Formal Methods in System Design (2020) 55:171–221
https://doi.org/10.1007/s10703-020-00347-z

Multi-scale verification of distributed synchronisation

Paul Gainer[1] · Sven Linker[1] · Clare Dixon[1,2] · Ullrich Hustadt[1] · Michael Fisher[1]

Published online: 20 September 2020

Abstract
Algorithms for the synchronisation of clocks across networks are both common and important within distributed systems. We here address not only the formal modelling of these algorithms, but also the formal verification of their behaviour. Of particular importance is the strong link between the very different levels of abstraction at which the algorithms may be verified. Our contribution is primarily the formalisation of this connection between individual models and population-based models, and the subsequent verification that is then possible. While the technique is applicable across a range of synchronisation algorithms, we particularly focus on the synchronisation of (biologically-inspired) pulse-coupled oscillators, a widely used approach in practical distributed systems. For this application domain, different levels of abstraction are crucial: models based on the behaviour of an individual process are able to capture the details of distinguished nodes in possibly heterogenous networks, where each node may exhibit different behaviour. On the other hand, collective models assume homogeneous sets of processes, and allow the behaviour of the network to be analysed at the global level. System-wide parameters may be easily adjusted, for example environmental factors inhibiting the reliability of the shared communication medium. This work provides a formal bridge across the "abstraction gap" separating the individual models and the population-based models for this important class of synchronisation algorithms.

Keywords Synchronisation · Pulse-coupled oscillators · Abstraction · Probabilistic verification · Weak bisimulation

1 Introduction

Small computing devices comprising networks, be it commercial wireless sensor networks, or communicating devices in the Internet of Things, are becoming increasingly common. However, to enable these devices to communicate efficiently, they have to employ methods to use the shared communication medium while avoiding conflicting messages on this medium,

This work was supported by the Sir Joseph Rotblat Alumni Scholarship at Liverpool and the Engineering and Physical Sciences Research Council, under Grants EP/N007565/1 (S4: Science of Sensor Systems Software), EP/L024845/1 (Verifiable Autonomy), and the FAIR-SPACE (EP/R026092/1) and RAIN (EP/R026084/1) RAI Hubs.

Extended author information available on the last page of the article

in particular in the form of collisions. Collisions occur if two or more devices simultaneously try to access the communication medium, and often result in neither message being delivered. Several protocols to organise shared medium access have been developed and analysed [1,51]. These protocols typically identify a common time frame and divide this frame into slots associated to each node. Thus every node has an allocated time slot that it may use to send its messages onto the shared medium.

Such an approach introduces the need for a common clock between the nodes, i.e., they need to synchronise. A valuable approach to achieve synchrony of nodes is the implementation of biologically-inspired *pulse-coupled oscillators* (PCOs) [38]. A network of PCOs synchronises in the following way: all oscillators have a similar *clock cycle* at the end of which they *fire*. That is, they transmit a broadcast message which is received by all oscillators in their communication range. These oscillators then adjust their own position within their clock cycle according to a *phase response function*. Depending on the concrete implementation, they may move their current position within the clock cycle closer to its end, or closer to its start.

Most analyses of the synchronisation behaviour of PCOs are concerned with continouous clock cycles, i.e., where clocks take real values from the interval [0, 1]. However, the smaller devices get, the more important it is to save memory and computing time for such a low-level functionality. Even a floating point number may need too much memory, compared to an implementation with, for example, a four-bit vector. Hence, in previous work, we chose to analyse the behaviour of *discrete time PCOs* [24].

In contrast to continuous time PCOs, networks of discrete time PCOs are not always guaranteed to synchronise. Instead, whether they synchronise or not depends on the type of coupling between the oscillators and their common phase-response function. We analysed the behaviour of such networks for different parameters via model-checking, to check both qualitatively for which parameters the networks synchronise, as well as quantitatively for how long they need to achieve a synchronised state and how much energy is used to achieve this [25]. In the context of large numbers of single oscillators, for example in the context of wireless sensor networks, the well-known state-space explosion problem of the model-checking approach is extremely important [14]. We formalised a network of oscillators as *population models* [20] which exploit the behavioural homogeneity of the nodes to encode the global state efficiently. This allows the network size to be increased above what would be feasible when distinguishing each node, with the restriction that only fully-connected networks, where all sensors can communicate with all other sensors, can be modelled. But the construction of a population model from a given oscillator specification is not straightforward, and in particular, it is not obvious whether the constructed population model correctly reflects the behaviour of the oscillators. This results in an 'abstraction gap': after abstracting into populations, how can we be sure that the abstraction process was correct and that the results of verification of population models actually hold for the concrete models on which they are based?

In this paper, we remedy this lack of certainty, by proving the correspondence of our population model with an explicit formalisation of the oscillators. To that end, we present the concrete oscillator model as well as its formalisation as a discrete-time Markov chain. Subsequently we describe the corresponding population model, and show how we can, in addition to the abstraction created by the populations, reduce the state space even further to facilitate the analysis. Finally, we prove that the behaviour of a network of concrete oscillators and the population model are probabilistically weakly bisimilar. We cannot prove a one-to-one correspondence, since the concrete model implicitly includes the possibility of

identifying individual oscillators, which is exactly what the population model abstracts from. In particular, our contributions are:

- the definition of a model for fully-connected networks of pulse-coupled oscillators (Sect. 4),
- the detailed definition of a population model (Sect. 5), based on previous work [24,25],
- a way of reducing the size of the formal population models (Sect. 5.5),
- a proof that these two models are probabilistic weak-bisimilar (Theorem 3), and
- an evaluation of synchronisation behaviour using probabilistic model-checking (Sect. 7).

The paper is structured as follows. In Sect. 2, we review a selection of related work, both for models of pulse-coupled oscillators, as well as approaches for their verification. After an introduction of preliminary notions in Sect. 3, we present the concrete model of single oscillators as a discrete-time Markov chain in Sect. 4. The abstract model in terms of population models and proofs about their properties are contained in Sect. 5. In Sect. 6, we prove the correspondence between the two types of models. The experimental evaluation of synchronisation behaviour is presented in Sects. 7, and 8 concludes the paper.

2 Related work

The canonical model of pulse-coupled oscillators, and their synchronisation, was formulated by Mirollo and Strogatz [38], and based on Peskin's model of a cardiac pacemaker [43]. Here the progression of an oscillator through its oscillation cycle is given by a real value in the interval [0, 1]. Mirollo and Strogatz proved that with a convex phase response function, a network of mutually coupled oscillators always converges, i.e., their position within the oscillation cycle eventually coincides. Such a model has been shown to be applicable to the clock synchronisation of wireless sensor nodes [47] and swarms of robots [41].

Synchronisation algorithms based on pulse-coupled oscillators are often beneficial in unreliable, decentralised networks, where other synchronisation algorithms are not appropriate. For example, the Flooding Time Synchronisation Protocol (FTSP) [36] requires the use of an arbitrary root node. In situations where the root becomes unavailable due to communication failure or power outage, FTSP will have to assign another root node. When implemented on unreliable, decentralised networks, FTSP may spend considerable resources on repeatedly assigning root nodes, which may slow down or prevent synchronisation [11]. Other algorithms such as the Berkeley algorithm [26] and Cristian's algorithm [16] require the use of centralised time servers, which is problematic for unreliable, decentralised networks.

Several decentralised network algorithms for synchronisation are based on pulse-coupled oscillators [47,50]. For example, the Gradient Time Synchronisation Protocol (GTSP) by Sommer and Wattenhofer [46] achieves synchronisation by having nodes send their current clock value to their neighbours. Each node then calculates the average of the clock values received and its own clock value. This process is then repeated to maintain synchronisation. Another approach to synchronisation, the Pulse-Coupled Oscillator Protocol [39], makes use of refractory periods after sending messages containing time information. During the refractory period, no more messages are sent, which reduces network bandwidth and energy usage. A similar approach is used in the FiGo protocol [11], which combines biologically inspired synchronisation with information distribution via gossiping. All of these approaches use different phase response functions.

In general, synchronisation algorithms based on PCOs are more robust for unreliable networks, as they do not require centralised nodes and can work with only partial network

connectivity [11]. They are particularly useful for battery-powered nodes in wireless networks, as the node can be placed in a low-power mode during the refractory period, thus reducing energy usage. (The clock keeps ticking even in low-power mode, thanks to the design of microcontrollers such as the 'Atmel ATmega128L' [6].)

Synchronisation of clocks for networks of nodes has been investigated from different perspectives. Heidarian et al. [29] analysed the behaviour of a synchronisation protocol based on time allocation slots for up to four nodes and different topologies, from fully connected networks to line topologies. They modelled the protocol as timed automata [3], and used the model-checker UPPAAL [10] to examine its worst-case behaviour. Their model is based on continuous time, and in particular, they did not model pulse-coupled oscillators.

Bartocci et al. [8] described pulse-coupled oscillators as extended timed automata with suitable semantics to model their peculiarities. They defined a dedicated logic to analyse the behaviour of a network of such automata along traces, and used a pacemaker as a case study to verify the eventual synchronisation and the time needed to achieve this.

Our models and methods are different from all of these approaches. A key difference in our work from that of others analysing PCOs is that we define the oscillation cycle to consist of discrete steps. To the best of our knowledge, with the exception of the paper by Webster et al. (including some of the authors of this paper) [49] and our previous work [24,25], there is no other work concerned with PCOs with discrete oscillation cycles. Furthermore, all of these approaches distinguish between single oscillators in the network, while the properties of interest relate to global behaviour. This discrepancy between local modelling and global analysis restricts the size of networks that can be analysed, due to the state-space explosion. To extend the size of analysable networks, we employ *population models*, a counting-abstraction of such networks [19]. Instead of identifying each oscillator on its own, we record how many oscillators are in each step of the oscillation cycle. This significantly reduces the state-space by exploiting the symmetries in the model [20], and we are hence able to extend the size of networks.

The notion of population models should not be confused with *population protocols* [5], a formalism to express distributed algorithms. In contrast to our setting, communication in population protocols is always between two agents, where one agent initiates the communication and the other responds. Furthermore, even though the agents cannot identify the other agents in the network, within the global model each agent is uniquely associated with a state. In our model, we cannot distinguish between two different agents sharing the same state, even at the global level. Finally, our oscillators may change their state without interacting with other oscillators, while the agents in a population protocol must communicate with another agent to change their internal state.

Other techniques have been used to model populations of processes. For example population-based models using PEPA, a stochastic process algebra, are discussed in [30]. The modelling of individuals using PEPA is introduced and if the identification of individuals is not necessary (similar to our work) a population-based approach is advocated to allow larger populations to be modelled. Unlike our approach the population-based models make use of a continuous approximation of the discrete behaviour.

Chemical Reaction Networks (CRNs), see for example [12,45] have been used to represent the behaviour of reactions between chemicals in a solution. These have been provided with different semantics including both deterministic and stochastic semantics. CRNs have been used to model asynchronous logic circuits with properties of the models being verified using the probabilistic model checker PRISM [13]. They have also been investigated to analyse their capacity to represent discrete probability distributions focusing on their steady state [12]. In our work we model both a fully connected network of oscillators and the population

models *directly* as stochastic processes, in particular discrete-time Markov chains. We focus on synchronisation properties rather than their steady state. Like [13] we use PRISM to verify these properties.

Similarly to typical definitions of counter abstractions [9,21], we use counters to model concurrent entities that are indistinguishable for our purposes. For example, to analyse the probability of eventually reaching a synchronised state, we are not interested in an order of oscillators, which would be artificial anyway. However, in contrast to these approaches, we do not include means to introduce new oscillators into a model. That is, the values within our population models are naturally bounded by the number of oscillators within the network.

3 Preliminaries

In this section we define *discrete-time Markov chains* (DTMCs), stochastic processes with discrete state space and discrete time, and introduce Probabilistic Computation Tree Logic (PCTL), a logic that can be used to reason about probabilistic reachability and rewards in these processes.

Throughout this paper, we use the notation $f \oplus [x \mapsto y]$, where f is a function, to express *updating f at x* by y. That is, the function that coincides with f, except for x, where it takes the value y.

3.1 Discrete-time Markov chains

DTMCs can be used to model systems where the discrete-time evolution of the system can be represented by a discrete probabilistic choice over several outcomes at each step.

Definition 1 A discrete-time Markov chain D is a tuple $(\mathcal{S}, \sigma_I, \mathbf{P}, L)$ where $\mathcal{S}$ is a finite set of states. σ_I is the initial state, and $L : \mathcal{S} \to \mathbb{P}(\mathcal{L})$ is a labelling function that assigns properties of interest from a set of labels $\mathcal{L}$ to states. $\mathbf{P} : \mathcal{S} \times \mathcal{S} \to [0, 1]$ is the *transition probability matrix* subject to $\sum_{\sigma' \in \mathcal{S}} \mathbf{P}(\sigma, \sigma') = 1$ for all $\sigma \in \mathcal{S}$, where $\mathbf{P}(\sigma, \sigma')$ gives the probability of transitioning from σ to σ'. We say that there is a *transition* between two states $\sigma, \sigma' \in \mathcal{S}$ if $\mathbf{P}(\sigma, \sigma') > 0$.

Intuitively, a DTMC is a state transition system where transitions between states are labelled with probabilities greater than 0 and where states are labelled with properties of interest. An execution *path* ω of a DTMC $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ is a non-empty finite, or infinite, sequence $\sigma_0\sigma_1\sigma_2 \cdots$ where $\sigma_i \in \mathcal{S}$ and $\mathbf{P}(\sigma_i, \sigma_{i+1}) > 0$ for $i \geqslant 0$. We denote the set of all paths starting in state σ by $Paths^D(\sigma)$, and the set of all finite paths starting in σ by $Paths^D_f(\sigma)$. For paths where the first state along that path is the initial state σ_I we will simply use $Paths^D$ and $Paths^D_f$. Furthermore, we will use $Paths$ and $Paths_f$ if D is clear from the context. For a finite path $\omega_f \in Paths_f(\sigma)$ for some state σ, the *cylinder set* of ω_f is the set of all infinite paths in $Paths(\sigma)$ that share ω_f as a prefix. The probability of taking a finite path $\sigma_0\sigma_1 \cdots \sigma_n$ is given by $\prod_{i=1}^{n} \mathbf{P}(\sigma_{i-1}, \sigma_i)$. This measure over finite paths can be extended to a probability measure Pr_σ over the set of infinite paths $Paths(\sigma)$, where the smallest sigma-algebra over $Paths(\sigma)$ is the smallest set containing all cylinder sets for paths in $Paths_f(\sigma)$. For the probability measure over paths where the first state is the initial state σ_I we will simply use Pr. For a detailed description of the construction of the probability measure we refer the reader to [31].

3.2 Probabilistic computation tree logic

Probabilistic Computation Tree Logic [28] (PCTL) is a probabilistic extension of the temporal logic CTL. Properties for DTMCs can be formulated in PCTL and then checked against the DTMCs using *model checking*.

Definition 2 The syntax of PCTL is given by:

$$\Phi = p \mid \neg\Phi \mid (\Phi \wedge \Phi) \mid \mathrm{P}_{\bowtie\lambda}[\Psi]$$
$$\Psi = \Phi \ \mathrm{U}\ \Phi$$

where p is an atomic proposition taken from the set of labels $\mathcal{L}$, $\bowtie \in \{<, \leqslant, \geqslant, >\}$ and $\lambda \in [0, 1]$.

Formulas denoted by Φ are *state formulas* and formulas denoted by Ψ are *path formulas*. A PCTL formula is always a state formula, and a path formula can only occur inside the P operator. We now give the semantics of PCTL over a DTMC.

Definition 3 Given a DTMC $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$, we inductively define the satisfaction relation $\models$ for any state $\sigma \in \mathcal{S}$ as follows:

$$\begin{array}{lll}
\sigma \models p & \text{iff} & p \in L(\sigma) \\
\sigma \models \neg\Phi & \text{iff} & \sigma \not\models \Phi \\
\sigma \models (\Phi \wedge \Phi') & \text{iff} & \sigma \models \Phi \text{ and } \sigma \models \Phi' \\
\sigma \models \mathrm{P}_{\bowtie\lambda}[\Psi] & \text{iff} & Pr\{\omega \in Paths(\sigma) \mid \omega \models \Psi\} \bowtie \lambda
\end{array}$$

where $p \in \mathcal{L}$, and for any path $\omega = \sigma_0\sigma_1\sigma_2 \cdots$ of D as follows:

$$\omega \models \Phi \ \mathrm{U}\ \Phi' \qquad \text{iff} \quad \text{there exists } i \in \mathbb{N} \text{ s.t. } \omega_i \models \Phi' \text{ and for all } j < i . \omega_j \models \Phi .$$

Disjunction, *true*, *false*, and implication are derived as usual, and we define eventuality as $\mathrm{F}\ \Phi \equiv \mathit{true}\ \mathrm{U}\ \Phi$. When model checking any PCTL formula of the form $\mathrm{P}_{\bowtie\lambda}[\Psi]$ the actual probability is first calculated and then compared to the bound $\bowtie \lambda$ [35]. We will denote this calculated probability value by $\mathrm{P}_{=?}[\Psi]$.

While probabilistic reachability properties allow us to quantitatively analyse models with respect to the likelihood of reaching some set of states, they do not allow us to reason about other properties of interest, for instance the expected time taken for a network to synchronise [24], or the expected energy consumption of a network [25]. Therefore, we will often want to augment the DTMC corresponding to a population model with rewards. We do this by annotating states and transitions with real-valued rewards (respectively costs, should values be negative) that are awarded when states are visited, or transitions taken.

Definition 4 Given a DTMC $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ a *reward structure* for D is a pair $\mathcal{R} = (R_s, R_t)$ where $R_s : \mathcal{S} \to \mathbb{R}$ and $R_t : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ are the *state reward* and *transition reward* functions that respectively map states and transitions in D to real valued rewards.

For any finite path $\omega = \sigma_0 \cdots \sigma_k$ of D we define the total reward accumulated along that path up to, but not including, σ_k as

$$\mathrm{tot}_{\mathcal{R}}(\sigma_0 \cdots \sigma_k) = \sum_{i=0}^{k-1} (R_s(\sigma_i) + R_t(\sigma_i, \sigma_{i+1})) . \tag{1}$$

Given a DTMC $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ augmented with a reward structure $\mathcal{R}$, and some state $\sigma_0 \in \mathcal{S}$, we will often want to reason about the reward that is accumulated along a path $\omega = \sigma_0\sigma_1\sigma_2 \cdots \in Paths(\sigma_0)$ that eventually passes through some set of target states $\Omega \subset \mathcal{S}$. We first define a random variable over the set of infinite paths starting in state σ_0, $V_{\Omega,\sigma_0} : Paths(\sigma_0) \to \mathbb{R} \cup \{\infty\}$. If $\sigma_0 = \sigma_I$ then we will simply use V_Ω. Given the set $\omega_\Omega = \{j \mid \sigma_j \in \omega \cap \Omega\}$ of indices of states in ω that are in Ω we define the random variable

$$V_{\Omega,\sigma_0}(\omega) = \begin{cases} \infty & \text{if } \omega_\Omega = \emptyset \\ \mathrm{tot}_{\mathcal{R}}(\sigma_0 \cdots \sigma_k) & \text{otherwise, where } k = \min \omega_\Omega, \end{cases}$$

and define the expectation of V_{Ω,σ_0} with respect to Pr_{σ_0} by

$$E[V_{\Omega,\sigma_0}] = \int_{\omega \in Paths(\sigma_0)} V_{\Omega,\sigma_0}(\omega)\, dPr = \sum_{\omega \in Paths_{\sigma_0}} V_{\Omega,\sigma_0}(\omega) Pr\{\omega\}.$$

The logic of PCTL can be extended to include reward properties by introducing the state formula $\mathrm{R}_{\bowtie r}[\mathrm{F}\ \Phi]$, where $\bowtie \in \{<, \leqslant, \geqslant, >\}$ and $r \in \mathbb{R}$ [33]. Given a state $\sigma \in \mathcal{S}$, a real value r, and a PCTL path formula Φ, the semantics of this formula is given by

$$\sigma \models \mathrm{R}_{\bowtie r}[\mathrm{F}\ \Phi] \quad \text{iff} \quad E[V_{Sat(\Phi)}] \bowtie r,$$

where $Sat(\Phi)$ denotes the set of states in $\mathcal{S}$ that satisfy Φ. Similarly to the operator P, for any PCTL formula of the form $\mathrm{R}_{\bowtie r}[\Psi]$ we will denote the calculated expected value by $\mathrm{R}_{=?}[\Psi]$.

4 Concrete model of a network of pulse-coupled oscillators

In this section we give a brief introduction to the formal model of a single pulse-coupled oscillator, as originally presented in previous work [24]. Subsequently, we encode fully-coupled networks of such oscillators as discrete time Markov chains.

4.1 Pulse-coupled oscillator model

A model of a pulse-coupled oscillator is composed of two features. Firstly, we need to model the *oscillation*, a periodic variation of the state of the oscillator. In our model the *phase* of an oscillator indicates its progression through an oscillation cycle, which is divided into discrete steps. We assume that the oscillation frequency is the same for all oscillators; the internal clocks of all oscillators are running at the same speed. Secondly, the model must encapsulate the interactions between the oscillators. Oscillators that are *pulse-coupled* interact with each other at discrete times during their oscillation cycles. At some distinguished point an oscillator transmits a message to other oscillators that react to the reception of the message by adjusting their phase. We assume that the duration of time corresponding to an increment of 1 to the phase of an oscillator is long enough to perceive all messages coming from other oscillators. That is, the only way a message may be lost is if the sending oscillator fails to transmit its message.

The *phase* of an oscillator u at time t is denoted by $\phi_u(t)$. The phase of each u progresses through a sequence of discrete integer values bounded by some $T \geqslant 1$, its maximal phase. The phase progression over time of a single uncoupled oscillator is determined by the successor function, where the phase increases over time until it exceeds T, at which point the oscillator will fire in the next moment in time and the phase will reset to one. The phase progression

of an uncoupled oscillator is therefore cyclic with period T, and we refer to one cycle as an *oscillation cycle*.

When an oscillator fires, it may happen that its firing is not perceived by any of the other oscillators coupled to it. We call this a *broadcast failure* and denote its probability by $\mu \in [0, 1]$. Note that μ is a global parameter, hence the chance of broadcast failure is identical for all oscillators. When an oscillator fires, and a broadcast failure does not occur, it perturbs the phase of all oscillators to which it is coupled; we use $\alpha_u(t)$ to denote the number of all other oscillators that are coupled to u and will fire at time t.

Definition 5 The *phase response function* is a positive increasing function $\Delta : \{1, \ldots, T\} \times \mathbb{N} \times \mathbb{R}^+ \to \mathbb{N}$ that maps the phase of an oscillator u, the number of other oscillators perceived to be firing by u, and a real value defining the strength of the coupling between oscillators, to an integer value corresponding to the perturbation to phase induced by the firing of oscillators where broadcast failures did not occur. We require $\Delta(\Phi, 0, \epsilon) = 0$ for all possible phase response functions, that is, oscillators are only perturbed if they perceive at least one other firing oscillator.

We can introduce a *refractory period* into the oscillation cycle of each oscillator. A refractory period is an interval of discrete values $[1, R] \subseteq [1, T]$ where $R \leqslant T$ is the size of the refractory period, such that if $\phi_u(t)$ is inside the interval, for some oscillator u at time t, then u cannot be perturbed by other oscillators to which it is coupled. If $R = 0$ then we set $[1, R] = \emptyset$, and there is no refractory period at all.

Definition 6 The *refractory function* $\mathrm{ref} : \{1, \ldots, T\} \times \mathbb{N} \to \mathbb{N}$ is defined as $\mathrm{ref}(\Phi, \delta) = \Phi$ if $\Phi \in [1, R]$, or $\mathrm{ref}(\Phi, \delta) = \Phi + \delta$ otherwise, and takes as parameters δ, the degree of perturbance to the phase of an oscillator, and Φ, the phase, and returns Φ if it is in the refractory period, or $\Phi + \delta$ otherwise.

