Timing configurations affect the macro-properties of multi-scale feedback systems

Patricia Mellodge
Electrical & Computer Engineering
University of Hartford
mellodge@hartford.edu

Ada Diaconescu
Computer Science & Networks
Telecom Paris, LTCI
diacones@telecom-paris.fr

Louisa Jane Di Felice
Institute of Environmental Science & Technology
Autonomous University of Barcelona
louisajane.difelice@uab.cat

***Abstract*—Multi-scale feedback systems, where information cycles through micro- and macro-scales leading to adaptation, are ubiquitous across domains, from animal societies and human organisations to electric grids and neural networks. Studies on the effects of timing on system properties are often domain specific. The Multi-Scale Abstraction Feedbacks (MSAF) design pattern aims to generalise the modelling of multi-scale systems where feedback occurs across scales. We expand on MSAF to include timing concerns and illustrate their effects via two models: a hierarchical oscillator (HO) and a hierarchical cellular automata (HCA). Results show how (i) different timing configurations significantly affect system macro-properties and (ii) different regions of time configurations can lead to the same macro-properties. These results contribute to theory, while also providing useful insights for designing and controlling such systems.**

***Index Terms*—Multi-scale feedback systems, time scales, oscillator, hierarchical cellular automata, micro-macro behaviour**

I. Introduction

Multi-scale systems are those systems where different scales of time, space, or information granularity interrelate via information flows. If information cycles through the system leading to the adaptation of system entities, they become *multi-scale feedback systems* [39]. E.g., workers in an organisation send information about their state to their managers, who send back commands leading to changes in their behaviours. Similarly, foraging ants lay pheromones, forming a trail that affects their behaviour. In autonomic systems, managed resources are monitored and commands are issued for self-adaptation [28]. In these examples, information from the micro-scale (workers, ants, resources) is *abstracted* onto a macro-scale, and some adaptation at the micro-scale occurs based on information flowing back down. Such feedback cycles can be repeated at recursively higher scales, with increasing abstraction tied to ever larger system parts, e.g., multi-level management organisations, ecosystems or autonomic systems [32], [42]. We use *level* and *scale* to refer to the granularity of observation of space, time, or information within a system [12].

While multi-scale feedback systems can be found across all domains, their generic properties remain under-explored. In previous work, we introduced the *Multi-Scale Abstraction Feedbacks* (MSAF) design pattern, as a means to generalise feedback cycles of information flows operating at multiple abstraction levels in various systems [11], [10]. MSAF identifies scales in relation to *information abstraction* – irrespective of how this is implemented and deployed. A macro-property at a higher scale can be tied to an *exogenous* macro-entity (e.g., an organisation's manager, different from the workers) but can also be *micro-distributed* among micro-entities at a lower scale (e.g., knowledge of power relations distributed across members of an animal society [21]), or *composed* of micro-entities (e.g., forest patch shapes affecting tree growth). Such multi-scale design allows coordinating increasingly larger systems, via a divide-and-conquer approach. Each scale may process similar amounts of information by making a different trade-off between information accuracy and control scope.

Another important trade-off is between a system's reaction time and the control scope considered, at different scales. Such timing aspects depend on inherent communication and processing delays, process execution frequencies, and adaptation lags (i.e., how long before adaptation takes effect). The question of how such timing aspects affect the behaviour of multi-scale systems has been approached primarily in domain-specific ways. General insights that would facilitate cross-domain transfer remain vague and untested. E.g., it is often said that higher levels must operate at a slower rate than lower ones, to ensure system stability [43], [32], [42]. Yet, excessive communication delays between macro- and micro-scales can cause dysfunction depending on system goals [20]. This paper aims to distill common timing concerns from domain-specific timing considerations across multi-scale systems, and to illustrate the effects of these time concerns via two examples.

Sec. III expands the MSAF pattern with common timing aspects. Sec. IV and V illustrate the effects of these aspects on system macro-properties, via two case studies of multi-scale oscillator models: a hierarchical oscillator (HO) [29] and a hierarchical cellular automata (HCA) [13], respectively. We focus on these as many real-world systems feature oscillating patterns, from collective behaviour in animal groups [17] and circadian cycles in the brain [27] to opinion dynamics in social networks [36], clock synchronisation in distributed computing systems, and the coupled motion of pendulum clocks [15].

Results highlight two key timing impacts. First, they show that different timing configurations can significantly affect system macro-properties. This allows using time delays as configuration parameters for changing system behaviours. Second, they show that different regions of time configurations lead to the same macro-property. This can help system robustness to time variations. We do not aim to claim these findings as new contributions to multi-scale oscillators. Rather, we propose them as generic time-related principles in multi-scale feedback systems, based on analysed literature and case studies.

II. Background & Related Work

A. Timing in Multi-Scale Systems

The role of timing in multi-scale systems has been explored mostly in either domain-specific ways (e.g., hierarchical smart grids [40], houses [4] and vehicles [2]) or in generic terms (e.g., multi-level design patterns in self-adaptive systems [42], autonomic systems [32], organic computing (OC) [38], self-aware systems [14], and multi-agent systems [33]). In both cases, results are difficult to reuse and transfer across domains. It is generally considered that lower levels should execute faster than higher ones. While this applies to most systems, the underlying constraints and variants are rarely discussed. Exceptions may also exist depending on desired behaviour (e.g., stock markets may not aim to reach a steady state).

Similar examples can be found in natural system studies. In the field of ecology, multi-scale systems are usually nested. As macro-properties arise from compositions of micro-entities, it is often taken for granted that higher levels operate at a slower rate than lower ones, and that this is necessary for system stability [3]. Similarly, research in biology and paleontology that has focused on nested hierarchies assumes that different timescales are inherent to such systems [41]. Institutional and policy studies, on the other hand, tend to focus on multi-scale systems with exogenous macro-entities (e.g., higher-level bureaucracies send commands to lower ones). Here, delays are often described as dysfunctional, as they can lead to policy ineffectiveness (as upper levels send out-dated commands to controlled resources) [7]. Building on the examples of coral reef formation and power dynamics in Macaque societies, [20] argues that slow-changing variables (at the macro-scale) reduce the amount of environmental uncertainty for micro-entities, but that if these variables are too slow, they cannot be detected by micro-entities. The fact that slow-changing macro-variables can be perceived as constant by micro-entities is also noted by [18], within adaptive neural code, showing how in the flies' vision system adaptation occurs at different timescales, with longer ones providing a separate information transmission channel. Ref. [6] links organism motor functions to primitive language, indicating that macro-properties (or 'symbols') allow to delay immediate reactions to external changes, so as to take into account previous experiences and generate more complex behaviours.

The control systems community has studied timing in multi-scale systems using different terminology, e.g., hierarchical, singularly perturbed, multiloop, nested, and cascade control systems. Hierarchical control systems are addressed in [19], where hierarchies are defined by functions or time horizons of the multilayer configuration. The highest layer necessarily has the longest time horizon to achieve optimal control for the system. Applications of singular perturbation theory to control systems were reviewed in [31], where systems are decomposed into parts with fast and slow dynamics. Multiloop [9], nested [5], and cascade [22] systems generally refer to systems in which multiple feedback loops control variables of importance at different scales. E.g., in aerospace applications different loops address (from micro- to macro-scale): attitude, attitude rate, and guidance. It is generally assumed that higher levels (i.e., the outer loops in the nested system) operate at a slower rate, or that optimal relative rates for stability and performance can be shown in specific situations.

