Field Based Behavior Regulation for Self-Organization in Cellular Mechanical Systems

Yan Jin and Chang Chen University of Southern California, USA

A Cellular Self-Organizing (CSO) approach is proposed for developing adaptive mechanical systems. The design of CSO systems however is difficult because the global effect emerges from local actions and interactions that can be hard to specify and control. In order to achieve high level resilience and robustness of CSO system and retain the capability of specifying desired global effects, we propose a field based regulative control mechanism, called Field based Behavior Regulation or FBR. FBR is a real-time, dynamical, distributed mechanism that regulates the emergence process for CSO systems to self-organize and self-reconfigure in complex operation environments. FBR characterizes the task environment in terms of "fields" and extend the system flexibility and robustness without imposing global control over local cells or agents. This paper describes the models of CSO and FBR and demonstrates their effectiveness by presenting simulation based case studies.

Introduction

As human society progresses and becomes more sophisticated, our demand for new ideals, new capabilities, and new environments intensifies, resulting in ever increasing complexity of man-made systems, including physical systems, technologies, organizations, and social, political and economic systems. Complexity has been recognized as an important feature and mechanism of bio-systems, societal systems and technology development processes. However, for engineered systems, the notion of complexity often points to unintended, undesirable and must-avoid system properties.

One may note that the increasing complexity of engineered systems comes from increasingly complex and highly sophisticated functional requirements. For example, increasing demands for performance, safety, ease of operation, comfort of ride, and least environmental impact have led to today's complex automobiles. Over the process of product evolution, the capabilities or functionalities of the automobile were added incrementally, with special cares being made to make sure that unintended actions of, and interactions between, the components be eliminated or at least minimized. A major issue with developing complex engineered systems is that the sheer number of and sophisticated interdependencies among the system components imply *uncertainty* and *unknowns* to the engineers, making it difficult for them to ensure the valid operation range for the system to survive its expected lifecycle.

On the other hand, it is intriguing to consider that the nature "faces" all the uncertainties and unknowns and yet the natural systems are "designed" to live with these uncertainties and unknowns as an inherent part of their capabilities. We observe that human design and natural "design" are very distinct from each other: human design is more purpose or function driven and takes a *top-down* approach to avoid possible complexity problems, while the nature "design" is arguably less purposeful and follows a bottomup approach by making complexity as a "solution" to deal with the arising uncertainties and unknowns (Ashby, 1958). The research on system biology (Kitano 2002), self-configurable systems (Subramanian and Katz 2000) and component-based design (Kopetz 1998) has explored the formation of adaptive systems from both natural and man-made perspectives. In our research, we introduce a Cellular and Self-Organizing (CSO) approach to building adaptive systems. In this approach, a mechanical system is composed of multiple mechanical cells, which can be either identical (for homogeneous systems) or distinct (for heterogeneous systems). Further, the formation of such systems is based on a set of bottom-up, dynamical and self-organized mechanisms. It is fully understood that the CSO approach will not be able to compete with the traditional methods in a short term and for many applications. However, the paradigm-shift from componentbased to cell-based and from top-down to bottom-up promises an alternative future for developing complex engineered systems.

Both *multi-agent* systems and *self-organizing* systems have been highlighted in many engineering fields, such as computer science, industrial engineering, and material science. Much research has been done to investigate the properties and benefits of such systems, and the ways to build them. One critical design research question is: *How can a designer connect the design of mechanical cells/agents, their interactions with functional requirements (or tasks)?* In our research, we propose a *field based behavior regulation* (FBR) approach as a basis for cells to interact with each other and with their task environment. In this approach, a mechanical cell Field Based Behavior Regulation for Self-Organization

is a *sensor-operator* unit that can perform a range of actions or behaviors. At a given time, a cell's behavior can be self-regulated based on the cell's *"field position"* at that moment. The field position of a cell is determined by the task requirements and the environmental situation that are sensible by the cell. When a CSO system is composed of multiple mechanical cells, at any given time, different cells may perform different or similar behaviors. This *cellular differentiation* is achieved locally through field based regulation, unlike conventional modular (Gu et al, 1997, Gershenson et al, 1999) or component-based (Kopetz, 1998) approaches in which differentiation of components is determined at design time and does not change during system operation.

In the rest of this paper, we first review the related work in Section 2 and then introduce our CSO (cellular self-organizing) framework in Section 3. In Section 4, we present field based behavior regulation (FBR) approach, and in Section 5 we demonstrate the effectiveness of our approach through simulation based case studies. Section 6 draws conclusions and points to future research directions.