The phase evolution of an oscillator u over time is then defined as follows, where the *update function* and *firing predicate*, respectively denote the updated phase and firing of oscillator u,

$$\begin{aligned} \mathit{update}_u(t) &= 1 + \mathrm{ref}(\phi_u(t), \Delta(\phi_u(t), \alpha_u(t), \epsilon)), \\ \mathit{fire}_u(t) &= \mathit{update}_u(t) > T, \\ \phi_u(t+1) &= \begin{cases} 1 & \text{if } \mathit{fire}_u(t) \\ \mathit{update}_u(t) & \text{otherwise.} \end{cases} \end{aligned}$$

For real deployments of synchronisation protocols it is often the case that the duration of a single oscillation cycle will be at least several seconds [15,42]. The perturbation induced by the firing of a group of oscillators may lead to groups of other oscillators to which they are coupled firing in turn. The firing of these other oscillators may then cause further oscillators to fire, and so forth, leading to a "chain reaction", where each group of oscillators triggered to fire is *absorbed* by the initial group of firing oscillators. Since the whole chain reaction of absorptions may occur within just a few milliseconds, and in our model the oscillation cycle is a sequence of discrete states, when a chain reaction occurs the phases of all perturbed oscillators should be updated in one single time step.

4.2 Modelling the network as a DTMC

In this section, we present our model of a fully-connected network of pulse-coupled oscillators. Observe that by the definition of single PCOs, the reaction of an oscillator u to incoming

communication of other oscillators depends on the *number* of oscillators communicating and the implementation of the perturbation function. We choose to present the full semantics of such a network, instead of defining it as the parallel composition of single oscillators. After presenting the semantics, we will discuss this decision.

We model the whole, fully-connected network of oscillators as a single DTMC $D = (\mathcal{S}, s_I, \mathbf{P}, L)$, where each state $s \in \mathcal{S}$ denotes a global state of the network, with the exception of s_I, which is a distinguished initial state. More precisely, each state of the DTMC contains the internal states of each oscillator, as well as an abstraction of the environment.

We model each transition of an oscillator as a single transition within the DTMC. However, since the oscillators may influence each other within a single time step (that is, when they are firing), we cannot simply allow for arbitrary sequences of transitions. For instance, as stated in Sect. 4.1, all oscillators run with the same clock-speed. Hence we need to prevent a single oscillator from taking a transition and thus progressing its phase without giving the other oscillators a chance to do the same. We achieve this by the following means:

- we divide the internal computation of each oscillator into two modes: *start* and *update*, and
- we add a counter to the model, containing the number of oscillators that fire.

The counter also possesses both modes, and resets at the start of each "round" of computation. First, in the *start* mode, each oscillator checks whether it would fire, according to its phase response function and the current number of oscillators that already fired, as given by the counter. If it does, it increases the counter and updates its mode to *update*, otherwise it just updates its mode. If all oscillators are in the update mode, they compute their new phases in a single step, according to the phase response function and the current state of the environment counter. Furthermore, we impose an order on the evaluation of the oscillators in the start mode if at least one oscillator fires, starting from the highest phase to the lowest. In this way, we model that firing oscillators are always perceived by the other nodes, and thus may lead to the firing of the latter. In particular, this reflects the *absorptions* of messages as defined in the previous section.

The general idea of the progress of the network of oscillators is visualised in Fig. 1. In the figure, each rounded rectangle shows a state of a network of four oscillators. The circles represent the nodes, where we inscribe their current phases and an abbreviation of their modes. A node that is about to fire is indicated by a starred circle, while a shaded circle indicates a node that is within the refractory period. The rectangle denotes the environment counter, with its corresponding value and mode. The phase response function is $\Delta(\Phi, \alpha, \epsilon) = [\Phi \cdot \alpha \cdot \epsilon]$, where $[\cdot]$ denotes rounding to the nearest integer. We set $\epsilon = 0.3$, and $\mu = 0.2$.

In the first state, all outgoing transitions only check whether to increase the counter. Since no oscillator is in the firing phase, all oscillators just update their mode (observe that the single arrow actually denotes four transitions). In the next step, all oscillators increase their phase by one, and reset their mode to *start*. In the next four transitions, oscillator 2 fires and increases the counter, which in turn is sufficient for oscillator 3 to fire as well, since $7 + [7 \cdot 1 \cdot 0.3] + 1 = 7 + [2.1] + 1 = 7 + 2 + 1 = 10 > T$. Oscillator 3, however, then fails to fire, and does not increase the counter. Neither oscillator 1 nor 4 get perturbed to fire. The latter because it is still in its refractory period, and thus ignores the firing oscillator, and the former since $4 + [4 \cdot 1 \cdot 0.3] + 1 = 4 + [1.2] + 1 = 4 + 1 + 1 = 6 \leqslant T$. During the last transition of the example, oscillator 2 and 3 reset their phase to one, while oscillator 1 is perturbed and increases its phase by two steps at once. Oscillator 4 is within its refractory period, which means that it is not perturbed, and simply increments its phase. In addition to these transitions, we also need some bookkeeping transitions, to ensure that the counter

is reset before the oscillators check their phase response. Furthermore, observe that in the example, it is crucial that oscillator 3 checks its response after oscillator 2 increased the counter, since otherwise oscillator 3 would not have been perturbed to fire.

Formally, we combine the states of the oscillators and the environment into a single state of the DTMC. Each oscillator can be described by a tuple consisting of the current *phase* Φ of the oscillator and the *mode* mode within this phase. The phase ranges from 1 to T, while the mode takes values from $\{start, update\}$. Furthermore, we use a single counter to keep track of the number of oscillators that fired successfully within a single phase computation.

For a network of N oscillators, a state of the DTMC consists of a function osc that associates a phase and mode with each oscillator,

$$\mathsf{osc}\colon \{1, \ldots, N\} \to (\{1, \ldots, T\} \times \{start, update\}),$$

and the state of the environment env counting the number of oscillators that fired,

$$\mathsf{env} \in \{0, \ldots, N\} \times \{start, update\}.$$

A state is therefore a tuple $s = (\mathsf{env}, \mathsf{osc})$, where env is the state of the environment, and osc is the state of the network. We denote the set of all concrete system states by $\mathcal{S}_c$. For simplicity, we will use the notation $\phi(s, u)$ to refer to the phase of oscillator u in state s, and similarly, $\mathsf{mode}(s, u)$ to refer to its mode. By abuse of notation, we will also write $\mathsf{mode}(s, \mathsf{env})$ to denote the mode of the environment and $\mathsf{count}(s)$ to refer to the value of the environment counter in state s. We sometimes need to denote that an oscillator changes neither its phase nor its mode in the definitions of the transitions. To that end, we define the stability of oscillator u between s and s' as

$$\mathsf{stable}_u(s, s') \equiv \phi(s, u) = \phi(s', u) \wedge \mathsf{mode}(s, u) = \mathsf{mode}(s', u) \ .$$

Similarly, we define stability of the environment:

$$\mathsf{stable}_{\mathsf{env}}(s, s') \equiv \mathsf{count}(s) = \mathsf{count}(s') \wedge \mathsf{mode}(s, \mathsf{env}) = \mathsf{mode}(s', \mathsf{env})$$

We use the notation $\mathsf{init}_\Phi(s) = \{u \mid \mathsf{mode}(s, u) = start \wedge \phi(s, u) = \Phi\}$ for the set of all oscillators sharing phase Φ and mode *start* in the state s. Furthermore, we also use the notation $\mathsf{init}(s) = \{u \mid \mathsf{mode}(s, u) = start\}$.

We now define the transition probabilities between states. To do this we first distinguish the following cases:

1. the environment resets its counter;
2. no oscillator has a clock value of T;
3. an oscillator is in the mode *start*, has a clock value lower than T, is perturbed, but not enough to fire;
4. an oscillator is in the mode *start*, has a clock value lower than T and is perturbed enough to fire;
5. an oscillator is in the mode *start*, has a clock value lower than T and is perturbed enough to fire, but fails to do so;
6. an oscillator is in the mode *start*, has a clock value of T, and broadcasts its pulse;
7. an oscillator is in the mode *start*, has a clock value of T, and fails to broadcast its pulse;
8. all oscillators are in the mode *update*, update their clock and reset their state to *start*.

We will impose an order on certain transitions for two reasons. Firstly, we will restrict transitions that are only used for bookkeeping purposes. For example, we will require that the reset transition of the environment is taken before any of the transitions for the oscillators

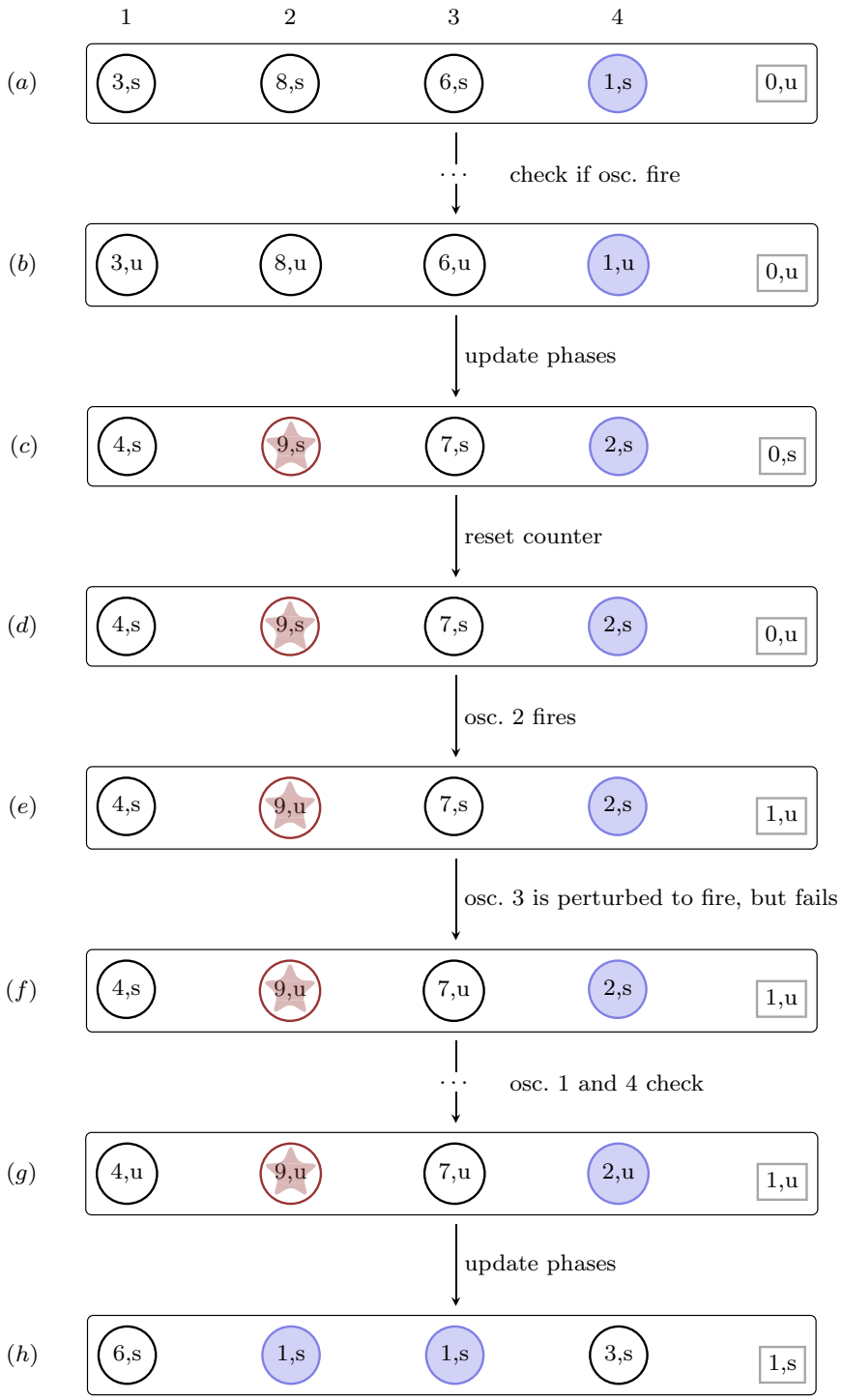


Fig. 1 Transitions in the Concrete Oscillator Model ($N = 4$, $T = 9$, $R = 2$)

within a phase are activated. In particular, this means that each computation starts with a transition of the type 1. Secondly, we need to ensure that, if at least one oscillator fires, the phase response of all oscillators is evaluated starting with oscillators in the highest phase, down to the lowest phase, as described above. The cases stated above are reflected in the following definitions for the transition probability between two states $s = (\mathsf{env}, \mathsf{osc})$ and $s' = (\mathsf{env}', \mathsf{osc}')$.

Case 1, where the environment resetting its counter is treated as follows. In the precondition, we require that the mode of the counter is *start*, and the state of the oscillators does not change from s to s'. Furthermore, the mode of the counter changes to *update* in s', and its value is set to 0. Since this transition is mandatory at the beginning of each round, its probability is 1. An example for this transition can be found in Fig. 1 in the transition from state (c) to (d).

$$\begin{aligned} &\text{If } \mathsf{mode}(s, \mathsf{env}) = \mathit{start} \wedge \mathsf{mode}(s', \mathsf{env}) = \mathit{update} \wedge \mathsf{count}(s') = 0 \\ &\quad \wedge \forall u\colon \mathsf{stable}_u(s, s'), \\ &\text{then } \mathbf{P}(s, s') = 1. \end{aligned} \tag{2}$$

If no oscillator is at the end of its cycle, that is, in case 2, we define the probability of one oscillator updating its mode as follows. Observe that we have to normalise the transition probability by the number of all oscillators that have not transitioned to their update mode yet. This is correct, since no oscillator fires, which also means that no oscillator can be activated beyond the maximum phase. This implies in particular that the order of oscillator transitions does not matter in this round. For example, all outgoing transitions from state (a) in Fig. 1 have probability $\frac{1}{4}$, while the subsequent transitions (which are not shown in the figure) occur with probability $\frac{1}{3}$, and so on.

$$\begin{aligned} &\text{If } \mathsf{mode}(s, \mathsf{env}) = \mathit{update} \text{ and there is a } w \text{ s.t.} \\ &\quad \mathsf{mode}(s, w) = \mathit{start} \wedge \mathsf{mode}(s', w) = \mathit{update} \wedge \phi(s, w) = \phi(s', w) \\ &\quad \wedge \forall u\colon \phi(s, u) < T \wedge \forall u\colon u \neq w \rightarrow \mathsf{stable}_u(s, s') \wedge \mathsf{stable}_{\mathsf{env}}(s, s') \\ &\text{then } \mathbf{P}(s, s') = \frac{1}{|\mathsf{init}(s)|}. \end{aligned} \tag{3}$$

Now we will consider the cases 3, 4 and 5, where some oscillator already fired (i.e., $\mathsf{count}(s) > 0$), and other oscillators are perturbed. In all three cases, one common precondition is that the counter is in its *update* mode and that there is an appropriate oscillator in the *start* mode. One complication arises: we have to ensure that the messages of firing oscillators lead to the perturbation of the other oscillators. Recall that the perturbation function is increasing, and thus a higher phase of an oscillator may result in a higher perturbation. That is, oscillators with a higher phase need to be perturbed by fewer firing oscillators before their phase is increased beyond the threshold and they in turn fire. However, if the oscillators with high phases fire, their additional messages may be enough to perturb oscillators with lower phases to fire as well. Hence, if we did not enforce an order from high to low phases, oscillators with a lower phase might not be perturbed when oscillators with a higher phase fire. To solve this, we only allow the oscillators to update their mode once all oscillators with a higher phase have been considered. Observe that we normalise the transition probabilities according to the number of oscillators satisfying similar conditions. Formally, this means that we need to normalise on the number of oscillators with the same phase in the *start* mode. To model case 3, we only change the mode of the corresponding oscillator to *update*, and keep the rest of the state. The precondition is, that all oscillators with higher phases have already

been considered, and the oscillator under consideration is not perturbed enough to fire, still within its refractory period, or both. For example, from state (f) on, the oscillators 1 and 4 are possibly perturbed, but not enough to fire. In this case, we will still first update oscillator 1, due to its higher phase.

$$\begin{aligned}
&\text{If } \mathsf{mode}(s, \mathsf{env}) = \mathit{update} \text{ and there is a } w \text{ s.t.}\\
&\quad \mathsf{mode}(s, w) = \mathit{start} \wedge \mathsf{mode}(s', w) = \mathit{update} \wedge \phi(s, w) = \phi(s', w)\\
&\quad \wedge \phi(s, w) < T \wedge \exists u\colon \phi(s, u) = T\\
&\quad \wedge \forall u\colon u \neq w \rightarrow (\mathsf{mode}(s, u) = \mathit{update} \vee \phi(s, u) \leqslant \phi(s, w))\\
&\quad \wedge \phi(s, w) + \Delta(\phi(s, w), \mathsf{count}(s), \epsilon) + 1 \leqslant T \vee \phi(s, w) \leqslant R\\
&\quad \wedge \forall u\colon u \neq w \rightarrow \mathsf{stable}_u(s, s')\\
&\quad \wedge \mathsf{stable}_{\mathsf{env}}(s, s')\\
&\text{then } \mathbf{P}(s, s') = \frac{1}{|\mathsf{init}_{\phi(s,w)}(s)|}.
\end{aligned} \tag{4}$$

If an oscillator is perturbed, and actually fires, we update its mode to *update*, and increase the counter of the environment. We only allow for this transition, if the oscillator is perturbed to fire, and is outside of refractory period. This definition corresponds to case 4. The probability of such a transition is the probability that a broadcaset failure does not occur, $1-\mu$, normalised by the number of oscillators in the same phase that have not yet been considered.

$$\begin{aligned}
&\text{If } \mathsf{mode}(s, \mathsf{env}) = \mathit{update} \text{ and there is a } w \text{ s.t.}\\
&\quad \mathsf{mode}(s, w) = \mathit{start} \wedge \mathsf{mode}(s', w) = \mathit{update} \wedge \phi(s, w) = \phi(s', w)\\
&\quad \wedge \phi(s, w) < T \wedge \exists u\colon \phi(s, u) = T\\
&\quad \wedge \forall u\colon u \neq w \rightarrow (\mathsf{mode}(s, u) = \mathit{update} \vee \phi(s, u) \leqslant \phi(s, w))\\
&\quad \wedge \phi(s, w) + \Delta(\phi(s, w), \mathsf{count}(s), \epsilon) + 1 > T \wedge \phi(s, w) > R\\
&\quad \wedge \forall u\colon u \neq w \rightarrow \mathsf{stable}_u(s, s')\\
&\quad \wedge \mathsf{count}(s') = \mathsf{count}(s) + 1\\
&\quad \wedge \mathsf{mode}(s, \mathsf{env}) = \mathsf{mode}(s', \mathsf{env})\\
&\text{then } \mathbf{P}(s, s') = \frac{1-\mu}{|\mathsf{init}_{\phi(s,w)}(s)|}
\end{aligned} \tag{5}$$

If the oscillator fails to fire, as described by case 5, the only differences to the preceeding case is that we do not increase the counter, and that the probability of the transition is the normalised broadcast failure probability. As an example, consider the transition in Fig. 1 from state e to f, where oscillator 3 is indeed perturbed to the end of its cycle, but fails to fire. That is, the environment counter is not increased. The probability of this transition is $\frac{\mu}{1} = 0.2$, since it is the only oscillator with the phase value 7.

$$\begin{aligned}
&\text{If } \mathsf{mode}(s, \mathsf{env}) = \mathit{update} \text{ and there is a } w \text{ s.t.}\\
&\quad \mathsf{mode}(s, w) = \mathit{start} \wedge \mathsf{mode}(s', w) = \mathit{update} \wedge \phi(s, w) = \phi(s', w)\\
&\quad \wedge \phi(s, w) < T \wedge \exists u\colon \phi(s, u) = T\\
&\quad \wedge \forall u\colon u \neq w \rightarrow (\mathsf{mode}(s, u) = \mathit{update} \vee \phi(s, u) \leqslant \phi(s, w))\\
&\quad \wedge \phi(s, w) + \Delta(\phi(s, w), \mathsf{count}(s), \epsilon) + 1 > T \wedge \phi(s, w) > R\\
&\quad \wedge \forall u\colon u \neq w \rightarrow \mathsf{stable}_u(s, s')\\
&\quad \wedge \mathsf{stable}_{\mathsf{env}}(s, s')
\end{aligned}$$

$$\text{then } \mathbf{P}(s, s') = \frac{\mu}{|\mathsf{init}_{\phi(s,w)}(s)|}. \tag{6}$$

Now we turn to the cases 6 and 7 where some oscillator is at the end of its cycle. The preconditions of both cases are similar to the preceeding cases: the counter is required to be in the *update* mode, and there is an oscillator w, whose phase is T and mode is *start*. Furthermore, in s', the mode of w is *update*, and the state of all other oscillators does not change. The difference between the cases is whether the counter is increased, that is, whether the oscillator manages to broadcast its signal. The probability of succeeding is $\frac{1-\mu}{|\mathsf{init}_T(s)|}$, since there may be more than one oscillator in phase T at state s. Hence we have to normalise the tranistion probability accordingly. Similarly, the probability of failing to fire is $\frac{\mu}{|\mathsf{init}_T(s)|}$. So, the probability of the transition from state (d) to (e) is $\frac{1-0.2}{1} = 0.8$, since it is the only oscillator in the firing phase.

$$\begin{aligned}
&\text{If } \mathsf{mode}(s, \mathsf{env}) = \mathit{update} \text{ and there is a } w \text{ s.t.}\\
&\quad \mathsf{mode}(s, w) = \mathit{start} \wedge \phi(s, w) = T \wedge \mathsf{mode}(s', w) = \mathit{update}\\
&\quad \wedge\, \phi(s, w) = \phi(s', w) \wedge \forall u\colon u \neq w \rightarrow \mathsf{stable}_u(s, s')\\
&\quad \wedge\, \mathsf{count}(s') = \mathsf{count}(s) + 1\\
&\text{then } \mathbf{P}(s, s') = \frac{1-\mu}{|\mathsf{init}_T(s)|}.
\end{aligned} \tag{7}$$

$$\begin{aligned}
&\text{If } \mathsf{mode}(s, \mathsf{env}) = \mathit{update} \text{ and there is a } w \text{ s.t.}\\
&\quad \mathsf{mode}(s, w) = \mathit{start} \wedge \phi(s, w) = T \wedge \mathsf{mode}(s', w) = \mathit{update}\\
&\quad \wedge\, \phi(s, w) = \phi(s', w) \wedge \forall u\colon u \neq w \rightarrow \mathsf{stable}_u(s, s')\\
&\quad \wedge\, \mathsf{count}(s') = \mathsf{count}(s)\\
&\text{then } \mathbf{P}(s, s') = \frac{\mu}{|\mathsf{init}_T(s)|}.
\end{aligned} \tag{8}$$

The final case 8, where all oscillators update their clock values simultaneously, is given by the following formula. It requires that all oscillators have finished their computation, whether they fire, and both the counter and the oscillators will reset their mode to *start* after the transition.

$$\begin{aligned}
&\text{If } \mathsf{mode}(s, \mathsf{env}) = \mathit{update} \text{ and } \mathsf{mode}(s', \mathsf{env}) = \mathit{start} \text{ and}\\
&\quad \text{for all } u \text{ we have } \mathsf{mode}(s, u) = \mathit{update} \wedge \mathsf{mode}(s', u) = \mathit{start} \wedge F_{\mathit{update}}\\
&\text{then } \mathbf{P}(s, s') = 1.
\end{aligned} \tag{9}$$

The formula F_{update} is an abbreviation for the conjunction of the following four conditions, which model the update of the phases of the oscillators, according to the phase response function. Observe that the phases of the oscillators had not been updated by the previously defined transitions. Hence, we now update the phases of all oscillators at once.

$$\begin{aligned}
&\forall u\colon \phi(s, u) = T \rightarrow\\
&\quad \phi(s', u) = 1 && \text{(9a)}\\
&\forall u\colon \phi(s, u) < T \wedge \phi(s, u) \leqslant R \rightarrow\\
&\quad \phi(s', u) = \phi(s, u) + 1 && \text{(9b)}\\
&\forall u\colon \phi(s, u) < T \wedge \phi(s, u) > R \wedge\\
&\quad \phi(s, u) + \Delta(\phi(s, u), \mathsf{count}(s), \epsilon) + 1 \leqslant T \rightarrow
\end{aligned}$$

$$\phi(s', u) = \phi(s, u) + \Delta(\phi(s, u), \mathsf{count}(s), \epsilon) + 1 \tag{9c}$$

$$\forall u\colon \phi(s, u) < T \wedge \phi(s, u) > R \wedge \\ \phi(s, u) + \Delta(\phi(s, u), \mathsf{count}(s), \epsilon) + 1 > T \rightarrow \\ \phi(s', u) = 1. \tag{9d}$$

In this formula, (9a) handles the simple case of firing oscillators, while (9b) defines the behaviour of oscillators within their refractory period. The formulas (9c) and (9d) reflect the two cases where oscillators are perturbed, either not exceeding their oscillation cycle, or firing, respectively. For example, in the transition from state (b) to (c) in the figure, oscillators 1 to 3 satisfy clause 9c (where $\Delta(\Phi, 0, 0.3) = 0$ by definition for all phases Φ), while oscillator 4 is updated due to clause 9b. In the transition from (g) to (h), however, oscillator 1 is again updated due to clause 9c, oscillator 2 due to clause 9a, oscillator 3 due to 9d and oscillator 4 again due to 9b.