B. Coupled Oscillators

Our two case studies build on existing models of coupled oscillators. Kim et al. [29] explores the synchronisation of coupled biochemical oscillations in cellular systems. Synchronisation is affected by the coupling strength and communication delay, with the parameter space showing three distinct behaviours: oscillation without synchronisation, oscillation with synchronisation, and no oscillation. This model was chosen because of the explicit inclusion of time delay between oscillators as a parameter and we expand it by connecting oscillators in multi-scale configurations, and adding micro-macro communication delays that affect synchronisation (sec. IV).

For the HCA case, we expand on the model in [13], where Cellular Automata (CAs) were organised in a multi-scale configuration, generating macro-structures from uniform micro-scale conditions. Here, we analyse how different combinations of execution frequencies at various scales affect micro and macro oscillating behaviours (sec. V).

As highlighted in [34], natural oscillatory processes tend to follow a multi-scale organisation, with macro-scale frequencies affecting micro-scale behaviour. In most oscillators, communication and adaptation are not instantaneous [37]. E.g., biological systems require a minimum interval to transmit information [8]. The same applies to most artificial systems. Hence, time-related questions are relevant both in the context of oscillator behaviour, within a wide range of applications, and for multi-scale feedback systems, more generally. Several studies discussed the impact of time delays on the behaviour of coupled oscillators (e.g., [26], [30]). These studies suggest that time delays can significantly affect system dynamics [16].

III. MSAF Design Pattern & Time Extensions

A. Overview of the MSAF Design Pattern

The MSAF design pattern in [11], [10] models feedback loops in multi-scale systems in terms of information flows that merge, split, and cycle through different abstraction levels (Fig. 1). Information flows are streams of changes (attached to a material substrate) which can be observed, interpreted, and used for adaptation in line with semantic definitions of information [25]. Such information flows merge and aggregate information at increasingly higher abstraction levels (bottom-up), then split and reify information again at more detailed levels (top-down), forming multi-scale feedback cycles.

A single feedback loop consists of the following steps (blue in Fig. 1): 1) acquisition and abstraction of state information;

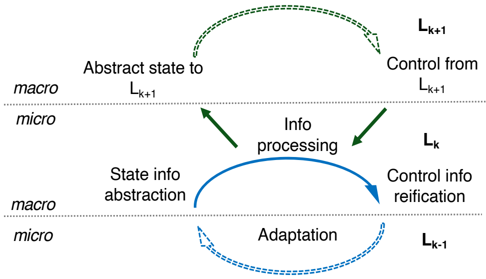

Fig. 1: MSAF Feedback Loops: All arrows are information flows. Each level L_k is a macro level relative to the one below L_{k-1} (except for bottom level, L_0) and a micro level for the one above L_{k+1} (except for top level, L_{M-1}).

2) information processing (e.g., decision); 3) information reification (control command); and 4) adaptation. These steps match existing feedback designs in autonomic (MAPE-K) [28] and organic computing [38], or feedback control systems [23].

Extending this design to multiple scales implies adding further feedback loops on top of each other. This involves two extra steps for connecting feedback loops between levels (in green): a) sending state abstraction of L_k to upper level L_{k+1}; and b) receiving control information from L_{k+1}, to be used as control input, or goal, in L_k's processing step (2). From L_k's perspective, all upper-level feedbacks can be modelled as a single one (dotted green arrow), at L_{k+1}; and all lower levels as one adaptation process (dotted blue arrow), at L_{k-1}.

Hence, a managed resource (at L_0) receives feedback controls that merge information from several scales, covering increasingly larger system scopes. Such multi-scale feedback design helps to control large systems by limiting the amount of processed information at each level and by mixing quick local reactions with slower coordinated responses.

B. Time Considerations in Multi-Scale Feedback Loops

Generalising from feedback systems [23] and control theory [35], we distill several key timing considerations impacting system behaviour: communication delay; processing time; adaptation lag; sample time (for digital systems). To simplify, we merge these into two main timing aspects, applicable to all MSAF steps: i) *execution delay* (τ), the step execution duration (including communication and processing); and ii) *execution interval* (Δt), how often the step executes. We group MSAF steps (1-3) (abstraction, processing, and control) into a single 'management flow' (including inter-level abstraction (a) and control (b) for higher levels), featuring an execution delay τ_{mng} and interval Δt_{mng}. The adaptation step (4) also features an execution delay τ_{adpt} and interval Δt_{adpt}.

All timing considerations from 'classic' feedback control systems apply here. We highlight some of these below without aiming for a comprehensive review. Delay in the management flow τ_{mng} implies the risk of providing a control command (output) based on an outdated monitored state (input). It may lead to oscillations, longer settling times, or instability [23]; and decrease reactivity to state disturbances. Yet, if $\tau_{mng} << \tau_{adpt}$ there is a risk of overreaction from the management flow, i.e., repeating or exacerbating a control command as it fails to perceive the effects of a previous command. This risk is removed when controls are not 'cumulative' (e.g., goal-oriented commands can be repeated with the same effect).

With respect to execution intervals, the smaller the Δt_{mng} (i.e., the management flow executes more often), the more reactive it can be to state changes, while again, risking to overreact if it executes before previous controls take effect. Overreaction is avoided if $\Delta_{adpt} < \Delta_{mng}$ (also considering delays); or when controls are repeated (without increased amplitudes) and the adaptation flow only executes the last one (if $\Delta_{adpt} > \Delta_{mng}$). Ideally, the management flow would execute rapidly ($\tau_{mng} \to 0$), but only at intervals large enough for the effects of its commands to take effect in the adaptation flow ($\Delta_{mng} \sim \tau_{adpt}$). Other combinations of execution delays and intervals are also viable. The above considerations become more complex when feedback cycles extend across multiple scales. When management flows at different scales execute in parallel, each flow at L_k gets abstract state information from L_{k-1} and control information from L_{k+1}, to issue commands back to L_{k-1}. For L_k, information from L_{k-1} is more recent than from L_{k+1}, as the latter would have crossed at least an extra scale. Yet, information from L_{k+1} includes abstractions about broader system scopes (under control levels from L_{k+1} to L_{M-1}). This allows L_k entities to coordinate their local actions based on wider system views. Control information for L_0 entities merges information flows from all system scales, with lower-scales information being narrower but more recent (or accurate) and higher-scales information being broader but more outdated. Hence, higher-level management flows would have larger delays than lower ones, as it takes more time for their input and output flows to travel to and from L_0. This situation is due to system implementation constraints (i.e., inherent communication and processing delays at each level), rather than being a desirable system design property. Still, in case of rapid management relative to adaptation delay ($\tau_{mng}<<\tau_{adpt}$) it makes sense to execute higher managers less often than lower ones ($\Delta_{mng,k}>\Delta_{mng,k-1}$), to avoid overreactions or instability. However, increasing $\Delta_{mng,k}$ may decrease the system's coordinated responses. Typical solutions combine fast, accurate, localised reactions from lower-scales, for avoiding disaster (e.g., reflexes in organisms, obstacle avoidance in autonomous cars) with slower, more context-aware responses, for coordinated behaviour (e.g., strategic planing in organisms, rerouting autonomous cars). Various combinations of cross-level execution delays and intervals lead to different system behaviours (macro-properties). We focus on examples illustrated via our two applications (HO and HCA).

