[Array 11 (2021) 100069](https://doi.org/10.1016/j.array.2021.100069)

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | Contents lists available at ScienceDirect |  |
| Array |
| journal homepage: www.elsevier.com/journals/array/2590-0056/open-access-journal |
|  | | |

On the stability of (self-)adaptive behaviour in continuously changing environments: A quantification approach   
Martin Goller, Sven Tomforde\*   
Christian-Albrechts-Universit€at zu Kiel, Intelligent Systems, Hermann-Rodewald-Str. 3, 24118, Kiel, Germany

|  |  |
| --- | --- |
| A R T I C L E I N F O | A B S T R A C T |
| Keywords:  Self-awareness  Self-adaptive and self-organising systems System analysis  Stability  Degree of self-adaptation  Autonomic computing  Organic computing | The concept of self-adaptation and self-organisation (SASO) is a modern approach to cope with the ever-increasing complexity and interconnectedness of large-scale component systems. The basic idea is to react to environmental dynamics and disturbances by re-configuring the productive behaviour and/or the relations to other systems. However, this may result in unstable and even oscillating macro-level behaviour, potentially rendering the adaptation efforts of the contained component systems inappropriate. We assume that such an unstable configuration is an indicator of unexpected behaviour which can lead to a reduced utility of the overall system. To enable the system to be self-aware about such events, we propose a concept to measure the config-uration stability of a SASO system by creating a derived time series based on the configurations. This is based on |

the application of the Kinoshita measure. We show the applicability of the concept and the observed behaviour in different simulated use-cases.

|  |  |
| --- | --- |
| 1. Introduction | adaptation decisions of the (sub-)systems will lead to decreased stability |

of the overall system and potentially to a decreased user acceptance.

Within the last decade, (self-)adaptivity increasingly became an in-tegral part of the design of next-generation information and communi-

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| cation | systems | [1]. | Concepts | such | as | intelligent | systems | [2], |

cyber-physical systems [3], Internet of Things [4] or self-aware computing systems [5] established a trend in systems engineering to-wards constellations of typically distributed autonomous subsystems that integrate dynamically into an overall system constellation and self-adapt their behaviour in response to changing situations [6]. We refer to sys-tems with abilities to autonomously modify their behaviour and their structural integration in terms of cooperation with other (sub-)systems as self-adaptive and self-organising (SASO) systems in the remainder of this

Although there might be a perfect reason for each decision about self-adaptation at sub-system-level, the resulting macro-level behaviour and the interplay of the distributed subsystems’ behaviour may lead to unstable or even oscillating conditions. Although there is some work on the quantification of system properties and runtime behaviour in SASO system (see section 2.2 for an overview), what is missing is an integrated approach to balancing the adaptation decision between possible perfor-mance and robustness gains on the one hand and system stability and user acceptance on the other. With this article, we propose a first step towards such a constructive trade-off – the basis for this is an awareness of the actual stability within the system federation, based on external

article. observation.

To equip a SASO system with the capability of context-aware self-adaptation, different approaches have been proposed [9], e.g. by the Automatic Computing [7] or Organic Computing [8] initiatives. gives an overview. These system designs often incorporate autonomous learning techniques to allow a continuous and unsupervised improvement of the self-adaptation behaviour [10]. The result of these efforts is that systems can react fast and continuously to changing conditions as well as dis-turbances [11] and consequently achieve higher robustness [12].

As outlined in previous work [13], we assume that too frequent

\* Corresponding author.

E-mail address: st@informatik.uni-kiel.de (S. Tomforde).

<https://doi.org/10.1016/j.array.2021.100069>

Based on the ideas to quantify a ‘degree of adaptation’, the authors have presented a first concept for measuring the stability of the system's configuration in Ref. [14]. The previous article evaluated the measure in an external disturbance scenario and a system failure scenario. This article extends the previous work by the evaluation of the measure in two new scenarios (see sections 4.2 and 4.4), determines novel insights about the behaviour and its interpretability, and gives an approach to deter-mine the necessary parameters for the evaluation (see section 3.3).

The remainder of this article is organised as follows: Section 2

Received 7 December 2020; Received in revised form 22 March 2021; Accepted 10 May 2021   
Available online 28 May 2021   
2590-0056/© 2021 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

M. Goller, S. Tomforde Array 11 (2021) 100069

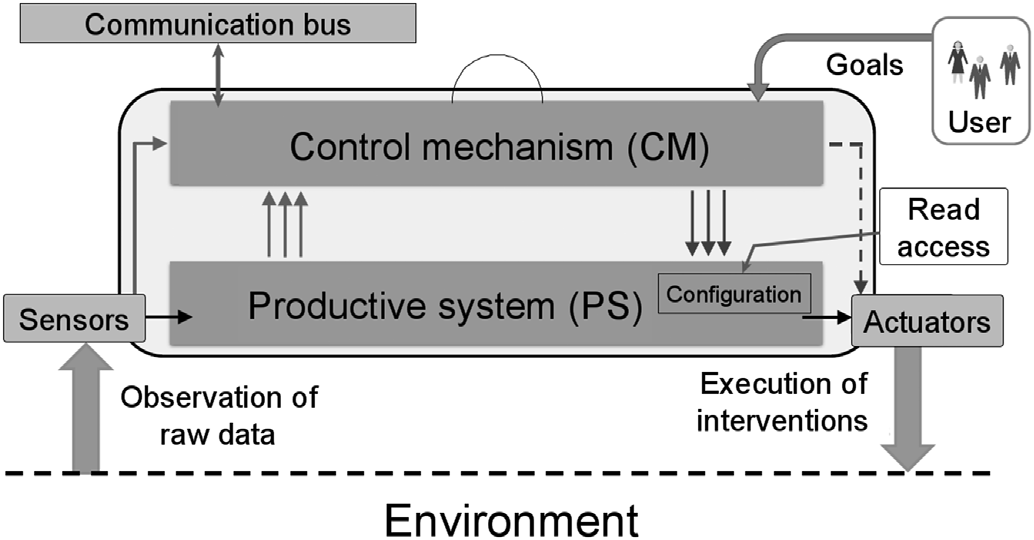


Fig. 1. Schematic illustration of a subsystem ai from Ref. [15]. The arrows from the sensors and PS to the CM indicate observation flows, while the arrows from the CM to the PS and the actuators indicate control flows. Dashed arrows emphasise a possible path that is typically not used. Not shown: The CM is able to communicate with other CMs in the shared environment to exchange infor-mation such as sensor reading and to negotiate policies.

explains the underlying system model for SASO systems including the assumptions made by the authors and describes related work. Section 3 presents an approach to the measurement of adaptation behaviour sta-bility, which is experimentally analysed in section 4. Finally, section 5 summarises the paper and gives an outlook on future work.

particular realisation of the parameter (e.g., as real value, boolean/flag, integer or categorical variable). Each subsystem has its own configura-tion space, i.e. an n-dimensional space defining all possible realisations of the configuration vector. The combination of the current configuration vectors of all contained subsystems of the overall system S defines the joint configuration of S. We assume that modifications of the configu-ration vectors are done by the different CM only, i.e. locally at each subsystem, and are the result of the self-adaptation process of the CM.

