Using the Phase Space to Design Complexity

Design Methodology for Distributed Control of Architectural Robotic Elements

Mary Katherine Heinrich¹, Phil Ayres²
¹,²Centre for Information Technology and Architecture, IBT, KADK
¹,²{mhei|Phil.Ayres}@kadk.dk

Architecture that is responsive, adaptive, or interactive can contain active architectural elements or robotic sensor-actuator systems. The consideration of architectural robotic elements that utilize distributed control and distributed communication allows for self-organization, emergence, and evolution on site in real-time. The potential complexity of behaviors in such architectural robotic systems requires design methodology able to encompass a range of possible outcomes, rather than a single solution. We present an approach of adopting an aspect of complexity science and applying it to the realm of computational design in architecture, specifically by considering the phase space and related concepts. We consider the scale and predictability of certain design characteristics, and originate the concept of a formation space extension to the phase space, for design to deal directly with materializations left by robot swarms or elements, rather than robots' internal states. We detail a case study examination of design methodology using the formation space concept for assessment and decision-making in the design of active architectural artifacts.

Keywords: phase space, complexity, attractor, distributed control

INTRODUCTION & BACKGROUND

The methodology detailed in this paper is developed as part of the EU research project flora robotica (Hamann et al. 2015, [1]), which uses tightly coupled plant-robot bio-hybrids to form architectural artifacts that are self-organizing and evolving, with distributed control mechanisms. In the flora robotica project, the robotic controllers incorporate approaches from evolutionary robotics (Bongard 2013), distributed embodied evolution (Watson et al. 2002), and interactive evolutionary computation (Secretan et al. 2008). This paper deals specifically with the problem of the architectural design of such a robotic system, and the question of how high-level architectural design requirements can be merged with low-level rules of robotic control.

Active architectural elements

In this context, the architectural design problem at hand is that of designing controllers for active/robotic architectural elements that have strong self-organizing behaviors.
Existing buildings containing active products (such as automated window shades or automated HVAC) are often controlled centrally, through readily available technology known as building automation systems or smart building management systems ([2]:Siemens 2012). These centralized systems deal with changes in environmental conditions by existing design methodologies of averaging or homogenizing (cf. Pavliotis and Stuart 2008), at the scale of perhaps a wall or a room. In addition to these industry products, there are existing buildings featuring bespoke active facade elements. There are many built examples incorporating matrices of LED lighting, such as the centrally controlled BIX facade of Kunsthaus Graz (Cook and Fournier 2004). There are also examples of built facades incorporating kinetic elements, including Media-TIC (Cloud 9 [3]), IBA Soft House (Splitterwerk [5]), Yeosu Theme Pavilion (SOMA [6]), Al Bahr Towers (AHR [7]), and MegaFaces (Asif Khan [8]). Some of these facades feature decentralized control but, to the knowledge of the authors, do not incorporate decentralized communication between elements. There are some departures from the centralized communication paradigm at the scale of installation, e.g. Sentient Chamber (PBAI [9]) (Chan et al. 2015). There are also relevant architectural demonstrators, prototypes, or probes (Thomsen and Tamke 2009) in literature (Frazer 1995, Beesley (ed) 2006, Thomsen et al. 2015). We therefore understand the question of designing distributed/decentralized control and communication in architectural elements to be relevant to CAAD (computer-aided architectural design) state of the art, in addition to its relevance to flora robotica.

**Evolving robotic controllers**

In prior flora robotica work (Wahby et al. 2016), we have used NEAT (Stanley and Miikkulainen 2004) to evolve Artificial Neural Network controllers in simulation for an evolutionary robotics (Bongard 2013) approach to the control of plant growth and motion in a bio-hybrid setup, and successfully crossed the reality gap to replicate our simulation results in physical experiments. Having demonstrated that, in the tested conditions, results of plant-robot bio-hybrid controllers evolved in simulation are able to cross the reality gap, we seek to extend the work to evolving distributed controllers. In this paper, we focus on the architectural design problems posed by the context of distributed controllers evolved in simulation.

**Common Encoding & Integrated Projection**

In the flora robotica project, in order to avoid a single model’s potential bias toward either the plant symbiont or the robot symbiont, we will take a pluralistic modelling (Helbing 2010) approach to the definition of the distributed bio-hybrid controller (see partial florarobotica models/analyses in Wahby et al. 2016, Zahadat et al. 2015, Soorati and Hamann 2015, Heinrich and Ayres 2016). As a general framework for approaching the problem of merging high-level architectural requirements with these pluralistic low-level rules of control, we propose a common encoding and integrated projection (see Figure 1) which will incorporate bidirectional communication between each of the low-level input models and a single common encoding of those models, will use the current state of the common encoding to generate an integrated projection of a late timestep, will assess the integrated projection according to high-level architectural objectives, and will use that assessment to pun-
ish or reward the evolution of the low-level input models.