IV. Biochemical Oscillator Model

A. HO Overview

We use the coupled biochemical oscillator model from [29] which is extended to: a) a flat network of more than two coupled oscillators and b) a hierarchy of oscillators. Coupled

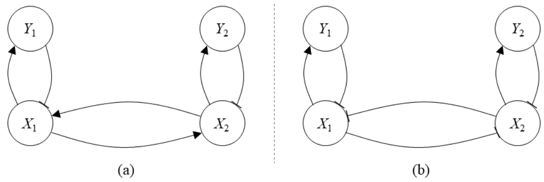

Fig. 2: Coupled oscillators with PP coupling (a) and NN coupling (b).

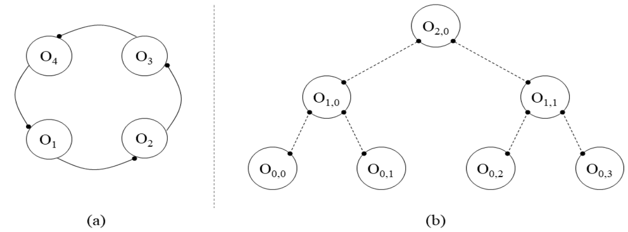

Fig. 3: (a) A flat network with $N = 4$ oscillators and (b) a hierarchy of oscillators with $M = 3$ levels and $C = 2$ children per oscillator. Each oscillator consists of X and Y components. The small circles indicate either P or N type coupling between the X components of each oscillator.

biochemical oscillators are observed throughout many systems in nature [29] (e.g., cellular processes involving circadian rhythms). A single oscillator consists of two interacting components X and Y (e.g., mRNA and protein) in which X inhibits its own synthesis while promoting that of Y and Y inhibits its own and X's synthesis, resulting in a feedback relationship that induces oscillation in the concentrations of both. In coupled biochemical oscillators, each oscillator consists of two interacting components X and Y, with coupling between the X components as shown in Fig. 2. There are two types of coupling: double-positive (PP) in which each X promotes the synthesis of the other, or double-negative (NN) in which each X inhibits the synthesis of the other. Further, the interactions have two parameters: coupling strength F between the two X's and communication time delay τ associated with their interaction. With respect to oscillation period, in [29] it was shown that PP coupling performed better with low τ, while NN was more robust with large τ. In this section, we extend the model to multiple oscillators in multiple configurations to determine if similar results are obtained.

To accommodate multiple oscillators, we generalise this model to include more oscillators arranged in a 'flat' configuration (i.e., no hierarchy). Fig. 3(a) shows this system with four oscillators, which can have either P or N type coupling. Each X_i promotes or inhibits X_{i+1} depending respectively on P or N type coupling. The HO design includes oscillators at different levels. Here, oscillators differ from their stand-alone forms by taking into account (detailed) state information from the lower levels and (abstract) control information from the upper levels. The macro-entities are oscillators whose models are modified from [29]. Fig. 3(b) shows the HO structure for three levels with two children per oscillator. The coupling occurs between the X components of the oscillators between levels. There is no direct communication between oscillators at a given level.

B. Flat and HO Models

The flat network of oscillators is modeled by the following differential equations; for $i = 1..N$, where N is the number of oscillators and $i-1$ is within the range $1..N$ (so if $i = 1$, then $i - 1 = N$). It is assumed that all oscillators are coupled with the same strength F, type P or N, and time delay τ.

$$\frac{dX_i}{dt} = \frac{1 + PP\left(FX_{i-1}(t-\tau)\right)^3}{1 + \left(FX_{i-1}(t-\tau)\right)^3 + \left(\frac{Y_i(t-2)}{0.5}\right)^3} \cdots \\ \cdots - 0.5X_i(t) + 0.1 \tag{1}$$

$$\frac{dY_i}{dt} = \frac{\left(\frac{X_i(t-2)}{0.5}\right)^3}{1 + \left(\frac{X_i(t-2)}{0.5}\right)^3} - 0.5Y_i(t) + 0.1 \tag{2}$$

For the HO, the differential equations for X depend on their level. At the highest level, oscillators only act as aggregators of information from the lower level. At the lowest level, they only receive feedback from their corresponding macro-entity. Oscillators at middle levels receive information from above and below. While the equation modelling the Y component of all oscillators is analogous to (2), the model for X is:

$$\frac{dX_{m,i}}{dt} = \frac{1 + PP\left(F_m\Gamma_{m,i}(t-\tau_m)\right)^3}{1 + \left(F_m\Gamma_{m,i}(t-\tau_m)\right)^3 + \left(\frac{Y_{m,i}(t-2)}{0.5}\right)^3} \cdots \\ \cdots - 0.5X_{m,i}(t) + 0.1 \tag{3}$$

where

$$\Gamma_{m,i}(\cdot) = WX_{m+1,p_{m,i}}(\cdot) + (1-W)\bar{X}_{m,i}(\cdot) \tag{4}$$

and $\bar{X}_{m,i}(\cdot)$ denotes the mean X concentration taken over the children of $X_{m,i}$; $m = 0..M-1$, where M is the number of levels; $i = 0..N_m - 1$, where N_m is the number of oscillators in level m; $p_{m,i}$ is the position of the parent of $X_{m,i}$; $K = M-1$; F_m is the coupling strength; τ_m is the time delay; and $W \in [0, 1]$. Parameters F_m and τ_m are constant across a level and W is constant across the system.

In the HO model, the abstracted information is the average concentration of a substance X in micro-entities given by $\bar{X}_{m,i}(\cdot)$. The feedback information that is sent down from macro to micro is the concentration of X in the macro-oscillator, which impacts the concentration of X in the micro-oscillators through (3). Function $\Gamma_{m,i}(t-\tau_m)$ in (4) communicates both the abstracted information from the lower level and feedback information from the higher level, with W controlling the relative importance of the feedback signal. For the top and bottom levels, Γ is modified to eliminate the first and second terms respectively due lack of feedback from above and lack of aggregated information from below.

C. Simulation Time and Sequence

The oscillators are continuous time systems described by differential equations, digitized using a numerical solver. Each differential equation was solved simultaneously for specified interval of time $[0, T_{end}]$, resulting in a discrete time series of X and Y concentration values.

With respect to timing aspects described in Section III-B, there is a micro-to-macro abstraction delay and macro-to-micro feedback delay, both characterised by τ_m, which reflects the time it takes for the concentration of X to be transmitted (i.e., transmission delay, in literature). This means that the abstracted state and feedback is based on old micro-state information. These semantics simulate communication delays τ_{mng} for abstracted states, with negligible adaptation delay ($\tau_{adpt} = 0$) and continuously executing feedback cycles (Δ_{mng} and Δ_{adpt}). Delays at higher levels are always higher than at lower levels due to delay accumulation.

D. Experimental Settings

A number of system configurations were simulated. Flat networks had from 4 to 11 oscillators. HOs had 64 bottom-level oscillators arranged in two ways: $M = 2$ levels, $C = 64$ children; and $M = 4$ levels, $C = 4$ children. For each configuration a range of parameter values was used: coupling strength $F = 0..8$, time delay $\tau = 1..15$, and coupling (PP, NN). Each configuration was simulated for 10 runs, with the initial X values of an oscillator randomly chosen from the interval (0,1) and the initial Y values set to 1. Values of oscillation frequency, amplitude, and synchronisation time were averaged over the 10 runs. Since in all cases, only small variations in these values were measured and overall system behavior was consistent, 10 runs was deemed sufficient to investigate the macro-properties.

E. Overall Behaviour

The coupled oscillator systems exhibit three basic types of emergent behaviour: unsynchronised oscillation, no oscillation, and synchronised oscillation with all levels in phase. In the case of the HO, there is an additional type: synchronised oscillation with levels out of phase. These behaviours are shown in Fig. 4 (for $M = 3$ levels and $C = 2$ children). The three stacked time series plots show the oscillator X concentrations at each level plotted together, with the top level having one oscillator, the middle level having two oscillators, and the bottom level having four oscillators.

