.: Almost optimal strategies in one clock priced timed games. In: Arun-Kumar, S., Garg, N. (eds.) FSTTCS 2006. LNCS, vol. 4337, pp. 345–356. Springer, Heidelberg (2006). doi:[10.1007/11944836_32](https://doi.org/10.1007/11944836_32)
 20. Brihaye, T., Bruyère, V., Raskin, J.-F.: On optimal timed strategies. In: Pettersson, P., Yi, W. (eds.) FORMATS 2005. LNCS, vol. 3829, pp. 49–64. Springer, Heidelberg (2005). doi:[10.1007/11603009_5](https://doi.org/10.1007/11603009_5)
 21. Bulychev, P., David, A., Larsen, K.G., Legay, A., Li, G., Poulsen, D.B.: Rewrite-based statistical model checking of WMTL. In: Qadeer, S., Tasiran, S. (eds.) RV 2012. LNCS, vol. 7687, pp. 260–275. Springer, Heidelberg (2013). doi:[10.1007/978-3-642-35632-2_25](https://doi.org/10.1007/978-3-642-35632-2_25)
 22. Bulychev, P., David, A., Larsen, K.G., Legay, A., Li, G., Poulsen, D.B., Stainer, A.: Monitor-based statistical model checking for weighted metric temporal logic. In: Bjørner, N., Voronkov, A. (eds.) LPAR 2012. LNCS, vol. 7180, pp. 168–182. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-28717-6_15](https://doi.org/10.1007/978-3-642-28717-6_15)
 23. Bulychev, P., David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B.: Checking and distributing statistical model checking. In: Goodloe, A.E., Person,

- S. (eds.) NFM 2012. LNCS, vol. 7226, pp. 449–463. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-28891-3_39](https://doi.org/10.1007/978-3-642-28891-3_39)
24. Cassez, F., David, A., Fleury, E., Larsen, K.G., Lime, D.: Efficient on-the-fly algorithms for the analysis of timed games. In: Abadi, M., Alfaró, L. (eds.) CONCUR 2005. LNCS, vol. 3653, pp. 66–80. Springer, Heidelberg (2005). doi:[10.1007/11539452_9](https://doi.org/10.1007/11539452_9)
 25. Cassez, F., David, A., Larsen, K.G., Lime, D., Raskin, J.-F.: Timed control with observation based and stuttering invariant strategies. In: Namjoshi, K.S., Yoneda, T., Higashino, T., Okamura, Y. (eds.) ATVA 2007. LNCS, vol. 4762, pp. 192–206. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-75596-8_15](https://doi.org/10.1007/978-3-540-75596-8_15)
 26. Cassez, F., Jessen, J.J., Larsen, K.G., Raskin, J.-F., Reynier, P.-A.: Automatic synthesis of robust and optimal controllers – an industrial case study. In: Majumdar, R., Tabuada, P. (eds.) HSCC 2009. LNCS, vol. 5469, pp. 90–104. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-00602-9_7](https://doi.org/10.1007/978-3-642-00602-9_7)
 27. Damm, W., Hermanns, H. (eds.): CAV 2007. LNCS, vol. 4590. Springer, Heidelberg (2007)
 28. David, A., Du, D., Larsen, K.G., Mikucionis, M., Skou, A.: An evaluation framework for energy aware buildings using statistical model checking. *Sci. China Inf. Sci.* **55**(12), 2694–2707 (2012)
 29. David, A., Jensen, P.G., Larsen, K.G., Legay, A., Lime, D., Sørensen, M.G., Taankvist, J.H.: On time with minimal expected cost! In: Cassez, F., Raskin, J.-F. (eds.) ATVA 2014. LNCS, vol. 8837, pp. 129–145. Springer, Heidelberg (2014). doi:[10.1007/978-3-319-11936-6_10](https://doi.org/10.1007/978-3-319-11936-6_10)
 30. David, A., Jensen, P.G., Larsen, K.G., Mikučionis, M., Taankvist, J.H.: UPPAAL STRATEGO. In: Baier, C., Tinelli, C. (eds.) TACAS 2015. LNCS, vol. 9035, pp. 206–211. Springer, Heidelberg (2015). doi:[10.1007/978-3-662-46681-0_16](https://doi.org/10.1007/978-3-662-46681-0_16)
 31. David, A., Larsen, K.G., Legay, A., Mikucionis, M., Poulsen, D.B.: Uppaal SMC tutorial. *STTT* **17**(4), 397–415 (2015)
 32. David, A., Larsen, K.G., Legay, A., Mikucionis, M., Poulsen, D.B., Sedwards, S.: Statistical model checking for biological systems. *STTT* **17**(3), 351–367 (2015)