2.2. Related work

Within the last two decades, some work has been presented that aims at measuring and assessing SASO system properties. The most prominent examples are self-organisation [17] and emergence [18]. However, most of these metrics require domain knowledge, are dedicated to a specific application, are restricted to simple, small-scale models [19], or describe architectural and implementational properties for the comparison of systems at design-time [20].

Kaddoum et al. [21] discuss the need to refine classical performance metrics to SASO purposes and present specific metrics for self-adaptive systems. They distinguish between “nominal” and “self-\*” situations and their relations. For instance, they measure the operation time about the adaptation time to determine the effort. Some of the developed metrics have been investigated in detail by Camara et al. for software architecture scenarios [22]. Besides, success and adaptation efforts and ways to measure autonomy have been investigated, see e.g. [23].

2. Background Only a few contributions focused on general metrics to determine the

effort and the benefit of (self-)adaptation in distributed collections of

In this section, we initially describe our system model that defines which characteristics and capabilities of a SASO system we argue. Af-terwards, we briefly summarise developments of related work.

autonomous subsystems, see Ref. [24] for an overview. Examples include the relation between working and adaptation time, the availability of subsystems for task processing, and the performance of the overall system (i.e., the degree to which a certain goal is achieved). This has been

2.1. System model accompanied by a transfer of traditional performance metrics, see

Ref. [21] or augmented with measures necessary for control strategies of

In this article, we refer to a SASO system S as a collection A of autonomous subsystems ai that are able to adapt their behaviour based on self-awareness of the internal and external conditions. We further assume that such a subsystem is an entity that interacts with other entities, i.e., other systems, including hardware, software, humans, and the physical world with its natural phenomena. These other entities are referred to as the environment of the given system. The system boundary is the common frontier between the system and its environment.

virtual). Internally, each ai consists of two parts: The productive system Each ai 2 A is equipped with sensors and actuators (both, physical or

part PS, which is responsible for the basic purpose of the system, and the control mechanism CM, which controls the behaviour of the PS (i.e., performs self-adaptation) and decides about relations to other sub-systems. In comparison to other system models, this corresponds to the

SASO systems [25]. However, the focus on an degree of adaptation, the stability of adaptations, and the foundation for deriving a trade-off be-tween possible performance gain of individual component systems and negative impact on macro-level behaviour has not been focused.

In general, there is only a very limited number of contributions on when (and if at all) to adapt in literature. Recently, Chen et al. presented a concept that is based on the so-called ‘technical debt’ [26]: The authors introduce a gate that either enables or disables the adaptation mecha-nism. This decision is based on deriving an integrated score for defining the current so-called ‘temporal interest’ and the expected ‘revenue’. This implicitly allows for a similar goal as outlined in this paper: More sta-bility rather than frequent self-adaptation. However, it does not come with a measurement that determines the degree of self-adaptation or the degree of stability, preferable in relation to the current context.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| separation of concerns between System under Observation and Control | In | [17] | Tomforde | et | al. | introduce | a | method | to | measure |

(SuOC) and Observer/Controller tandem [15] in the terminology of Organic Computing (OC) [16] or Managed Resource and Autonomic Manager in terms of Autonomic Computing [7]. Fig. 1 illustrates this concept with its input and output relations. The user describes the system purpose by providing a utility or goal function U which determines the behaviour of the subsystem. The User usually takes no further action to influence the decisions of the subsystem. Actual decisions are taken by the productive system and the CM based on the external and internal conditions and messages exchanged with other subsystems. We model each subsystem to act autonomously, i.e., there are no control hierarchies in the overall system. Please note that for the context of this article an explicit local configuration of the PS is necessary – which in turn limits the scope of the applicability of the proposed method. Furthermore, each subsystem must provide a read access to the configuration.

At each point in time, the productive system of each ai is configured using a vector ci. This vector contains a specific value for each control variable that can be altered to steer the behaviour, independently of the

2

M. Goller, S. Tomforde Array 11 (2021) 100069

in this article), variability, or acceptability (which is subject to future work). Technically, our probabilistic approach differs fundamentally from other approaches as it models each aspect of the distributed sub-systems’ configurations as random variables without any assumptions on the impact of the performance, the causes for adaptation decisions, or temporal implications. We compare the observations using divergence measures, which then allows for defining different measures for self-adaptation on-top of this quantification.

3. Configuration stability

If the CM decides that an adaptation is necessary it will change the configuration of the productive system. Since the decision models used by the CM are hidden to external observers, we model the configuration changes as a random process with the configuration vector as a random variable. Therefore, the observed configurations form a random distri-bution. For this distribution, an estimated density can be associated and new configurations can be assigned a probability concerning that density.

Our approach is based on the idea that if new configurations with a high probability are chosen the system works as expected and the un-derlying adaptation is desirable or ‘normal’. In such a case, we call the configuration state ‘stable’. If a new configuration has a low probability we still assume that the system is working as usual and that this adap-tation is due to a minor disturbance. Only when configurations with low probability are chosen over a longer course of time, we presume that a major disturbance or a system failure is on hand.

If it comes down to the implementation, configuration parameters are usually represented as real numbers. Since a single vector of real numbers has a probability of P ¼ 0 in a continuous density, we will look at the probability densities that are created by several vectors.

By comparing the configurations of a subsystem in a current time window with those of a previous window, we can identify abnormal adaptation activity in a single subsystem. To identify abnormal changes in a SASO system at a global scale (i.e. at macro-level, we take all configuration comparisons into account and then apply a measure based on the “macroscopic measure for detecting abnormal changes in a multi-agent system” as defined by Kinoshita [28].

3.1. Definition of the Kinoshita measures

Kinoshita proposes a measure which he uses as an indicator for un-usual activity changes in a distributed multi-agent system. Kinoshita defines two values, the activity factor at a given time and the variance of fluctuation of the activity factor. The activity factor is based on the classification of each agent as either active or inactive. Let N denote the total number of agents in the multi-agent system and let nt be the number of active agents at a given time t. The activity factor zt is defined as

|  |  |  |
| --- | --- | --- |
| zt :¼2 � nt ~~�~~ N ~~þ~~ 1 ~~2~~ � ~~N~~  (1)  For a given window size M, the fluctuation ξt of the activity factor at  time t is calculated as | | |
| ξt :¼ zt �1 M� | X | zt�i |
| and finally, we calculate the variance νt of the fluctuation: | | |

3.2. Application of the Kinoshita measures

To apply the Kinoshita measure, we need to define what active and inactive agents are. Obviously, the subsystems of our SASO system are the agents in Kinoshita's measure. Our goal is to determine whether the configuration of each subsystem is stable or not. Therefore, we define a subsystem as active if and only if its configuration is unstable. That is that the divergence of the density of the latest configurations and the density of previous configurations is greater than a certain threshold. That means that for the latest configurations values were chosen that follow a dis-tribution which significantly differs from the previous one.