F. Experimental Results

In flat networks, it was found that synchronisation occurred consistently in PP coupled systems having no more than 5 oscillators and for NN coupled systems having no more than 10 oscillators in the network. Furthermore, the region of the parameter space that achieved synchronisation was a relatively small subset. To achieve consistent synchronisation in systems with more than 10 oscillators and for a larger range of F and τ values, it is necessary to have a hierarchical structure.

For HO systems, we present two kinds of results relevant to our contribution (with analogs in the HCA model): the effect of system parameters on a) generated macro patterns and b) oscillation periods. Other results on oscillation amplitudes and synchronisation settling time are also briefly discussed.

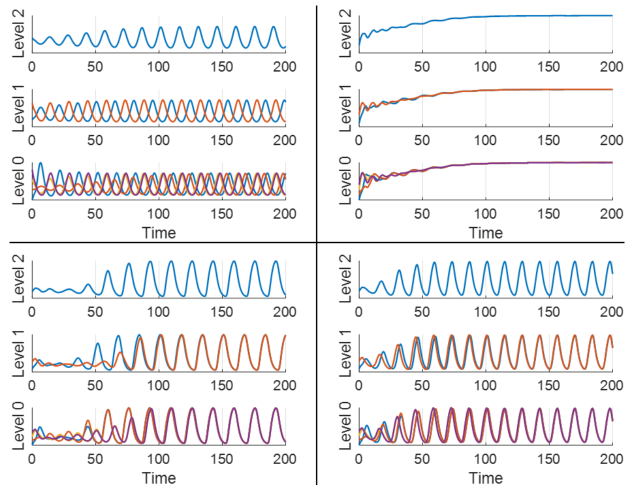

Fig. 4: Different types of emergent behaviour for an HO with $M = 3$ levels and $C = 2$ children. Upper left: unsynchronised oscillation. Upper right: no oscillation. Lower left: synchronised oscillation with adjacent levels out of phase. Lower right: synchronised oscillation with all levels in phase.

1) Impact of time on generated macro patterns: In HO systems, Fig. 5 shows the emergent behaviour of the system for all configurations tested. In all cases, the bottom level synchronised for the middle range values of F (yellow region). For low F values, there was oscillation, but no synchronisation (light blue region). For high F values, there were no oscillations (dark blue region). The smaller hierarchy ($M = 2$) achieved synchronisation in a larger part of the parameter space. There are two distinct transition regions: from unsynchronised to synchronised oscillations and from synchronised oscillations to no oscillations. The transition from synchronised to no oscillations was deterministic. For both PP systems, oscillations only occurred for $0 \leq F \leq 3.5$. This transition was unaffected by time delay. In contrast, for both NN systems, the transition from oscillations to no oscillations occurred for F between 3 and 6, depending on the value of time delay. The transition from unsynchronised to synchronised oscillations was stochastic, as indicated by the color transition from light blue to yellow: yellow, synchronisation happened in each run; light blue, it did not happen in any run; in-between colors, synchronisation occurred only in some runs, according to the color scale to the right of the plot.

2) Impact of time on oscillation periods: Fig. 6 shows how the period of oscillation varies with the coupling strength and time delay. Time delay has a larger impact on the period for PP coupled systems while the effect is negligible for NN coupled systems. The number of levels had no effect on the period.

Oscillation amplitude varies with time delay within the synchronised region, with a larger impact occurring in NN systems (in contrast to the effect on period). The number of levels ($M = 2$ and $M = 4$) had a negligible impact on amplitude; but fewer levels lead to faster synchronisation, for both PP and NN coupling. The effect of time delay on

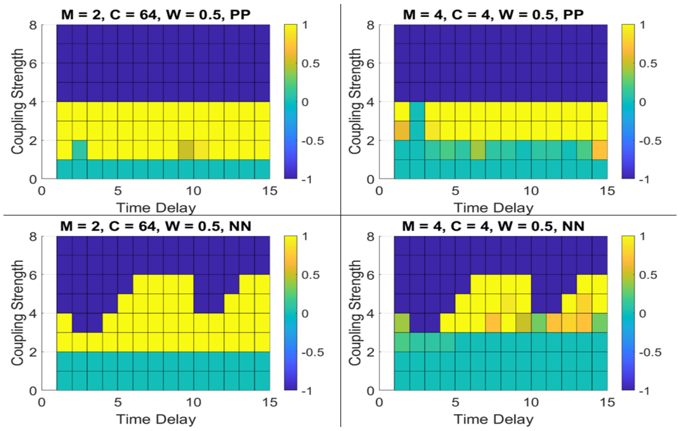

Fig. 5: Each plot shows the emergent behaviour for $F = 0..8$ and $\tau = 1..15$. Colors indicate regions in the parameter space where different behaviours occur: Dark blue, no oscillations; yellow, synchronised oscillations (with levels in or out of phase); light blue, unsynchronised oscillation; color gradients from yellow to light blue, synchronisation only in some runs.

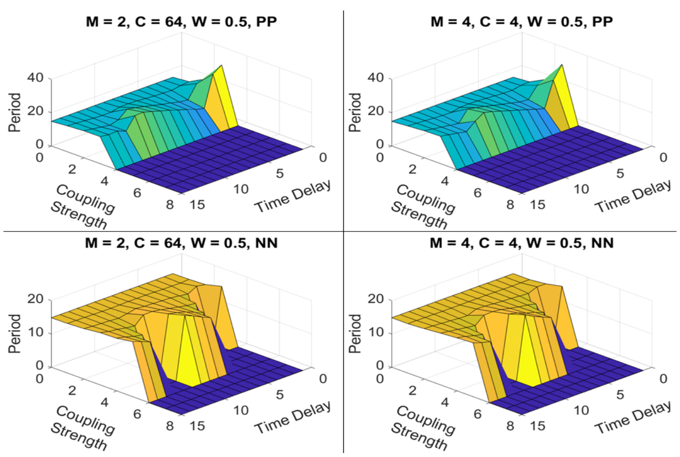

Fig. 6: Each plot shows the oscillation period for $F = 0..8$ and $\tau = 1..15$. Upper left: $M = 2$ levels, $C = 64$ children, PP coupling. Upper right: $M = 4$ levels, $C = 4$ children, PP coupling. Lower left: $M = 2$ levels, $C = 64$ children, NN coupling. Lower right: $M = 4$ levels, $C = 4$ children, NN coupling. Zero values mean there is no oscillation.

synchronisation time is more pronounced for NN coupling. Full results can be found at [1].

G. Discussion

These results show the effect of time delay on the system's macro patterns and their properties. For NN coupling τ affects the type of synchronisation, but not the amplitude. Conversely for PP coupling, τ affects the oscillation period, but not the synchronisation type. For both types, the number of levels M affects unsynchronised to synchronised transition, due to the increased time delay caused by larger M. Further, these results indicate that HO systems are advantageous compared to flat networks: a) HO systems are able to synchronise more oscillators: 64 oscillators for HO, compared to a maximum of 5 and 10 for flat networks (with PP and NN, respectively). b) The desired synchronisation behaviour occurs in a larger region of the parameter space as noted by the large yellow regions in Fig. 4. In contrast, at their maximum size, flat networks achieved synchronisation for a single combination of coupling strength and time delay. c) Due to their large synchronised region, HO systems are more robust to parameter variations. For NN, a change in one parameter (time delay) can also be compensated by changing another parameter (coupling strength) to reach synchronisation without affecting the amplitude.