 33. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B., Vliet, J., Wang, Z.: Statistical model checking for networks of priced timed automata. In: Fahrenberg, U., Tripakis, S. (eds.) FORMATS 2011. LNCS, vol. 6919, pp. 80–96. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-24310-3_7](https://doi.org/10.1007/978-3-642-24310-3_7)
 34. David, A., Larsen, K.G., Legay, A., Mikučionis, M., Wang, Z.: Time for statistical model checking of real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 349–355. Springer, Heidelberg (2011). doi:[10.1007/978-3-642-22110-1_27](https://doi.org/10.1007/978-3-642-22110-1_27). [41]
 35. David, A., Larsen, K.G., Legay, A., Nyman, U., Traonouez, L.-M., Wasowski, A.: Real-time specifications. *STTT* **17**(1), 17–45 (2015)
 36. David, A., Larsen, K.G., Legay, A., Nyman, U., Wasowski, A.: Timed I/O automata: a complete specification theory for real-time systems. In: Johansson, K.H., Yi, W. (eds.) Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2010, Stockholm, Sweden, 12–15 April 2010, pp. 91–100. ACM (2010)
 37. David, A., Larsen, K.G., Legay, A., Mikučionis, M.: Schedulability of herschel-planck revisited using statistical model checking. In: Margaria, T., Steffen, B. (eds.) ISoLA 2012. LNCS, vol. 7610, pp. 293–307. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-34032-1_28](https://doi.org/10.1007/978-3-642-34032-1_28)

38. David, A., Larsen, K.G., Legay, A., Nyman, U., Wasowski, A.: ECDAR: an environment for compositional design and analysis of real time systems. In: Bouajjani, A., Chin, W.-N. (eds.) ATVA 2010. LNCS, vol. 6252, pp. 365–370. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-15643-4_29](https://doi.org/10.1007/978-3-642-15643-4_29)
39. David, N., David, A., Hansen, R.R., Larsen, K.G., Legay, A., Olesen, M.C., Probst, C.W.: Modelling social-technical attacks with timed automata. In: Bertino, E., You, I. (eds.) Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats, MIST 2015, Denver, Colorado, USA, 16 October 2015, pp. 21–28. ACM (2015)
40. Dierks, H., Kupferschmid, S., Larsen, K.G.: Automatic abstraction refinement for timed automata. In: Raskin, J.-F., Thiagarajan, P.S. (eds.) FORMATS 2007. LNCS, vol. 4763, pp. 114–129. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-75454-1_10](https://doi.org/10.1007/978-3-540-75454-1_10). [57]
41. Gopalakrishnan, G., Qadeer, S. (eds.): CAV 2011. LNCS, vol. 6806. Springer, Heidelberg (2011)
42. Herbretreau, F., Srivathsan, B., Walukiewicz, I.: Better abstractions for timed automata. *Inf. Comput.* **251**, 67–90 (2016)
43. Jensen, P.G., Larsen, K.G., Srba, J., Sørensen, M.G., Taankvist, J.H.: Memory efficient data structures for explicit verification of timed systems. In: Badger, J.M., Rozier, K.Y. (eds.) NFM 2014. LNCS, vol. 8430, pp. 307–312. Springer, Heidelberg (2014). doi:[10.1007/978-3-319-06200-6_26](https://doi.org/10.1007/978-3-319-06200-6_26)
44. Jessen, J.J., Rasmussen, J.I., Larsen, K.G., David, A.: Guided controller synthesis for climate controller using UPPAAL TIGA. In: Raskin, J.-F., Thiagarajan, P.S. (eds.) FORMATS 2007. LNCS, vol. 4763, pp. 227–240. Springer, Heidelberg (2007). doi:[10.1007/978-3-540-75454-1_17](https://doi.org/10.1007/978-3-540-75454-1_17). [57]
45. Jørgensen, K.Y., Larsen, K.G., Srba, J.: Time-darts: a data structure for verification of closed timed automata. In: Cassez, F., Huuck, R., Klein, G., Schlich, B. (eds.) Proceedings Seventh Conference on Systems Software Verification, SSV 2012, Sydney, Australia, 28–30 November 2012. EPTCS, vol. 102, pp. 141–155 (2012)