system a at time t. We model ca,t as a random variable. Let M > 1 be the Let ca;t 2 Rnbe the n-dimensional configuration vector of the sub-

window size and L > 0 be the delay parameter. Using an appropriate density estimation technique, we define two probability densities DCa,t and DPa,t based on parts of the time series of ca,t:

DCa;t :¼ densityðca;t�m; ca;t�mþ1; …; ca;tÞ

DPa;t :¼ densityðca;t�l�m; ca;t�l�mþ1; …; ca;t�lÞ   
 DC is the density of the current window and DP the density of a previous window in the time series. To these two densities, we now apply the Kullback-Leibler divergence [29] measure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| KL2ðP; QÞ | ¼ | Z ∞ | pðxÞlog�pðxÞ��2 | (3) |
| � |
| Z ∞ | qðxÞlog�qðxÞ��2 |

and get a new time series

da;t :¼ KL2ðDCa;t; DPa;tÞ (4)

Finally, we define a subsystem a to be active at time t if da,t > ϵ for a threshold value ϵ. Otherwise, the subsystem is called inactive.

3.3. Finding the right parameters

Finding suitable values for the parameters M, L and ϵ is crucial for the significance of the calculated Kinoshita measure. A low ϵ will mark configuration changes as unstable which at a closer look are still acceptable as stable. On the other hand, too high values for M and L would mask short term occurrences of high amplitude changes. Our ex-periments show that good choices for these parameters highly depend on the actual SASO-System. In general, the configuration of these parame-ters is consequently application-specific and should be customised at runtime using hyper-parameter-tuning. However, we aim at a ‘good-enough’ reference configuration.

A first approach to find such acceptable values is to create a simula-tion of the SASO system and gather the configuration time series of the simulated subsystems. In the next step, change these data samples for some of the agents such that at controlled time points for a few steps high amplitude changes occur. Then we create an optimisation problem: Find the best values for M, L and ϵ such that the controlled changes create unique peaks in the Kinoshita measure.

Our experiments show that the best values vary depending on where we set the artificial changes. However, they vary only in a very limited

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ν1 |  | ξ2 | 1 |  | ξ | !2 | (2) | range. Therefore, a reasonable next step is to run this optimisation |
| several times with different points for the changes. This gives a distri- |
| t :¼ M | X | t�i� | M | X | t�i |  |  | bution for M, L and ϵ. The values that occur most often usually are in fact |
| good values for the actual evaluation. |

Kinoshita argues that unusual peaks in the time series of νt are an indi-cator of abnormal changes in the underlying distribution of active and inactive subsystems. Consequently, we aim at using this measure to detect such peaks that then serve as an indicator for undesired adaptation behaviour that needs to be suppressed.

3

M. Goller, S. Tomforde Array 11 (2021) 100069

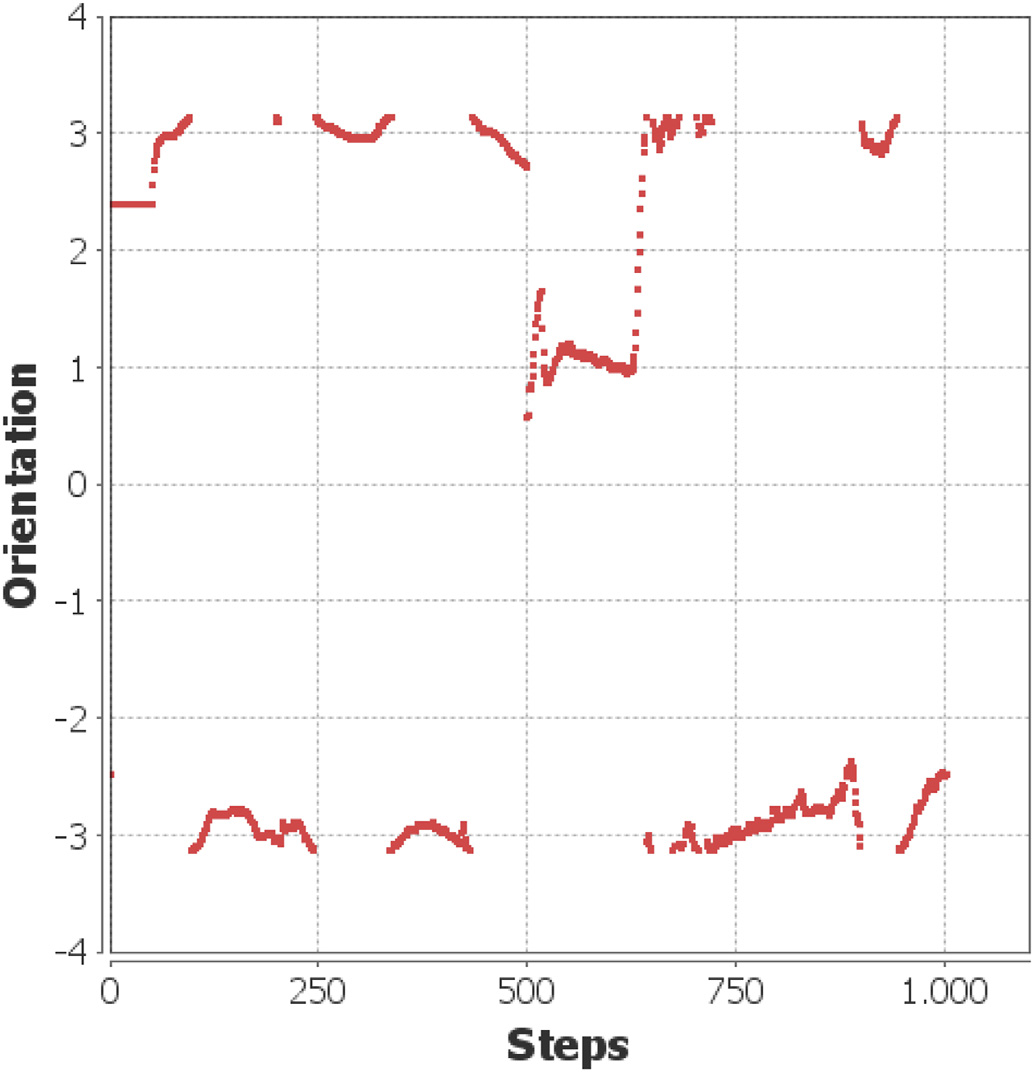


Fig. 2. The orientation time series of one of the affected birds. The values wrap around at � π. The disturbance is visible as the drop at t ¼ 500.

configuration of the system as unstable from the beginning. In fact, normal and abnormal adaptation behaviour cannot be distinguished in these scenarios, which renders our measure not applicable. However, current experiments show that this is not the case for the investigated scenarios (where future work aims also at defining the limits for the application).

4. Experiments and evaluation

We analyse the behaviour of the proposed measure in two artificial flocking [30] and two basic traffic simulations. All simulations are implemented in the MASON [31] simulation environment.