V. Hierarchical Cellular Automata Case Study

A. HCA Overview

Cellular Automata (CA) are discrete models where the state of each entity (cell) at t depends on the cell's previous state and on its neighbours' states, at t-1. Cells are usually arranged in a grid and their inter-dependency modelled via a rule set. CA, including coupled CA, have been employed to model a wide range of complex systems, including multi-scales [24]. To analyse timing effects on such multi-scale systems, we reuse the Hierarchical Cellular Automata (HCA) simulator in [13]. It organises multiple CA into several levels. Cross-level CA interactions follow the MSAF pattern: a) abstract state information (bottom-up) and b) control commands or goals (top-down). Each CA (except the top) has two rule sets: Expansive rules (R_E) increase the CA's number of live cells; Regressive rules (R_R) decrease them. The control goal from above dictates the CA's active rules (to execute). CAs at different levels have different R_E-R_R rule-pairs.

Each CA at a lower level L_k is mapped bidirectionally to a single cell of a CA at a higher level L_{k+1}. In the bottom-up mapping, the entire state of a lower CA is abstracted (based on the percentage of its live cells relative to a threshold Th_k) and sets the binary state of its mapped cell in a higher CA. In the top-down mapping, the state of each cell in a higher CA controls the rule activation of its mapped lower CA (i.e., sets R_E or R_R). These bidirectional interactions form inter-level feedbacks, replicated at successive levels, up to the top (which only executes static rules). Simulations are deterministic.

B. HCA Notation & Inter-level Mapping

Table I summarises the main HCA concepts and notations (details in [13]). HCA consists of several levels (L_k), each with one or several CA ($CA_{k,i}$) (Fig. 7). Each $CA_{k,i}$ at a micro-level is mapped bidirectionally to one cell $C_{k+1,j,s}$ of a $CA_{k+1,j}$ at the macro-level: a) the state abstraction ($AS_{k,i}$) of $CA_{k,i}$ (micro) is set as the state ($CS_{k+1,j,s}$) of its mapped cell $C_{k+1,j,s}$ (macro) via (5) and (6) and b) the control goal ($G_{k,i}$) from the cell state $CS_{k+1,j,s}$ (macro) sets the active rule of its mapped $CA_{k,i}$ (micro) via (7) and (8).

$$CS_{k+1,i} \leftarrow AS_{k,i}\ ,\quad i = 1..N_k\ ,\ map(C_{k+1,i}; CA_{k,i}) \tag{5}$$

$$AS_{k,i} = \begin{cases} 1, & \text{if } \sum_{s=1}^{S_{k,i}} CS_{k,i,s} >= Th_k \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

TABLE I: Main HCA Concepts and Notations

Notation	Description
L_k	Level k, with $k = 0..M$-1, M the N° of HCA levels
$CA_{k,i}$	Cellular Automata i at level L_k, $i = 0..N_k$-1, N_k the N° of CA at L_k
$CA_{k,i} \Rightarrow < state >$	$CA_{k,i}$ converges to steady state $< state >$: either O_P (oscillate with period P) or S_X (stuck with X live cells)
$CS_{k,i,s}$	State of Cell $C_{k,i,s}$ of $CA_{k,i}$ ($S_{k,i}$ cells), s=1..$S_{k,i}$ $CS_{k,i,s} \in \{0, 1\}$, $0 \equiv$ false/dead, $1 \equiv$ true/live
$AS_{k,i}$	Abstract State of $CA_{k,i}$, $AS_{k,i} \in \{0, 1\}$
Th_k	Threshold for calculating Abstract States of $CA_{k,i}$ at L_k
$G_{k,i}$	Goal of $CA_{k,i}$; $G_{k,i} \in \{0, 1\}$
$R_{k,i}$	Active Rules (executing) of automaton $CA_{k,i}$
$map(C_{k,i,s}; CA_{k-1,j})$	Mapping between cell $C_{k,i,s}$ and automaton $CA_{k-1,j}$; implies transfer of abstract state (up) & cntrl. goal (down)
Fq_k	Activation frequency of level L_k– the number of activations of L_k after which $CA_{k,i}$ actually execute $R_{k,i}$.
$Fq = ...$	Activation *frequency pattern* across HCA levels; E.g., Fq= 1-2-3 means that Fq_0=1, Fq_1=2, Fq_2=3

$$G_{k,i} \leftarrow CS_{k+1,i}, \quad map(C_{k+1,i}; CA_{k,i}) \tag{7}$$

$$R_{k,i} = \begin{cases} R_{k,E} \text{ (Expansive rules)}, & \text{if } G_{k,i} == 1 \\ R_{k,R} \text{ (Regressive rules)}, & \text{if } G_{k,i} == 0 \end{cases} \tag{8}$$

C. Simulation Time & Sequence

A HCA simulation proceeds in discrete cycles, each one executing all levels successively, from bottom L_0 to top L_{M-1}. A cycle consists of M discrete steps t_k (k=0..M-1), each one executing all CAs at a corresponding level L_k.

Each $CA_{k,i}$ in an active level L_k: **i) exchanges information with its macro-CA** (sends $AS_{k,i}$, (6); gets a control goal $G_{k,i}$, (7)); **ii) sets its active rules** ($R_{k,i}$) **depending on its goal** ($G_{k,i}$, (8)); and **iii) steps** (executes its active rules). As exceptions, $CA_{0,i}$ (bottom) do not get abstracted states from below, using their previous state instead; and CA_{M-1} (top) do not get goals from above, using a static rule. During a step t_k, all CAs at L_k execute in parallel; the step ends when all $CA_{k,i}$ have finished executing.

With respect to timing aspects in subsec. III-B, HCA considers state abstraction delays as negligible and control communication delays of one cycle between each two levels (i.e., $\tau_{mng,k}$=1 cycle). Hence, abstract state is always up to date (i.e., travels across all levels in one cycle) yet controls take M steps to arrive from top to bottom. Adaptation delay is also negligible (τ_{adpt}=0). Each level has an activation frequency Fq_k (i.e., execution interval $\Delta_{mng,k}$): Fq_k=d means that L_k only activates at every d cycles. Finally, control commands G_k are cumulative (repeating them exacerbates the effect), yet do not increase their values if inactive micro-CAs ignore them.

D. Experimental Settings

We set-up a three-level HCA: L_0 (bottom), L_1 (middle) and L_2 (top), Fig. 7. L_0 has 32 CAs (4x8 matrix), of 441 (21x21) cells each. This maps to a 32 (4x8) cell CA at L_1 which maps to a one-cell CA at L_2. To simplify HCA behaviour, we only experiment here with inversible rule-pairs (e.g., from any CA state, executing R_E and R_R leads to the same state).

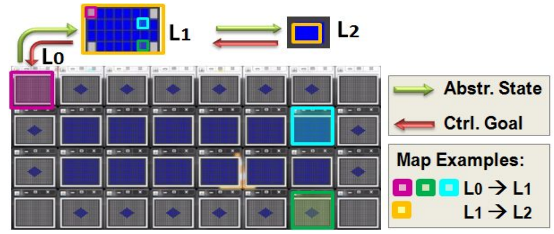

Fig. 7: 3-level HCA, with 3 differentiated states at L_0: 4x Corners CA_{Cr} (purple), 16x Borders CA_{Bo} (green), 12x Core CA_{Co} (cyan).

a) $R_{0,E}$ & $R_{1,E}$ State	Neighbour Count 0	1	2	3	4
0	0	1	1	1	1
1	1	1	1	1	1

b) $R_{0,R}$ State	Neighbour Count 0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	1	1

c) $R_{1,R}$ State	Neighbour Count 0	1	2	3	4
0	0	0	0	0	1
1	0	0	0	0	1

Fig. 8: Diamond Rules: a) R_E for L_0 &L_1; b) R_R for L_0; c) R_R for L_1.