Related Work

Much research has been done to investigate multi-agent and selforganizing systems and to develop methods for designing such systems. Self-organization and emergent behavior as two major features of such systems have been popular research topics in the research field of complex systems (von Neumann, 1966; Fukuda & Kawauchi, 1990; Weisbuch, 1991; Bojinov et al, 2000; Butler et all, 2001; Wolfram, 2002; Zouein, 2009). Self-organization is the large scale organization through the limited local interactions of the constituent components. Emergence represents the concept of the patterns, often unpredictable ones, which are exhibited in the large scale organization. Holland (1992) and Gell-Mann (1994) extended the research to non-homogeneous system and pointed out the nonlinearity between local and global which becomes the biggest challenge of such systems. To further address the problem, the Game of Life (Garden, 1970) and more Cellular Automata based fractals have been explored (Wolfram, 2002).

In the field of engineering design, design for adaptability and design of reconfigurable systems have been investigated in the past decade. In their work focusing on vehicle design, Ferguson and Lewis (2006) introduced a method of designing effective reconfigurable systems that focuses on determining how the design variables of a system change, as well as investi-

gating the stability of a reconfigurable system through the application of a state-feedback controller. This method is based on multi-objective optimization and allows systems to adjust their design variable by dynamically optimizing in response to changing conditions. The adaptability of such systems is limited by the range of change of the variables and by the preconceivable changing situations. Martin and Ishii (2002) proposed a design for variety (DFV) approach that allows quick reconfiguration of products but mainly aims to reduce time to market by addressing generational product variation. Indices have been developed for generational variance to help designers reduce the development time of future evolutionary products (Martin and Ishii, 2002). In addition to developing design methods for reconfigurable systems, various reconfigurable robotics have been developed mostly by computer scientists. Fukuda and Nakagawa (1988) developed a dynamically reconfigurable robotic system known as DRRS. Unsal et al (2001) focused on creating very simplistic i-Cube systems (with cubes being able to attached to each other) in order to investigate whether they can fully realize the full potential of this class of systems. PolyBot has gone through several updates over the years (Yim et al., 2000, 2002) but acquired notoriety by being the first robot that "demonstrated sequentially two topologically distinct locomotion modes by self-configuration. SuperBot (Shen et al., 2006) is composed of a series of homogeneous modules each of which has three joints and three points of connection. Control of SuperBot is naturally inspired and achieved through what the authors describe as the "hormone" control algorithm (Shen et al., 2002, 2004: Salemi et al., 2001).

Bio-mimetic design methods allow designers to identify appropriate natural systems or mechanisms from which to draw design inspirations. The idea of using DNA and genes to capture genotype of systems is not new. Inspired by the nature's evolution process, genetic algorithm (GA) (Goldberg, 1989) and genetic programming (GP) (Koza, 1992) have been established to model problems using bit string (GA) or functional tree (GP) genes and to solve problems by evolving the best solution(s) from a population through reproduction, mutation, recombination, natural selection and survival of fitness. This approach has been taken to solve various engineering problems including design optimization, configuration design, and design automation (Maher, 2001; Koza, 1992; Fogel et al, 1996; Parmee, 1997; Bentley, 1999; Bonnie and Malaga, 2000; Lee et al 2001; Koza et al, 1999; Vajna and Clement, 2002; Fan et al, 2003). In addition to direct encoding where genotype codes map to the phenotypes directly, recently researchers have explored indirect coding method, called computational embryogeny (Kumar and Bentley, 2000), to evolve rules that build or develop corresponding phenotypes (Yogev and Antonsson, 2007). Although these computational methods have been successfully applied to solve optimization problems with specific fitness functions, effectively integrating the methods into our proposed CSO systems design and development framework remains a key challenge. Aiming to develop machines that can replicate and repair themselves, Lipson (2007) and his colleagues (Zyokov et al, 2005) investigated and demonstrated autonomous self-replication in the context of homogeneously composed systems comprised of cube modules, and the systems that have the capability if damaged to construct a detached functional copy of its non-functioning self through a technique called continuous self-modeling (Bongard et al. 2006).