4.1. Flocks with external disturbance

This flocking simulation consists of 50 birds on a toroidal plane with random starting points and random initial orientations. We consider two cases: First a normal simulation and second a disturbed one. In both scenarios, the birds follow the usual rules of flocking:

● Alignment: A bird will align its direction with the average direction of its neighbours  
● Cohesion: A bird will steer towards the centre of all neighbouring birds  
● Avoidance: A bird will steer away from neighbours that are too close

For each of these rules, a direction vector vx is computed, weighted (with factor wx) and then added together. This sum is then normalised and eventually added to the current direction vcurrent vector of the given bird to determine it's direction vnext in the next time step:

vnext ¼ vcurrent þ normalise  
 � þw\* w\* alignmentvalignment

cohesionvcohesion   
 (5)

For the two simulations, the values are wcohesion ¼ walignment ¼ 1.0, þ w\* avoidancevavoidance�

4

M. Goller, S. Tomforde Array 11 (2021) 100069

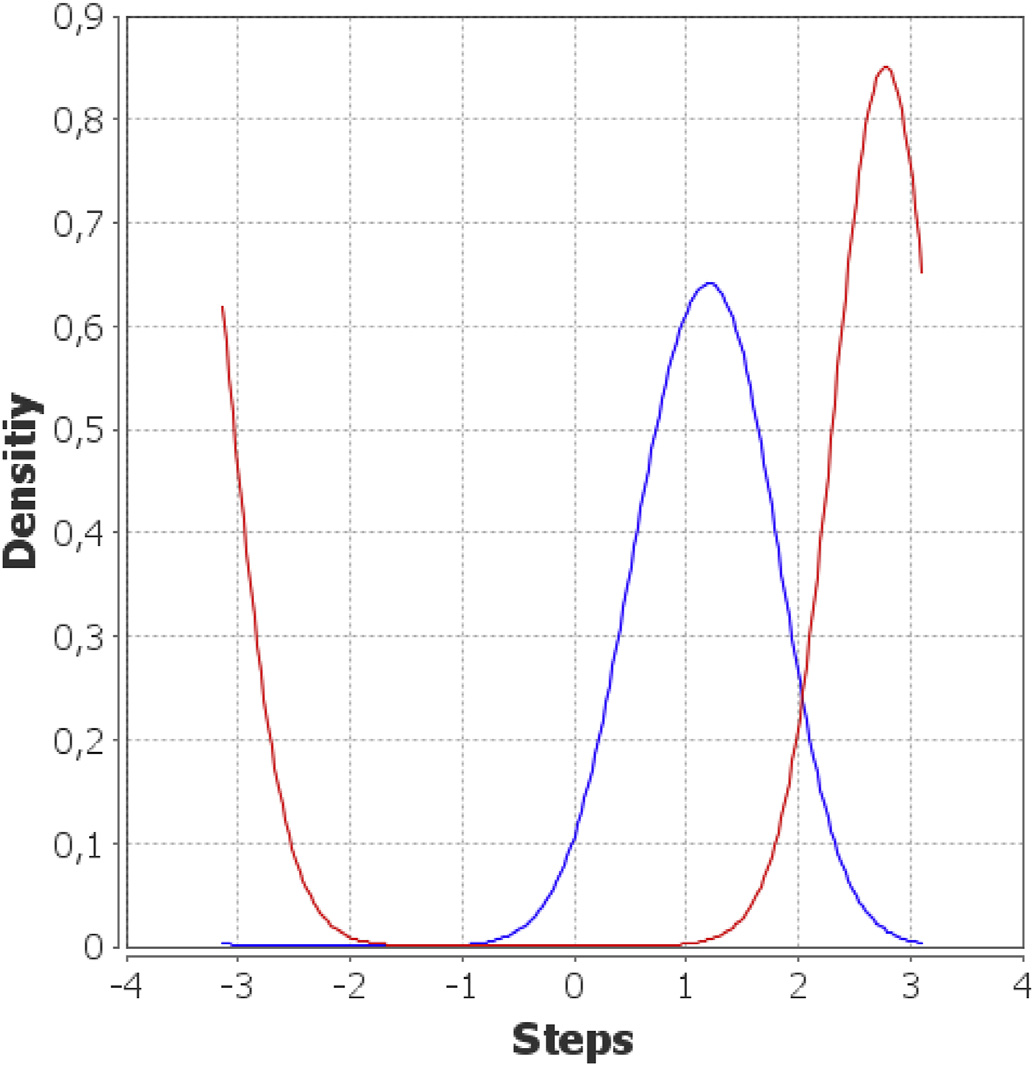


Fig. 5. Densities for the orientation distributions of bird from Fig. 2 in the

disturbed flocking simulation. Red: during t ¼ 480 to t ¼ 500, blue: during t ¼ 500 to t ¼ 520.

the orientation of one of the affected birds. Most of the time, this bird has

an orientation near � π, but when the disturbance occurs it changes it's orientation abruptly to a value near 1. It then takes the bird about 120

steps to realign to the majority of the flock.

To apply the Kinoshita measure, we need to fix values for M, L and ϵ.

For this, we put the undisturbed simulation into the heuristic from sec-

tion 3.3 and change the configuration values for 10% of the agents. From

t ¼ 200 to t ¼ 210 these agents get a constant configuration value of 1 and from t ¼ 300 to t ¼ 310 they get a value of 0. Therefore, we expect the heuristic to find values such that one peak at t ¼ 200 � L and one at t ¼ 300 � L is sufficiently prominent.

Fig. 3 shows results for a fixed ϵ ¼ 1. For M ¼ L ¼ 10 (red graph) we see the peaks at t ¼ 190 and t ¼ 290 but also two additional peaks at t ¼ 200 and t ¼ 210 which are almost the same height as the first one. M ¼ L ¼ 25 (grey graph) we have no sufficiently prominent peak at t ¼ 175. The blue graph shows the result for M ¼ L ¼ 20 which meets our expectation fairly well.Fig. 4 gives an impression on the effects of changing the

ϵ-parameter. A low ϵ allows too many agents to be counted as active, even

those who are not using artificially changed configuration data. There-

fore, the grey graph (ϵ ¼ 0.5 gives too many peaks. The blue graph shows the results for ϵ ¼ 1, where we can identify the desired peaks alongside a few smaller ones. The red graph shows ϵ ¼ 2. Here, no other peaks are present. We then repeat this process with other time points for the arti-

ficially changed data. Finally, the heuristic tells us that good values for M

and L are in fact near M ¼ L ¼ 20 and that ϵ � 1 is suitable. We keep in mind that a high value for ϵ could suppress important agents. Therefore,

we choose M ¼ L ¼ 20 and ϵ ¼ 1 for the evaluation of this simulation.

disturbed simulation, we need to calculate the distribution densities To determine whether the bird a is count as active at t ¼ 520 in the

DCa,520 ¼ density(ca,500, …, ca,520) and DPa,520 ¼ density(ca,480, …, ca,500) of its orientation. To do so, we use the common kernel density estimation

methods from Ref. [32]. Since the orientation is circular (wrapping

around at � π), we need to use a circular kernel function. In this case, we chose a von Mises kernel [33]. Fig. 5 shows the resulting densities with

the centre of distribution near 2.7 in the first time window and near 1.2 in

the second window.