All experiments start with $CA_{k,0}$ in the same initial state, executing $R_{0,E}$; and CA_1 and CA_2 in dead state (sending G=1 control goals until first changing to live states). Experiments vary in configurations for the two thresholds (Th_0 and Th_1) for calculating abstract states for L_1 and L_2 via (6) and the three activation frequencies (Fq_0, Fq_1 and Fq_2), setting the delay between subsequent level activations (for L_0, L_1 & L_2).

To show the rule-independence of our results, we tested two inversible rule-pairs at L_0: 1) **Diamond** (Fig. 8: (a) R_E & (b) R_R; non-toroidal), with $CA_{0,i}$ initialised with one central live cell, generating an expanding and regressing diamond shape (Fig. 9-top); and 2) **Line** (R_E: cell 'dead' $\rightarrow$ 'live' if at least one live neighbour, and R_R: cell 'live' $\rightarrow$ 'dead' when less than 4 live neighbours; toroidal), with $CA_{0,i}$ initialised with a central horizontal line of live cells, generating a vertically expanding and retracting rectangle (Fig. 9-bottom).

For CA_1 we always use a non-inversible rule-pair (Fig. 8: (a) R_E & (c) R_R), which generates four possible states (Fig. 10): *Null* (0 live cells), *Core* (12 live cells), *Cross* (28 live cells), and *Full* (32 live cells). L_2 has a single rule-set, inverting the current state (i.e., live $\rightarrow$ dead; dead $\rightarrow$ live).

We define three experimental sets, with Th_0=0.1 (i.e., 10% live cells) and $Th_1 \in \{0.3, 0.7, 0.9\}$. These three values fall in between the four CA_1 states; all other values are redundant.

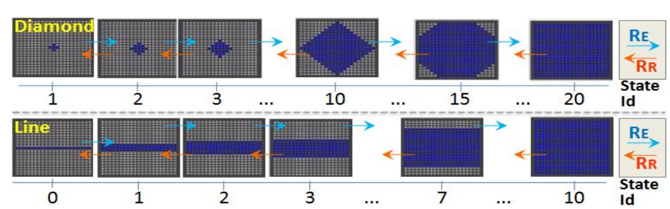

Fig. 9: 21x21 CA_0 States: *expand* left-to-right, R_E; *regress* right-to-left, R_R **Diamond (top)**: generates 20 states, with N° of live cells: 1, 5, 13, 25, 41, 61, 85, 113, 145, 181, 221, 261, 297, 329, 357, 381, 401, 417, 429, 437, 441; **Line (bottom):** 11 states: 21, 63, 105, 147, 189, 231, 273, 315, 357, 399, 441.

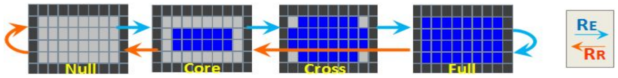

Fig. 10: CA_1 States: expand left-to-right (R_E): Null $\rightarrow$ Core $\rightarrow$ Cross $\rightarrow$ Full; regress right-to-left (R_R): Null $\leftarrow$ Core $\leftarrow$ Full (skip Cross, non-inversible).

Within each set, we run tests with varying activation frequencies: Fq_0=1..2, Fq_1=1..5, Fq_2=1..5. A test with Fq=1-3-5 means that Fq_0=1, Fq_1=3, Fq_3=5. This leads to about 300 tests (2 rules x 3 Th_1 values x 50 Fq values).

E. Overall Behaviour

A finite CA can only converge ($\Rightarrow$) to three behaviours: i) *dead* (S_0), all cells set to 0; ii) *live-stuck* (S_X), blocked in a state with X live cells (set to 1); iii) *oscillating* (O_P), cycling through a set of states, with the state sequence repeating every P steps. At L_0, a $CA_{0,i}$'s behaviour depends on the goal pattern received from L_1 (i.e., 1 & 0 sequence activating R_E & R_R). If a goal pattern has more 0s than 1s, activating R_R more than R_E, then $CA_{0,i} \Rightarrow S_0$. If R_E activates more than R_R, then $CA_{0,i} \Rightarrow S_{441}$. For 'balanced' R_E &R_R patterns, $CA_{0,i} \Rightarrow O_P$.

Superposing goal patterns from CA_1's four states differentiates $CA_{0,i}$ into a maximum of three groups (Fig. 7): 1) Core $CA_{0,Co}$, the 12 $CA_{0,i}$ at the core of L_0's 4x8 matrix; mapped to the 12 live cells in CA_1's Core state; 2) Corner CA_{0-Cr}, the 4 $CA_{0,i}$ at the corners of L_0's matrix; mapped to the 4 dead cells in CA_1's Border state; and 3) Border CA_{0-Bo}, 16 remaining $CA_{0,i}$ on the borders of L_0's matrix (no corners).

In brief, $CA_{0,i}$ have an expanding or regressing tendency (i.e., growing or shrinking N° of live cells) depending on the active rule set, R_E or R_R, respectively. When crossing Th_0, this tendency is propagated (and accentuated) upwards through CA_1. When crossing Th_1, it reaches CA_2, which inverses it. The inverse tendency is propagated downwards, back to $CA_{0,i}$, which crosses Th_0 the other way. The propagation process is repeated upwards with the opposite tendency, then inverted again at CA_2. This creates an expansion-regression oscillation across levels. Because CA_1 executes its own rule-pair $CA_{0,i}$ differentiate, following different behaviours and converging to different states (e.g., 3 $CA_{0,i}$ states in Fig. 7).

F. Experimental Results

We present two kinds of results, relevant to our contribution. Firstly, we show how different activation frequencies constrain the possible oscillation periods P that may occur at HCA levels. We also note that many frequency combinations generate the same oscillation period P (macro-property), though not necessarily through the same state set. Secondly, we show how different activation frequencies lead to different macro-patterns amongst $CA_{0,i}$, i.e., whether $CA_{0,Co}$, $CA_{0,Bo}$, $CA_{0,Cr}$ converge to O_P, S_0 or S_X. Full results can be found at [1].

1) Impact of time on oscillation periods: Table II summarises results for **Diamond rules**, with $Th_0 = 0.1$, $Th_1 = 0.7$, $Fq_0 = 1$, and Fq_1 & Fq_2 varying between 1 and 5. At L_0, some $CA_{0,i}$ oscillate and some end in a static state. To simplify, we only show here the O_P value for the $CA_{0,i}$ that do oscillate and discuss differentiated macro-patterns in the next subsection. Equivalent results were obtained for $Fq_0 = 2$.

TABLE II: Oscillations for Diamond rules: Th_0=0.1, Th_1=0.7, Fq_0=1, Fq_1&Fq_2=1..5

Max Fq Val.	Fq0-Fq1-Fq2	Final State	
MaxFq=1		**L0 & L1**	**L2**
Common case:	111	O2	O2
MaxFq=2		**L0 & L1**	**L2**
Common cases:	112,121,122	O4	O4
MaxFq=3		**L0 & L1**	**L2**
Common cases:	113,131,123,132,133	O6	O6
MaxFq=4		**L0 & L1**	**L2**
Common cases:	114,141,124,142,,144	O8	O8
Exceptions:	134	O24	O8
	143	O8	O24
MaxFq=5		**L0 & L1**	**L2**
Common cases:	115,151,125,152,,155	O10	O10
Exceptions:	135	O30	O10
	153	O10	O30
	145	O20	O10
	154	O10	O20

Results show clear correlations between activation frequencies (Fq pattern) and ensuing O_P values. We generalise these for P_0 and P_1 via (9) based on empirical analysis (P_2 features an equivalent formula, replacing Fq_1 with Fq_2 in (9)).