Our previous work on CSO generated a design DNA concept and associated system formation mechanisms (Zouein, 2008, Jin et al, 2010). This research extends the previous research by first expanding the concept of design DNA from a static specification to a dynamic and probabilistic representation and then introducing a new *field based control mechanism* to utilize the potentials of such systems for increased robustness and resilience. In addition, while most current approaches for multiagent systems design requires agents to have a global unique identifiers for cooperation and some methods such as DHM (digital hormone model) (Shen et al 2004) require explicit local interactions, our field based behavior regulation approach allows agents to respond to the field of the task environment *spontaneously* and interact with other cells or agents only implicitly, rather than deliberately, as a result of their actions in the task field.

CSO: A Naturalistic Approach to Design

The goals of our research on CSO systems are two-fold. First, we aim to develop systems that are *flexible* in responding to various known or unknown tasks, *robust* in achieving given tasks under changing environment situations, and *resilient* in dealing with partial system failures. Secondly, we are interested in understanding how nature does "design" and developing a similar *bottom-up* and *self-organizing* based design method for future complex engineered systems, i.e., we intent to *design the design* that our CSO systems can carry out by themselves through self-organizing.

Before introducing the fundamental concepts of our proposed CSO systems, we first compare engineered systems with natural systems from a *design* perspective. As shown in Figure 1, in this comparison, we divide the natural systems into two categories, dynamical systems (e.g., planetary system) and biological systems (e.g., animals and plants), and the "design perspective" is captured by dividing analysis into four levels:

Y. Jin and C. Chen

- *Physical substrate*: the physical units that constitute the system;
- Mechanism: the ways by which the system attain its behavior;
- *Capability*: the manifestation of external effects of the system, desired or not; and
- Adaptation: the ways by which the system changes itself.

gical ms
tic Natural ion
val: and luce
ed self- ation
ls

Figure 1: Comparison of Engineered Systems and Natural Systems

As shown in Figure 1, conventional engineered systems is designed and built based on physical components that can be structured in various ways. The mechanism is realized by the design based organization of the behaviors of the components. The desired functions are achieved through the working mechanisms of the organized component behaviors. These systems cannot change themselves in any explicit or implicit way in response to the changes of operation environment, meaning that unconventional ways of designing engineered systems is needed to achieve system adaptability.

Natural dynamical systems are formed based on objects such as planets or particles. Their mechanisms are completely self-organized based on the relationships, such as gravity, between the objects. While natural dynamical systems do not perform their "functions" per se, they do exhibit their "capabilities" by reaching their attractors and maintaining stability around these attractors through spontaneous processes of the individual objects. Furthermore, the chaotic attractors of dynamical systems can be considered as the mechanism that can increase the variety of the stable states of

6

the system, hence the richness of strange attractors can be considered as a feature of adaptability. On the other hand, one may also consider the landscape of all attractors of a dynamical system is determined when the system is formed.

Common to all biological systems, *cellular* structure is indispensable for these systems to grow into complex configuration. Unlike dynamical systems where no memory or shared information is present from an object's point of view, each cell in a biological system possesses a "description," called DNA, of the whole system and is able to *interpret* this locally shared information to generate local actions, i.e., producing adequate proteins. The self-organizing behavior is still spontaneous but guided by DNA. The separation of description of the system from the system itself makes it possible to "copy" and "vary" the description independently from changing the system. Therefore, mutation and natural selection together create an evolution framework for open-ended adaptation. Even before the structure of the DNA molecule was discovered by Watson and Crick (1953), von Neumann (1966) proposed a self-replicating scheme indicating the complexity threshold, after which the system can increase its complexity in an open-ended fashion. Separation of the system description from the system and combination of "copying" and "interpreting" are the two key features of such a scheme.

Learning from what nature "does" has led us to treat *self-organizing* as the key concept that needs to be implemented in future adaptive engineered system. Self-organizing has profound implications in dealing with complexity. First, it is spontaneous hence does not require pre-specifying "who should do what in what ways," allowing high level uncertainty under unpredictable situations. Secondly, if arranged properly, increasing system complexity can be a solution to dealing with high level environment complexity. The challenge, however, is how we can devise and guide selforganizing so that desired system level emergent behaviors and functions can be achieved.

In our proposed Cellular Self-Organizing (CSO) systems framework, shown in Figure 2, three concepts are fundamental, namely, *mechanical cells, fields*, and *design-DNA*. Mechanical cells constitute the physical substrate for system formation and they are the entities that self-organize themselves for emergent system behaviors and functions. The concept of *field* is needed to bring tasks and environmental constraints into the mechanical cells' self-organizing framework. Like dynamical systems, where gravity fields are basic means for planets to self-organize, we need some fields in which our mechanical cells can self-organize. Unlike dynamical systems, our fields must be able to capture of tasks and make completing tasks part of the "attractor landscape." Since our *fields* are artifacts to be

designed, the self-organizing behavior of our mechanical cells must be guided. For this, we introduce design-DNA that contains both system information and the information needed for finding and evolving into "attractors". Again, the explicit description of the system using design-DNA allows open-ended adaptation through genetic evolution. The details of these concepts together with the elaborations are described in the next section.