5

M. Goller, S. Tomforde Array 11 (2021) 100069

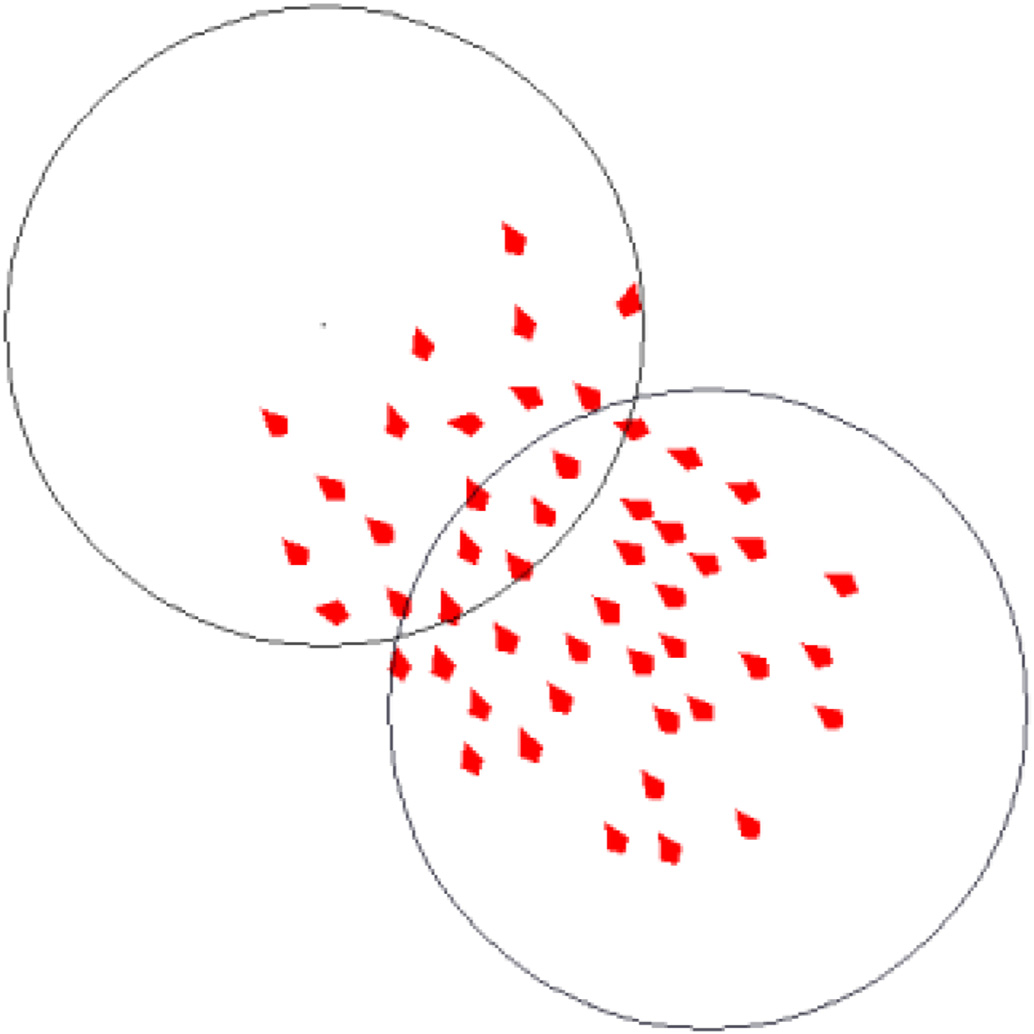


Fig. 8. The second flocking simulation at time step t ¼ 250. The agents are heading towards the upper left area which has a higher utility value than the lower right area since t ¼ 205.

undisturbed (blue) simulation. Except for minimal differences due to the arithmetic precision flaws of the Java double type [34] both graphs are identical until t ¼ 480. At that point, the time windows incorporate the disturbance. The disturbance creates a clear peak in the graph. Shortly after the disturbance, the birds group in three isolated flocks, two of them merge at t ¼ 900 which results in a second peak. Other than the activity factor, the fluctuation variance emphasises these two events.

4.2. Flocks with change of utility

In this simulation, we model 50 agents that again roam a toroidal plane with random starting points and random initial orientations. Next, we assign each point of the plane a utility value such that the utility values for all points in the plane are zero except for two circular, inter-secting areas. The agents follow the rule of avoidance as in the previous scenarios (but not alignment or cohesion, i.e. wcohesion ¼ walignment ¼ 0) and they can evaluate the utility value of their current position. The goal for each agent is to stay inside the area of the highest utility value. If it is not inside that area it will steer towards the centre of that area until the area is entered (see Fig. 8). This can be viewed as a flock of predators running around in an area with the best supply of prey or as a group of vacuum cleaning robots that clean the dirtiest area in the house.

In analogy to the real world, where the predators will decimate the prey and the robots will remove the dirt, we introduce a dynamic in the utility values. The values U for the two non-zero utility areas change over time following a sinus function with a constant phase p and the time step variable t:

UðtÞ :¼ ðsinð0:001\*p\*tÞ þ 1Þ=2

The phase parameters for the two areas are p ¼ 7 and p ¼ 9. Therefore, their utility values oscillate between 0 and 1 with different speeds. By this means, we force the agents to migrate to the other area which creates a notable overlay over the usual chaotic motion behaviour. At the beginning of the simulation, the lower right area is the one with the highest utility value. Within 150 steps, all agent have reached this area. At t ¼ 205 the upper left area will become the one with the higher value