In brief, we note that for Fq=1-1-1 we get O_2 at all levels. This is due to L_2's state-inversion rule and the inter-level delay τ_{mng} of 1 cycle, which changes active rules (between P_E & P_R) at each cycle, going back-and-forth between two states and producing O_2, the double of the rule-changing frequency. When one of the level frequencies doubles, the rule-changing frequency also doubles, and P doubles this value. E.g., for $Fq \in \{$1-1-2,1-2-1,1-2-2$\}$ we get O_4 at all levels. We observe equivalent results in all cases marked in green in Table II, where P is the double of the maximum frequency of all levels. When frequencies at L_1 and L_2 differ (red in Table II), the level with the highest frequency still features P that doubles that frequency; while in the level with lower frequency P is a multiple of the lowest common multiplier (LCM) of all level frequencies. This makes sense as that is the only cycle where all levels execute. L_0 always follows the same O_P as L_1.

$$P_0 = P_1 = \begin{cases} a * Fq_1, \text{ if } Fq_1 \geqslant Fq_2 \\ b * LCM(Fq_0, Fq_1, Fq_2), \text{ if } Fq_1 < Fq_2 \end{cases} \quad a, b \in \mathbb{N}_{>0} \setminus \{1\} \tag{9}$$

Similar results were obtained when increasing Th_1 to 0.9. The main difference was that for Fq=1-3-4, we obtained O_{12} at all levels; rather than O_{24}-O_{24}-O_8 as when Th_1=0.7. Results for Th_1=0.3 were also similar, the only difference occurring for $Fq \in \{$1-4-5, 1-2-3$\}$, where the HCA $\Rightarrow S_0$. Using **Line rules** produced equivalent $O_{k,P}$ types when testing the same configuration ranges. A difference here is that the toroidal configuration means that a $CA_{0,i}$ that grows (R_E) to all live cells can no longer regress (R_R), hence staying in S_{441} (e.g.,for Fq_0=1, Fq_1=5, Fq_2=1..5 & any thresholds).

2) Impact of time on generated macro patterns: At L_0, macro-patterns occur as different behaviours in the three $CA_{0,i}$

TABLE III: Macro-patterns for Diamond (left) & Line (right)

Diamond Rules, Th0=0.1, Th1=0.9			
Fq0-Fq1-Fq2	CA0,Cr	CA0,Bo	CA0,Co
1-1-3,2-1-3,3-1-2, 3-1-3	S0	O6	S441
3-2-1,3-2-2	O12	O12	S441
1-3-1,1-3-3,1-2-3, 1-3-2,2-3-1,2-2-3, 2-3-2,2-3-3,3-1-1, 3-3-1,3-2-3,3-3-2, 3-3-3	O6	O6	S441

Line Rules, Th0=0.1, Th1=0.3			
Fq0-Fq1-Fq2	CA0,Cr	CA0,Bo	CA0,Co
1-1-3,3-1-1,3-3-1, 3-3-2,3-3-3	S0	S0	O6
2-1-3	S0	S0	S0
3-1-2,3-1-3,	S0	O6	S441
1-3-1,1-3-3,1-2-3, 1-3-2,2-3-1,2-2-3, 2-3-2,2-3-3	S0	S0	S441

groups: $CA_{0,Cr}$ (corners), $CA_{0,Bo}$ (borders), and $CA_{0,Co}$ (core). Different Fq patterns generate different macro-patterns. E.g., Table III shows macro-patterns for the Diamond and Line rules for all frequency patterns with at least one $Fq_k = 3$.

G. Discussion

Results show how timing adjustments (e.g., activation frequencies) can be used to control macro-behaviours (e.g., oscillation periods P). Notably, P_k only depends on cross-level frequency patterns (9), while the rule-pairs and threshold configurations only impact the state set that such oscillations cycle through and the macro-pattern. Furthermore, a wide range of frequency combinations lead to similar oscillation behaviour, e.g., all combinations with a maximum frequency of 2 lead to O_4 forms and so on, all the way to maximum of Fq=5 leading to O_{10} (and multiples, e.g., O_{20}, O_{30}). These are key properties for controlling oscillations, while being robust to disturbances (e.g., in thresholds or frequencies).

Results also show how varying frequencies that lead to the same oscillation periods at L_0 (as above) may do so via different macro-pattern formations (i.e., different $CA_{0,i}$ states). Hence, activation frequencies become configuration parameters for shifting overall system behaviour (i.e., macro-pattern productions). As above, different frequency configurations can lead to the same macro-pattern, which can enhance robustness.

VI. General discussion

The above examples brought to the fore two significant aspects about how timing may impact system macro-behaviours: i) different combinations of execution delays and frequencies impact resulting macro-properties (all other parameters being the same) – hence timing properties represent system configurations in their own right, to be studied or designed as such; and ii) multiple timing configurations may result in the same macro-property, which may help enhance system robustness against time-related variations. These results are in line with existing insights across specific fields, including control theory. Here, we provided a common framework to test and discuss such insights in a more generic manner, allowing to formalise them more rigorously and to compare them across fields. The two case studies thus illustrated how generic timing concepts distilled from the vast domain-specific literature could play out in similar ways across various application domains.

TABLE IV: Comparison of HO and HCA case studies

	HO	**HCA**
info abstraction (bottom-up)	averaging (of X conc.)	binary threshold-crossing (number of live cells)
control commands (top-down)	macro state (average X conc.)	macro state (rule-activating goal)
macro-properties analysed	oscillator synchronisation type and period	oscillation period and macro-pattern
simulation time	continuous	discrete cycles
inter-level delays	abstraction & control delays (transmit X conc.)	control delay (transmit goal)
execution frequencies	NA	configurable at each level
configuration parameters	multi-scale structure, abstraction delays, coupling strength	rule-pairs, initial states, abstraction thresholds, execution frequencies

While still theoretical, the two examples illustrated different instances of multi-scale oscillation systems, with many potential applications. They differ in their functions, multi-scale design choices, and analysed variations (see Table IV), but despite these differences, the examples featured highly similar generic results with respect to the timing impacts on macro-properties.

VII. Conclusions

This paper identified common timing-related concerns from domain-specific literature (i.e., execution delays and frequencies) and illustrated their impact on system macro-properties via two concrete examples: hierarchical biochemical oscillators (HO) and cellular automata (HCA). We selected these case studies as both generic and applicable to various domains. Experimental results from both examples show how timing confers a configuration parameter just as powerful as any other variable. I.e., changing execution delays or intervals, in various cross-scale combinations, generates different outcomes, e.g., synchronisation types and oscillation periods in HO; macro-patterns and oscillation periods in HCA. Also, several time configuration regions produce equivalent macro-behaviours, possibly improving system robustness to time disturbances. This contribution sets a basis for developing a comprehensive theory of timing in multi-scale feedback systems, helping practitioners to transfer and apply key insights across domains.

References

[1] *ACSOS repository*. 2021. URL: https://gitlab.telecom-paris.fr/ada.diaconescu/msaf/-/tree/master/acsos21.