**COMPLEXITY & PREDICTABILITY FOR ARCHITECTURAL DESIGN**

In the description and study of dynamic systems, the 'phase space' is the representation of all possible instantaneous states that can occur in a physical system (Butkovskiy 1990, Sayama 2015). Each of these states, and its associated characteristics, corresponds to a single point in the mathematical phase space.

In existing design rhetoric, we sometimes refer to the parameter space and the solution space of a model. These spaces can be approximated and understood analytically, because any particular combination of parameters always results in the same unique solution. The phase space is distinct from these in that it describes the full set of possible states that may eventually arise from a set of behaviors, and therefore, in certain types of systems, cannot be derived purely analytically (see Bar-Yam 1997). In the context of architectural design, this means that design solutions must be modeled and simulated before visual interrogation and analysis of the design can occur.

These types of systems, which can be described through modelling and the analysis of model results, are categorically complex systems (or complex adaptive systems). The scientific study of complex systems deals in part with the principles governing the emergence of such systems from simple components (Bar-Yam 1997). This paper details an approach of applying a particular aspect of complexity science to the realm of computational design (or design computation), specifically by employing the mathematical phase space as an approach for architectural design, modelling, and simulation. Due to the context of architectural design, we assume that only spatially distributed models of systems are relevant to this design methodology, which can be an important distinction in modelling (e.g. work on the breakdown of mean-field approximation and work on spatially distributed models, Werfel and Bar-Yam 2004, Bar-Yam and Sayama 2006, Dieckmann and Law 2000).

In this paper, we consider the complex systems and behaviors of architecture that is responsive, adaptive, or interactive— in other words, architecture that contains sensor-actuator systems. We consider sensor-actuator systems that are controlled through decentralized control and communication, thereby allowing self-organizing behavior. We also adopt some of the vocabulary of complexity science, so that 'complexity' refers to the number of possibilities in a system and the length of description thereby required to encompass it (Bar-Yam 1997).

It is sometimes assumed in architectural discourse that incorporating methods of emergence, distributed control, and evolutionary robotics will largely equate to unpredictability in the architectural result (Frazer 1995, Kolarevic and Malkawi (eds) 2005, Kretzer and Hovestadt (eds) 2014). However, we know from complexity science that this varies depending on which scale of the system you are describing, which characteristics you are concerned with, and at which point in time you are viewing the system. For example, in the classic agent-based Boids model (Reynolds 1987) of flocking birds, though the positions of agents in relation to one another may be very unpredictable at early timesteps, they will be much more predictable at later timesteps after they have formed flocks. Similarly, if the system's behavior is described at the scale of individual agents within...
a flock, it is more unpredictable because each run of the system will have birds with differing start positions and orientations. However, if the model is described at the scale of the flock, then it is much more predictable, because the system will tend toward similar flocking behaviors, regardless of start position.

The amount of predictability varies greatly by system. Some complex systems have attractors, which will cause the system to move toward a particular configuration or set of configurations, depending on the number of attractors and the expanse of the basins associated with those attractors. For instance, if a particular system has two attractors, and the basins of attraction from these two cover the phase space equally and entirely, then the system will tend to one of these two states with equivalent probability, despite an expansive array of possible start conditions (see Hurley 1982, Sayama 2015). In other words, the phase space of possible configurations is predictable, if not the exact configuration, and the expansiveness of that predictable phase space depends on the complexity of the system. The amount of predictability of the exact configuration that a system with attractors will move toward depends on the number of attractors and basins of attraction. Therefore, in looking to design rules for distributed control and communication in architectural sensor-actuator systems, we do not design a solution by finding and evaluating individual instances. We instead work toward designing a system’s (i.e. set of behaviors’) full set of possibilities (the phase space) and its attractors.