Levels	Cellular Self-Organizing (CSO) Systems
Adaptation	Distributed and design-DNA based evolution
Capability	Field based "attractors"
Mechanism	dDNA guided and field based self- organizing
Physical Substrated	Mechanical Cells, Fields,

Figure 2: The Cellular Self-Organizing Systems Framework

The Model and Concepts

In this section, we elaborated the discussion of the last section by introducing definitions of the concepts that were introduced. Through the process of describing definitions, we also introduce the field based mechanisms that are needed to realize self-organization. Since in this paper we focus on self-organization aspect of the CSO systems, we will skip the detailed discuss on of the definition of design-DNA, of which more information can be found in (Zouein 2008) and (Jin et al 2009).

A mechanical Cell (*mCell*) is the basic element or unit of a mechanical CSO system:

Definition 1 (Mechanical Cell): mCell = {Cu, S, A, B};

where *Cu*: control unit; $S = \{s_1, s_2, ...\}$: sensors/sensory information; $A = \{a_1, a_2, ...\}$: actuators/actions; *B*: designed behavior, or design information (see definition 4 below).

Field Based Behavior Regulation for Self-Organization

Almost all existing cellular systems, such as Superbot (Shen et al., 2006) and Miche (Gilpin et al., 2008), can be modeled using this definition. *mCell* is the smallest structural and functional unit of a CSO system. Although for a CSO system design, either homogeneous with identical *mCells* or heterogeneous with different *mCells*, the appearance or the structure of its *mCells* may be different, a *mCell* should be able to sense the environment around it and process material, energy and/or information as their actions. We make following assumptions about *mCells*:

- Assumption 1 (Cellular Capability): A mCell has the ability to execute predefined programs, sensing the world around it, process sensory information and the incoming communication, and decide on its action and interaction with others.
- Assumption 2 (Cellular Limitation): mCells have limited sensors, limited range for each sensor, limited communication range with others, and limited number of possible actions.

Definition 2 (State): State = $\{S_C, A_C\}$

where $S_C \subseteq S$ and $A_C \subseteq A$ are currently sensory information and actions, respectively.

State is used to represent the situation which the current *mCell* is in. It is the combination of the current sensor information Sc and current actions Ac. This definition of state parallels the sensor-motor description of cognitive systems (von Foerster 1977).

Definition 3 (Behavior): $b = \{S_E, A_E\} \rightarrow A_N$

where $S_E \subseteq S$ and $A_E \subseteq A$ are existing sensor information and actions, respectively; and $A_N \subseteq A$ are next step actions.

A behavior b is the designed action for given situations or states. The Cu of the *mCell* should be able to judge the situation and make decisions on next actions. The design information (like a complete drawing in a conventional system design) of a CSO system is the fully developed behaviors for each *mCell*.

Definition 4 (Behaviors of System): $BoS = \{B_1, B_2, ..., B_n\}$; where $B_1, B_2, ..., B_n$ are behavior sets of each *mCell* in the system.

The design information for such a system is the set of all the behaviors which local *mCells* should follow; also this *BoS* is supposed to be designed by a designer or designers. If all *mCells* share the same behavior set B, i.e.,

 $B_1 = B_2 = ... = B_n = B$, then we have a homogeneous CSO system. Otherwise, the CSO system is said to be heterogeneous.

From the above four definitions, one may see that the concept of *mCell* is similar to that of biological cell. A biological cell serves its purpose by producing proteins, which parallels a *mCell* producing local actions; the biological cell can only process the signals that the receptor on the membrane can catch, similar to *mCells*. Furthermore, all biological cells hold a full "design information" stored in DNA. Similarly, *mCells* hold the same design information through a design-DNA, or *dDNA*, which is captured by the associated behavior set.

As mentioned above, incorporating task requirements into the selforganizing system as "attractors" is a challenge. We address this challenge with the following definitions of *function requirements* and *fields*.

Definition 5 (Functional Requirement): $FR_i = \{S_i, A_i\}$

where S_i , and A_i form a specific state or situation.