[2] J. Albus et al. *4DRCS: a reference model architecture for unmanned vehicle systems version 2.0*. Tech. rep. National Institute of Standards and Technology, 2002.

[3] T.F.H. Allen and T.B. Starr. *Hierarchy: perspectives for ecological complexity*. Univ. of Chicago Press, 2017.

[4] F. Allerding, B. Becker, and H. Schmeck. *Decentralised Energy Management for Smart Homes*. Ed. by C. Müller-Schloer et al. Springer, 2011.

[5] D.J. Allwright and W.M. Wonham. "Time Scales Hierarchical in Stably Control Nested Systems". In: *IFAC Proc. Vol.* 13.6 (1980), pp. 85–91. ISSN: 1474-6670.

[6] K.L. Bellman and L.J. Goldberg. “Common origin of linguistic and movement abilities”. In: *American Journal of Physiology* 246.6 (1984), R915–R921.

[7] R.L. Brinkman and J.E. Brinkman. “Cultural lag: a relevant framework for social justice”. In: *International Journal of Social Economics* (2005).

[8] M.Y. Choi et al. “Synchronization in a system of globally coupled oscillators with time delay”. In: *Physical Review E* 61.1 (2000), p. 371.

[9] T.C. Coffey and I.J. Williams. “Stability analysis of multiloop, multirate sampled systems.” In: *AIAA Journal* 4.12 (1966), pp. 2178–2190.

[10] A. Diaconescu, L.J. Di Felice, and P. Mellodge. “Exogenous coordination in multi-scale systems: How information flows and timing affect system properties”. In: *Future Gen. Compu. Sys.* 114 (2021), pp. 403–426.

[11] A. Diaconescu, L.J. Di Felice, and P. Mellodge. “Multi-scale feedbacks for large-scale coordination in self-systems”. In: *Intl. Cnf. Self-Adaptive and Self-Organizing Systems (SASO)*. IEEE. 2019, pp. 137–142.

[12] A. Diaconescu, L.J. Di Felice, and P. Mellodge. “An Information-oriented View of Multi-Scale (Feedback) Systems”. In: *SISSY workshop (with ACSOS’21)*. 2021.

[13] A. Diaconescu, S. Tomforde, and C. Müller-Schloer. “Holonic cellular automata: modelling multi-level self-organisation of structure and behaviour”. In: *Intl. Cnf. Artificial Life*. MIT Press. 2018, pp. 186–193.

[14] A. Diaconescu et al. *Generic architectures for collective self-aware computing systems*. Ed. by S. Kounev et al. Springer, 2017, pp. 191–235.

[15] R. Dilão. “Antiphase and in-phase synchronization of nonlinear oscillators: The Huygens’s clocks system”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 19.2 (2009), p. 023118.

[16] M.G. Earl and S.H. Strogatz. “Synchronization in oscillator networks with delayed coupling: A stability criterion”. In: *Physical Review E* 67.3 (2003), p. 036204.

[17] B. Ermentrout. “An adaptive model for synchrony in the firefly Pteroptyx malaccae”. In: *Journal of Mathematical Biology* 29.6 (1991), pp. 571–585.

[18] A.L. Fairhall et al. “Efficiency and ambiguity in an adaptive neural code”. In: *Nature* 412.6849 (2001), pp. 787–792.

[19] W. Findeisen. *Hierarchical Control Systems: An Introduction*. Tech. rep. IIASA, Laxenburg, Austria: Intl. Insti. for Applied Systems Analysis, Apr. 1978.

[20] J. Flack et al. “Timescales, symmetry, and uncertainty reduction in the origins of hierarchy in biological systems”. In: *Evolution cooperation and complexity* (2013), pp. 45–74.

[21] J.C. Flack. “Coarse-graining as a downward causation mechanism”. In: *Philosophical Transactions of the Royal Society A* 375.2109 (2017), p. 20160338.

[22] R.G. Franks and C.W. Worley. “Quantitative Analysis of Cascade Control”. In: *Ind. Eng. Chem. Res.* 48.6 (1956), pp. 1074–1079.

[23] J.L. Hellerstein et al. *Feedback Control of Computer Systems*. J. Whiley, 2004.

[24] A.G. Hoekstra et al. “Complex Automata: Multi-scale Modeling with Coupled Cellular Automata”. In: *Simulating Complex Sys. by CA*. Springer, 2010, pp. 29–57.

[25] E. Jablonka. “Information: Its interpretation, its inheritance, and its sharing”. In: *Philosophy of science* 69.4 (2002), pp. 578–605.

[26] S.O. Jeong, T.W. Ko, and H.T. Moon. “Time-delayed spatial patterns in a two-dimensional array of coupled oscillators”. In: *Physical review letters* 89.15 (2002), p. 154104.

[27] T.H. Kang et al. “Circadian oscillation of nucleotide excision repair in mammalian brain”. In: *Proc. of National Academy of Sciences* 106.8 (2009), pp. 2864–2867.

[28] J.O. Kephart and D.M. Chess. “The vision of autonomic computing”. In: *Computer* 36.1 (2003), pp. 41–50.

[29] J.R. Kim et al. “A design principle underlying the synchronization of oscillations in cellular systems”. In: *Journal of Cell Science* 123.4 (2010), pp. 537–543.

[30] T.W. Ko, S.O. Jeong, and H.T. Moon. “Wave formation by time delays in randomly coupled oscillators”. In: *Physical Review E* 69.5 (2004), p. 056106.

[31] P.V. Kokotovic. “Applications of Singular Perturbation Techniques to Control Problems”. In: *SIAM Rev.* 26.4 (1984), pp. 501–550.

[32] J. Kramer and J. Magee. “Self-Managed Systems: an Architectural Challenge”. In: *FOSE*. 2007, pp. 259–268.

[33] M. Minsky. *Society of Mind*. Simon&Schuster, 1988.

[34] D. Monsivais-Velazquez et al. “Dynamics of hierarchical weighted networks of van der Pol oscillators”. In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30.12 (2020), p. 123146.

[35] N.S. Nise. *Control Systems Engineering*. 8th. USA: John Wiley & Sons, Inc., 2019.

[36] A. Pluchino et al. “Opinion dynamics and synchronization in a network of scientific collaborations”. In: *Physica A: Statistical Mechanics and its Applications* 372.2 (2006), pp. 316–325.

[37] J.P. Ramírez et al. “Effects of time delay in the synchronized motion of oscillators with Huygens’ coupling”. In: *IFAC Proceedings Volumes* 45.12 (2012), pp. 159–164.

[38] H. Schmeck et al. *Adaptivity and Self-organisation in Organic Computing Systems*. Ed. by C Müller-Schloer et al. Springer, 2011, pp. 1–32.

[39] H.A. Simon. “The architecture of complexity”. In: *Facets of systems science*. Springer, 1991, pp. 457–476.

[40] J-P. Steghöfer et al. “HiSPADA: Self-Organising Hierarchies for Large-Scale Multi-Agent Systems”. In: *Autonomic and Autonomous Systems*. 2013, pp. 259–268.

[41] J.W. Valentine and C.L. May. “Hierarchies in biology and paleontology”. In: *Paleobiology* (1996), pp. 23–33.

[42] D. Weyns et al. *Patterns for Decentralised Control in SAS*. Ed. by R.deLemos et al. LNCS, 2013, pp. 76–107.

[43] J. Wu. “Hierarchy theory”. In: *Linking ecology and ethics for a changing world* (2013), pp. 281–301.