6

M. Goller, S. Tomforde Array 11 (2021) 100069

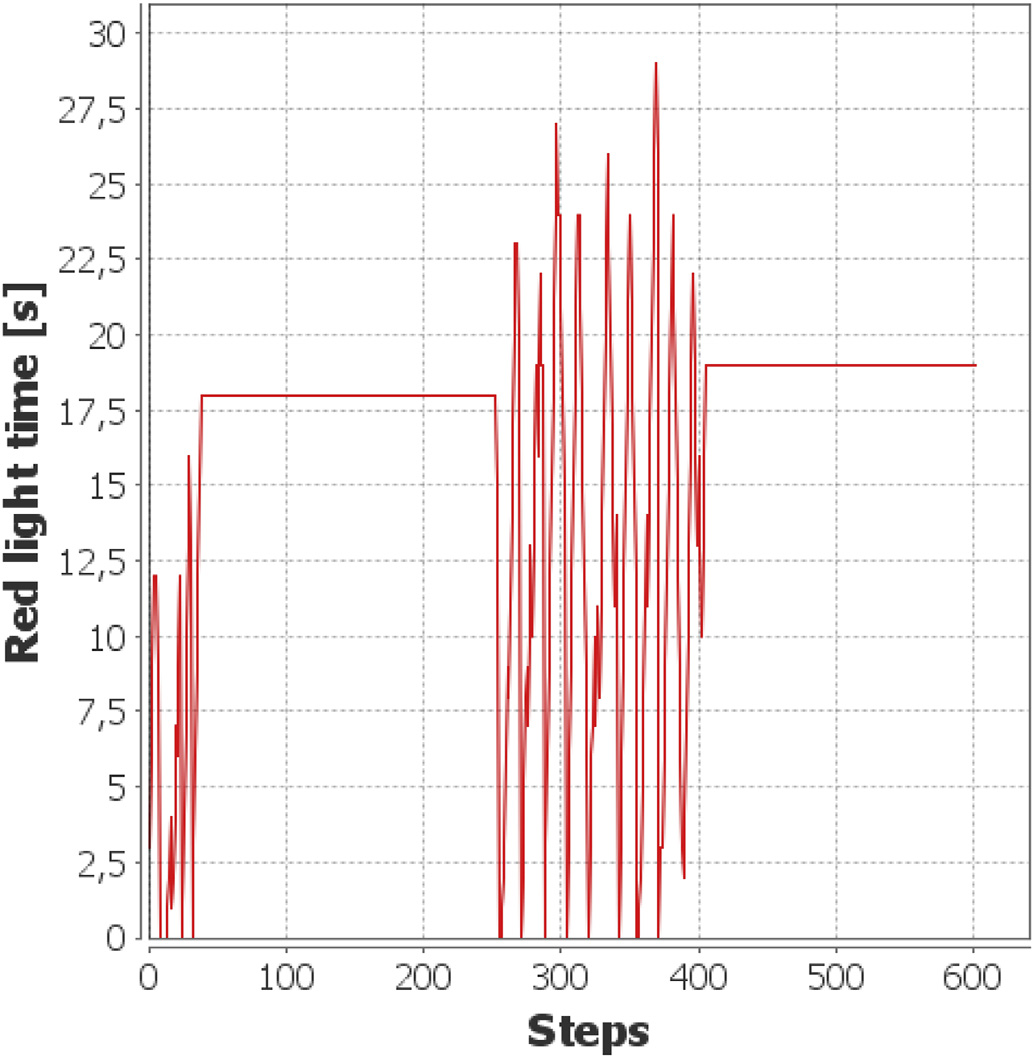


Fig. 11. The red light times at intersection D2 for the lane E2 → D2.

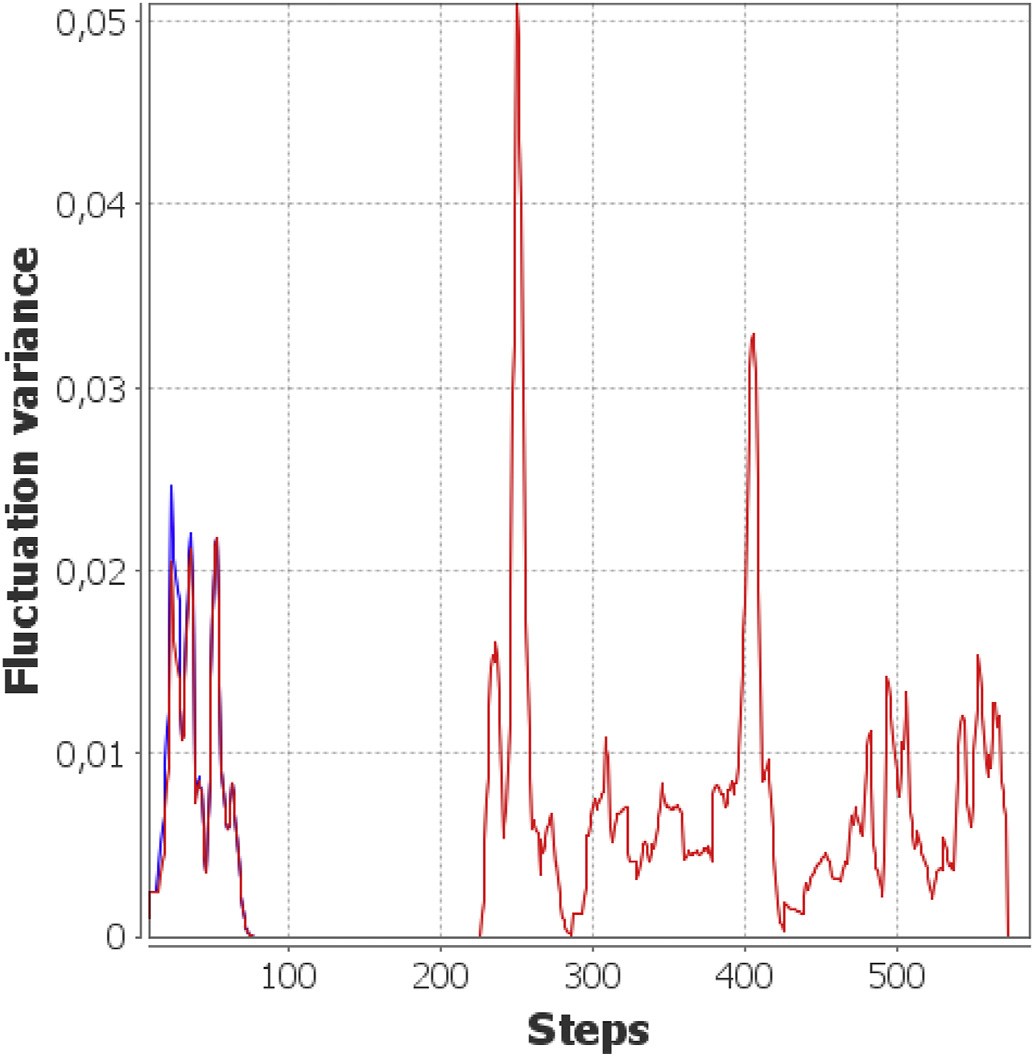


Fig. 12. The Kinoshita fluctuation variance of the first traffic simulation. Red: with a blocked intersection from t ¼ 250 to t ¼ 400, blue: undisturbed.

network.