Finally, we define the transitions from the initial state, that form a distribution over all possible initial configurations for the network. As explained above, states where any component (i.e., an oscillator or the environment) is in the mode *update* are considered intermediate states. Hence, we only allow for transitions from the initial state to states, where every component is in the *start* mode, and furthermore, the counter of the environment is set to 0. Let us denote this set of states by $\mathcal{S}'$, i.e.,

$$\mathcal{S}' = \{s \in \mathcal{S} \mid \mathsf{count}(s) = 0 \wedge \mathsf{mode}(s, \mathsf{env}) = \mathit{start} \wedge \forall u\colon \mathsf{mode}(s, u) = \mathit{start}\} \ .$$

Then, for each state $s \in \mathcal{S}'$, we have

$$\mathbf{P}(s_I, s) = \frac{1}{|\mathcal{S}'|} \ . \tag{10}$$

As stated above, this model directly models a fully-connected network of N oscillators. It would certainly be possible to define such a model as a parallel composition of single oscillators, but we will in the following argue that such a definition would not increase the readability of the definitions, and even obscure the behaviour of the network. In particular, the probability for an oscillator to end up in a certain state after such a step is dependent on the overall number of oscillators in the system. For example, consider the third state shown in the example network of Fig. 1. That is, oscillator 2 is about to fire and oscillator 4 is still in the refractory period. If we added another oscillator, which we name "5", to the network, the possible transitions from this state depend on the phase of this oscillator. If it is not at the end of its cycle, the only outgoing transitions are due to oscillator 2 firing (and either succeeding or failing). However, oscillator 5 may be perturbed enough to fire, thus increasing the chance of oscillator 1 being perturbed enough to fire as well. Similarly, if oscillator 5 is also at the end of its cycle, the activation of oscillators 1 and 3 is increased, depending on the perturbation function. In the final transition of the sequence (i.e., after all oscillators changed their mode from *start* to *update*), the resulting state will be different then the corresponding state in the original model with four oscillators. Even in this small example, the necessary case distinctions for the definition of a parallel composition grow quite large, and would still not allow us to refer to already existing transitions, but require the creation of completely new ones.

With this model, we could begin to analyse the synchronisation behaviour with respect to different phase response functions or broadcast failure probabilities. However, the state space of the model increases exponentially with the number of oscillators, which makes an analysis beyond small numbers infeasible. To overcome this restriction, we increase the level of abstraction as presented in the next section.

5 Population model

In this section, we define a *population model* of a network of pulse-coupled oscillators for parameters as defined in Sect. 4.1 as $\mathcal{P} = (\Delta, N, T, R, \epsilon, \mu)$. Oscillators in our model have identical dynamics, and two oscillators are indistinguishable if they share the same phase. That is, we can reason about groups of oscillators, instead of individuals. We therefore encode the global state of the model as a tuple $\langle k_1, \ldots, k_T \rangle$ where each k_Φ is the number of oscillators sharing a phase value of Φ. The population model does not account for the introduction of additional oscillators to a network, or the loss of existing coupled oscillators. That is, the population N remains constant.

Definition 7 A global state of a population model $\mathcal{P} = (\Delta, N, T, R, \epsilon, \mu)$ is a T-tuple $\sigma \in \{0, \ldots, N\}^T$, where $\sigma = \langle k_1, \ldots, k_T \rangle$ and $\sum_{\Phi=1}^{T} k_\Phi = N$. The set of all global states of $\mathcal{P}$ is $\Gamma(\mathcal{P})$, or simply Γ when $\mathcal{P}$ is clear from the context.

Example 1 Figure 2 shows four global states for an instantiated population model of $N = 8$ oscillators with $T = 10$ discrete values for their phase and a refractory period of length $R = 2$. We use the linear phase response function $\Delta(\Phi, \alpha, \epsilon) = [\Phi \cdot \alpha \cdot \epsilon]$ where $[\cdot]$ denotes rounding to the closest integer. Furthermore, let $\epsilon = 0.115$. For example $\sigma_0 = \langle 0, 0, 2, 1, 0, 0, 5, 0, 0, 0 \rangle$ is the global state where two oscillators have a phase of three, one oscillator has a phase of four, and five oscillators have a phase of seven. The starred node indicates the number of oscillators with phase ten that will fire in the next moment in time, while the shaded nodes indicate oscillators with phases that lie within the refractory period (one and two). If no oscillators have some phase Φ then we omit the 0 in the corresponding node. Observe that, while going from σ_{i-1} to σ_i ($1 \leqslant i \leqslant 3$), the oscillator phases increase by one. In the next section, we will explain how transitions between these global states are made. Note that directional arrows indicate cyclic direction, and do not represent transitions.

With every state $\sigma \in \Gamma$ we associate a non-empty set of *failure vectors*, where each failure vector is a tuple of broadcast failures that could occur in σ.

Definition 8 A *failure vector* is a T-tuple $F = \langle f_1, \ldots, f_T \rangle \in (\{0, \ldots, N\} \cup \{\star\})^T$, where $f_i = \star$ implies $f_j = \star$ for all $1 \leqslant j \leqslant i$. We denote the set of all possible failure vectors by $\mathcal{F}$.

Given a failure vector $F = \langle f_1, \ldots, f_T \rangle$, $f_\Phi \in \{0, \ldots, N\}$ indicates the number of broadcast failures that occur for all oscillators with a phase of Φ. If $f_\Phi = \star$ then no oscillators with a phase of Φ fire. Semantically, $f_\Phi = 0$ and $f_\Phi = \star$ differ in that the former indicates that all (if any) oscillators with phase Φ fire and no broadcast failures occur, while the latter indicates that all (if any) oscillators with a phase of Φ do not fire. If no oscillators fire at all in a global state then we have only one possible failure vector, namely $\{\star\}^T$.

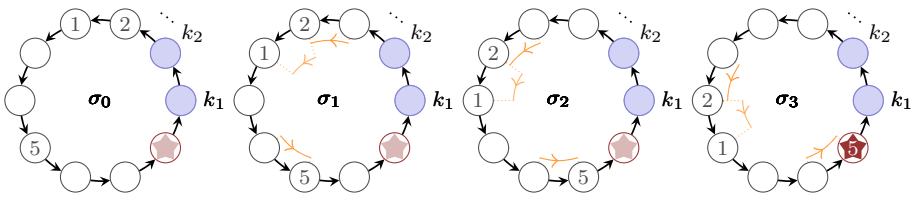


Fig. 2 Evolution of the global state over four discrete time steps

5.1 Transitions

In Sect. 5.2 we will describe how we can calculate the set of all possible failure vectors for a global state, and thereby identify all of its successor states. However we must first show how we can calculate the single successor state of a global state σ, given some failure vector F.

Absorptions Since we are considering a fully connected network of oscillators, two oscillators sharing the same phase will have their phase updated to the same value in the next time step. They will always perceive the same number of other oscillators firing. Therefore, for each phase Φ we define the function $\alpha^{\Phi} : \Gamma \times \mathcal{F} \to \{0, \ldots, N\}$, where $\alpha^{\Phi}(\sigma, F)$ is the number of oscillators with a phase greater than Φ perceived to be firing by oscillators with phase Φ, in some global state, incorporating the broadcast failures defined in the failure vector F. This allows us to encode the aforementioned chain reactions of firing oscillators. Note that our encoding of chain reactions results in a global semantics that differs from typical parallelisation operations, for example, the construction of the cross product of the individual oscillators. Observe that, in the concrete model of Sect. 4.2, we modelled such a behaviour by case 4.

Given a global state $\sigma = \langle k_1, \ldots, k_T \rangle$ and a failure vector $F = \langle f_1, \ldots, f_T \rangle$, the following mutually recursive definitions show how we calculate the values $\alpha^1(\sigma, F), \ldots, \alpha^T(\sigma, F)$, and how functions introduced in Sect. 4.1 are modified to indicate the update in phase, and firing, of *all* oscillators sharing the same phase Φ. Observe that to calculate any $\alpha^{\Phi}(\sigma, F)$ we only refer to definitions for phases greater than Φ and the base case is $\Phi = T$, that is, values are computed from T down to 1. The function ref is the refractory function as defined in Sect. 4.1.

$$update^{\Phi}(\sigma, F) = 1 + \mathrm{ref}(\Phi, \Delta(\Phi, \alpha^{\Phi}(\sigma, F), \epsilon)) \tag{11}$$

$$fire^{\Phi}(\sigma, F) = update^{\Phi}(\sigma, F) > T \tag{12}$$

$$\alpha^{\Phi}(\sigma, F) = \begin{cases} 0 & \text{if } \Phi = T \\ \alpha^{\Phi+1}(\sigma, F) + k_{\Phi+1} - f_{\Phi+1} & \text{if } \Phi < T,\ f_{\Phi+1} \neq \star \text{ and } fire^{\Phi+1}(\sigma, F) \\ \alpha^{\Phi+1}(\sigma, F) & \text{otherwise} \end{cases} \tag{13}$$

Transition function We now define the transition function that maps phase values to their updated values in the next time step. Note that since we no longer distinguish different oscillators with the same phase we only need to calculate a single value for their evolution and perturbation.

Definition 9 The *phase transition function* $\tau : \Gamma \times \{1, \ldots, T\} \times \mathcal{F} \to \mathbb{N}$ maps a global state σ, a phase Φ, and some possible failure vector F for σ, to the updated phase in the next discrete time step, with respect to the broadcast failures defined in F, and is defined as

$$\tau(\sigma, \Phi, F) = \begin{cases} 1 & \text{if } fire^{\Phi}(\sigma, F) \\ update^{\Phi}(\sigma, F) & \text{otherwise.} \end{cases} \tag{14}$$

Let $\mathcal{U}_{\Phi}(\sigma, F)$ be the set of phase values Ψ where all oscillators with phase Ψ in σ will have their phase updated to Φ in the next time step, with respect to the broadcast failures defined in F. Formally,

$$\mathcal{U}_{\Phi}(\sigma, F) = \{\Psi \mid \Psi \in \{1, \ldots, T\} \wedge \tau(\sigma, \Psi, F) = \Phi\}. \tag{15}$$

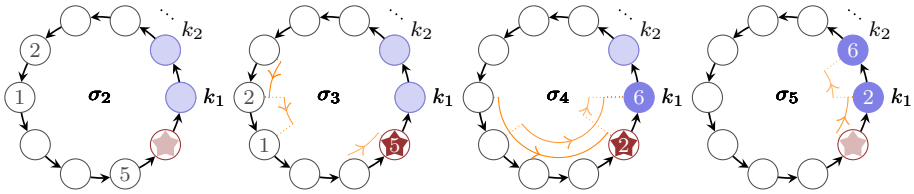


Fig. 3 Evolution of the global state over four discrete time steps

We can now calculate the successor state of a global state σ and define how the model evolves over time.

Definition 10 The *successor function* $\overrightarrow{\text{succ}} : \Gamma \times \mathcal{F} \rightarrow \Gamma$ maps a global state σ and a failure vector F to a state σ', and is defined as $\overrightarrow{\text{succ}}(\langle k_1, \ldots, k_T \rangle, F) = \langle k'_1, \ldots, k'_T \rangle$, where $k'_\Phi = \sum_{\Psi \in \mathcal{U}_\Phi(\sigma, F)} k_\Psi$ for $1 \leqslant \Phi \leqslant T$.

Example 2 Recall that the perturbation function of our example was given as $\Delta(\Phi, \alpha, \epsilon) = [\Phi \cdot \alpha \cdot \epsilon]$, where $[\cdot]$ denotes rounding and $\epsilon = 0.115$. Consider the global state σ_2 of Fig. 3 where no oscillators will fire since $k_{10} = 0$. We therefore have one possible failure vector for σ_0, namely $F = \{\star\}^{10}$. Since no oscillators fire the dynamics of the oscillators are determined solely by their standalone evolution, and all oscillators simply increase their phase by 1 in the next time step. Now consider the global state σ_3 and $F = \langle \star, \star, \star, \star, \star, \star, 1, 0, 0, 0 \rangle$, a possible failure vector for σ_3, indicating that oscillators with phases of 7 to 10 will fire and one broadcast failure will occur for the single oscillator that will fire with phase 7. Here a chain reaction occurs as the perturbation induced by the firing of the 5 oscillators causes the single oscillator with a phase of 7 to also fire. A broadcast failure occurs when this single oscillator fires, and the perturbation of the 5 firing oscillators is insufficient to cause the 2 oscillators with a phase of 6 to also fire. In the next state the oscillator with phase 7 has been absorbed by the group of the 5 oscillators that had phase 10.

More explicitly, since $\mathit{fire}^{10}(\sigma_3, F)$ holds we have that $\alpha^9(\sigma_3, F) = \alpha^{10}(\sigma_3, F) + k_{10} - f_{10} = 0 + 5 - 0 = 5$. Now, since $\Delta(9, 5, 0.115) = [9 \cdot 5 \cdot 0.115] = [5.175] = 5$, we have $\mathit{update}^9(\sigma_3, F) = 15 > 10$, and thus, fire^9 holds. Hence, we have that $\alpha^8(\sigma_3, F) = \alpha^9(\sigma_3, F) + k_9 - f_9 = 0 + 5 - 0 = 5$, and similarly, due to $\Delta(8, 5, 0.115) = 5$, fire^8 holds. That is, we have that $\alpha^7(\sigma_3, F) = \alpha^8(\sigma_3, F) + k_8 - f_8 = 0 + 5 - 0 = 5$. We then continue calculating $\alpha^\Phi(\sigma_3, F)$ for $6 \geqslant \Phi \geqslant 1$, and noting that $\Delta(6, 5, 0.115) = [3.45] = 3$. Hence $\mathit{fire}^6(\sigma_3, F)$ does not hold, and we obtain $\alpha^1(\sigma_3, F) = \alpha^2(\sigma_3, F) = \alpha^3(\sigma_3, F) = \alpha^4(\sigma_3, F) = \alpha^5(\sigma_3, F) = \alpha^6(\sigma_3, F) = \alpha^7(\sigma_3, F) = 5$. We conclude that $\mathcal{U}_1(\sigma_3, F) = \{10, 9, 8, 7\}$, $\mathcal{U}_{10}(\sigma_3, F) = \{6, 5\}$, $\mathcal{U}_9(\sigma_3, F) = \{4, 3\}$, and $\mathcal{U}_\Phi(\sigma_3, F) = \emptyset$ for $9 > \Phi > 3$. Since $R = 2$ we have that $\mathcal{U}_3(\sigma_3, F) = \{2\}$ and $\mathcal{U}_2(\sigma_3, F) = \{1\}$. We calculate the successor of σ_3 as $\sigma_4 = \overrightarrow{\text{succ}}(\langle \star, \star, \star, \star, \star, \star, 1, 0, 0, 0 \rangle, F) = \langle k_{10} + k_9 + k_8 + k_7, k_1, k_2, 0, 0, 0, 0, 0, k_4 + k_3, k_6 + k_5 \rangle = \langle 6, 0, 0, 0, 0, 0, 0, 0, 0, 2 \rangle$.

Lemma 1 *The number of oscillators is invariant during transitions, i.e., the successor function only creates tuples that are states of the given model. Formally, let $\sigma = \langle k_1, \ldots, k_T \rangle$ and $\sigma' = \langle k'_1, \ldots, k'_T \rangle$ be two states of a model $\mathcal{P}$ such that $\sigma' = \overrightarrow{\text{succ}}(\sigma, F)$, where F is some possible failure vector for σ. Then $\sum_{\Phi=1}^{T} k_\Phi = \sum_{\Phi=1}^{T} k'_\Phi = N$.*

Proof Observe that the range of the function τ is bound by T. By construction we can see that for any σ, for any possible failure vector F for σ, and for all $\Phi \in \{1, \ldots, T\}$,

we have that $1 \leqslant \tau(\sigma, \Phi, F) \leqslant T$. Hence for all Ψ with $1 \leqslant \Psi \leqslant T$, there is a Φ such that $\Psi \in \mathcal{U}_\Phi(\sigma, F)$. This implies $\bigcup_{\Phi=1}^{T} \mathcal{U}_\Phi(\sigma, F) = \{1, \ldots, T\}$. Furthermore, there cannot be more than one Φ such that $\Psi \in \mathcal{U}_\Phi(\sigma, F)$, since τ is functional. Now we have $\sum_{\Phi=1}^{T} k'_\Phi = \sum_{\Phi=1}^{T} \sum_{\Psi \in \mathcal{U}_\Phi(\sigma, F)} k_\Psi = \sum_{\Phi=1}^{T} k_\Phi = N$. □

5.2 Failure vector calculation

We construct all possible failure vectors for a global state by considering every group of oscillators in decreasing order of phase. At each stage we determine if the oscillators would fire. If they fire then we consider each outcome where any, all, or none of the firings result in a broadcast failure. We then add a corresponding value to a partially calculated failure vector and consider the next group of oscillators with a lower phase. If the oscillators do not fire then there is nothing left to do, since by Definition 5 we know that Δ is increasing, therefore all oscillators with a lower phase will also not fire. We can then pad the partial failure vector with $\star$ appropriately to indicate that no failure could happen since no oscillator fired.

Table 1 illustrates how a possible failure vector for global state σ_3 in Fig. 3 is iteratively constructed. The first three columns respectively indicate the current iteration i, the global state σ_3 with the currently considered oscillators underlined, and the elements of the failure vector F computed so far. The fourth column is *true* if the oscillators with phase $T+1-i$ would fire given the broadcast failures in the partial failure vector. We must consider all outcomes of any or all firings resulting in broadcast failure. The final column therefore indicates whether the value added to the partial failure vector in the current iteration is the only possible value (*false*), or a choice from one of several possible values (*true*).

Initially we have an empty partial failure vector. At the first iteration there are 5 oscillators with a phase of 10. These oscillators will fire so we must consider each case where 0, 1, 2, 3, 4 or 5 broadcast failures occur. Here we choose 0 broadcast failures, which is then added to the partial failure vector. At iterations 2 and 3 the oscillators would have fired, but since there are no oscillators with a phase of 9 or 8 we only have one possible value to add to the partial failure vector, namely 0. At iteration 4 a single oscillator with a phase of 7 fires, and we choose the case where the firing resulted in a broadcast failure. In the final iteration oscillators with a phase of 6 do not fire, hence we can conclude that oscillators with phases less than 6 also do not fire, and can fill the partial failure vector appropriately with $\star$.

Formally, we define a family of functions *fail* indexed by Φ, where each $fail_\Phi$ takes as parameters some global state σ, and V, a vector of length $T - \Phi$. V represents all broadcast failures for all oscillators with a phase greater than Φ. The function $fail_\Phi$ then computes

Table 1 Construction of a possible failure vector for a global state $\sigma_3 = \langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5 \rangle$

iteration (i)	π_1	Failure vector B	Fired	Branches
0	$\langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5 \rangle$	$\langle\rangle$	–	*false*
1	$\langle 0, 0, 0, 0, 0, 2, 1, 0, 0, \underline{5} \rangle$	$\langle 0 \rangle$	*true*	*true*
2	$\langle 0, 0, 0, 0, 0, 2, 1, 0, \underline{0}, 5 \rangle$	$\langle 0, 0 \rangle$	*true*	*false*
3	$\langle 0, 0, 0, 0, 0, 2, 1, \underline{0}, 0, 5 \rangle$	$\langle 0, 0, 0 \rangle$	*true*	*false*
4	$\langle 0, 0, 0, 0, 0, 2, \underline{1}, 0, 0, 5 \rangle$	$\langle 1, 0, 0, 0 \rangle$	*true*	*true*
5	$\langle 0, 0, 0, 0, 0, \underline{2}, 1, 0, 0, 5 \rangle$	$\langle \star, \star, \star, \star, \star, \star, 1, 0, 0, 0 \rangle$	*false*	–

the set of all possible failure vectors for σ with suffix V. Here we use the notation $v^\frown v'$ to indicate vector concatenation.

Definition 11 We define $\mathit{fail}_\Phi : \Gamma \times \{0, \ldots, N\}^{T-\Phi} \to \mathbb{P}((\{0, \ldots, N\} \cup \{\star\})^T)$, for $1 \leqslant \Phi \leqslant T$, as the family of functions indexed by Φ, where $\sigma = \langle k_1, \ldots, k_T \rangle$ and

$$\mathit{fail}_\Phi(\sigma, V) = \begin{cases} \bigcup_{k=0}^{k_\Phi} \mathit{fail}_{\Phi-1}(\sigma, \langle k \rangle^\frown V) & \text{if } 1 < \Phi \leqslant T \text{ and } \mathit{fire}^\Phi(\sigma, \{\star\}^{\Phi\frown} V) \\ \bigcup_{k=0}^{k_1} \{\langle k \rangle^\frown V\} & \text{if } \Phi = 1 \text{ and } \mathit{fire}^1(\sigma, \langle \star \rangle^\frown V) \\ \left\{\{\star\}^{\Phi\frown} V\right\} & \text{otherwise} \end{cases}$$

Observe that the result of fail_T is always a set of well defined failure vectors, since whenever $\star$ is introduced into a failure vector at index Φ, all preceding indices are also filled with $\star$, as required by Definition 8.

Definition 12 Given a global state $\sigma \in \Gamma$, we define $\mathcal{F}_\sigma$, the set of all possible failure vectors for that state, as $\mathcal{F}_\sigma = \mathit{fail}_T(\sigma, \langle\rangle)$, and define $\mathit{next}(\sigma)$, the set of all successor states of σ, as $\mathit{next}(\sigma) = \{\overrightarrow{\mathrm{succ}}(\sigma, F) \mid F \in \mathcal{F}_\sigma\}$.

Note that for some global states $|\mathit{next}(\sigma)| < |\mathcal{F}_\sigma|$, since we may have that $\overrightarrow{\mathrm{succ}}(\sigma, F) = \overrightarrow{\mathrm{succ}}(\sigma, F')$ for some $F, F' \in \mathcal{F}_\sigma$ with $F \neq F'$.

Given a global state σ and a failure vector $F \in \mathcal{F}_\sigma$, we will now compute the probability of a transition being made to state $\overrightarrow{\mathrm{succ}}(\sigma, F)$ in the next time step. Recall that μ is the probability with which a broadcast failure occurs. Firstly we define the probability mass function $\mathsf{PMF} : \{1, \ldots, N\}^2 \to [0, 1]$, where $\mathsf{PMF}(k, f)$ gives the probability of f broadcast failures occurring given that k oscillators fire, as $\mathsf{PMF}(k, f) = \mu^f (1-\mu)^{k-f} \binom{k}{f}$. We then denote by $\mathsf{PFV} : \Gamma \times \mathcal{F}_\sigma \to [0, 1]$ the function mapping a possible broadcast failure vector F for σ, to the probability of the failures in F occurring. That is,

$$\mathsf{PFV}(\langle k_1, \ldots, k_T \rangle, \langle f_1, \ldots, f_T \rangle) = \prod_{\Phi=1}^{T} \begin{cases} \mathsf{PMF}(k_\Phi, f_\Phi) & \text{if } f_\Phi \neq \star \\ 1 & \text{otherwise} \end{cases} \tag{16}$$

Lemma 2 *For any global state σ, PFV is a discrete probability distribution over $\mathcal{F}_\sigma$. Formally, $\sum_{F \in \mathcal{F}_\sigma} \mathit{PFV}(\sigma, F) = 1$.*

Proof Given a global state $\sigma = \langle k_1, \ldots, k_T \rangle$ we can construct a tree of depth T where each leaf node is labelled with a possible failure vector for σ, and each node Λ at depth Φ is labelled with a vector of length Φ corresponding to the last Φ elements of a possible failure vector for σ. We denote the label of a node Λ by $V(\Lambda)$. We label each node Λ_ω with $\langle \omega \rangle^\frown V(\Lambda)$. We iteratively construct the tree, starting with the root node, *root*, at depth 0, which we label with the empty tuple $\langle\rangle$. For each node Λ at depth $0 \leqslant \Phi < T$ we construct the children of Λ as follows:

1. If oscillators with phase Φ fire we define the sample space $\Omega = \{0, \ldots, n_\Phi\}$ to be a set of disjoint events, where each $\omega \in \Omega$ is the event where ω broadcast failures occur, given that k_Φ oscillators fired. For each $\omega \in \Omega$ there is a child Λ_ω of Λ with label $\langle \omega \rangle^\frown V(\Lambda)$, and we label the edge from Λ to Λ_ω with $\mathsf{PMF}(k_\Phi, \omega)$.
2. If oscillators with phase Φ do not fire then Λ has a single child $\Lambda_\star$ labelled with $\langle \star \rangle^\frown V(\Lambda)$, and we label the edge from Λ to $\Lambda_\star$ with 1.