There are two reasons why the *functional requirement* holds similar construct of *state* described above. First, this representation allows us to specify "desired states" of the system. These desired states can be *goal states* or *transient states* that a designer deems to be necessary. Second, using the state construct to represent functional requirements allows *mCells* to recognize whether the function is achieved by examining combinations their sensor information and actions.

It is worth mentioning that our definition of *functional requirement*, or *function*, of using both sensory information S_i and action A_i is more general than the conventional *function* definition that uses only action A_i . When $S_i = \emptyset$ our definition is consistent with the conventional one. Our general representation allows designers to specify circumstances (i.e., sensory information) in addition to actions, leading to more precise functional specification.

At present we explore CSO systems with homogeneous *mCells*. Following the stem cell analogy, we consider that the initial homogeneous *mCells* with multiple behavioral capabilities will, during the process of emergence, differentiate and find their "specialty" behaviors during the period of task execution. We expect that this *self-organized* emergence may create functional blocks consisting of multiple *mCells*, as organs forming in biological systems or attractors in dynamical systems. Once a task is accomplished, or the environment changes, the functional blocks may dissolve by themselves and the *mCells* will continue to renew their differentiation and form new functional blocks.

10

The high level self-organizing and redundancy ensure the flexibility, robustness and resilience of the system, i.e.,

- The enormous size of the potential behaviors resulted from the cellular formation of the system provides functional basis for "unforeseeable" functional requirements and environment changes, increasing the system *flexibility* and *robustness*, respectively, and
- The redundancy of *mCells* together with the large number of *mCells* makes the role of single *mCell* insignificant during the emergence of the system behaviors. Failures with a single *mCell* can be dealt with by other similar *mCells*, leading to high level system *resilience*.

From a design perspective, however, developing CSO systems is a challenging task. As much as we attempt to understand how biological systems develop their *emergence*, we face enormous challenges in developing such fruitful *emergence* in our engineered systems. In our research, we attempt to generate "guided emergence" by providing rules for *mCells* to self-organize and for desired system behaviors to emerge. Two fundamental issues must be addressed. The first relates to design information representation. We have introduced a design DNA or *dDNA* based representation scheme to capture CSO system information at the cellular level (Jin et al, 2008; Zouein et al 2011). The second issue has to do with devising mechanisms to guide self-organization. In the following, we introduce a *field* based approach to allow mCells to self-regulate their behaviors in order to induce system level emergence of reaching "attractors".

Field driven Behavior Regulation (FBR)

In dynamical systems, the concept of field is everywhere, e.g., gravity field, electrical field, magnetic field and electromagnetic field. Objects operating in the fields can "sense" the field and react to it by following physical principles. In the biological world, the function of an organism is realized by a collection of different types of cells working together. The distribution of the chemical signals, called *morphogen*, controls the biological regulation hence the shape and organ formation. Through the developmental process stem cells differentiate into different cell types by responding to specific *morphogen* distribution.

In our CSO systems, *mCells* need the similar differentiation capability in order to self-organize and collectively become a functional system. Instead of producing different *proteins*, differentiated *mCells* produce different *actions*. Instead of being triggered by chemical signals, our *mCells* differentiation must be triggered by the *functional requirements* and *environmental constraints.* To realize such behaviors, we extend the concept of physical field and chemical field into more general "fields" and introduce a *field driven behavior regulation* (FBR), for guiding cellular self-organization and building CSO systems.

For a CSO system, the sensory capabilities of its *mCells* are pre-defined and given. In this case, whenever a task (defined by its FRs) and an operation environment (may or may not be fully known) are given, we can define a *task field* which captures the external world to a *mCell* encompassing both task requirements and environmental conditions. We have:

Definition 6 (Sensory Info & Sensing): sInfo := SNS (FR, Env) where, FR: functional requirements; Env: environment situation; SNS: sensory operator.■

Definition 7 (Task Field & Field Formation): tField := FLD (sInfo)where, $sInfo = (s_1, ..., s_n)$: sensory information; FLD: filed formation operator.

From Definition 7 it can be seen that we define the task field relative to a specific mCell and its sensing capability. Figure 3(a) shows a simple example of *tField*. A mCell *m* is moving to its destination *d* with the potential of encountering an obstacle *obs* in a two dimensional space. In this case, the destination *d* can be considered as an attractor that creates an *attraction field*, capturing the task requirements; and in a similar way, the obstacle, *obs*, creates a *repelling field*, characterizing the operation environment. It can be seen from Figure 3(a) that the task field *tField* serves as a "complete" context for a *mCell* to operate in this specific example.