Each intersection tries to minimise the waiting time for all cars at all incoming lanes by optimising the red light times for each lane. There are 250 cars in the simulation. They are part of the environment and not of the SASO system itself. Each car picks its destination randomly and chooses the shortest path to it. When it reaches the destination, the process is repeated and therefore all cars in motion constantly. In contrast to OTC system, the intersections of the SASO system in this simulation do not communicate with each other. Their reconfiguration results are only

7

M. Goller, S. Tomforde Array 11 (2021) 100069

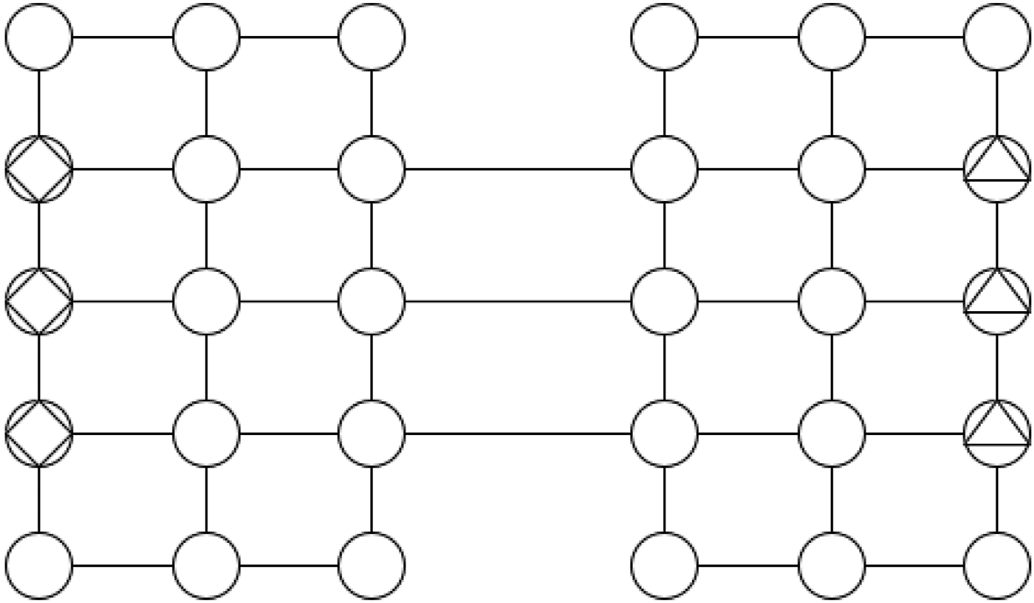


Fig. 14. The street network for the second traffic simulations. Home steads are marked with a square, work places with a triangle.

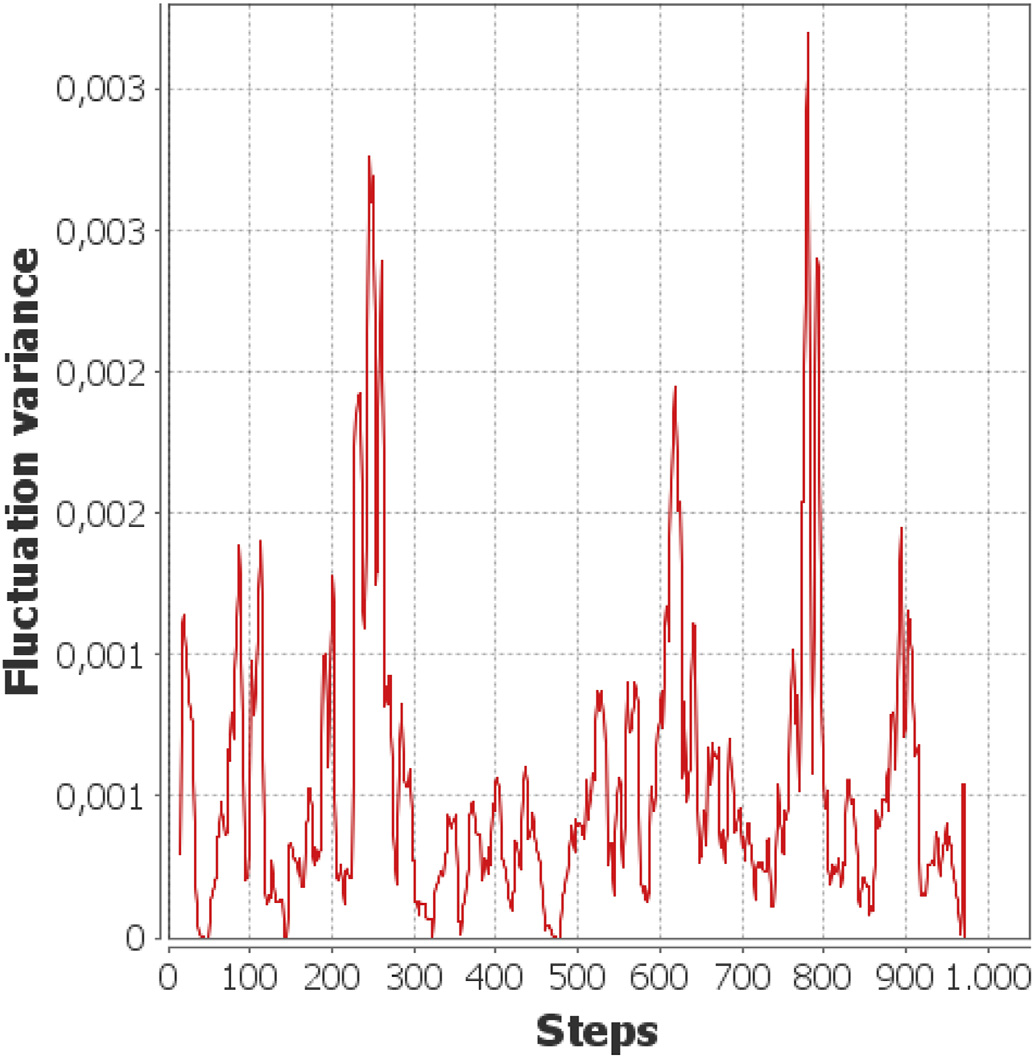


Fig. 15. The Kinoshita fluctuation variance of the rush hour simulation with increased traffic flow starting at t ¼ 250 and t ¼ 750.

4.4. Network of traffic lights with changes in the environment

t ¼ 750 this is repeated but this time the new cars will go from the workplaces towards home.

Fig. 15 shows the time series νt for this scenario with parameters M ¼ L ¼ 15 and ϵ ¼ 2. Since the random background traffic has no preferred routes, we see a more noisy time series with more peaks. The peaks occurring after t ¼ 250 and t ¼ 750 are due to the increased configuration changes that are caused by the higher traffic flow. These peaks are still clearly visible but are not as outstanding as in the previous simulations. The other peaks are a result of the random traffic. This shows that the measure is still applicable in such an environment although its perfor-mance to identify major events is not as good as in situations with less noise.

5. Conclusion

This article presented a step towards an integrated framework for the quantification of runtime SASO behaviour. Based on preliminary work on a probabilistic approach to assess a degree of self-adaption and self-organisation, this article presented a method to measure the stability of configurations of autonomous subsystems within an overall SASO constellation. Taking the design concepts of Organic Computing and Autonomic Computing as basis, we found our approach on the interface between adaption mechanism and productive system by assuming read access to the current configurations of the productive units. We showed that the proposed measure building upon the Kinoshita approach can be used to identify global anomalies in the system's configuration which serve as indicators for abnormal adaptation processes. The capability to identify such anomalies relies on the choices for the incorporated hyper-parameters.