We denote the label of an edge from a node Λ to its child Λ' by $L(\Lambda, \Lambda')$. For case 2 we can observe that if oscillators with phase Φ do not fire then we know that oscillators with

any phase $\Psi < \Phi$ will also not fire, since from Definition 5 we know that Δ is an increasing function. Hence, all descendants of Λ will also have a single child, with an edge labelled with 1, and each node is labelled with the label of its parent, prefixed with $\langle\star\rangle$.

After constructing the tree we have a vector of length T associated with each leaf node, corresponding to a failure vector for σ. The set $\mathcal{F}_\sigma$ of all possible failure vectors for σ is therefore the set of all vectors labelling leaf nodes. We denote by $\mathsf{P}^{\downarrow}(\Lambda)$ the product of all labels on edges along the path from Λ back to the root. Given a global state $\sigma = \langle k_1, \ldots, k_T\rangle$ and a failure vector $F = \langle f_1, \ldots, f_T\rangle \in \mathcal{F}_\sigma$ labelling some leaf node Λ at depth T, we can see that

$$\mathsf{P}^{\downarrow}(\Lambda) = 1 \cdot \prod_{\Phi=1}^{T} \begin{cases} \mathsf{PMF}(k_\Phi, f_\Phi) & \text{if } f_\phi \neq \star \\ 1 & \text{otherwise} \end{cases} = \mathsf{PFV}(\sigma, F).$$

Let D^Φ denote the set of all nodes at depth Φ. We show $\sum_{d\in D^\Phi} \mathsf{P}^{\downarrow}(d) = 1$ by induction on Φ. For $\Phi = 0$, i.e., $D^\Phi = \{root\}$, the property holds by definition. Now assume that $\sum_{d\in D^\Phi} \mathsf{P}^{\downarrow}(d) = 1$ holds for some $0 \leqslant \Phi < T$. Let Λ be some node in D^Φ, and let C^Λ be the set of all children of Λ. Consider the following two cases: If oscillators with phase Φ do not fire then $|C^\Lambda| = 1$, and for the only $c \in C^\Lambda$ we have that $L(\Lambda, c) = 1$. If oscillators with phase Φ fire observe that PMF is a probability mass function for a random variable defined on the sample space $\Omega = \{0, \ldots, k_\Phi\}$. In either case we can see that $\sum_{c\in C^\Lambda} L(\Lambda, c) = 1$. Note that $D^{\Phi+1} = \bigcup_{d\in D^\Phi} C^d$, and recall that $L(d, c) \cdot \mathsf{P}^{\downarrow}(d) = \mathsf{P}^{\downarrow}(c)$. Therefore,

$$\sum_{d\in D^{\Phi+1}} \mathsf{P}^{\downarrow}(d) = \sum_{d\in D^\Phi} \sum_{c\in C^d} L(d, c) \cdot \mathsf{P}^{\downarrow}(d) = \sum_{d\in D^\Phi} \left(\mathsf{P}^{\downarrow}(d) \sum_{c\in C^d} L(d, c) \right).$$

Since $\sum_{c\in C^d} L(d, c) = 1$ for each $d \in D^\Phi$, and from the induction hypothesis, we then have that

$$\sum_{d\in D^\Phi} \left(\mathsf{P}^{\downarrow}(d) \sum_{c\in C^d} L(d, c) \right) = \sum_{d\in D^\Phi} \mathsf{P}^{\downarrow}(d) = 1.$$

We have already shown that $\mathsf{P}^{\downarrow}(\Lambda) = \mathsf{PFV}(\sigma, F)$ for any leaf node Λ labelled with a failure vector F, and since the set of all labels for leaf nodes is $\mathcal{F}_\sigma$ we can conclude that

$$\sum_{F\in\mathcal{F}_\sigma} \mathsf{PFV}(\sigma, F) = \sum_{d\in D^T} \mathsf{P}^{\downarrow}(d) = 1.$$

This proves the lemma. □

Example 3 We consider again the global states $\sigma_3 = \langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5\rangle$ and $\sigma_4 = \langle 6, 0, 0, 0, 0, 0, 0, 0, 0, 2\rangle$, given in Fig. 3, of the population model instantiated in Example 1, and the failure vector $F = \langle \star, \star, \star, \star, \star, \star, 1, 0, 0, 0\rangle$ given in Example 2, noting that $F \in \mathcal{F}_{\sigma_3}$, $\overrightarrow{\mathrm{succ}}(\sigma_3, F) = \sigma_4$, and $\mu = 0.1$. We calculate the probability of a transition being made from σ_3 to σ_4 as

$$\begin{aligned} &\mathsf{PFV}(\langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5\rangle, \langle \star, \star, \star, \star, \star, \star, 1, 0, 0, 0\rangle) \\ &\quad = 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot 1 \cdot \mathsf{PMF}(1, 1) \cdot \mathsf{PMF}(0, 0) \cdot \mathsf{PMF}(0, 0) \cdot \mathsf{PMF}(5, 0) \\ &\quad = (0.1^1 \cdot 0.9^0 \cdot 1) \cdot (1) \cdot (1) \cdot (0.1^0 \cdot 0.9^5 \cdot 1) = 0.059049 \end{aligned}$$

We now have everything we need to fully describe the evolution of the global state of a population model over time. An execution path of a population model $\mathcal{P}$ is an infinite sequence of global states $\omega = \sigma_0\sigma_1\sigma_2\sigma_3\cdots$, where σ_0 is called the *initial state*, and $\sigma_{k+1} \in next(\sigma)$ for all $k \geqslant 0$.

5.3 Synchronisation

When all oscillators in a population model have the same phase in a global state we say that the state is *synchronised*. Formally, a global state $\sigma = \langle k_1, \ldots, k_T\rangle$ is *synchronised* if, and only if, there is some $\Phi \in \{1, \ldots, T\}$ such that $k_\Phi = N$, and hence $k_{\Phi'} = 0$ for all $\Phi' \neq \Phi$. We will often want to reason about whether some particular run ω of a model leads to a global state that is synchronised. We say that a path $\omega = \sigma_0\sigma_1\cdots$ synchronises if, and only if, there exists some $k \geqslant 0$ such that σ_k is synchronised. Once a synchronised global state is reached any successor states will also be synchronised. Finally we can say that a model synchronises if, and only if, all runs of the model synchronise.

5.4 Model construction

Given a population model $\mathcal{P} = (\Delta, N, T, R, \epsilon, \mu)$ we construct a DTMC $D(\mathcal{P}) = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ where L ranges over the singleton $\{synch\}$. We define the set of states $\mathcal{S}$ to be $\Gamma(\mathcal{P}) \cup \{\sigma_I\}$, where σ_I is the initial state of the DTMC. For each $\sigma = \langle k_1, \ldots, k_T\rangle \in \Gamma(\mathcal{P})$, we set $L(\sigma) = \{synch\}$ if there is some $1 \leqslant i \leqslant T$ such that $k_i = N$.

To calculate the likelihood of permutations of a population of oscillators we will make use of multinomial coefficients, for which we provide the standard definition.

Definition 13 The *multinomial coefficient* is an extension of the binomial coefficient that gives the number of ordered permutations of the elements of a multiset. Given a finite multiset $\mathcal{M}$, a permutation is an ordered arrangement of its elements, where each element appears a number of times equal to its multiplicity in $\mathcal{M}$. The number of permutations of $\mathcal{M}$ is given by

$$\binom{n}{m_1, m_2, \ldots, m_i} = \frac{n!}{m_1!m_2!\cdots m_i!} = \binom{m_1}{m_1}\binom{m_1+m_2}{m_2}\cdots\binom{m_1+m_2+\ldots+m_i}{m_i},$$

where $m_1, \ldots m_i$ are the multiplicities of the elements of $\mathcal{M}$ and n is the sum of those multiplicities.

In the initial state all oscillators are *unconfigured*. That is, oscillators have not yet been assigned a value for their phase. For each $\sigma = \langle k_1, \ldots, k_T\rangle \in \mathcal{S} \setminus \{\sigma_I\}$ we define

$$\mathbf{P}(\sigma_I, \sigma) = \frac{1}{T^N}\binom{N}{k_1, \ldots, k_T} \tag{17}$$

to be the probability of moving from σ_I to a state where k_i arbitrary oscillators are configured with the phase value i for $1 \leqslant i \leqslant T$. The multinomial coefficient defines the number of possible assignments of phases to distinct oscillators that result in the global state σ. The fractional coefficient normalises the multinomial coefficient with respect to the total number of possible assignments of phases to all oscillators. In general, given an arbitrary set of initial configurations (global states) for the oscillators, the total number of possible phase assignments can be calculated by computing the sum of the multinomial coefficients for each

configuration (global state) in that set. Since Γ is the set of all possible global states, we have that

$$\sum_{\langle k_1,\ldots,k_T\rangle\in\Gamma} \binom{N}{k_1,\ldots,k_T} = T^N.$$

We assign probabilities to the transitions as follows: for every $\sigma \in \mathcal{S} \setminus \{\sigma_I\}$, we consider each $F \in \mathcal{F}_\sigma$, and set $\mathbf{P}(\sigma, \overrightarrow{\mathrm{succ}}(\sigma, F)) = \mathsf{PFV}(\sigma, F)$. For every combination of σ and σ' where $\sigma' \notin \mathit{next}(\sigma)$ we set $\mathbf{P}(\sigma, \sigma') = 0$.

5.5 Model reduction

We now describe a reduction of the population model that results in a significant decrease in the size of the model, but is equivalent to the original model with respect to the unbounded-time reachability of synchronised states.

We first distinguish between states where one or more oscillators are about to fire, and states where no oscillators will fire at all. We refer to these states as *firing states* and *non-firing* states respectively. In the following, we fix a population model $\mathcal{P}$ and simply refer to the states by Γ, instead of $\Gamma(\mathcal{P})$.

Definition 14 Given a population model $\mathcal{P}$, a global state $\langle k_1, \ldots, k_T\rangle \in \Gamma$ is a *firing state* if, and only if, $k_T > 0$. We denote by Γ^{F} the set of all firing states of $\mathcal{P}$, and denote by $\Gamma^{\mathsf{NF}} = \Gamma \setminus \Gamma^{\mathsf{F}}$ the set of all non-firing states of $\mathcal{P}$. We will again omit $\mathcal{P}$ if it is clear from the context

The reduction is constructed by collapsing deterministic sequences of transitions. This allows non-firing states to be eliminated from the model. While this approach may seem simple, to the best of our knowledge such a reduction is not automatically applied by existing tools for the automatic analysis of such a model, for example the model checkers PRISM [34], Storm [18], and IscasMC [27]. In addition, collapsing sequences of deterministic transitions from the initial unconfigured state to firing states is not straightforward, and it is unclear how this could be easily inferred by automated tools.

Given a DTMC $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ let $|\mathbf{P}| = |\{(t, t') \mid t, t' \in \mathcal{S}^2 \text{ and } \mathbf{P}(t, t') > 0\}|$ be the number of non-zero transitions in $\mathbf{P}$, and $|D| = |\mathcal{S}| + |\mathbf{P}|$ be the total number of states and non-zero transitions in D.

Theorem 1 *For every population model $\mathcal{P}$ and its corresponding DTMC $D(\mathcal{P}) = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$, there is a reduced model $D'(\mathcal{P}) = (\mathcal{S}', \sigma_I, \mathbf{P}', L')$ where $|D'(\mathcal{P})| < |D(\mathcal{P})|$ and unbounded-time reachability properties with respect to synchronised firing states in $D(\mathcal{P})$ are preserved in $D'(\mathcal{P})$. In particular, the states and transitions in $D(\mathcal{P})$ are reduced in $D'(\mathcal{P})$ such that $\mathcal{S}' = \mathcal{S} \setminus \Gamma^{NF}$ and*

$$|\mathcal{S}'| = 1 + \frac{T^{(N-1)}}{(N-1)!},$$
$$|\mathbf{P}'| \leqslant |\mathbf{P}| - 2|\Gamma^{NF}|$$

where $x^{(n)}$ is the rising factorial $x^{(n)} = x(x+1)\cdots(x+n-1)$.

Observe that only unbounded-time reachability properties are preserved in the reduction of a model. In the unreduced model precisely T transitions correspond to one oscillation of a

standalone oscillator. For the reduced model this is not the case, since the length of removed deterministic transitions sequences are not encoded in the reduction.

We now proceed to prove this theorem. To that end, we need some preliminary properties of non-firing states and their relation to firing states.

Lemma 3 *Every non-firing state $\sigma \in \Gamma^{NF}$ has exactly one successor state, and in that state all oscillator phases have increased by* 1.

Proof Given a non-firing state $\sigma = \langle k_1, \ldots, k_T \rangle$ observe that as $k_T = 0$ there is only one possible failure vector for σ, namely $\{\star\}^T$. The set of all successor states of σ is then the singleton $\{\overrightarrow{\text{succ}}(\sigma, \{\star\}^T)\}$. By construction we can then see that $update^{\Phi}(\sigma, \{\star\}^T) = 1$ and $\mathcal{U}_{\Phi}(\sigma, \{\star\}^T) = \{\Phi - 1\}$ for $1 \leqslant \Phi \leqslant T$. The single successor state is then given by $\overrightarrow{\text{succ}}(\sigma, \{\star\}^T) = \langle 0, k_1, \ldots, k_{T-1} \rangle$. □

Corollary 1 *A transition from any non-firing state is taken deterministically, since for any $\sigma \in \Gamma^{NF}$ we have $PFV(\sigma, \{\star\}^T) = 1$.*

Reachable state reduction Given a path $\omega = \sigma_0 \cdots \sigma_{n-1}\sigma_n$ where $\sigma_i \in \Gamma^{\mathsf{NF}}$ for $0 < i < n$ and $\sigma_0, \sigma_n \in \Gamma^{\mathsf{F}}$, we omit transitions (σ_i, σ_{i+1}) for $0 \leqslant i < n$, and instead introduce a direct transition from σ_0, the first firing state, to σ_n, the next firing state in the sequence. For any $\sigma = \langle k_1, \ldots, k_T \rangle \in \Gamma$ let $\delta_\sigma = \max\{\Phi \mid k_\Phi > 0 \text{ and } 1 \leqslant \Phi \leqslant T\}$ be the highest phase of any oscillator in σ. The successor state of a non-firing state is then the state where all phases have increased by $T - \delta_\sigma$. Observe that $T - \delta_\sigma = 0$ for any $\sigma \in \Gamma^{\mathsf{F}}$.

Definition 15 The *deterministic successor function* $\overrightarrow{\text{succ}} : \Gamma \to \Gamma^{\mathsf{F}}$, given by

$$\overrightarrow{\text{succ}}(\langle k_1, \ldots, k_T \rangle) = \{0\}^{T-\delta_\sigma} \frown \langle k_1, \ldots, k_{\delta_\sigma} \rangle,$$

maps a state $\sigma \in \Gamma$ to the next firing state reachable by taking $T - \delta_\sigma$ deterministic transitions. Observe that for any firing state σ we have $\delta_\sigma = T$, and hence that $\overrightarrow{\text{succ}}(\sigma) = \sigma$.

We now update the definition for the set of all successor states for some global state $\sigma \in \Gamma$ to incorporate the deterministic successor function.

Definition 16 Given a global state $\sigma \in \Gamma$, we define $\overrightarrow{next}(\sigma)$ to be the set of all successor states of σ, where

$$\overrightarrow{next}(\sigma) = \{\overrightarrow{\text{succ}}(\overrightarrow{\text{succ}}(\sigma, F)) \mid F \in \mathcal{F}_\sigma\}.$$

Definition 17 Given a firing state $\sigma \in \Gamma^{\mathsf{F}}$ let $pred(\sigma)$ be the set of all non-firing predecessors of σ, where σ is reachable from the predecessor by taking some positive number of transitions deterministically. Formally,

$$pred(\sigma) = \{\sigma' \mid \sigma' \in \Gamma^{\mathsf{NF}} \text{ and } \overrightarrow{\text{succ}}(\sigma') = \sigma\}.$$

We refer to all states $\sigma' \in pred(\sigma)$ as *deterministic predecessors* of σ.

Then given $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ with $\mathcal{S} = \{\sigma_I\} \cup \Gamma$, we define $\mathcal{S}' = \mathcal{S} \setminus \bigcup_{\sigma \in \Gamma^{\mathsf{F}}} pred(\sigma)$ to be the reduction of $\mathcal{S}$ where all non-firing states from which a firing state can be reached deterministically are removed.

Lemma 4 *For any $D(\mathcal{P}) = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ with $\mathcal{S} = \Gamma \cup \{\sigma_I\}$, the reduction $\mathcal{S}'$ is equal to $\Gamma^F \cup \{\sigma_I\}$.*

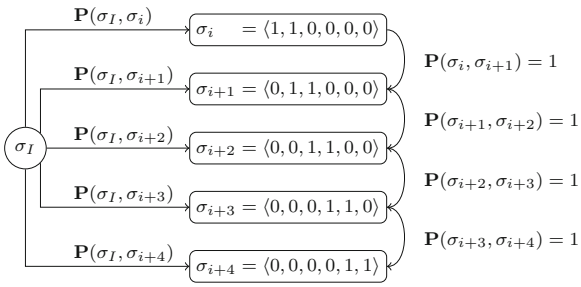


Fig. 4 Five possible initial configurations in $\mathcal{S}$ for $N = 2$, $T = 6$

Proof Let $P = \bigcup_{\sigma \in \Gamma^{\mathsf{F}}} pred(\sigma)$ be the set of all predecessors of firing states in Γ^{F}. Since $\mathcal{S} = \Gamma \cup \{\sigma_I\}$ and $\mathcal{S}' = \mathcal{S} \setminus P$ we can see that $\mathcal{S}' = \Gamma^{\mathsf{F}} \cup \{\sigma_I\}$ if, and only if, $P = \Gamma^{\mathsf{NF}}$. From Definition 17 it follows that $P \subseteq \Gamma^{\mathsf{NF}}$. In addition, for any $\sigma \in \Gamma^{\mathsf{NF}}$ there is some state σ' such that $\sigma \in pred(\sigma')$ and $\sigma' = \overrightarrow{\mathrm{succ}}(\sigma) \in \Gamma^{\mathsf{F}}$, hence $\Gamma^{\mathsf{NF}} \subseteq P$ and the lemma is proved. □

Lemma 5 *For a population model $\mathcal{P} = (\Delta, N, T, R, \epsilon, \mu)$ and its corresponding DTMC $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ with $\mathcal{S} = \Gamma \cup \{\sigma_I\}$, the number of states in the reduction of $\mathcal{S}$ is given by $|\mathcal{S}'| = 1 + \frac{T^{(N-1)}}{(N-1)!}$.*

Proof Observe that there are $\binom{N+T-1}{N}$ ways to assign T distinguishable phases to N indistinguishable oscillators [23]. Since $\mathcal{S} = \Gamma \cup \{\sigma_I\}$ and Γ is the set of all possible configurations for oscillators we can see that $|\mathcal{S}| = \binom{N+T-1}{N} + 1$. For any non-firing state $\sigma = \langle k_1, \ldots k_T \rangle \in \Gamma^{\mathsf{NF}}$ we know from Definition 7 that $\sum_{\Phi=1}^{T} k_\Phi = N$ and from Definition 14 that $k_T = 0$, so it must be the case that $\sum_{\Phi=1}^{T-1} k_\Phi = N$. That is, there must be $\binom{N+T-2}{N}$ ways to assign $T-1$ distinguishable phases to N indistinguishable oscillators, and so $|\Gamma^{\mathsf{NF}}| = \binom{N+T-2}{N}$. From Lemma 4 we know that $\mathcal{S}' = \mathcal{S} \setminus \Gamma^{\mathsf{NF}}$ so it must be the case that $|\mathcal{S}'| = |\mathcal{S}| - |\Gamma^{\mathsf{NF}}| = 1 + \binom{N+T-1}{N} - \binom{N+T-2}{N} = 1 + \frac{T^{(N-1)}}{(N-1)!}$. □

Transition matrix reduction Here we describe the reduction in the number of non-zero transitions in the model. We ilustrate how initial transitions to non-firing states are removed by using a simple example, and then describe how we remove transitions from firing states to any successor non-firing states..

Figure 4 shows five possible initial configurations $\sigma_i, \ldots, \sigma_{i+4} \in \mathcal{S}$ for $N = 2$ oscillators with $T = 6$ values for phase, where a transition is taken from σ_I to each σ_k with probability $\mathbf{P}(\sigma_I, \sigma_k)$. Any infinite run of D where a transition is taken from σ_I to one of the configured states $\sigma_i, \ldots, \sigma_{i+3}$ will pass through σ_{i+4}, since all transitions $(\sigma_{i+k}, \sigma_{i+k+1})$ for $0 \leqslant k \leqslant 3$ are taken deterministically. Also, observe that states $\sigma_i, \ldots, \sigma_{i+3}$ are not in $\mathcal{S}'$, since σ_{i+4} is reachable from each by taking some number of deterministic transitions. We therefore set the probability of moving from σ_I to σ_{i+4} in $\mathbf{P}'$ to be the sum of the probabilities of moving from σ_I to σ_{i+4} and each of its predecessors in $\mathbf{P}$. Generally, given a state $\sigma \in \mathcal{S}'$ where $\sigma \neq \sigma_I$, we set $\mathbf{P}'(\sigma_I, \sigma) = \mathbf{P}(\sigma_I, \sigma) + \sum_{\sigma' \in pred(\sigma)} \mathbf{P}(\sigma_I, \sigma')$.

We now define how we calculate the probability with which a transition is taken from a firing state to each of its possible successors. For each firing state $\sigma \in \mathcal{S}'$ we consider each possible successor $\sigma' \in \overrightarrow{next}(\sigma)$ of σ and define $\mathcal{F}_{\sigma \to \sigma'}$ to be the set of all possible failure vectors for σ for which the successor of σ is σ', given by $\mathcal{F}_{\sigma \to \sigma'} = \{F \in \mathcal{F}_\sigma \mid$

$\overrightarrow{\text{succ}}(\overrightarrow{\text{succ}}(\sigma, F)) = \sigma'\}$. We then set the probability with which a transition from σ to σ' is taken to $\mathbf{P}'(\sigma, \sigma') = \sum_{F \in \mathcal{F}_{\sigma \to \sigma'}} \mathsf{PFV}(\sigma, F)$.