The description used to design an architectural complex system should occur at the scale that is pertinent to the particular architectural design problem at hand (this does not change the scale of rule-based interactions; scale of rule and outcome are still distinct). The system should be designed so that coherent behaviors or correlated behaviors (see complexity profile and related concepts, Bar-Yam 1997) occur at that pertinent scale (see Figure 2). As we saw in the example of the classic Boids model (Reynolds 1987), there could be a prohibitive amount of randomness at a certain timestep and scale, making it seem too unpredictable for architectural design. At such a state, a comprehensive description of the system would have to include a description of each individual agent in order to describe the behavior of the system. But in that same system, there could be coherent behavior at a different timestep and scale. At this state, a comprehensive description could be comprised of very little information, because it would need to describe only one behavior: that of the coherent group (see random, coherent and correlated behaviors, Bar-Yam 1997).

The balance of complexity and simplicity in a designed system may likely depend on which scale is being described, and at which point in time. Therefore, the task of designing architecture that is emergent, distributed, or evolutionary is not a task of designing behaviors with primarily unpredictable results. It is rather the task of designing a relevant phase space; it is the task of designing the scale at which coherent or correlated behavior occurs, without resorting to high-level control.

**DESIGN METHODOLOGY CASE STUDY: CONTROLLERS FOR SOLAR RESPONSIVE SELF-ORGANIZING GROWTH**

The case study selected is the task of designing an exterior plant-robot bio-hybrid system (similar to the bio-hybrid setup in our prior flora robotica work, Wahby et al. 2016) that responds to solar exposure. The design task of the response is to shade occupiable spaces from the sun by growing taller where needed, but to only grow where necessary, so as to leave as much occupiable space on the site as possible. However, the system should maintain some presence, even if the site receives no sunlight, so that the open area is divided into smaller occupiable spaces, and so that the system has a chance to react if the lighting conditions change at a later time. The controller should assume that the bio-hybrid system generally grows upward, to both align with plant dynamics and to be self-structuring. The bio-hybrid system should be able to be placed on any site and...
correctly interpret whether each element is in sun or shade.

<table>
<thead>
<tr>
<th>predetermined</th>
<th>centralized</th>
<th>decentralized</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="sunny.png" alt="Sunny" /></td>
<td><img src="sunny.png" alt="Sunny" /></td>
<td><img src="sunny.png" alt="Sunny" /></td>
</tr>
<tr>
<td><strong>succeeds</strong></td>
<td><strong>succeeds</strong></td>
<td><strong>succeeds</strong></td>
</tr>
<tr>
<td><img src="shady.png" alt="Shady" /></td>
<td><img src="shady.png" alt="Shady" /></td>
<td><img src="shady.png" alt="Shady" /></td>
</tr>
<tr>
<td><strong>fails</strong></td>
<td><strong>fails</strong></td>
<td><strong>succeeds</strong></td>
</tr>
</tbody>
</table>

A predetermined controller in this case (which would bring bio-hybrid growth to a certain height based on solar geometry derived from latitude and longitude) is not sufficient, because the controller would be unable to react to shade cast on the site by clouds, trees, or neighboring buildings (see Figure 3). A centralized sensor-controller would also not be sufficient in this case because, though the sensor would be able to correctly interpret its local neighborhood’s solar condition, it would assume that the solar condition is the same across the site, and would fail in conditions where neighboring buildings (for instance) cast shadow on half of the site (see Figure 3). Decentralized sensor-controllers, by contrast, would not fail in any of the above mentioned scenarios (see Figure 3) and has the spatial consequence and design opportunity of increased heterogeneity.

If the system has not only distributed control, but distributed communication, it has the added advantage of being able to create spatial clusters as it grows, to provide as much occupiable space on the site as possible. Therefore, with the design problem presented in the case study, it is appropriate to design a system that features distributed control and distributed communication.

**Cellular Automata (CA) setup**

We select Cellular Automata [10] as a typology to address this design problem, because it models decentralized communication in a way that is befitting of non-mobile architectural elements, it is discrete in both time and space, and the possible states of the system and the possible transitions it can undergo are finite (Wolfram 1983, Wolfram 1984, Sayama 2015). We use a Moore neighborhood setup [11], in which, as each cell determines its state, it looks at the cells directly below, above, and beside it, and it also looks at the 4 cells diagonal to it. We use a standard majority rule (Wolfram 1983), in which each cell decides its state by conforming to whichever state holds the majority in its neighborhood.