The presented approach is a step towards the overall goal of devel-oping an integrated measurement framework for a continuous runtime analysis of SASO system properties. Future work follows two major research directions: On the one hand, we aim at an application to further, more complex scenarios of distributed SASO constellations and assess the behaviour of the measure. This includes an evaluation of this method in real-world applications and finding more constructive methods to iden-tify suitable parameters. On the other hand, we build on-top of the in-dicators provided by the measure to establish a runtime approach to using this information within the adaption logic, e.g., to better balance the trade-off between possible performance gain due to adaptation de-cisions and the impact on the stability and correspondingly the users’acceptance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

|  |  |
| --- | --- |
| We created this additional simulation to test the measures in a more noisy and larger environment where a smaller part of the subsystems | Acknowledgements |

experience a direct influence from external events. The scenario simu-lates rush hours in a city street network. The traffic lights and the cars operate in the same way as in the previous scenario, but we use a different street network. The network represents two islands each with a Manhattan-type network of sizes 3 by 5. The islands are connected with three bridges. The connections between two intersections provide one

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| This | research | is | partly | funded | by | the | Deutsche | For- |

schungsgemeinschaft (DFG, German Research Foundation) under grant TO 843/5-1 (project ‘InTURN’). The publication of this article was financially supported by DFG within the funding programme Open Ac-cess Publizieren. The authors acknowledge the support by the DFG.

|  |  |
| --- | --- |
| lane for each direction (see Fig. 14).  During the whole simulation, there are 250 cars driving around | References |

randomly. Furthermore, three intersections on one island are designated to be homesteads and three intersections on the other island are labelled as work places. At t ¼ 250 a total count of 500 new cars will appear randomly in the three homesteads and start driving towards one of the three workplaces. Therefore, the bridges have to handle an increasing flow from one side to the other in the following time steps. When the new cars arrive at their work places, they are removed from the simulation. At

8

M. Goller, S. Tomforde Array 11 (2021) 100069

[3] [Rajkumar R, Lee I, Sha L, Stankovic J. Cyber-physical systems: the next computing](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref3)

[4] [Atzori L, Iera A, Morabito G. The internet of things: a survey. Comput Network](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref4) [revolution. In: Design automation conference. IEEE; 2010. p. 731](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref3)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref4)[6](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref3).

[5] [Kounev S, Lewis P, Bellman K, Bencomo N, Camara J, Diaconescu A, Esterle L,](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref5) [2010;54(15):2787](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref4)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref5)[805](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref4).

[Geihs K, Giese H, et al. The notion of self-aware computing. In: Self-aware](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref5)

[6] [Bellman K, Tomforde S, Würtz R. Interwoven systems: self-improving systems](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref6) [computing systems. Springer; 2017. p. 3](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref5)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref6)[16](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref5).

[integration. In: 8th IEEE int. Conf. on self-adaptive and self-organizing systems](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref6)

[7] [Kephart J, Chess D. The vision of autonomic computing. IEEE Computer 2003;](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref7) [workshops; 2014. p. 123](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref6)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref7)[7](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref6).

[36(1):41](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref7)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref8)[50](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref7).

[8] [Müller-Schloer C, Tomforde S. Organic computing](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref8) [– techncial systems for survival in the real world, autonomic systems. Birkh](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref8)€a[user Verlag; 2017](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref8).

[9] [Weyns D, Schmerl B, Grassi V, Malek S, Mirandola R, Prehofer C, Wuttke J, Andersson J, Giese H, G](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref9)€o[schka K. On patterns for decentralized control in self-adaptive systems. In: Software engineering for self-adaptive systems II. Springer;](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref9)

[10] [D'Angelo M, Gerasimou S, Ghahremani S, Grohmann J, Nunes I, Pournaras E,](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref10) [2013. p. 76](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref9)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref10)[107](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref9).

[Tomforde S. On learning in collective self-adaptive systems: state of practice and a 3d framework. In: 2019 IEEE/ACM 14th international symposium on software](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref10)

[11] [Krupitzer C, Roth F, VanSyckel S, Schiele G, Becker C. A survey on engineering](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref11) [engineering for adaptive and self-managing systems. IEEE; 2019. p. 13](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref10)–[24](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref10).

[12] [Tomforde S, Kantert J, Müller-Schloer C, B](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref12)€o[delt S, Sick B. Comparing the effects of](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref12) [approaches for self-adaptive systems. Pervasive Mob Comput 2015;17:184](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref11)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref12)[206](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref11).

[disturbances in self-adaptive systems - a generalised approach for the quantif](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref12)i[cation](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref12)

[13] [Tomforde S, Goller M. To adapt or not to adapt: a quantif](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref13)i[cation technique for](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref13) [of robustness. Trans. Comput. Collect. Intell. 2018;28:193](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref12)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref13)[220](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref12).

[measuring an expected degree of self-adaptation. Computers 2020;9(1):21](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref13).

[14] M. Goller, S.Tomforde, Towards a continuous assessment of stability in (Self-) Adaptation behaviour, Proceedings of the 1st IEEE international conference on autonomic computing and self-organising systems workshops (ACSOS-W), 6th Workshop on Self-Aware Computing (SeAC20).

[15] [Tomforde S, Prothmann H, Branke J, H](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref15)€a[hner J, Mnif M, Müller-Schloer C, Richter U, Schmeck H. Observation and control of organic systems. In: Müller-Schloer C, Schmeck H, Ungerer T, editors. Organic computing - a paradigm shift for](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref15)

[16] Tomforde S, Sick B, Müller-Schloer C. Organic computing in the spotlight, CoRR [complex systems, autonomic systems. Birkh](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref15)€a[user Verlag; 2011. p. 325](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref15)–[38](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref15).

abs/1701.08125. <http://arxiv.org/abs/1701.08125>.

[17] [Tomforde S, Kantert J, Sick B. Measuring self-organisation at runtime - a](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref17)   
[quantif](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref17)i[cation method based on divergence measures. In: Proc. of 9th int. Conf. on agents and artif](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref17)i[cial intelligence; 2017. p. 96](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref17)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref18)[106](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref17).

[18] [Fisch D, J](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref18)€a[nicke M, Sick B, Müller-Schloer C. Quantitative emergence–a ref](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref18)i[ned approach based on divergence measures. In: IEEE int. Conf. on self-adaptive and](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref18)

[19] [Chan W. Interaction metric of emergent behaviours in agent simulations. In: Proc.](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref19) [self-organizing systems. IEEE; 2010. p. 94](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref18)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref19)[103](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref18).

[20] [Raibulet C, Masciadri L. Metrics for the evaluation of adaptivity aspects in software](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref20) [of the winter sim. Conf.; 2011. p. 357](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref19)[–](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref20)[68](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref19).

[systems. Int. J. Adv. Softw. 2010;3(1](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref20) & [2):238](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref20)–[51](http://refhub.elsevier.com/S2590-0056(21)00017-5/sref20).

9