Lemma 6 *For a population model $\mathcal{P} = (\Delta, N, T, R, \epsilon, \mu)$, the corresponding DTMC $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ with $\mathcal{S} = \{\sigma_I\} \cup \Gamma$, and its reduction $D'(\mathcal{P}) = (\mathcal{S}', \sigma_I, \mathbf{P}', L')$, the transitions in $\mathbf{P}$ are reduced in $\mathbf{P}'$ such that $|\mathbf{P}'| \leqslant |\mathbf{P}| - 2|\Gamma^{\mathsf{NF}}|$*

Proof From Lemma 4 we know that $|\mathcal{S}'| = |\mathcal{S} \setminus \Gamma^{\mathsf{NF}}|$, and hence that $|\Gamma^{\mathsf{NF}}|$ transitions from σ_I to non-firing states are not in $\mathbf{P}'$, and from Lemma 3 we also know that there is one transition from each non-firing state to its unique successor state that is not in $\mathbf{P}'$. Since no additional transitions are introduced in the reduction it is clear that $|\mathbf{P}'| \leqslant |\mathbf{P}| - 2|\Gamma^{\mathsf{NF}}|$. □

Lemma 7 *For every population model DTMC $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$, unbounded-time reachability properties with respect to synchronised firing states in D are preserved in its reduction D'.*

Proof We want to show that for every $\bowtie \in \{<, \leqslant, \geqslant, >\}$ and every $\lambda \in [0, 1]$, if $\sigma_I \models \mathrm{P}_{\bowtie\lambda}[\mathrm{F}\ synch]$ holds in D then it also holds in D'. From the semantics of PCTL over a DTMC we have

$$\sigma_I \models \mathrm{P}_{\bowtie\lambda}[\mathrm{F}\ synch] \quad \Leftrightarrow \quad Pr\{\omega \in Paths^D \mid \omega \models \mathrm{F}\ synch\} \bowtie \lambda.$$

Therefore we need to show that

$$Pr^D\{\omega \in Paths^D \mid \omega \models \mathrm{F}\ synch\} =_\sigma Pr^{D'}\{\omega' \in Paths^{D'} \mid \omega' \models \mathrm{F}\ synch\},$$

where Pr^D and $Pr^{D'}$ denote the probability measures with respect to the sets of infinite paths from σ_I in D and D' respectively.

Given a firing state $\sigma^{\mathsf{F}} \in \mathcal{S}$ we denote by $Paths^D_{\sigma^{\mathsf{F}}}$ the set of all infinite paths of D starting in σ_I where the first firing state reached along that path is σ^{F}. All such sets for all firing states in $\mathcal{S}$ form a partition, such that $\bigcup_{\sigma^{\mathsf{F}} \in \Gamma^{\mathsf{F}}} Paths^D_{\sigma^{\mathsf{F}}} = Paths^D$. That is, for all firing states $\sigma^{\mathsf{F}}, \sigma^{\mathsf{F}\prime} \in \mathcal{S}$ where $\sigma^{\mathsf{F}} \neq \sigma^{\mathsf{F}\prime}$ we have that $Paths^D_{\sigma^{\mathsf{F}}} \cap Paths^D_{\sigma^{\mathsf{F}\prime}} = \emptyset$.

Now observe that any infinite path ω of D can be written in the form $\omega = \sigma_I \omega_1^{\mathsf{NF}} \sigma_1^{\mathsf{F}} \omega_2^{\mathsf{NF}} \sigma_2^{\mathsf{F}} \cdots$ where σ_i^{F} is the ith firing state in the path and each $\omega_i^{\mathsf{NF}} = \sigma_i^1 \sigma_i^2 \cdots \sigma_i^{k_i}$ is a possibly empty sequence of k_i non-firing states. Then for every such path in D there is a corresponding path ω' of D' without non-firing states, and of the form $\omega' = \sigma_I \sigma_1^{\mathsf{F}} \sigma_2^{\mathsf{F}} \sigma_3^{\mathsf{F}} \cdots$, as for any i we have $\sigma_i^j \in pred(\sigma_i^{\mathsf{F}})$ for all $1 \leqslant j \leqslant k^i$. As only deterministic transitions have been removed in D' we can see that $Pr^D_{\sigma_1^{\mathsf{F}}}\{\sigma_1^{\mathsf{F}} \omega_2^{\mathsf{NF}} \sigma_2^{\mathsf{F}} \cdots\} = Pr^{D'}_{\sigma_1^{\mathsf{F}}}\{\sigma_1^{\mathsf{F}} \sigma_2^{\mathsf{F}} \sigma_3^{\mathsf{F}} \cdots\}$. Hence, we only have to consider the finite paths from σ_I to σ_1^{F}. To that end, observe that there are $\left|pred(\sigma_1^{\mathsf{F}})\right|$ possible prefixes for each path from σ_I to σ_1^{F} where the initial transition is taken from σ_I to some non-firing predecessor of σ_1^{F}, plus the single prefix where the initial transition is taken to σ_1^{F} itself. Overall there are exactly $\left|pred(\sigma_1^{\mathsf{F}})\right| + 1$ distinct finite prefixes that have ω' as their corresponding path in D'. We denote the set of these prefixes for a path ω' in D' by $Pref(\omega')$. Since the measure of each finite prefix extends to a measure over the set of infinite paths sharing that prefix, it is sufficient to show that the sum of the probabilities for these finite prefixes is equal to the probability of the unique prefix $\sigma_0, \sigma_1^{\mathsf{F}}$ of ω', that is $Pr^D Pref(\omega') = Pr^{D'}\{\sigma_I, \sigma_1^{\mathsf{F}}\}$. We can then write

$$Pr^D Pref(\omega') = \mathbf{P}(\sigma_I, \sigma_1^{\mathsf{F}}) + \sum_{\sigma' \in pred(\sigma_1^{\mathsf{F}})} \mathbf{P}(\sigma_I, \sigma') \cdot 1^{k_{\sigma'}}$$

$$= \mathbf{P}(\sigma_I, \sigma_1^{\mathsf{F}}) + \sum_{\sigma' \in pred(\sigma_1^{\mathsf{F}})} \mathbf{P}(\sigma_I, \sigma'),$$

where $k_{\sigma'}$ is the number of deterministic transitions that lead from σ' to σ_1^{F} in D. Now recall that for any $\sigma \in \mathcal{S}' \setminus \{\sigma_I\}$ we have

$$\mathbf{P}'(\sigma_I, \sigma) = \mathbf{P}(\sigma_I, \sigma) + \sum_{\sigma' \in pred(\sigma)} \mathbf{P}(\sigma_I, \sigma).$$

So we have shown that $Pr^D Pref(\omega') = Pr^{D'}\{\sigma_I, \sigma_1^{\mathsf{F}}\}$ and the lemma is proved. □

Proof of Theorem 1 Follows from Lemmas 5 and 6 for the reduction of states and transitions respectively, and from Lemma 7 for the preservation of unbounded time reachability properties. □

5.6 Reward structures for reductions

We now show how a reward structure for a reduced model can be derived from any reward structure for the unreduced model, and prove that these reward structures are equivalent with respect to the reachability of synchronised firing states.

Theorem 2 *For every population model $\mathcal{P}$ with corresponding DTMC $D = (\mathcal{S}, \sigma_I, \mathbf{P}, L)$ and a reduction $D' = (\mathcal{S}', \sigma_I, \mathbf{P}', L')$ of D, and for every reward structure $\mathcal{R} = (R_s, R_t)$ for D, there is a reward structure $\mathcal{R}' = (R'_s, R'_t)$ for D' such that unbounded-time reachability reward properties with respect to synchronised firing states in D are preserved in D'.*

Given any reward structure $\mathcal{R} = (R_s, R_t)$ on D we construct the corresponding reward structure $\mathcal{R}' = (R'_s, R'_t)$ as follows:

- There is no reward for the initial state and we set $R_s(\sigma_I) = 0$.
- For every firing state σ^{F} in $\mathcal{S}$ with $R_s(\sigma^{\mathsf{F}}) = r$ we set $R'_s(\sigma^{\mathsf{F}}) = r$.
- For every pair of distinct firing states $\sigma_1^{\mathsf{F}}, \sigma_2^{\mathsf{F}} \in \mathcal{S}'$, where there is a non-zero transition from σ_1^{F} to σ_2^{F} in D', there is a (possibly empty) sequence $\sigma_1^{\mathsf{NF}} \cdots \sigma_k^{\mathsf{NF}}$ of k deterministic predecessors of σ_2^{F} in $\mathcal{S}$ such that $k > 0$ implies $\mathbf{P}(\sigma_1^{\mathsf{F}}, \sigma_1^{\mathsf{NF}}) > 0$, $\mathbf{P}(\sigma_k^{\mathsf{NF}}, \sigma_2^{\mathsf{F}}) = 1$, and $\mathbf{P}(\sigma_i^{\mathsf{NF}}, \sigma_{i+1}^{\mathsf{NF}}) = 1$ for $1 \leqslant i < k$. We set the reward for taking the transition from σ_1^{F} to σ_2^{F} in D' to be the sum of the rewards that would be accumulated across that sequence by a path in D, formally

$$R'_t(\sigma_1^{\mathsf{F}}, \sigma_2^{\mathsf{F}}) = \mathrm{tot}_{\mathcal{R}}(\sigma_1^{\mathsf{F}} \sigma_1^{\mathsf{NF}} \cdots \sigma_k^{\mathsf{NF}} \sigma_2^{\mathsf{F}}).$$

- For every firing state σ^{F} in $\mathcal{S}'$ there is a non-zero transition from the initial state σ_I to σ^{F} in $\mathbf{P}'$. Therefore, all paths of D' where σ^{F} is the first firing state along that path share the same prefix, namely $\sigma_I, \sigma^{\mathsf{F}}$. For paths of D this is not necessarily the case, since σ^{F} is the first firing state not only along the path where the initial transition is taken to σ^{F} itself, but also along any path where the initial transition is taken to a non-firing state from which a sequence of deterministic transitions leads to σ^{F} (that state is a deterministic predecessor of σ^{F}). We therefore set the reward along a path $\omega' = \sigma_I \sigma_1^{\mathsf{F}} \sigma_2^{\mathsf{F}} \cdots$ for taking the initial transition to σ^{F} in D' to be the sum of the total rewards accumulated along all distinct path prefixes of the form $\sigma_I \omega^{\mathsf{NF}} \sigma^{\mathsf{F}}$, normalised by the total probabilitiy of taking any of these paths, where ω^{NF} is a possibly empty sequence of deterministic predecessors of

σ^{F}, and where the total reward for each prefix is weighted by the probability of taking the transitions along that sequence,

$$R'_i(\sigma_I, \sigma^{\mathsf{F}}) = \frac{\sum_{\omega_{pre} \in Pref(\omega')} \mathrm{tot}_{\mathcal{R}}(\omega_{pre}) Pr^D\{\omega_{pre}\}}{Pr^{D'}\{\sigma_I \sigma_1^{\mathsf{F}}\}} \tag{18}$$

Proof of Theorem 2 We want to show that for every reward structure $\mathcal{R}$ for D and corresponding reward structure $\mathcal{R}'$ for D', every $\bowtie \in \{<, \leqslant, \geqslant, >\}$ and every $r \in \mathbb{R}$, if $\sigma_I \models \mathrm{R}_{\bowtie r}[\mathrm{F}\ synch]$ holds in D then it also holds in D'. Let $V_{Sat(\mathrm{F}synch)}$ and $V'_{Sat(\mathrm{F}synch)}$ respectively denote the random variables over $Paths^D$ and $Paths^{D'}$ whose expectations correspond to $\mathcal{R}$ and $\mathcal{R}'$. From the semantics of PCTL over a DTMC we have

$$\begin{aligned} \sigma_I \models \mathrm{R}_{\bowtie r}[\mathrm{F}\ synch] \quad &\Leftrightarrow E[V_{Sat(synch)}] \bowtie r \\ &\Leftrightarrow \sum_{\omega \in Paths} V_{Sat(synch)}(\omega) Pr^D\{\omega\} \bowtie r. \end{aligned}$$

Therefore, we need to show that

$$\sum_{\omega \in Paths^D} V_{Sat(synch)}(\omega) Pr^D\{\omega\} = \sum_{\omega' \in Paths^{D'}} V'_{Sat(synch)}(\omega') Pr^{D'}\{\omega'\}, \tag{19}$$

where Pr^D and $Pr^{D'}$ denote the probability measures with respect to the sets of infinite paths from σ_I in D and D' respectively. There are two cases:

Firstly, if there exists some path of D that does not synchronise then by definition $V_{Sat(synch)} = \infty$. Also, from Lemma 7 we know that there is a corresponding path of D' that does not synchronise, and hence that $V'_{Sat(synch)} = \infty$. By definition the probability measure of all paths of D and D' are strictly positive. Therefore, all summands of Equation 19 are defined, and the expectation of both $V_{Sat(synch)}$ and $V'_{Sat(synch)}$ is ∞.

Secondly, we consider the case where all possible paths of D and D' synchronise. First we define the function $reduce : Paths^D \to Paths^{D'}$ that maps paths of D to their corresponding path in the reduction D',

$$reduce(\sigma_I \omega_1^{\mathsf{NF}} \sigma_1^{\mathsf{F}} \omega_2^{\mathsf{NF}} \sigma_2^{\mathsf{F}} \cdots) = \sigma_I \sigma_1^{\mathsf{F}} \sigma_2^{\mathsf{F}} \cdots,$$

where ω_i^{NF} is the (possibly empty) sequence of deterministic predecessors of the firing state σ_i^{F}. Let $reduce^{-1}(\omega)$ denote the preimage of ω under $reduce$. Then, we can rewrite the left side of (19) to

$$\sum_{\omega' \in Paths^{D'}} \sum_{\omega \in reduce^{-1}(\omega')} V_{Sat(synch)}(\omega) Pr^D\{\omega\}.$$

For any path ω of D or D' let $pre_s(\omega)$ be the prefix of that path whose last state is the first firing state along that path that is in the set $Sat(synch)$. So we want to show that the following holds for any path ω' of D',

$$\begin{aligned} \sum_{\omega \in reduce^{-1}(\omega')} V_{Sat(synch)}(\omega) Pr^D\{\omega\} &= V'_{Sat(synch)}(\omega') Pr^{D'}\{\omega'\} \\ \sum_{\omega \in reduce^{-1}(\omega')} \mathrm{tot}_{\mathcal{R}}(pre_s(\omega)) Pr^D\{\omega\} &= \mathrm{tot}_{\mathcal{R}'}(pre_s(\omega')) Pr^{D'}\{\omega'\}. \end{aligned} \tag{20}$$

Given some path ω let $\omega[i : j]$ denote the sequence of states in ω from the ith firing state to the jth firing state along that path (inclusively). The notation $\omega[- : j]$ indicates that no states are removed from the start of the path i.e. the first state is σ_I, and the notation $\omega[i : -]$ indicates that no states are removed from the end of the path. By recalling that $Pr(\sigma_0\sigma_1 \cdots \sigma_n) = \prod_{i=1}^{n} \mathbf{P}(\sigma_{i-1}, \sigma_i)$ we can see that $Pr(\sigma_0\sigma_1 \cdots \sigma_n) = Pr(\sigma_0 \cdots \sigma_i)Pr(\sigma_i \cdots \sigma_n)$ for any $0 < i < n$. Also from (1) it is clear that for any reward structure $\mathcal{R}$, $\text{tot}_{\mathcal{R}}(\sigma_I \cdots \sigma_n) = \text{tot}_{\mathcal{R}}(\sigma_I \cdots \sigma_i) + \text{tot}_{\mathcal{R}}(\sigma_i \cdots \sigma_n)$ holds for all $0 < i < n$. Now we can rewrite (20) to

$$\begin{aligned}&\sum_{\omega \in reduce^{-1}(\omega')} \left(\text{tot}_{\mathcal{R}}(pre_s(\omega)[- : 1]) + \text{tot}_{\mathcal{R}}(pre_s(\omega)[1 : -])\right) Pr^D\{\omega[- : 1]\} \\ &= \left(\text{tot}_{\mathcal{R}'}(pre_s(\omega')[- : 1]) + \text{tot}_{\mathcal{R}'}(pre_s(\omega')[1 : -])\right) Pr^{D'}\{\omega'[- : 1]\}.\end{aligned} \tag{21}$$

By the definition of $\mathcal{R}'$ we can write the right hand side of (21) as

$$\begin{aligned}&\left(\left(\frac{\sum_{\omega_{pre} \in Pref(\omega')} \text{tot}_{\mathcal{R}}(\omega_{pre}) Pr^D\{\omega_{pre}\}}{Pr^{D'}\{\omega'[- : 1]\}}\right) + \text{tot}_{\mathcal{R}'}(pre_s(\omega')[1 : -])\right) Pr^{D'}\{\omega'[- : 1]\} \\ &= \sum_{\omega_{pre} \in Pref(\omega')} \left(\text{tot}_{\mathcal{R}}(\omega_{pre}) Pr^D\{\omega_{pre}\}\right) + \text{tot}_{\mathcal{R}'}(pre_s(\omega')[1 : -]) Pr^{D'}\{\omega'[- : 1]\}.\end{aligned}$$

From Lemma 7 we know that

$$Pr^{D'}\{\omega'[- : 1]\} = Pr^D Pref(\omega') = \sum_{\omega_{pre} \in Pref(\omega')} Pr^D\{\omega_{pre}\},$$

and hence obtain

$$\begin{aligned}&\sum_{\omega_{pre} \in Pref(\omega')} \left(\text{tot}_{\mathcal{R}}(\omega_{pre}) Pr^D\{\omega_{pre}\}\right) + \sum_{\omega_{pre} \in Pref(\omega')} \text{tot}_{\mathcal{R}'}(pre_s(\omega')[1 : -]) Pr^D\{\omega_{pre}\} \\ &= \sum_{\omega_{pre} \in Pref(\omega')} \left(\text{tot}_{\mathcal{R}}(\omega_{pre}) + \text{tot}_{\mathcal{R}'}(pre_s(\omega')[1 : -])\right) Pr^D\{\omega_{pre}\}.\end{aligned} \tag{22}$$

Since $Pref(\omega')$ is the set of all possible finite prefixes from the initial state σ_I to the first firing state σ_1^{F}, and since $\omega[- : 1] = pre_s(\omega)[- : 1]$ clearly holds, we know that

$$\bigcup_{\omega_{pre} \in Pref(\omega')} \{\omega_{pre}\} = \bigcup_{\omega \in reduce^{-1}(\omega')} \{\omega[- : 1]\}. = \bigcup_{\omega \in reduce^{-1}(\omega')} \{pre_s(\omega)[- : 1]\}.$$

Using this fact, and by observing that by definition

$$\text{tot}_{\mathcal{R}'}(pre_s(\omega')[1 : -]) = \text{tot}_{\mathcal{R}}(pre_s(\omega)[1 : -]),$$

we can write (22) as

$$\sum_{\omega \in reduce^{-1}(\omega')} \left(\text{tot}_{\mathcal{R}}(pre_s(\omega)[- : 1]) + \text{tot}_{\mathcal{R}'}(pre_s(\omega')[1 : -])\right) Pr^D\{\omega[- : 1]\}.$$

This is the same as the left hand side of (21) and the theorem is proved. □

6 Connecting the concrete model and the population model

In this section, we define the abstraction function to connect a concrete model with a population model. To that end, let $D_c = (\mathcal{S}_c, s_I, \mathbf{P}_c)$ be a concrete model of a network of N PCOs with a clock cycle length T, a refractory period R, a phase response function Δ, a coupling ϵ and broadcast failure probability of μ. Furthermore, let $D_p = (\mathcal{S}_p, \sigma_I, \mathbf{P}_p)$ be the DTMC of a population model for the same parameters. For a finite path $\omega = \sigma_0, \ldots, \sigma_n$ we denote its last element by $last(\omega) = \sigma_n$.

The correspondence we want to show is that the initial states of the two models are weakly bisimulation equivalent [7]. Since we do not employ any actions except for the silent action in our models, we first define a simplified version of weak bisimulations. To that end, we use the definition of Baier and Hermanns [7], but with slightly altered notations to fit to our setting, and by ignoring all references to sequences of actions.

Definition 18 (Weak Bisimulation [7]) A *weak bisimulation* on $D = (\mathcal{S}, s_I, \mathbf{P})$ is an equivalence relation R on $\mathcal{S}$ such that for all $(s, s') \in R$, and all equivalence classes $E \in \mathcal{S}/R$, we have

$$\mathbf{P}(s, E) = \mathbf{P}(s', E)$$

where $\mathbf{P}(s, E) = \sum_{\omega \in Paths_f^D(s) \wedge last(\omega) \in E} \mathbf{P}(\omega)$ for any set $E \subseteq \mathcal{S}$. We say that two states s and s' are weakly bisimilar, if, and only if, there is a weak bisimulation R such that $(s, s') \in R$.

6.1 Proving the correspondence between concrete and population models

In this section, we will formally define a weak bisimulation relation between states in the concrete model D_c and the corresponding population model D_p. Since weak bisimulations are defined on a single DTMC, we will define a single DTMC combining both the concrete and the population model.

Definition 19 Let $D_c = (\mathcal{S}_c, s_I, \mathbf{P}_c)$ be a concrete model of a network and let $D_p = (\mathcal{S}_p, \sigma_I, \mathbf{P}_p)$ be the DTMC of a corresponding population model. The *combination of* D_c *and* D_p is the DTMC $D = (\mathcal{S}, i, \mathbf{P})$, where $\mathcal{S} = \mathcal{S}_c \cup \mathcal{S}_p \cup \{i\}$, i is a new state ($i \notin \mathcal{S}_c \cup \mathcal{S}_p$), and $\mathbf{P}$ is defined as follows:

$$\mathbf{P}(s, s') = \begin{cases} \mathbf{P}_c(s, s') & \text{if } s, s' \in \mathcal{S}_c \\ \mathbf{P}_p(s, s') & \text{if } s, s' \in \mathcal{S}_p \\ \frac{1}{2} & \text{if } s = i \text{ and } s' \in \{s_I, \sigma_I\} \\ 0 & \text{otherwise.} \end{cases}$$

The initial transitions from the new state i form a uniform distribution over the two behaviours, and hence the combination DTMC could behave like either model. However, for simplicity we will often refer to the original models D_c and D_p.

We need to associate states in D_c to states in D_p. In general, several concrete states will be mapped to a single population state, since we do not distinguish between different orders of oscillators in the latter, while we do in the former.

Furthermore, we want to abstract from different modes of the oscillators. However, it is not sensible to associate all modes within a phase to the same population state, since in the transitions from one mode to the next the system chooses, whether an oscillator fails to broadcast its pulse or not. If we want to be able to define a probabilistic weak bisimulation

relation, we need to represent the failures described by the transitions in the population model. To have an exact correspondence, we first collect all the concrete states where the counter and all oscillators are at the start mode into a single set.

$$\mathcal{S}_c' = \{s \in \mathcal{S}_c \mid \mathsf{mode}(s, \mathsf{env}) = \mathit{start} \wedge \forall u \colon \mathsf{mode}(s, u) = \mathit{start}\}$$

The abstraction function $h \colon \mathcal{S}_c' \to \mathcal{S}_p$ takes a concrete state s and counts the number of oscillators sharing the same phase, mapping $s = (\mathsf{env}, \mathsf{osc})$ to the corresponding state of the population model,

$$h(s) = \langle |\{u \mid \phi(s, u) = 1\}|, \ldots, |\{u \mid \phi(s, u) = T\}| \rangle.$$

Now we can define the weak bisimulation relation between the models.

Definition 20 Let $D = (\mathcal{S}, i, \mathbf{P})$ be the combination of a concrete model and its population model according to Definition 19. Then let $R' \subseteq \mathcal{S} \times \mathcal{S}$ be defined by

1. $(s_I, \sigma_I) \in R'$ and
2. $(s, \sigma) \in R'$ if, and only if, $s \in \mathcal{S}_c'$, $\sigma \in \mathcal{S}_p$ and $h(s) = \sigma$.

The weak bisimulation $R \subseteq \mathcal{S} \times \mathcal{S}$ is then the reflexive, symmetric and transitive closure of R'.

The partition induced by the relation R has three types of equivalence classes:

1. the class $E_{init} = \{s_I, \sigma_I\}$ consisting of the initial states;
2. classes $E = \{\sigma\} \cup \{s \mid s \in \mathcal{S}_c' \wedge h(s) = \sigma\}$ that contain exactly one state σ from the population model and all states s of the concrete model, whose abstraction is σ and where all components are in the mode *start*;
3. the class $E_\emptyset$ containing all states that are not in relation with other states. This class in particular does *not* contain any state of the population model.

We now proceed to show that this relation indeed is a weak bisimulation on D, with a minor caveat. Consider a state of the population model σ and a state of the concrete model s in the same equivalence class. Then we have $\mathbf{P}(s, E_\emptyset) = 1$, since every transistion starting in s immediately leads into a state in $E_\emptyset$. However, we also have $\mathbf{P}(\sigma, E_\emptyset) = 0$, since none of the successors of σ is an element of $E_\emptyset$. We could remedy this discrepancy by introducing a new state σ' for each state in the population model. Then all of these states would be in $E_\emptyset$, and every σ of the original population model would have a unique successor in $E_\emptyset$. Since this addition does not change the overall behaviour of the population model we chose instead to ignore the transitions ending in $E_\emptyset$, and only consider the probabilities of paths into the other equivalence classes.