The majority rule means that, if a cell is dark, 2 of its neighbors are dark, and 6 of its neighbors are light, the cell will become light in the next timestep (see Figure 4). Likewise, if a cell is light, 5 of its neighbors are dark, and 3 of its neighbors are light, it will become dark in the next timestep (see Figure 4). The majority rule was chosen because its dynamics are known to often lead to clusters, which are desirable in the given case study. In this setup, we use dimensions of 100 x 100 and periodic space, and in initial conditions use equal probability of randomized cell state distribution. (All setups are completed in a combination of PyCXsimulator [12], IronPython [13], and Grasshopper3d/Rhino3d [14], with reference to the PyCX Project [12].)

---

**Extending Phase Space to Formation Space**

The phase space of a Cellular Automaton is an enumeration of all possible configurations in that setup, according to the dimensions of the cellular automaton and the number of possible cell states. For instance, in the first part of Figure 5, we can see all the possible configurations in a 2-cell by 2-cell two-dimensional cellular automaton with binary cell states (two possible cell states). The number of possible configurations in this setup is few, so they are easy to imagine, visualize, and analyze. In the second part of Figure 5, we see a visualization of the phase space of this system (visualization method from Sayama...
2015). Each possible configuration of the system is subjected to the majority rule (in a Moore neighborhood in periodic space), and the configuration it goes to in the next timestep is recorded. Then, a graph is constructed showing each configuration transition as an edge connecting the two configurations. Each component (cluster) in the graph reveals a basin of attraction in the system, and the hub of that component (center of the cluster) reveals the attractor (see Sayama 2015). The phase space visualization for this system shows that it has two equally large basins of attraction, going to the attractor configurations of fully dark or fully light. There are several other configurations that have no predecessors and do not change configuration when subjected to the rule. The only way for the system to achieve these states is to begin there, and, once there, it can proceed nowhere else (see Sayama 2015). When using these systems in architectural design, however, we are unlikely to be concerned with the internal states of members of the system, which is what the phase space description is based upon. We are much more likely to be concerned with the material manifestation left behind as the system progresses. For this reason, we originate the concept of the mathematical formation space, as an extension of the phase space intended specifically for the realm of architectural design. The formation space is an isometric mapping onto the existing phase space (hence being considered an extension). It is a mapping of a particular characteristic of the configuration at a larger scale than the individual cell, ideally at a scale at which some coherence or correlation of behavior is applicable to the design problem at hand. In the third portion of Figure 5, we look at the particular characteristic of length of boundaries between opposing states throughout the system's entirety, isometrically mapped onto the phase space to form a formation space. In this mapping, we can see that the two large basins of attraction now lead to the same attractor, boundary length "0", and that the 7 states which used to be completely isolated can now be grouped together into boundary lengths "2" and "4". This method of visualization shown in Figure 5 (for phase space and for formation space) is fine for this very small system. However, the number of possible configurations in a CA system is defined as the number of possible cell states, raised to the power of the number of cells (Wolfram 1983, Sayama 2015). So, while this system shown in Figure 5 contains only 16 (2^4) configurations, a system with only 16 cells would contain 65,536 (2^16) configurations. It is easy to see how this visualization approach quickly becomes unmanageable.

Rather than interrogating every possible configuration in the phase space or formation space, we must find ways to approximate effectively and make reliable design decisions. The following lays out reasoning for the usefulness of the formation space (and related concepts), in architectural design, as motiva-
Figure 6
Progression of two setups. Top setup reaches its steady states very quickly, bottom setup does not yet reach its steady states at the 100th step.

Figure 7
The 100th timestep of 20 different setups, with 5 separate simulation runs shown for each setup. Setups are indicated by number of states (s) and size of neighborhood radius (r). This side-by-side comparison is meant as a method of visual analysis exploring dependencies in the system through changing variables.
tion for further work to verify usefulness and find effective approximations. For this case study, we include 20 setups of the CA, and conduct 5 simulation runs for each setup. The 20 setups cover each possible combination if number of possible cell states (s) ranges from 2 to 5 and radius size of neighborhood (r) ranges from 1 to 5 (these setup choices are strongly informed by PyCX Project [12] and Sayama 2015). Individual timesteps of single runs of two of the setups are shown in Figure 6, and the 100th timestep of every run is shown in Figure 7. In Figure 6, we see that the upper setup settles on its steady states extremely quickly, while the lower setup has not yet reached its steady states even by the 100th timestep. In Figure 7, we see that setups with smaller neighborhood radii reliably reach a steady state with many small clusters, with the initial possible states approximately equally represented. We also see that setups with larger neighborhood radii tend to end with the entire system in a homogeneous state, with only one of the starting cell states represented.