(a) *m* moves to *d* in *tField*



(b) *m* moves to *d* in *bField*

Figure 3: An Example of Task Field and Behavior Field

Since *mCell* differentiation is about behavior distribution, a *mCell* must be able to determine its behavior based on the given *task field*. Therefore,

we introduce a concept called *behavior field*, or *bField*, to capture the potential distribution of preferable behaviors a *mCell* can choose in a given *task field*. We further use FBR_{FD} to denote the transformation from a *task field* into a *behavior field* and introduce the following definition:

Definition 8 (Behavior Field & Field Transformation): $bField = FBR_{FT}(tField, B)$ where, FBR_{FT} : field based regulation (FBR) operator for field transformation; bField: behavior field; tField: task field.

According to Definition 8, how behaviors should be distributed is largely dependent on the field transformation operator FBR_{FT} . There can be different ways to represent *bField*. One may associate "rewards", "risks", or "probability" with different "locations" for a *mCell* to perform different behaviors. The "locations" can be defined as real 2- or 3-dimention spaces or n-dimension virtual spaces depending on the task domain and *mCell* properties. Figure 3(b) shows a simple example of *bField*. A *mCell m* is moving in the *task field* caused by the destination *d's attraction field* and the obstacle *obs' repelling field*. Based on some given field transformation operator, *FBR_{FT}*, the *mCell m* creates a *bField* around itself denoted by the curved dark line around *m*.

In this research, we associate a *mCell*'s "behavior distribution" with its surrounding "locations", and we further call this distribution *behavior pro-file*, or *bProfile*. Therefore, we introduce the following definition, which is a specific case of Definition 8.

Definition 8b (Behavior Profile & Field Transformation): $bProfile = FBR_{FT}$ (tField, B) where: $bProfile := \{(b_1, p_1), ..., (b_n, p_n)\}; \& [b_i \in B, 0 \le p_i \le 1, 1 \le i \le n]$ indicates (behavior, probability) pairs for a mCell to choose its actions, and n is the total number of possible behaviors that the mCell can perform; tField: task field; B: mCell's behavior set. ■

The dark line in Figure 3(b) mentioned above indicates the "behavior profile" for mCell m. Given a behavior profile at a given point of time, a mCell still need to "make a decision" to select a behavior. We introduce the second field based behavior regulation operator as follows.

Definition 9 (Behavior Selection & Behavior Selection): $b = FBR_{BS}(bProfile)$ where: *FBR_{BS}*: field based regulation (FBR) operator for behavior selection; *bProfile* := {(b_1 , p_1), ..., (b_n , p_n)}; & [$b_i \in B$, $0 \le p_i \le 1$, $1 \le i \le n$]; *b*: selected behavior $b \in B$.

Summarizing the above definitions, for a mCell m at time t under given functional requirements FR and environmental situation Env, the behavior or action of the mCell is chosen by following the following self-organizing operations:

$$b_{t+1} = FBR_{BS} \left(FBR_{FT} \left(FLD \left(SNS(FR_t, Env_t) \right) \right) \right)$$
(1)

From Equation (1), one can see that a mCell's sensing capability (*SNS*), its capability of forming a internally useful information field (*FLD*), and its field based behavior regulation based on field transformation (*FBR_{FT}*) and behavior selection (*FBR_{BS}*) completely determine the mCells self-organizing behavior. Interactions between mCells can be introduced by devising constraints between mCells that influences a mCell's *SNS*, *FLD*, *FBR* capabilities and consequently its behavior. The system stabilities and functions are achieved around the "attractors" of the mCells. For different task domains, these capabilities should be designed and devise differently so that the overall performance of the emergent behavior is desirable. In the following, we describe examples of how these self-organizing capabilities can be design and implemented in computer simulations.

Case Studies and Discussion

To investigate how such our approach can be applied to CSO systems design, a set of computer simulation based case studies were performed with the intention of addressing the following questions:

- What constitutes the task and behavior fields?
- What is the benefit of using the concept of behavior field?
- How will locally regulated behaviors emerge into desired global effects?
- How will the field transformation (*FBR_{FT}*) and behavior selection (*FBR_{BS}*) impact the global system behavior?

In the following subsection, we present two case studies. The first case study is designed to investigate the concept of *field* and the second one for demonstrating FBR effectiveness.