46. Kupferschmid, S., Hoffmann, J., Dierks, H., Behrmann, G.: Adapting an AI planning heuristic for directed model checking. In: Valmari, A. (ed.) SPIN 2006. LNCS, vol. 3925, pp. 35–52. Springer, Heidelberg (2006). doi:[10.1007/11691617_3](https://doi.org/10.1007/11691617_3)
47. Kupferschmid, S., Wehrle, M., Nebel, B., Podelski, A.: Faster than UPPAAL? In: Gupta, A., Malik, S. (eds.) CAV 2008. LNCS, vol. 5123, pp. 552–555. Springer, Heidelberg (2008). doi:[10.1007/978-3-540-70545-1_53](https://doi.org/10.1007/978-3-540-70545-1_53)
48. Larsen, K.G., Mikučionis, M., Muñiz, M., Srba, J., Taankvist, J.H.: Online and compositional learning of controllers with application to floor heating. In: Chechik, M., Raskin, J.-F. (eds.) TACAS 2016. LNCS, vol. 9636, pp. 244–259. Springer, Heidelberg (2016). doi:[10.1007/978-3-662-49674-9_14](https://doi.org/10.1007/978-3-662-49674-9_14)
49. Larsen, K.G., Behrmann, G., Brinksma, E., Fehnker, A., Hune, T., Pettersson, P., Romijn, J.: As cheap as possible: efficient cost-optimal reachability for priced timed automata. In: Berry, G., Comon, H., Finkel, A. (eds.) CAV 2001. LNCS, vol. 2102, pp. 493–505. Springer, Heidelberg (2001). doi:[10.1007/3-540-44585-4_47](https://doi.org/10.1007/3-540-44585-4_47)
50. Larsen, K.G., Larsson, F., Pettersson, P., Yi, W.: Efficient verification of real-time systems: compact data structure and state-space reduction. In: Proceedings of the 18th IEEE Real-Time Systems Symposium (RTSS 1997), San Francisco, CA, USA, 3–5 December 1997, pp. 14–24. IEEE Computer Society (1997)
51. Larsen, K.G., Mikucionis, M., Nielsen, B.: Online testing of real-time systems using UPPAAL. In: Grabowski, J., Nielsen, B. (eds.) FATES 2004. LNCS, vol. 3395, pp. 79–94. Springer, Heidelberg (2005). doi:[10.1007/978-3-540-31848-4_6](https://doi.org/10.1007/978-3-540-31848-4_6)

52. Larsen, K.G., Mikucionis, M., Nielsen, B., Skou, A.: Testing real-time embedded software using UPPAAL-TRON: an industrial case study. In: Wolf, W.H. (ed.) Proceedings of the 5th ACM International Conference On Embedded Software EMSOFT 2005, Jersey City, NJ, USA, 18–22 September 2005 , pp. 299–306. ACM (2005)
53. Larsen, K.G., Mikučionis, M., Taankvist, J.H.: Safe and optimal adaptive cruise control. In: Meyer, R., Platzer, A., Wehrheim, H. (eds.) Correct System Design. LNCS, vol. 9360, pp. 260–277. Springer, Heidelberg (2015). doi:[10.1007/978-3-319-23506-6_17](https://doi.org/10.1007/978-3-319-23506-6_17)
54. Larsen, K.G., Pettersson, P., Yi, W.: UPPAAL in a nutshell. *STTT* **1**(1–2), 134–152 (1997)
55. Maler, O., Larsen, K.G., Krogh, B.H.: On zone-based analysis of duration probabilistic automata. In: Chen, Y.-F., Rezine, A. (eds.) Proceedings 12th International Workshop on Verification of Infinite-State Systems, INFINITY 2010, Singapore, Singapore, 21 September 2010. EPTCS, vol. 39, pp. 33–46 (2010)
56. Mikucionis, M., Larsen, K.G., Nielsen, B.: T-UPPAAL: online model-based testing of real-time systems. In: 19th IEEE International Conference on Automated Software Engineering (ASE 2004), Linz, Austria, 20–25 September 2004, pp. 396–397. IEEE Computer Society (2004)
57. Raskin, J.-F., Thiagarajan, P.S. (eds.): FORMATS 2007. LNCS, vol. 4763. Springer, Heidelberg (2007)
58. Tretmans, J.: A formal approach to conformance testing. In: Rafiq, O. (ed.) Protocol Test Systems, VI, Proceedings of the IFIP TC6/WG6.1 Sixth International Workshop on Protocol Test systems, Pau, France, 28–30 September 1993. IFIP Transactions, vol. C-19, pp. 257–276. North-Holland (1993)
59. van Glabbeek, R.J., Höfner, P., Portmann, M., Tan, W.L.: Modelling and verifying the AODV routing protocol. *Distrib. Comput.* **29**(4), 279–315 (2016)