If we look at the resulting steady states from two runs of the same setup, such as those in Figure 8, it is clear that it would be impossible to predict which state an individual cell will end up in, or which overall configuration the system will end up in, if the starting conditions are randomized and unknown. In this way, considering the phase space of the system, its result seems incredibly unpredictable, and perhaps not useful for architectural design. However, if we do not consider the cell states, system configuration, and phase space, and instead consider the formation space and spatially-useful characteristics at the scale most relevant to the design problem at hand, the system suddenly seems drastically more predictable.

**Probe study into evolving controllers**

If, instead of designing for a particular set of these criteria, we desire that the density and scale of these occupiable areas can adjust, just like height, to achieve shading on the site, then we could subject the control rules of the system to evolution. In Figure 9 we see the results of an example probe study (Thomsen and Tamke 2009) into evolving the controller in simulation (in Galapagos [14]). The evolution of the controller is rewarded for maximizing open area, shading that open area, and maximizing length of boundaries. The results of this probe (Figure 9) show different results on each site, suggesting response to differences in shade provided by neighboring buildings. The first site, shaded only from the south, results in high density of boundary condition, tall growths, and open areas in the center, where they are shaded by growths. The second site, more shaded by buildings, results in shorter, fewer growths. The third site, already very shaded, results in low density of boundary conditions, and growths that slope downward and outward, thereby providing little additional shade to open areas. These preliminary outcomes show promise in fulfilling the high-level design requirements of maintaining open occupiable space and system growths as spatial dividers, and of shading occupiable space that is not already shaded by neighboring buildings, suggesting that the probe into evolving controllers merits further investigation.
CONCLUSION
In conclusion, we have presented a design methodology for applying phase space considerations to the realm of architecture, as an alternative to methods of averaging and homogenization. We have presented the framework of the *florarobotsica* common encoding and integrated projection. We have originated the formation space, as a concept to deal with characteristics of active architectural artifacts at a scale that is pertinent to a specific design requirement. We have used the formation space as a framework for design analysis of case study simulation results in the context of a specific design task, and used that analysis to construct a probe study into evolving controllers in simulation. The results of the probe point toward the possibility of future work using the formation space to evolve low-level control rules for high-level design tasks, with specificity to individual site. In the context of the prior *florarobotsica* work on evolutionary robotics and crossing the reality gap with bio-hybrids, this shows the potential usefulness of the formation space in looking at high-level architectural design requirements and merging them with distributed control and distributed communication. Future work includes pursuing substantiation of this specific potential usefulness, as well as general formalization, development, and verification of the formation space concept.

ACKNOWLEDGEMENTS
*florarobotsica* has received funding from the European Union’s Horizon 2020 research and innovation programme under the FET grant agreement no. 640959.

The authors would like to thank collaborators Heiko Hamann and Payam Zahadat for guidance in swarm robotics and artificial life, and Mohammad Divband Soorati for his idea of common encoding. The authors would like to thank Yaneer Bar-Yam, Hiroki Sayama, and the NECSI Winter School for instruction in complex systems and modelling. The paper relies heavily on the example of Sayama's PyCX Project.

REFERENCES
Bar-Yam, Y 1997, *Dynamics of complex systems*, Addison-Wesley, Reading, MA
Bongard, JC 2013, 'Evolutionary robotics', *Communications of the ACM*, 56(8), pp. 74-83
Butkovskiy, AG 1990, *Phase Portraits of Control Dynamical Systems*, Springer Netherlands
Chan, MTK, Gorbet, R, Beesley, P and Kulič, D 2015 'Curiosity-Based Learning Algorithm for Distributed


Sayama, H 2015, *Introduction to the modeling and analysis of complex systems*, Open SUNY Textbooks, Milne Library


Soorati, MD and Hamann, H 2015 'The Effect of Fitness Function Design on Performance in Evolutionary Robotics: The Influence of a Priori Knowledge', *Proc. of the 24th Int. Conf. on Genetic and Evolutionary Computation (GECCO '15)*, pp. 153-160


Thomson, MR and Tamke, M 2009 'Narratives of Making: thinking practice led research in architecture', *proceedings of the conference "COMMUNICATING (BY) DESIGN"


Wahby, M, Hofstadler, DN, Heinrich, MK, Zahadat, P and Hamann, H 2016 'An Evolutionary Robotics Approach to the Control of Plant Growth and Motion: Modeling Plants and Crossing the Reality Gap', *2016 10th IEEE International Conference on Self-Adaptive and Self-Organizing Systems*