Theorem 3 *Let $D_c = (\mathcal{S}_c, s_I, \mathbf{P}_c)$ and $D_p = (\mathcal{S}_p, \sigma_I, \mathbf{P}_p)$ be a concrete network of oscillators and its abstraction as a population model, respectively. Furthermore, let $D = (\mathcal{S}, i, \mathbf{P})$ be the combination of both of these models from Definition 19 and $R \subseteq \mathcal{S} \times \mathcal{S}$ be the relation from Definition 20. Then R is a weak bisimulation relation, and s_I and σ_I are weakly bisimilar.*

Proof **From transition sequences in D_p to transition sequences in D_c.** Let $(s_1, \sigma_1) \in R$, where $s_1 \in \mathcal{S}_c$, $\sigma_1 \in \mathcal{S}_p$, and let E be an equivalence class such that $\mathbf{P}(\sigma_1, E) > 0$, i.e., for the single state $\sigma_2 \in E$ of the population model, we have $\mathbf{P}(\sigma_1, \sigma_2) > 0$. Furthermore, for s_1, we have by the definition of R that $\mathsf{mode}(s_1, \mathsf{env}) = \mathit{start}$, $\mathsf{mode}(s_1, u) = \mathit{start}$ for all $1 \leqslant u \leqslant N$ and $h(s_1) = \langle |\{u \mid \phi(s_1, u) = 1\}|, \ldots, |\{u \mid \phi(s_1, u) = T\}| \rangle = \sigma_1$. Note

that there is only a single outgoing transition from s_1 according to condition (2). That is, in the successor state s of s_1, we have $\mathsf{count}(s) = 0$ and $\mathsf{mode}(s, \mathsf{env}) = \mathit{update}$, while the oscillator states are not changed. To keep the notation tidy, we identify this successor state with s_1 in the following.

Now consider two cases. If $|\{u \mid \phi(s_1, u) = T\}| = 0$, then $\sigma_2 = \langle 0, |\{u \mid \phi(s_1, u) = 1\}|, \ldots, |\{u \mid \phi(s_1, u) = T - 1\}|\rangle$, since no set of oscillators in σ_1 is perturbed by a firing oscillator. In particular, there is no u such that $\phi(s_1, u) = T$. Hence, for all possible successors s' of s_1, we have that only condition (3) is satisfied. Furthermore, this is the case until all oscillators changed their mode to *update*. Let us call this state s_1'. Now, the environment was not changed from s_1 to s_1', i.e., $\mathsf{count}(s_1) = \mathsf{count}(s_1') = 0$.

Hence, condition (9b) is satisfied for all oscillators. Since $\Delta(\Phi, 0, \epsilon) = 0$ for all Φ, the phase of each oscillator is increased by one. This implies that there is a single successor of s_1', which we call s_2 and that for all u, $\phi(s_2, u) = \phi(s_1', u) + 1$. In particular, we have that for all oscillators u, $\phi(s_2, u) > 0$, and $\{u \mid \phi(s_2, u) = \Phi\} = \{u \mid \phi(s_1', u) = \Phi - 1\}$ for all $0 < \Phi \leqslant T$. Hence $s_2 \in E$, and thus $\mathbf{P}(s_1, E) > 0$.

Now let $|\{u \mid \phi(s_1, u) = T\}| > 0$. Then, each transition in the population model is induced by a failure vector $F = \langle f_1, \ldots, f_T\rangle$. In particular, there is a maximal number k, such that for all $l < k$, we have $f_l = \star$. That is, k denotes the lowest phase in which oscillators possibly fire.

First, we introduce some notation, where $\Phi > R$.

$$\begin{aligned}
\mathcal{N}^{\mathrm{F}}(s) &= \{u \mid \phi(s, u) = T\} \\
\mathcal{N}^{\mathrm{PF}}_{\Phi}(s) &= \{u \mid \phi(s, u) = \Phi \wedge \Delta(\phi(s, u), \alpha^{\phi(s,u)}(\sigma_1, F), \epsilon) + 1 > T\} \\
\mathcal{N}^{\mathrm{P}}_{\Phi}(s) &= \{u \mid \phi(s, u) = \Phi \wedge \Delta(\phi(s, u), \alpha^{\phi(s,u)}(\sigma_1, F), \epsilon) + 1 \leqslant T\}
\end{aligned}$$

That is, $\mathcal{N}^{\mathrm{F}}(s)$ denotes the set of oscillators possibly firing in s. The sets $\mathcal{N}^{\mathrm{P}}_{\Phi}(s)$ and $\mathcal{N}^{\mathrm{PF}}_{\Phi}(s)$ denote the sets of oscillators being perturbed but not firing (since the perturbation is not sufficient for the oscillators to reach the end of their cycle), and possibly firing, respectively. We can only say that elements of $\mathcal{N}^{\mathrm{F}}(s)$ and $\mathcal{N}^{\mathrm{PF}}_{\Phi}(s)$ *possibly* fire, since they may be affected by a broadcast failure.

We now have to construct a sequence of transitions, where we draw the firing oscillators from the sets $\mathcal{N}^{\mathrm{F}}(s_1)$ and $\mathcal{N}^{\mathrm{PF}}_{\Phi}(s_1)$, according to the broadcast failure vector F. Furthermore, all elements of $\mathcal{N}^{\mathrm{F}}(s_1)$ and the sets $\mathcal{N}^{\mathrm{PF}}_{\Phi}(s_1)$ have to take transitions such that their phase value in the next iteration is 1.

Let $\sigma_1 = \langle k_1, k_2, \ldots, k_T\rangle$. Now consider an arbitrary sequence $u_1, \ldots, u_{k_T}$ of all k_T elements from $\mathcal{N}^{\mathrm{F}}(s_1)$. Additionally, let $C_T \subseteq \mathcal{N}^{\mathrm{F}}(s_1)$ be the set of oscillators in phase T with a broadcast failure, i.e., $|C_T| = f_T$. Observe that $\phi(s_1, u_j) = T$ for all $1 \leqslant j \leqslant k_T$. Furthermore, let $r_0^T = s_1$. Then we define a sequence of successors of $r_0^T = (\mathsf{env}_0^T, \mathsf{osc}_0^T)$ as follows, where $1 \leqslant j \leqslant k_T$. If $u_i \notin C_T$, then

$$\begin{aligned}
\mathsf{osc}_j^T &= \mathsf{osc}_{j-1}^T \oplus [u_j \mapsto (T, \mathit{update})] \\
\mathsf{env}_j^T &= (\mathsf{count}(r_{j-1}^T) + 1, \mathit{update})
\end{aligned}$$

otherwise

$$\begin{aligned}
\mathsf{osc}_j^T &= \mathsf{osc}_{j-1}^T \oplus [u_j \mapsto (T, \mathit{update})] \\
\mathsf{env}_j^T &= (\mathsf{count}(r_{j-1}^T), \mathit{update})
\end{aligned}$$

Observe that these states define a sequence of transitions from r_0^T to $r_{k_T}^T$ according to conditions (7) and (8).

Now, for each phase Φ, with $k \leqslant \Phi < T$, we proceed similarly. That is, we first choose a sequence $u_1^\Phi, \ldots, u_{k_\Phi}^\Phi$ of oscillators and a set $C_\Phi \subseteq \mathcal{N}_\Phi^{\mathrm{PF}}(s_1)$ with $|C_\Phi| = f_\Phi$.

Subsequently, we define each r_j^Φ to be

$$\begin{aligned} \mathsf{osc}_j^\Phi &= \mathsf{osc}_{j-1} \oplus [u_j^\Phi \mapsto (\Phi, \mathit{update})] \\ \mathsf{env}_j^\Phi &= \begin{cases} (\mathsf{count}(s_{j-1}^\Phi) + 1, \mathit{update}), & \text{if } u_j \notin C_\Phi \\ (\mathsf{count}(s_{j-1}^\Phi), \mathit{update}), & \text{otherwise} \end{cases} \end{aligned}$$

where $r_0^\Phi = r_{k_\Phi}^{\Phi+1}$. Observe again, that these sequences exhaust $\mathcal{N}_\Phi^{\mathrm{PF}}(s_1)$ for each phase Φ. Furthermore, the number of firing oscillators that are not inhibited by a broadcast failure in the concrete model coincides with the number of perceived firing oscillators in the population model in this phase.

$$\text{For each } \Phi \text{ with } k \leqslant \Phi \leqslant T, \text{ we have } \mathsf{count}(r_0^\Phi) = \alpha^\Phi(\sigma_1, F) \ . \tag{23}$$

To prove this let us consider the different cases: for $\Phi = T$, we have $\mathsf{count}(r_0^T) = 0 = \alpha^T(\sigma_1, F)$. Now let $\Phi < T$ and assume $\mathsf{count}(r_0^{\Phi+1}) = \alpha^{\Phi+1}(\sigma_1, F)$. By definition, we have

$$\alpha^\Phi(\sigma_1, F) = \alpha^{\Phi+1}(\sigma, F) + k_{\Phi+1} - f_{\Phi+1},$$

since $\Phi < T$ and $f_{\Phi+1} \neq \star$.

Now, in the sequence $r_0^{\Phi+1}, \ldots, r_{k_{\Phi+1}}^{\Phi+1}$, we increase $\mathsf{count}(r_0^{\Phi+1})$ exactly $k_{\Phi+1} - f_{\Phi+1}$ times, i.e,

$$\mathsf{count}(r_0^\Phi) = \mathsf{count}(r_0^{\Phi+1}) + k_{\Phi+1} - f_{\Phi+1}.$$

By assumption, we then get

$$\mathsf{count}(r_0^\Phi) = \alpha^{\Phi+1}(\sigma_1, F) + k_{\Phi+1} - f_{\Phi+1} = \alpha^\Phi(\sigma_1, F),$$

and property (23) holds.

This property in particular states that the perturbation within the population model and the concrete model is the same.

Since Δ is a monotonically increasing function in α, every oscillator in $\mathcal{N}_\Phi^{\mathrm{PF}}(s_1)$ is still perturbed to firing after other oscillators in the same phase fired. Hence, for each pair of states u_{j-1}^Φ and u_j^Φ with $1 \leqslant j \leqslant k_\Phi - f_\Phi$, a transition according to condition (6) is well-defined. Similarly, for oscillators that should fire, but are affected by a broadcast failure, u_{j-1}^Φ and u_j^Φ with $k_\Phi - f_\Phi + 1 \leqslant j \leqslant k_\Phi$, the transition is defined according to condition (5).

Now, for every $\Phi < k$, we know that $\Phi + \Delta(\Phi, \alpha^\Phi(\sigma_1, F), \epsilon) + 1 \leqslant T$ and $\alpha^{\Phi-1}(\sigma_1, F) = \alpha^\Phi(\sigma_1, F)$, according to equation (13). Hence, for every phase $\Phi < k$, we arbitrarily enumerate the oscillators of $\mathcal{N}_\Phi^{\mathrm{P}}(s_1) = \{u_1^\Phi, \ldots, u_{k_\Phi}^\Phi\}$ and define the following sequence of states r_j^Φ, where $r_0^\Phi = r_{k_{\Phi+1}}^{\Phi+1}$.

$$\begin{aligned} \mathsf{osc}_j^\Phi &= \mathsf{osc}_{j-1} \oplus [u_j^\Phi \mapsto (\Phi, \mathit{update})] \\ \mathsf{env}_j^\Phi &= (\mathsf{count}(r_{j-1}^\Phi), \mathit{update}) \end{aligned}$$

For each Φ and pair of states r_j^Φ and r_{j+1}^Φ, there is a transition according to condition (4). So, all in all, we have a sequence of transitions from s_1 to $r_{k_0}^0$.

Now, in $r^0_{k_0}$, we have that $\mathsf{mode}(r^0_{k_0}, \mathsf{env}) = \mathit{update}$ and for all u, $\mathsf{mode}(r^0_{k_0}, u) = \mathit{update}$. Then let s_2 be defined by the following formulas.

$$\mathsf{mode}(s_2, \mathsf{env}) = \mathit{start} \text{ and } \mathsf{count}(s_2) = \mathsf{count}(r^0_{k_0}) \tag{24}$$

$$\forall u\colon \mathsf{mode}(s_2, u) = \mathit{start} \tag{25}$$

$$\forall u\colon \phi(r^0_{k_0}, u) = T \rightarrow \phi(s_2, u) = 1 \tag{26}$$

$$\forall u\colon \phi(r^0_{k_0}, u) < T \wedge \phi(r^0_{k_0}(u), \leqslant)R \rightarrow \phi(s_2, u) = \phi(r^0_{k_0}, u) + 1 \tag{27}$$

$$\begin{aligned}&\forall u\colon \phi(r^0_{0}, u) < T \wedge \phi(r^0_{k_0}, u) > R \wedge \\ &\phi(r^0_{k_0}, u) + \Delta(\phi(r^0_{k_0}, u), \mathsf{count}(r^0_{k_0}), \epsilon) + 1 \leqslant T \\ &\quad \rightarrow \phi(s_2, u) = \phi(r^0_{k_0}, u) + \Delta(\phi(r^0_{k_0}, u), \mathsf{count}(r^0_{k_0}), \epsilon) + 1\end{aligned} \tag{28}$$

$$\begin{aligned}&\forall u\colon \phi(r^0_{k_0}, u) < T \wedge \phi(r^0_{k_0}, u) > R \wedge \\ &\phi(r^0_{k_0}, u) + \Delta(\phi(r^0_{k_0}, u), \mathsf{count}(r^0_{k_0}), \epsilon) + 1 > T \\ &\quad \rightarrow \phi(s_2, u) = 1\end{aligned} \tag{29}$$

Then $r^0_{k_0}$ and s_2 satisfy all parts of condition (9). Hence, we have a sequence of transitions from s_1 to s_2, and $s_2 \in \mathcal{S}'_c$. To prove $h(s_2) = \sigma_2$, we need to show that the number of oscillators possessing a phase Φ in s_2 matches the Φ-th entry of $\sigma_2 = \langle k'_1, \ldots, k'_T\rangle$. To that end, recall that by Definition 10, each $k'_\Phi = \sum_{\Psi \in \mathcal{U}_\Phi(\sigma, F)} k_\Psi$, where $\sigma_1 = \langle k_1, \ldots, k_T\rangle$ and $\mathcal{U}_\Phi(\sigma, F) = \{\Psi \mid \Psi \in \{1, \ldots, T\} \wedge \tau(\sigma, \Psi, F) = \Phi\}$. Observe that both the concrete model and the population model use the same perturbation function Δ and that τ is defined in terms of Δ. In particular, we have

$$\tau(\sigma, \Phi, F) = \begin{cases} 1 & \text{if } \mathit{fire}^\Phi(\sigma, F) \\ \mathit{update}^\Phi(\sigma, F) & \text{otherwise.}\end{cases}$$

Now let us distinguish three cases for Φ.

1. If $\Phi \leqslant R$, then $\mathit{update}^\Phi(\sigma_1, F) = \Phi + 1$, due to the definition of the refractory function ref. Similarly, for all u such that $\phi(s_1, u) = \Phi$, we get that $\phi(s_2, u) = \Phi + 1$. Hence, for all $\Phi \leqslant R$, we have that $|\{u \mid \phi(s_2, u) = \Phi + 1\}| = |\{u \mid \phi(s_1, u) = \Phi\}|$.
2. If $\Phi > R$ and $\mathit{update}^\Phi(\sigma_1, F) = \Psi$, with $\Psi \leqslant T$. Then $\phi(s_2, u) = \Psi$, by formula (28). Observe that the number of oscillators in s_1 with a phase of Φ is k_Φ. So, the number of oscillators that get perturbed to be in Ψ is the union of the oscillators u in phases Φ, where $\Delta(\Phi, \mathsf{count}(s_2), \epsilon) + 1 = \Psi$. That is, $\{u \mid \phi(s_2, u) = \Psi\} = \{u \mid \Delta(\phi(s_1, u), \mathsf{count}(s_2), \epsilon) + 1 = \Psi\}$. By the definition of τ and property (23), we get that $\tau(\sigma_1, \Phi, F) = \Psi$. That is, for a specific Ψ, we have that the phases Φ of oscillators perturbed to Ψ are in $\mathcal{U}_\Psi(\sigma_1, F)$. Hence, since the sets of oscillators in each phase are disjoint, $|\{u \mid \phi(s_2, u) = \Psi\}| = \sum_{\Phi \in \mathcal{U}_\Psi(\sigma_1, F)} k_\Phi$.
3. Finally, let $\mathit{update}^\Phi(\sigma_1, F) = \Psi$ and $\Psi > T$. Then $\tau(\sigma_1, \Phi, F) = 1$. Furthermore, by formulas (26) and (29), we have $\phi(s_2, u) = 1$ for all u with phase Φ. With similar reasoning as above, we get that $|\{u \mid \phi(s_2, u) = 1\}| = \sum_{\Phi \in \mathcal{U}_1(\sigma_1, F)} k_\Phi$.

Hence, we get $h(s_2) = \sigma_2$, that is $s_2 \in E$. Together with the existence of the transition sequence from s_1 to s_2 we get $\mathbf{P}(s_1, E) > 0$.

From transition sequences in D_c to transition sequences in D_p. Now we turn our attention to the other direction. That is, if we have a sequence of transitions in the concrete model, we can find a corresponding transition sequence in the population model. Let

$(s_1, \sigma_1) \in R$, and let E be an equivalence class such that $\mathbf{P}(s_1, E) > 0$. If $s_1 \in E$, then the theorem holds, since $\sigma_1 \in E$ by definition of R. Otherwise, let $\omega = s_1 \ldots s_2$ be an execution sequence from s_1 to s_2 and assume that for all s' on ω different from s_1 and s_2, we have $\mathsf{mode}(s', \mathsf{env}) = \mathit{update}$. Recall that there is a unique $\sigma_2 \in E$. By definition of R, we have

$$\sigma_1 = \langle |\{u \mid \phi(s_1, u) = 1\}|, \ldots, |\{u \mid \phi(s_1, u) = T\}| \rangle$$
$$\sigma_2 = \langle |\{u \mid \phi(s_2, u) = 1\}|, \ldots, |\{u \mid \phi(s_2, u) = T\}| \rangle$$

We now distinguish two cases. First, assume that $\{u \mid \phi(s_1, u) = T\} = \emptyset$, and let s be a state on ω such that $\mathsf{mode}(s, u) = \mathit{update}$ for all u. Then there is exactly one transition from s to s_2, which is defined according to Eq. (9). Furthermore, due to the assumption that no oscillator fires, we have $\mathsf{count}(s) = 0$, which implies $\Delta(\Phi, \mathsf{count}(s), \epsilon) = 0$ for all Φ by Definition 5. Hence, for all u, we have $\phi(s_2, u) = \phi(s, u) + 1 = \phi(s_1, u) + 1$. That is,

$$\begin{aligned}\sigma_2 &= \langle 0, |\{u \mid \phi(s_1, u) + 1 = 2\}|, \ldots, |\{u \mid \phi(s_1, u) + 1 = T\}| \rangle \\ &= \langle 0, |\{u \mid \phi(s_1, u) = 1\}|, \ldots, |\{u \mid \phi(s_1, u) = T - 1\}| \rangle \ .\end{aligned}$$

That is, we have $\mathbf{P}(\sigma_1, \sigma_2) > 0$ due to a deterministic transition, which, in particular, implies $\mathbf{P}(\sigma_1, E) > 0$.

The second case is more involved. Let us assume $\{u \mid \phi(s_1, u) = T\} \neq \emptyset$, that is, at least one oscillator fires. Hence, due to the preconditions of the transitions, we can divide the transition sequence from s_1 to s_2 as follows:

$$\omega = s_1, \ldots, r_T, \ldots, r_{T-1}, \ldots, r_1, \ldots, s_2 \ ,$$

where r_Φ denotes the state where all oscillators with phase Φ changed their mode to *update*. Our goal now is to find a broadcast failure vector F, such that $\overrightarrow{\mathrm{succ}}(\sigma_1, F) = \sigma_2$. To that end, let

$$f_\Phi = |\{u \mid \phi(s_1, u) \geqslant \Phi\}| - \left(\mathsf{count}(r_\Phi) + \sum_{\Psi=\Phi+1}^{T} f_\Psi \right)$$

for all Φ where $\Phi + \Delta(\Phi, \mathsf{count}(r_\Phi), \epsilon) + 1 > T$. For the remaining Φ, set $f_\Phi = \star$. Then $F = \langle f_1, \ldots, f_T \rangle$. With this broadcast failure vector at hand, we now have to show that

$$\sum_{\Psi \in \mathcal{U}_\Phi(\sigma_1, F)} |\{u \mid \phi(s_1, u) = \Psi\}| = |\{u \mid \phi(s_2, u) = \Phi\}| \ .$$

Recall that $\mathcal{U}_\Phi(\sigma_1, F) = \{\Psi \mid \tau(\sigma_1, \Psi, F) = \Phi\}$. Hence, we need to show $\phi(s_2, u) = \tau(\sigma_1, \phi(s_1, u), F)$ for all oscillators u. To this end, we now need to distinguish several cases, according to the different cases of the transition defined by condition (9).

First, let u be such that $\phi(s_1, u) \leqslant R$, i.e., oscillator u is within its refractory period. If $\phi(s_1, u) = T$, then we have

$$\begin{aligned}\phi(s_2, u) &= 1 && \{\text{Cond. } (9a)\} \\ &= \tau(\sigma_1, \phi(s_1, u), F) && \{\text{Definition } 9\}\end{aligned}$$

Otherwise, if $\phi(s_1, u) < T$, we have

$$\begin{aligned}\phi(s_2, u) &= \phi(s_1, u) + 1 && \{\text{Cond. } (9b)\} \\ &= \tau(\sigma_1, \phi(s_1, u), F) && \{\text{Definition } 9\}\end{aligned}$$

Now assume that $\phi(s_1, u) \geqslant R$, i.e., oscillator u is outside of its refractory period and thus will be perturbed by firing oscillators. If $\phi(s_1, u) = T$, then we proceed as in the previous case.

So, let us assume $\phi(s_1, u) < T$. To show that the transition function of the population model coincides with the result within the concrete model, we need to ensure that the perceived firing oscillators are equal in both models for each oscillator.

Within each phase, the perceived oscillators in the population model coincide with the oscillators that fired up to the next higher phase in the concrete model. Formally, for each $1 \leqslant \Phi < T$, we have

$$\mathsf{count}(r_{\Phi+1}) = \alpha^{\Phi}(\sigma_1, F) \ . \tag{30}$$

To show that this is true, let $\sigma_1 = \langle k_1, \ldots, k_T \rangle$. Let $f_\Phi \neq \star$. Then for $\Phi = T - 1$, we have

$$\begin{aligned}
\mathsf{count}(r_T) &= |\{u \mid \phi(s_1, u) = T\}| - f_T && \{\text{Definition of } f_\Phi\} \\
&= 0 + k_T - f_T && \{\text{Definition of } \sigma_1\} \\
&= \alpha^T(\sigma_1, F) + k_T - f_T && \{\text{Definition of } \alpha^\Phi, \text{Eq. (13)}\} \\
&= \alpha^{T-1}(\sigma_1, F)
\end{aligned}$$

Now assume that $\mathsf{count}(r_{\Phi+1}) = \alpha^{\Phi}(\sigma_1, F)$ for some $1 < \Phi < T$ with $f_\Phi \neq \star$. Then

$$\begin{aligned}
\mathsf{count}(r_\Phi) &= |\{u \mid \phi(s_1, u) \geqslant \Phi\}| - \left(\sum_{\Psi=\Phi}^{T} f_\Psi\right) && \{\text{Definition of } f_\Phi\} \\
&= \left(\sum_{\Psi=\Phi}^{T} k_\Psi\right) - \left(\sum_{\Psi=\Phi}^{T} f_\Psi\right) && \{\text{Definition of } \sigma_1\} \\
&= \left(\sum_{\Psi=\Phi+1}^{T} k_\Psi\right) - \left(\sum_{\Psi=\Phi+1}^{T} f_\Psi\right) + k_\Phi - f_\Phi \\
&= \mathsf{count}(r_{\Phi+1}) + k_\Phi - f_\Phi && \{\text{Definition of } f_\Phi\} \\
&= \alpha^{\Phi}(\sigma_1, F) + k_\Phi - f_\Phi && \{\text{Ass.}\} \\
&= \alpha^{\Phi-1}(\sigma_1, F) && \{\text{Definition of } \alpha^\Phi, \text{Eq. (13)}\}
\end{aligned}$$

If $f_\Phi = \star$, then the equality immediately holds.