Case Study 1: Single Exploration Cell

The overall task for this case study is for one mCell to travel to a given destination in an unknown environment. The two functional requirements are:

FR1 = "move to destination", and

FR2 = "avoid obstacle".

The *mCell* can decide the *direction* of movement, so the two behaviors are:

 b_1 = "move to the direction toward destination", and

 b_2 = "move away from the direction to obstacle".

We further assume that the obstacles between the *mCell* and the destination can be everywhere with any density and that the *mCell* can always sense the location of the destination and can sense the locations of the obstacles only when they are within a certain range. Given the two functional requirements, the sensor information and current actions, a *mCell* needs to decide which "action", i.e., direction, to take.

Task Field

The task field for this example is composed of the *attraction field* of the destination and the *repelling fields* of various obstacles, and more than one obstacle can exist at any time. We use parameter θ to represent the *attraction field* and β the *repelling field*, as show in Figure 3. Combining the two, we have task field for *mCell m*:

 $tFiled_m = \{ \theta; \beta_1, \beta_2, ..., \beta_n \};$ where, n = no. of obstacles



Figure 4: Tasks Field for mCell *m*

Behavior Regulation

As described above, in CSO systems field-driven behavior regulation has two steps, i.e.,

Step1: Transform *tField* into *bFiled* through *FBR*_{FT}

Step2: Select a specific behavior/action through FBR_{BS}.

Behavior field and *FBR_{FT}*: In this example, the *bField* or *bProfile* determines the likelihood in which a *mCell* is taking its next move into direction α , and the likelihood the *mCell* is avoiding this direction due to the existence of obstacles. The distribution of these two likelihoods around the 360 degree circle around a *mCell* constitutes the *bField* or *bProfile* of the *mCell*. Specifically, for one destination and on obstacle, we introduce the following *FBR_{FD}*:

$$bField_{m}(\alpha) = FRB_{FT}(tField_{m}, B) = \{\alpha, p_{\alpha}, q_{\alpha}\}$$
$$= \{\alpha, \frac{1}{\sqrt{2\pi}}e^{-\frac{(\alpha-\theta)^{2}}{2}}, \frac{1}{\sqrt{2\pi}}\left(1 - e^{-\frac{(\alpha-\beta)^{2}}{2}}\right)\}$$

where, α : direction for the next move

 p_{α} : probability that direction α should be taken

 q_{α} : probability that direction α should be avoided

Behavior selection and FBR_{BS} : After the behavior field is established, a *mCell* needs a mechanism for behavior selection. In this case study, we define two types of behavior selections: "select the best" and "select any one good enough", as indicated below.

 $FBR_{BS-B} = [Select the action with the highest probability in the bField]$

 FBR_{BS-G} = [Select any action, randomly from the actions that has a bigger than threshold probability in the *bField*]

In this case study, we will show how the above mentioned behavior field can be useful and the effective of applying different behavior selection strategies.

Figure 4 shows the time sequence of screen dumps of one of our simulation runs, with time steps indicated at the bottom of each box. As shown in Figure 4, a single explorer mCell can travel from a randomly assigned position on the left to a given destination on the upper right. Both the mCell's initial position and the positions of all obstacles are randomly generated for each simulation run.

In this case study, the *mCell* acts solely based on the task assignment (represented as FRs) and its sensory information without memory and planning. The FBR_{FT} constantly trans-forms the perceived *task field* into local *behavior field*, allowing the *mCell* to "know" what are possible valid behaviors that can be performed at each moment. Furthermore, the FBR_{BS} converts behavior or action potential into specific actions. By splitting the

16

process of FBR into two steps, a designer can make various combinations and find the good ones for his/her task domain.



Figure 5: Simulation Results of a Single *mCell* Exploring in a Random Obstacle Field Simulation Results

As one may imagine, when the density of obstacles increase, the *mCell* may be trapped on its way and not be able to reach the destination. Our simulation results verified this statement. To investigate how different FBR strategies may influence the "success rate" of the simulation runs, we examined two "behavior selection" strategies, i.e., FBR_{BS-B} (select the best) and FBR_{BS-G} (select from good enough, i.e. top 40%, randomly). We ran 500 test runs for each obstacle density for FBR_{BS-B} and FBR_{BS-G} , respectively, and calculate the success rate based on the 500 runs. Figure 5 shows the comparison result with 40 to 120 randomly assigned obstacles.