Furthermore, observe that for all Φ, if $f_\Phi = \star$, then we have $\mathsf{count}(r_{\Phi+1}) = \mathsf{count}(r_1)$. We can now proceed to prove the final two cases.

First, let $\phi(s_1, u) + \Delta(\phi(s_1, u), \mathsf{count}(r_1), \epsilon) + 1 \leqslant T$. This implies $f_{\phi(s_1,u)} = \star$. Then, by the observation and the claim above, we also get

$$\begin{aligned}
&\tau(\sigma_1, \phi(s_1, u), F) \\
&\quad = 1 + \mathrm{ref}(\phi(s_1, u), \Delta(\phi(s_1, u), \alpha^{\phi(s_1,u)}(\sigma_1, F), \epsilon)) && \{\phi(s_1, u) > R\} \\
&\quad = 1 + \phi(s_1, u) + \Delta(\phi(s_1, u), \alpha^{\phi(s_1,u)}(\sigma_1, F), \epsilon) && \{\text{Eq. (30)}\} \\
&\quad = 1 + \phi(s_1, u) + \Delta(\phi(s_1, u), \mathsf{count}(r_{\phi(s_1,u)+1}), \epsilon) && \{\text{Obs.}\} \\
&\quad = \phi(s_1, u) + \Delta(\phi(s_1, u), \mathsf{count}(r_1), \epsilon) + 1 && \{\text{Eq. (9c)}\} \\
&\quad = \phi(s_2, u) \ .
\end{aligned}$$

Finally, let $\phi(s_1, u) + \Delta(\phi(s_1, u), \mathsf{count}(r_1), \epsilon) + 1 > T$, i.e. $\phi(s_2, u) = 1$. Then it has also to be the case that $\phi(s_1, u) + \Delta(\phi(s_1, u), \mathsf{count}(r_{\phi(s_1,u)+1}), \epsilon) + 1 > T$. By the claim above, this means $\phi(s_1, u) + \Delta(\phi(s_1, u), \alpha^{\phi(s_1,u)}(\sigma_1, F), \epsilon) + 1 > T$. Hence, $\tau(\sigma_1, \phi(s_1, u), F) = 1$ as well.

Now recall that we assumed initially that for all intermediate states s of the transition sequence, we have $\mathsf{mode}(s, \mathsf{env}) = \mathit{update}$. If this is not the case, we can partition the sequence into distinct subsequences, where this assumption holds for each subsequence, and apply the arguments above.

Comparing the probabilities of transition sequences. Let $(s_1, \sigma_1) \in R$ and E be an equivalence class with $\mathbf{P}(s_1, E) > 0$ and $\mathbf{P}(\sigma_1, E) > 0$. Furthermore, let $\sigma_2, s_2 \in E$ be the unique state of the population model in E and a state of the concrete model, respectively. Let $\sigma_1 = \langle k_1, \ldots, k_T \rangle$. Furthermore, let $N = \sum_{i=1}^{T} k_i$. If no oscillator fires in σ_1 this also the case in s_1. Then we have $N!$ possibilities to create an transition sequence starting in s_1, each of which has a probability of $\frac{1}{N!}$ to happen. Hence, the probability that one of these transitions happen is $N! \cdot \frac{1}{N!} = 1$, which coincides with the definition in the population model.

For the case that at least one oscillator fires and thus perturbs the other oscillators, we consider the construction of a transition sequence from s_1 with respect to a failure vector $F = \langle f_1, \ldots, f_T \rangle$ for σ_1. During each phase Φ, we have to choose the particular order of the k_Φ oscillators to create a sequence from r_0^Φ to $r_0^{\Phi-1}$ and in addition, we have to choose the set C_Φ. That is, we have $k_\Phi!$ possible orders, and $\binom{k_\Phi}{f_\Phi}$ possibilities for the choice of C_Φ. Furthermore, the combined probability for the transitions of the oscillators that should fire but are inhibited by a broadcast failure is

$$\frac{1}{|\mathsf{init}_\Phi(s_1)|!} \cdot (1-\mu)^{k_\Phi - f_\Phi} \cdot \mu^{f_\Phi} \ .$$

Observe that at the start of the construction of each phase, $|\mathsf{init}_\Phi(s_1)| = k_\Phi$. Hence the probability above simplifies to

$$\frac{1}{k_\Phi!} \cdot (1-\mu)^{k_\Phi - f_\Phi} \cdot \mu^{f_\Phi} \ .$$

Due to the possible choices during the construction of the transition sequence, we have that the probability of *one* of these sequences to happen is

$$\binom{k_\Phi}{f_\Phi} \cdot k_\Phi! \cdot \frac{1}{k_\Phi!} \cdot (1-\mu)^{k_\Phi - f_\Phi} \cdot \mu^{f_\Phi} = \binom{k_\Phi}{f_\Phi} \cdot (1-\mu)^{k_\Phi - f_\Phi} \cdot \mu^{f_\Phi} \ ,$$

which is exactly the function $\mathsf{PMF}(k_\Phi, f_\Phi)$ as in the population model. Furthermore, with similar reasoning as above, the transition probability for the sequences, where no oscillator is perturbed anymore, is 1. Hence, $\mathbf{P}(s_1, E) = \mathbf{P}(\sigma_1, E)$, and the sum of the probabilities of the paths from s_1 of a population model state σ_1 to the elements of the equivalence class E is equal to the probability of the the transition from σ_1 to σ_2.

Finally, consider the transitions from the initial states into the corresponding equivalence classes. As stated in Sect. 5.4, Eq. (17), for each state of the population model $\sigma = \langle k_1, \ldots, k_T \rangle$, we have

$$\mathbf{P}(\sigma_I, \sigma) = \frac{1}{T^N} \binom{N}{k_1, \ldots, k_T} \ ,$$

which by construction equals $\mathbf{P}(\sigma_I, E)$, where E is the equivalence class with $\sigma \in E$. The multinomial coefficient gives the number of possible assignments of values k_i to the oscillators. This is exactly the number of states in the set $\mathcal{S}'_c$, where k_i oscillators are in phase i and the environment counter is set to 0. Since T^N is the total number of such states, the

probability of reaching the equivalence class E from s_I is also

$$\mathbf{P}(s_I, E) = \frac{1}{T^N}\binom{N}{k_1, \ldots, k_T},$$

and R is a weak bisimulation. □

By this theorem, we can use population models to analyse the global properties of a network of pulse-coupled oscillators following the concrete model as defined in Sect. 4 without loss of precision. In particular, this allows us to increase the size of the network to check such properties, while still giving us the opportunity to analyse the internal behaviour of nodes, if we restrict the network size.

7 Empirical analysis

In this section we first compare the model checking times for an analysis of the concrete and the population model, followed by an analysis of the impact of the reduction of the population model as defined in Sect. 5.5. Subsequently, we present the results of an empirical analysis of the population model. This evaluation is based on previous work where parametric influence [24] and power consumption [25] were investigated. For all analyses the perturbation function we used is a discretisation of the Mirollo and Strogatz model of pulse-coupled oscillator synchronisation [38]. In particular, we set the perturbation function to be $\Delta(\Phi, \alpha, \epsilon) = [\Phi \cdot \alpha \cdot \epsilon]$, where $[\cdot]$ denotes rounding to the nearest integer; the perturbation induced by the firing of another oscillator increases linearly with the phase of the perturbed oscillator. We use the probabilistic model checker PRISM [34] to formally verify properties of our models.

7.1 Model specification

Both the concrete models and population models are encoded using the guarded command language of PRISM, a state-based language, based on the Reactive Modules formalism of Alur and Henzinger [4]. Each model consists of a set of modules, which in turn consist of a set of local variables over finitely bound integers and Booleans and a set of commands that define its behaviour. A command consists of a predicate over the set of local variables for all modules and a set of possible transitions that will occur with some given probability should the predicate hold. A transition is an assignment of values to each of the local variables of the module. The local state space of a module is given by the set of all valuations for its local variables, and the global space is the product of all local state spaces. For further details we refer the reader to [40].

Concrete model The specification of the concrete model within PRISM is a straightforward implementation. Each individual oscillator, as well as the environment, is defined as a module. All of these modules only synchronise on a single message *sync*, to ensure that all oscillators were considered to update their phases.

The module for oscillator i contains local variables $mode_i$ and $phase_i$ for its mode and phase. Each module specifies transitions for a single oscillator according to Sect. 4.2. All of these transitions are similar in structure in each module, and hence we can employ the template mechanism of PRISM. To that end, we define the behaviour of a single oscillator within the

network, and can then replace the names of the local variables suitably.[1] That is, while we still need to explicitly define the dependencies between the oscillators, we do not need to manually write all module specifications. With the exception of the number of oscillators, the parameters of the network (the lengths of the oscillation cycle and the refractory period, the coupling constant and the broadcast failure probability) are also parameters within the input language of PRISM. That is, we can instantiate them and automatically run the model-checking engine on sets of parameter combinations.

Population model A single module is used to specify an instance of the population model. The global state is encoded using T finitely bound integer variables ranging over N discrete values, where each variable records the number of oscillators sharing some phase value in $1, \ldots, T$. In contrast to the specification of the concrete model, specifying an instance of the population model is not straightforward. For the concrete model the length of the oscillation cycle T could be easily specified as a parameter within the input language of PRISM, and moving between two models with different values for T, but sharing the same value for all other parameters, is as simple as changing the value of the parameter in the specification. For the population model this was not possible, as T variables are needed to count the oscillators with different phases, and hence changing T leads to a change in the number of variables. The changes that arise from the introduction or removal of these new variables propagate throughout the specification, which for large models may consist of many thousands of lines of code. Further complications also arose when encoding the transitions from the initial unconfigured state σ_I to every possible configured state for the oscillators, since a single model may have many thousands of such transitions for larger values of N and T.

To facilitate the analysis of families of parameter-wise different oscillator population models we developed a Python script[2] that allows the user to define ranges for N, T, R, ϵ and μ, for some fixed definitions for the perturbation function Δ. Then, given a list of properties, for each combination of parameters the script generates a specification, checks all the given properties against that specification using PRISM, and writes user specified output (e.g. result, model checking time, etc.) to a file that can be used by statistical analysis tools. The reduction introduced in Sect. 5.5 could not be implemented with the existing high-level specification language of PRISM. It was only possible to specify the individual rewards labelling some transitions, namely those from the initial unconfigured state to each of its possible successors, by introducing additional variables to the model. Therefore, at the time of writing, and to the best of our knowledge, it is not possible to implement a script for the generation of PRISM code for all the reduced models. The results shown in Table 3 were obtained using a prototypical version of the model checker ePMC, formerly known as IscasMC [27], which had been modified to accept a low level representation of a model as input (as a set of states and transitions), where each transition could be individually labelled with a reward.

7.2 Comparison between concrete and population

In this section, we compare the differences between the concrete and population models. We restrict our comparison to the analysis of the probability to achieve synchronisation,

[1] The implementation can be found at https://github.com/PaulGainer/mc-bio-synch/tree/master/multi-scale-verification/concrete.

[2] The implementation can be found at https://github.com/PaulGainer/mc-bio-synch/tree/master/multi-scale-verification/population.

Table 2 Model construction and model checking times for the concrete and population model with $T = 10$, $\mu = 0.2$ and $\epsilon = 0.1$ (in seconds)

R	$N = 4$				$N = 5$			
	Concrete Model		Population Model		Concrete Model		Population Model	
	Constr.	Check	Constr.	Check	Constr.	Check.	Constr.	Check
0	0.069	0.941	0.385	0.017	0.181	4.163	1.436	0.028
1	0.061	0.882	0.388	0.017	0.221	3.888	1.71	0.027
2	0.055	0.89	0.544	0.027	0.159	3.746	1.486	0.028
3	0.072	0.838	0.442	0.025	0.15	3.789	1.471	0.034
4	0.047	0.96	0.436	0.024	0.208	4.1	1.604	0.042
5	0.057	10.111	0.42	0.047	0.174	289.73	1.401	0.113
6	0.059	5.134	0.38	0.045	0.279	153.75	1.343	0.107
7	0.053	3.218	0.413	0.048	0.148	72.525	1.337	0.107
8	0.056	1.417	0.356	0.013	0.055	25.188	1.194	0.019
9	0.048	0.02	0.347	0.006	0.15	0.074	1.163	0.007
10	0.013	0.005	0.35	0.007	0.042	0.037	1.158	0.008

excluding the necessary time or power to achieve this (cf. Sect. 7.4). Model checking this property against both models yields identical results, as was expected in light of the results of Sect. 6. However, as expected from the definition of the concrete model in Sect. 4, the increase in the size of the model is much more pronounced for the concrete model. Hence, the performance of the model checking procedure differs strongly between the models.

Table 2 shows the model construction and checking times for some exemplary parameter combinations of the models, as reported by PRISM.[3] In the table, the model construction time denotes the time PRISM needs to construct a DTMC representation from the specification. In the concrete model, the bulk of time is spent in the model checking phase, while the construction is much faster. For the analysis of the population model, however, the situation is reversed. The model construction phase is at least an order of magnitude longer than the model checking phase. As expected, the model checker needs less time for the analysis of the population model, even if the times needed for model construction and checking were considered together. However, we can also see that while checking both the concrete and population model, refractory periods around $R = \frac{T}{2}$ are harder for the model checker to analyse. This is because for very low and very high refractory periods the synchronisation probability is either 1 or 0, and such qualitative results are trivially, and efficiently obtained in a precomputation step by the model checker.

For properties relating to global behaviour a population model is beneficial, However, it requires all nodes to be behaviourally identical. While we defined our concrete model in a similar fashion, we could in principle relax this restriction. That is, we could allow for different perturbation functions for each node, or for partitions of nodes.

[3] The experiments were run on a computer equipped with an Intel Core i7-7700 CPU at 3.6 GHz and with 16GB of RAM. The version of PRISM used was 4.4 beta.

Table 3 Reduction in state space and transitions where $R = 1$ and $\epsilon = 0.1$ for different population sizes and maximal phase values

N	T	D		D'		Reduction (%)	
		States	Transitions	States	Transitions	States	Transitions
3	6	113	188	22	52	80.5	72.3
5	6	505	1030	127	389	74.9	62.2
8	6	2575	7001	793	3154	69.2	54.9
3	8	241	410	37	97	84.6	76.3
5	8	1585	3250	331	1097	79.1	66.2
8	8	12871	34615	3433	14519	73.3	58.1
3	10	441	752	56	156	87.3	79.3
5	10	4005	8114	716	2484	82.1	69.4
8	10	48,621	128,6936	11,6441	50,6883	76.5	60.5

7.3 Reduction analysis

In this section we present the effect of the reduction of population models as defined in Sect. 5.5. Table 3 shows the number of reachable states and transitions of the DTMC D, and corresponding reduction D', for different population sizes (N) and oscillation cycle lengths (T), using the Mirollo and Strogatz model of synchronisation presented at the start of this section. The number of reachable states is stable under changes to the parameters R, ϵ, and μ, since every possible firing state is always reachable from the initial state. For the results shown here the parameters were arbitrarily set to $R = 1$, $\epsilon = 0.1$. The underlying graph of the DTMC, and hence the number of transitions, is stable under changes to the parameter μ, and is not of interest here. Larger values of N and T were not investigated, due to the large model sizes when generating the unreduced model.

Table 4 shows the number of transitions of the DTMC, and corresponding reduction, for various population model instances, and again uses the Mirollo and Strogatz model of synchronisation. Increasing the length of the refractory period (R) results in an increase in the reduction of transitions in the model. A longer refractory period leads to more firing states where the firing of a group of oscillators is ignored. This results in successor states having oscillators with lower values for phase, and hence a longer sequence of deterministic transitions (later removed in the reduction) leading to the next firing state. Conversely, increasing the strength of the coupling between oscillators (ϵ) results in a decrease in the reduction of transitions in the model. For the Mirollo and Strogatz model of synchronisation used here, increasing the coupling strength results in a linear increase in the pertubation to phase induced by the firing of an oscillator. This results in successor states of firing states having oscillators with higher values for phase, and hence a shorter sequence of deterministic transitions leading to the next firing state.

Table 4 Reduction in transitions for $N = 5$ and $T = 10$ with different values of R and ϵ

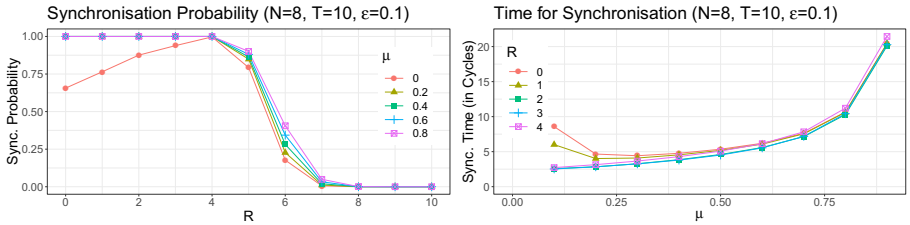
R	ϵ	Transitions		Reduction (%)
		D	D'	
1	0.1	8114	2484	69.4
3	0.1	7928	2391	69.8
5	0.1	7568	2211	70.8
7	0.1	6976	1915	72.5
9	0.1	6006	1430	76.2
1	0.01	6006	1430	76.2
1	0.05	6426	1640	74.5
1	0.1	8114	2484	69.4
1	0.25	8950	2902	67.6
1	0.5	9382	3118	66.7

7.4 Population model evaluation

In this section we discuss the influence of different parameters of the population model defined in Sect. 5 on both the likelihood that a network of oscillators will eventually synchronise, and the requisite time and power consumption to achieve this. For a real deployment of synchronising nodes, for example a Wireless Sensor Network (WSN), communication is costly with respect to energy consumption. Therefore, minimising power consumption is a critical consideration for their design [2,44]. Once deployed, a WSN is generally expected to function independently for long periods of time. In particular, regular battery replacement can be costly and impractical for remote sensing applications. Hence, it is important to reduce the power consumption of the individual nodes by choosing low-power hardware and/or energy efficient protocols. However, to make informed choices, it is also necessary to have good estimations of the power consumption for individual nodes. While the general power consumption of the hardware can be extracted from data sheets, estimating the overall power consumption of different protocols is more demanding. Communication between nodes in the network is either active when sending a message, i.e., when a node fires, or passive, when receiving messages from other nodes. Hence, during periods where a sensor does neither, the antenna can be shut down to save energy. In our models, this interval of inactivity corresponds to the refractory period. That is, the longer the refractory period is, the less energy will be consumed.

First, we consider a binary metric where we are only interested in states where all oscillators share precisely the same phase. Second, we derive a synchronisation metric from the complex order parameter of Kuramoto [32], that captures the degree of synchrony of a fully connected network of oscillators as a real value in the interval [0, 1].

Binary synchronisation metric Here we use the binary notion of synchronisation introduced in Sect. 5.3, where a global state $\sigma = \langle k_1, \ldots, k_T \rangle$ is synchronised if, and only if, there is some $\Phi \in \{1, \ldots, T\}$ such that $k_\Phi = N$. We created concrete input models for PRISM for different parameters of the model, for example the number of oscillators and different coupling strengths. Each of these models was subsequently checked with respect to different



(a) (b)

Fig. 5 Synchronisation probabilities for different refractory periods (**a**), and synchronisation times for different rates of broadcast failure (**b**)

properties. Other case studies could also be considered for alternative models of synchronisation where the dynamics of oscillators, and their interactions, can be described by some perturbation function.

We are interested in the probability of eventual synchronisation and in the expected time needed to achieve synchronisation. The probability of eventual synchronisation is given by the PCTL property

$$\mathrm{P}_{=?}\left[\mathrm{F}\bigvee_{i=1}^{T}(k_i = N)\right].$$

we first define a reward structure that associates a value of $\frac{1}{T}$ with every unsynchronised state in $\Gamma \setminus \{\sigma_I\}$, that records the number of cycles taken to achieve synchrony.

To determine the expected time taken for a population model to synchronise we accumulate a reward along a path until some synchronised global state is reached, and define a reward structure $\mathcal{R}_{\text{time}} = (R_s, R_t)$. We set $R_s(\sigma) = \frac{1}{T}$ for every unsynchronised state in $\Gamma \setminus \{\sigma_I\}$, $R_s(\sigma') = 0$ for all other $\sigma' \in \Gamma$, and $R_t(\sigma_1, \sigma_2) = 0$ for all $\sigma_1, \sigma_2 \in \Gamma$. Intuitively, we expect a reward of 1 along a path where synchronisation occurs after T transitions (one complete oscillation cycle). This is achieved by assigning a reward of $\frac{1}{T}$ to each unsynchronised state, since a transition of the model from one state to the next corresponds to a step of $\frac{1}{T}$ oscillation cycles. In this way we obtain a measure of synchronisation time in oscillation cycles.

The expectation of time to achieve synchronisation is then given by the PCTL property

$$\mathrm{R}_{=?}\left[\mathrm{F}\bigvee_{i=1}^{T}(k_i = N)\right],$$

with respect to the reward structure $\mathcal{R}_{\text{time}}$. We note here that a result of *Infinity* is obtained for accumulating this reward along a path where the probability of reaching a synchronised state is less than 1.

We generated models for different numbers of oscillators $3 \leqslant N \leqslant 8$, cycle lengths $4 \leqslant T \leqslant 10$, coupling constants $\epsilon \in \{0, 0.1, \ldots, 1.0\}$, refractory periods $0 \leqslant R \leqslant T$, and message loss probabilities $\mu \in \{0, 0.1, \ldots, 1.0\}$, and analysed the models with respect to the two properties of interest.

Figure 5a plots the probability of synchronisation for different rates of broadcast failure against the refractory period for $N = 8$, $T = 10$, and $\epsilon = 0.1$. We can observe a trade-off between a high refractory period and high synchronisation probability. As long as the refractory period is less than half the oscillation cycle, synchronisation will be achieved in

almost all cases. Higher values for R result in a rapid drop in synchronisation probability. The exception is the edge case $\mu = 0$, which may seem surprising. If $\mu = 0$ a model is deterministic. The results for $\mu = 1$ are omitted here as, unsurprisingly, if all firings result in broadcast failures the synchronisation probability is almost zero. In fact, the only runs that synchronise in this case are runs where the first configured state is already synchronised.

Figure 5b shows us that a higher refractory period results in shorter synchronisation times when the probability for broadcast failure is low. In general, a longer refractory period up to half the cycle length improves the rate of convergence to synchrony, which is consistent with the findings of [17]. Furthermore, for high values of μ the differences in synchronisation times for different refractory period lengths are negligible. Hence, a refractory period of slightly less than half the cycle, with a low coupling constant ϵ, is optimal for this model of synchronisation. As ϵ is increased the results remain similar, but with a decrease in synchronisation times.

Order parameter synchronisation metric In the previous section a binary metric of synchrony for a population model was employed, where a state was synchronised if, and only if, all oscillators in that state shared the same phase. However, it is clear that some global states appear to be closer to achieving a truly synchronised state than others. Consider the global states $\sigma_1 = \langle 0, 2, 0, 2, 0, 2 \rangle$ and $\sigma_2 = \langle 0, 0, 0, 0, 1, 5 \rangle$ of some population model for a network of $N = 6$ nodes with an oscillation cycle over $T = 6$ discrete values. Using the binary notion of synchrony all that is known is that both states are not synchronised, yet it is clear that for nearly all models of synchronisation, encoded as some perturbation function, σ_2 appears to be closer to converging to a state where all oscillators share the same phase.

The binary notion of synchrony can be extended by introducing a *phase coherence* metric for the level of synchrony of a global state. The metric is derived from the *order parameter* introduced by Kuramoto [32] as a measure of synchrony for a population of coupled oscillators. If the phases of the oscillators are considered as positions on the unit circle in the complex plane, they can be represented as complex numbers with magnitude 1.

Definition 21 The function $\phi^{\mathbb{C}} : [1 \ldots T] \to \mathbb{C}$ maps a phase value to its corresponding position on the unit circle in the complex plane, and is defined as $\phi^{\mathbb{C}}(\Phi) = e^{i\theta_\Phi}$, where $\theta_\Phi = \frac{2\pi}{T}(\Phi - 1)$.

A measure of synchrony $\eta \in [0, 1]$ can then be obtained by calculating the magnitude of the complex number corresponding to the mean of the phase positions. A global state has a maximal value of $\eta = 1$ when all oscillators are synchronised and share the same phase Φ, mapped to the position defined by $\phi^{\mathbb{C}}(\Phi)$. It then follows that the mean position is also $\phi^{\mathbb{C}}(\Phi)$ and $|\phi^{\mathbb{C}}(\Phi)| = 1$. A global state has a minimal value of $\eta = 0$ when all of the positions mapped to the phases of the oscillators are uniformly distributed around the unit circle. This also occurs when their positions achieve mutual counterpoise, for example when $\frac{N}{2}$ oscillators share some phase value Φ and the remaining $\frac{N}{2}$ oscillators have a phase value whose position on the complex plane is the negation of $\phi^{\mathbb{C}}(\Phi)$.

Definition 22 The *phase coherence function* $\mathsf{PCF} : \Gamma \to [0, 1]$ maps a global state to a real value in the interval $[0, 1]$, and is given by

$$\mathsf{PCF}(\langle k_1, \ldots, k_T \rangle) = \left| \frac{1}{N} \sum_{\Phi=1}^{T} k_\Phi \phi^{\mathbb{C}}(\Phi) \right|,$$

where $|\cdot|$ denotes the complex modulus.

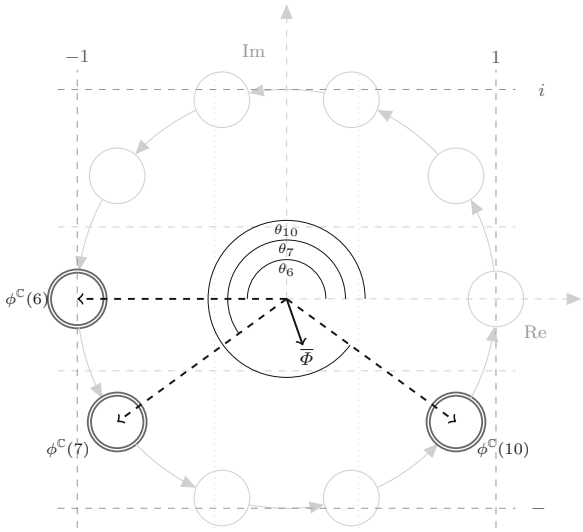


Fig. 6 Argand diagram of the phase positions for the global state $\langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5\rangle$

Example 4 Figure 6 shows a plot on the complex plane of the positions of the phases for some global state $\sigma = \langle 0, 0, 0, 0, 0, 2, 1, 0, 0, 5\rangle$ of a population model where $N = 8$, $T = 10$. The phase positions are given by $\phi^{\mathbb{C}}(6) = e^{i\pi}$ for 2 oscillators with phase 6, $\phi^{\mathbb{C}}(7) = e^{\frac{6i\pi}{5}}$ for 1 oscillator with phase 7, and $\phi^{\mathbb{C}}(10) = e^{\frac{9i\pi}{5}}$ for 5 oscillators with phase 10. The phase coherence can then be determined as

$$\mathsf{PCF}(\sigma) = \left|\frac{1}{8}\left(2e^{i\pi} + e^{\frac{6i\pi}{5}} + 5e^{\frac{9i\pi}{5}}\right)\right| = 0.4671.$$

The mean phase position is indicated on the diagram by $\overline{\Phi}$.

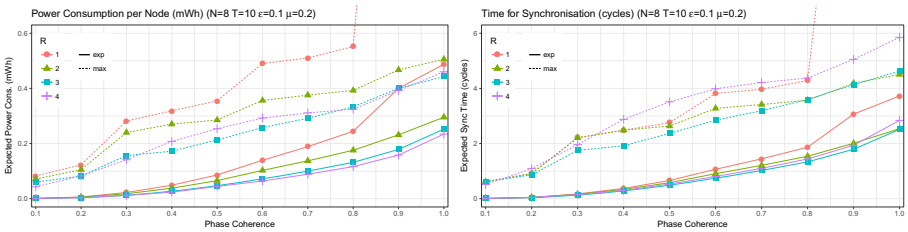
The two new properties of interest are firstly, the time taken for the network to reach a state where some desirable degree of synchronisation with respect to the new metric has been achieved, and secondly, the power consumed by the network to reach that state. In addition to the expected time/power consumption we will also investigate the maximal time/power consumption.[4]

We now define a reward structure $\mathcal{R}_{\text{pow}} = (R_s, R_t)$ that annotates a model with rewards corresponding to power consumption. Firstly, let I_{id}, I_{rx}, and I_{tx} be the current draw in amperes for the idle, receive, and transmit modes of a synchronising node in a network, V be the voltage, C be the length of the oscillation cycle in seconds, and M_t be the time taken to transmit a synchronisation message in seconds. We now define

$$W_{id} = \frac{I_{id}VC}{3600T}, \qquad W_{rx} = \frac{I_{rx}VC}{3600T}, \qquad W_{tx} = \frac{I_{tx}VM_t}{3600},$$

where W_{id} is the power consumption in Watt-hours of one node for one discrete step within its refractory period (the oscillator is in the idle mode), W_{rx} is the power consumption in Watt-hours of one node for one discrete step outside of its refractory period (in receive mode), and W_{tx} is the power consumption in Watt-hours to transmit one synchronisation message. The power consumption of the network consists of the power necessary to transmit

[4] This can be achieved by using the *filter* construct of the model checker PRISM.



(a) (b)

Fig. 7 Power (**a**) and time (**b**) per node to achieve synchronisation

the synchronisation messages, and that of the oscillators in the idle and receive modes. For synchronisation messages, we consider each firing state σ, and assign a reward of $R_t(\sigma, \sigma') = k_1 W_{tx}$ to every transition from σ to a successor state $\sigma' = \langle k_1, \ldots, k_T \rangle$. This corresponds to the total power consumption for the transmission of k_1 synchronisation messages. For each state $\sigma = \langle k_1, \ldots, k_T \rangle$ where the oscillators are configured, the total power consumption for oscillators in the idle and receive modes is

$$\rho_{\mathcal{S}}(\sigma) = \sum_{\Phi=1}^{R} k_\Phi W_{id} + \sum_{\Phi=R+1}^{T} k_\Phi W_{rx} \ .$$

For our experiments we set $\epsilon = 0.1$ and $\mu = 0.2$. We could have conducted analyses for different values for these parameters. For a real system, the probability μ of broadcast failure occurrence is highly dependent on the deployment environment. For deployments in benign environments we would expect a relatively low rate of failure, for instance a WSN within city limits under controlled conditions, whilst a comparably high rate of failure would be expected in harsh environments such as a network of off-shore sensors below sea level. The coupling constant ϵ is a parameter of the system itself. Our results suggest that higher values for ϵ are always beneficial, however this is because we only model fully connected networks. High values for ϵ may be detrimental when considering different topologies, since firing nodes may perturb synchronised subcomponents of a network. However we defer such an analysis to future work.

As an example we analyse the power consumption for values taken from the datasheet of the *MICAz* mote [37]. For the transmit, receive and idling mode, we assume $I_{tx} = 17.4\, mA$, $I_{rx} = 19,7\, mA$, and $I_{id} = 20\mu A$, respectively. Furthermore, we assume that the oscillators use a voltage of $3.0\ V$. We define $\mathsf{coherent}_\lambda$ to be a predicate that holds for any state σ in $\Gamma \setminus \{\sigma_I\}$ with $\mathsf{PCF}(\sigma) \geqslant \lambda$. The properties of interest are then given by the PCTL property

$$\mathrm{R}_{=?}[\mathrm{F}\ \mathsf{coherent}_\lambda] \tag{31}$$

Figure 7a, b show both the average and maximal power consumption per node (in mWh) and time (in cycles) needed to synchronise, in relation to the phase coherence of the network with respect to different lengths of the refractory period, where $\epsilon = 0.1$ and $\mu = 0.2$. That is, they show how much power is consumed (time is needed, resp.) for a system in an arbitrary state to reach a state where some degree of phase coherence has been achieved.

The average, and maximal values are obtained using the *avg* and *max* filters of the PRISM model checker. These filters give the average (maximum, resp.) expected reward across all paths starting in states that satisfy some given predicate. The desired values are therefore obtained by checking property (31) against all states of the model where oscillators are

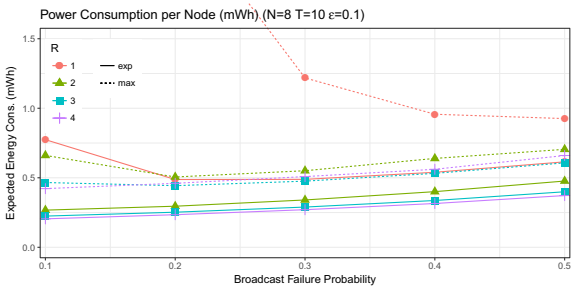
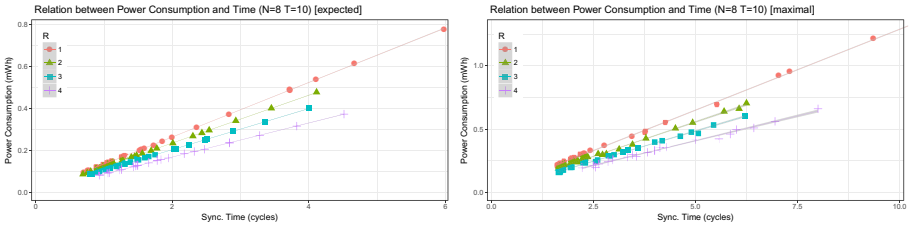


Fig. 8 Power consumption in relation to broadcast failure probability



(a) (b)

Fig. 9 Expected (**a**) and maximal (**b**) power consumption vs. time taken to achieve synchrony

configured i.e. all $\sigma \in \mathcal{S} \setminus \{\sigma_I\}$. For both cases the total values obtained were divided by the number of nodes in the network.

The much larger values obtained for $R = 1$ and phase coherence $\geqslant 0.9$ are not shown here, to avoid distortion of the figures. The energy consumption for these values is roughly 2.4 mWh, while the time needed is around 19 cycles. Observe that we only show values for the refractory period R with $R < \frac{T}{2}$. For larger values of R not all runs synchronise [24], resulting in an infinitely large reward being accumulated for both the maximal and average cases.

As expected when starting from an arbitrary state, the time and power consumption increases monotonically with the order of synchrony to be achieved. On average, networks with longer refractory periods require less power for synchronisation, and take less time to achieve it. The only exception is that the average time to achieve synchrony with a refractory period of four is higher than for two and three. However, if lower phase coherence is sufficient then this trend is stable. In contrast, the maximal power consumption of networks with $R = 4$ is consistently higher than of networks with $R = 3$. In addition, the maximal time needed to achieve synchrony for networks with $R = 4$ is higher than for lower refractory periods, except when the phase coherence is greater than or equal to 0.9. We find that networks with a refractory period of three will need the smallest amount of time to synchronise, regardless of whether we consider the maximal or average values. Furthermore, the average power consumption for full synchronisation (phase coherence one) differs only slightly between $R = 3$ and $R = 4$ (less than 0.3 mWh). Hence, for the given example, $R = 3$ gives the best results. These relationships are stable even for different broadcast failure probabilities μ, while the concrete values increase only slightly, as illustrated in Fig. 8, which shows the power consumption for different values of μ when $\epsilon = 0.1$.

The general relationship between power consumption and time needed to synchronise is shown in Fig. 9a, b. Within these figures, we do not distinguish between different coupling

constants and broadcast failure probabilities. We omit the two values for $R = 1, \epsilon = 0.1$ and $\mu \in \{0.1, 0.2\}$ in Fig. 9b to avoid distortion of the graph, since the low coupling strength and low probability of broadcast failure leads to longer synchronisation times and hence higher power consumption. While this might seem surprising it has been shown that uncertainty in discrete systems often aids convergence [22].

The relationship between power consumption and time to synchronise is linear, and the slope of the relation decreases for higher refractory periods. While the linearity is almost perfect for the average values, the maximal values have larger variation. The figures again suggest that $R = 3$ is a sensible and reliable choice, since it provides the best stability of power consumption and time to synchronise. In particular, if the broadcast failure probability changes, the variations are less severe for $R = 3$ than for other refractory periods.

8 Conclusion

In this paper we have introduced a formal concrete model for a network of nodes synchronising their clocks over a set of discrete values. Furthermore, we developed a population model that can alleviate state-space explosion when reasoning about significantly larger networks. We encoded both models as discrete-time Markov chains, and evaluated them for several parameter combinations. Furthermore, we formally connected the models by showing that a concrete model of a network and a population model of same network are probabilistically weakly bisimilar.

Formalising the individual nodes of a network allows for the analysis of their internal properties. Even though we did not give explicit definitions, a concrete network could be instantiated to incorporate different topologies by explicit encoding of possible perturbances in the nodes' transitions. However, the internal structure also complicates the verification of global network properties. Modelling the whole network as the product of the models for the individual nodes quickly, and unsurprisingly, results in a model that is too large to analyse with existing tools and techniques. While the use of appropriate collective abstractions, such as population models, allow for the analysis of larger networks, they often impose restrictions on the topologies of the network that can be considered. We could, of course, simply take the product of individual population models to represent network structures more specialised than the fully-connected graphs considered here, but again we face the consequences of this approach when trying to analyse the resulting model, in particular the explosion of the state space. Furthermore, this also means that every node of a population model may influence all nodes of a connected population model. Finally, our abstract relation would need to take the mapping of single nodes into different components into account. When using population models we lose the possibility to distinguish between nodes having the same internal state. However, this does not restrict our analysis when considering networks of homogeneous nodes where the properties of interest relate to global behaviours of the network itself.

Our current definition of pulse-coupled oscillators only allows for non-negative results of the phase response function. However, there are also oscillator definitions with phase response functions with possibly negative values [48]. That is, instead of shifting the state of an oscillator towards the end of the cycle, the perturbation may reduce the value of the oscillator's state. It would be interesting to study the impact of negative-valued phase response functions in the setting of discrete clock values.

References

1. Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) Wireless sensor networks: a survey. Comput Netw 38(4):393–422
2. Albers S (2010) Energy-efficient algorithms. Commun ACM 53(5):86–96
3. Alur R, Dill DL (1994) A theory of timed automata. Theor Comput Sci 126(2):183–235
4. Alur R, Henzinger TA (1999) Reactive modules. Form Methods Syst. Design 15(1):7–48
5. Angluin D, Aspnes J, Diamadi Z, Fischer MJ, Peralta R (2006) Computation in networks of passively mobile finite-state sensors. Distrib Comput 18(4):235–253
6. Atmel Corporation (2018) ATmega128L: 8-bit Atmel microcontroller with 128 kBytes in-system programmable flash. http://www.atmel.com/images/doc2467.pdf, last accessed 6th April 2018
7. Baier C, Hermanns H (1997) Weak bisimulation for fully probabilistic processes. In: Grumberg O (ed) Computer aided verification. Springer, Berlin, pp 119–130
8. Bartocci E, Corradini F, Merelli E, Tesei L (2010) Detecting synchronisation of biological oscillators by model checking. Theor Comput Sci 411(20):1999–2018
9. Basler G, Mazzucchi M, Wahl T, Kroening D (2009) Symbolic counter abstraction for concurrent software. In: CAV 2009, Springer, Heidelberg, LNCS, vol 5643, pp 64–78, https://doi.org/10.1007/978-3-642-02658-4_9
10. Behrmann G, David A, Larsen KG, Hakansson J, Petterson P, Yi W, Hendriks M (2006) Uppaal 4.0. In: QEST 2006, IEEE Computer Society, pp 125–126
11. Breza M (2013) Bio-inspired tools for a distributed wireless sensor network operating system. PhD thesis, Imperial College, London
12. Cardelli L, Kwiatkowska M, Laurenti L (2018a) Programming discrete distributions with chemical reaction networks. Nat Comput 17(1):131–145. https://doi.org/10.1007/s11047-017-9667-5
13. Cardelli L, Kwiatkowska M, Whitby M (2018b) Chemical reaction network designs for asynchronous logic circuits. Nat Comput 17(1):109–130. https://doi.org/10.1007/s11047-017-9665-7
14. Chen Z, Zhang D, Zhu R, Ma Y, Yin P, Xie F (2013) A review of automated formal verification of ad hoc routing protocols for wireless sensor networks. Sensor Lett 11(5):752–764
15. Christensen AL, Grady RO, Dorigo M (2009) From fireflies to fault-tolerant swarms of robots. IEEE Trans Evolut Comput 13(4):754–766
16. Cristian F (1989) Probabilistic clock synchronization. Distrib Comput 3(3):146–158
17. Degesys J, Basu P, Redi J (2008) Synchronization of strongly pulse-coupled oscillators with refractory periods and random medium access. In: Proceedings of the SAC 2008, ACM, pp 1976–1980
18. Dehnert C, Junges S, Katoen JP, Volk M (2017) A storm is coming: a modern probabilistic model checker. In: CAV, Springer, pp 592–600
19. Delzanno G (2003) Constraint-based verification of parametrized cache coherence protocols. Form. Methods Syst. Design 23(3):257–301
20. Donaldson AF, Miller A (2006) Symmetry reduction for probabilistic model checking using generic representatives. In: ATVA 2006, Springer, Berlin, LNCS, vol 4218, pp 9–23. https://doi.org/10.1007/11901914_4
21. Emerson EA, Trefler RJ (1999) From asymmetry to full symmetry: new techniques for symmetry reduction in model checking. In: CHARME 1999, Springer, Berlin, LNCS, vol 1703, pp 142–156. https://doi.org/10.1007/3-540-48153-2_12
22. Fatès N (2015) Remarks on the cellular automaton global synchronisation problem. In: Proceedinsg of the AUTOMATA 2015, Springer, LNCS, vol 9099, pp 113–126
23. Feller W (1968) An introduction to probability theory and its applications, vol 3. Wiley, New York
24. Gainer P, Linker S, Dixon C, Hustadt U, Fisher M (2017) Investigating parametric influence on discrete synchronisation protocols using quantitative model checking. In: QEST 2017, Springer, Cham, LNCS, vol 10503, pp 224–239. https://doi.org/10.1007/978-3-319-66335-7_14

25. Gainer P, Linker S, Dixon C, Hustadt U, Fisher M (2018) The power of synchronisation: formal analysis of power consumption in networks of pulse-coupled oscillators. In: 20th international conference on formal engineering methods, ICFEM 2018, pp 160–176. https://doi.org/10.1007/978-3-030-02450-5_10
26. Gusella R, Zatti S (1989) The accuracy of the clock synchronization achieved by TEMPO in Berkeley UNIX 4.3BSD. IEEE Trans Soft Eng 15(7):847–853
27. Hahn EM, Li Y, Schewe S, Turrini A, Zhang L (2014) IscasMC: a web-based probabilistic model checker. In: Proceeding of the FM 2014, Springer, pp 312–317
28. Hansson H, Jonsson B (1994) A logic for reasoning about time and reliability. Formal Aspects Comput 6(5):512–535
29. Heidarian F, Schmaltz J, Vaandrager F (2012) Analysis of a clock synchronization protocol for wireless sensor networks. Theor Comput Sci 413(1):87–105
30. Hillston J, Tribastone M, Gilmore S (2012) Stochastic process algebras: from individuals to populations. Comput J 55(7):866–881. https://doi.org/10.1093/comjnl/bxr094
31. Kemeny JG, Snell JL, Knapp AW (2012) Denumerable Markov chains: with a chapter of Markov random fields by David Griffeath, graduate texts in mathematics, vol 40. Springer, New York
32. Kuramoto Y (1975) Self-entrainment of a population of coupled non-linear oscillators. In: International symposium on mathematical problems in theoretical physics, Springer, LNP, vol 39, pp 420–422
33. Kwiatkowska M, Norman G, Parker D (2007) Stochastic model checking. In: SFM 2007, Springer, Berlin, LNCS, vol 4486, pp 220–270. https://doi.org/10.1007/978-3-540-72522-0_6
34. Kwiatkowska M, Norman G, Parker D (2011a) Prism 4.0: verification of probabilistic real-time systems. In: Proceedings of the CAV 2011, Springer, LNCS, vol 6806, pp 585–591
35. Kwiatkowska M, Parker D, Qu H (2011b) Incremental quantitative verification for markov decision processes. In: international conference on dependable systems and networks, IEEE, pp 359–370
36. Maróti M, Kusy B, Simon G, Lédeczi A (2004) The flooding time synchronization protocol. In: Proceedings of SenSys 2004, ACM, pp 39–49
37. MEMSIC, Inc (2017) MICAz wireless measurement system. http://www.memsic.com/userfiles/files/Datasheets/WSN/micaz_datasheet-t.pdf. Accessed 3rd March 2017
38. Mirollo RE, Strogatz SH (1990) Synchronization of pulse-coupled biological oscillators. SIAM J App Math 50(6):1645–1662
39. Pagliari R, Scaglione A (2007) Design and implementation of a PCO-based protocol for sensor networks. In: Proceedings of SenSys 2007, pp 387–388. ACM
40. Parker DA (2003) Implementation of symbolic model checking for probabilistic systems. PhD thesis, University of Birmingham
41. Perez-Diaz F, Zillmer R, Groß R (2015) Firefly-inspired synchronization in swarms of mobile agents. In: Proceedings of AAMAS 2015, international foundation for autonomous agents and multiagent systems, pp 279–286
42. Perez-Diaz F, Trenkwalder SM, Zillmer R, Groß R (2018) Emergence and inhibition of synchronization in robot swarms. In: DARS 2016, Springer, Cham, pp 475–486. https://doi.org/10.1007/978-3-319-73008-0_33
43. Peskin C (1975) Mathematical aspects of heart physiology. Courant Lecture Notes, Courant Institute of Mathematical Sciences, New York University
44. Rhee S, Seetharam D, Liu S (2004) Techniques for minimizing power consumption in low data-rate wireless sensor networks. In: Proceedings of the WCNC 2004, IEEE, pp 1727–1731
45. Soloveichik D, Cook M, Winfree E, Bruck J (2008) Computation with finite stochastic chemical reaction networks. Nat Comput 7(4):615–633. https://doi.org/10.1007/s11047-008-9067-y
46. Sommer P, Wattenhofer R (2009) Gradient clock synchronization in wireless sensor networks. In: Proceedings of IPSN 2009, IEEE, pp 37–48
47. Tyrrell A, Auer G, Bettstetter C (2006) Fireflies as role models for synchronization in ad hoc networks. In: Proceedings of Bionetics 2006, ACM, pp 1–7
48. Wang Y, Nuñez F, Doyle FJ (2012) Energy-efficient pulse-coupled synchronization strategy design for wireless sensor networks through reduced idle listening. IEEE Trans Sig Proc 60(10):5293–5306
49. Webster M, Breza M, Dixon C, Fisher M, McCann J (2018) Formal verification of synchronisation, gossip and environmental effects for critical IoT systems. In: Proceedings of AVoCS 2018, EasyChair, EasyChair Preprint no. 377
50. Werner-Allen G, Tewari G, Patel A, Welsh M, Nagpal R (2005) Firefly-inspired sensor network synchronicity with realistic radio effects. In: Proceedings of SenSys 2005, ACM, pp 142–153
51. Yick J, Mukherjee B, Ghosal D (2008) Wireless sensor network survey. Comput Netw 52(12):2292–2330

Affiliations

Paul Gainer[1] · Sven Linker[1] · Clare Dixon[1,2] · Ullrich Hustadt[1] · Michael Fisher[1]

✉ Sven Linker
s.linker@liverpool.ac.uk

Paul Gainer
p.gainer@liverpool.ac.uk

Clare Dixon
cldixon@liverpool.ac.uk; clare.dixon@manchester.ac.uk

Ullrich Hustadt
u.hustadt@liverpool.ac.uk

Michael Fisher
mfisher@liverpool.ac.uk

1 Department of Computer Science, University of Liverpool, Liverpool, UK

2 Department of Computer Science, The University of Manchester, Manchester, UK