Figure 5 shows that overall the "select from good enough randomly" works better than "select the best" and that as the density of obstacles increases the advantage of the former increases. From a CSO system development perspective, the result is interesting in two ways. First it indicates that behavior regulation strategies have profound impact on individual



mCell's performance, and secondly the "randomness" seems to bring "in-telligence" into the system mechanically.

Figure 6. Comparison of "Select the Best" (*FBR*_{BS-B}) and "Select from Top 40% Randomly" (*FBR*_{BS-G})

With the "select the best" strategy, a *mCell* always targets on one single best direction in deciding on their next move. When the obstacle density is low, this strategy can likely produce ideal performance in which both time and energy can be saved. The reason behind is that with limited number of obstacles distributed sparsely, there is close to zero likelihood that the *mCell* may get trapped by its own "best" calculation. When the density of obstacles increases, however, much more likely the "traps" exist in the field, resulting in lower success rate for this strategy.

The "select from top 40% randomly" strategy may not work perfectly in terms of saving time and energy. However, when the environment becomes more unpredictable and unfriendly, the *mCell* can robustly sustain the environmental change and maintain its performance. Thanks to the randomness of behavior selection, the "traps" may be overcome by the *mCell* through internal variability. Only the intrinsic variety of the system (i.e., *mCell* in this case) can concur the variety of the environment (Ashby 1958).

Case Study 2: CSO Mover System

In the single *mCell* case study, we demonstrated how *tField* can be defined and how *bField* can be generated and *behavior selection* be carried out through field-driven regulation (FBR). To investigate how FBR may impact on the emergence when multiple such *mCells* work together for a Field Based Behavior Regulation for Self-Organization

single task, we conducted the second case study. In Case Study 2, the task for multiple identical *mCells* is to move an object from a start point to a destination point in a two dimensional unknown environment with all the obstacles randomly distributed in the field in the same way as in Case Study 1. The *mCells* are limited in action: they only push the object from their center to the object's center. At a given time, a *mCell* must decide on which direction to push the object. The overall movement of the object will be the result of the emergent behavior of all the *mCells* pushing the object.

In this case study, all *mCells* can only push from their centers to the object's center with the same force, and the overall movement of the object is the emergence of all *mCells*' relative locations. The behavior of each *mCell* is to choose a "right" location to push the object. The three functional requirements are:

FR1 = "stay close to the object",

FR2 = "push object to destination", and

FR3 = "avoid obstacles".

A *mCell* can choose a relative location to the object, so the three behaviors are:

 b_1 = "move to locations as close as possible to the object",

 b_2 = "push the object towards destination", and

 b_3 = "push the object away from obstacles".

We assume that all the *mCells* have similar setup as the previous case study; they can sense the destination anywhere and they can only sense the obstacles within a certain range.

Task Field

Similar to the previous case study, we also use parameter θ to represent the *attraction field* and β the *repelling field*. In addition to those two, this case study introduces a new *attraction field d* as the relative distance from *mCell* to the Object. The related task field is shown in Figure 7 and besides *mCell* m there are *mCells* i, j and k in dash line. Combining the three, we have task field for *mCell* m:

tFiled_m = { d, θ ; $\beta_1, \beta_2, ..., \beta_n$ }; where, n = no. of obstacles

Behavior Regulation

The two steps behavior regulation described in the previous case study is still valid in this case:



Figure 7: Tasks Field for *mCell m*

Behavior field and *FBR_{FT}*: In this example, the *bField* or *bProfile* determines the likelihood in which a *mCell* is taking its next move to either stay in the current location to push the object or move to other locations because the relative distance is too far, the pushing direction is towards a collision or the pushing direction is away from destination. The relative location for *mCell* is represented by α and *d*. For one destination and on obstacle, we introduce the following *FBR_{FT}*:

$$bField_{m}(\alpha, d) = FRB_{FD}(tField_{m}, B)$$

= {\alpha, d, p_{\alpha}, p_{\alpha}, q_{\alpha}}
= {\alpha, d, \frac{1}{\sqrt{2\pi}}e^{-\frac{d^{2}}{2}}, \frac{1}{\sqrt{2\pi}}e^{-\frac{(\beta-\alpha)^{2}}{2}}, \frac{1}{\sqrt{2\pi}} \left(1 - e^{-\frac{(\beta-\alpha)^{2}}{2}}\right)}

where $\boldsymbol{\alpha}:$ the angle corresponding to an arbitrary predefined coordinate

d: the related distance.

p_d: probability that distance d should be taken

 p_{α} : probability that pushing direction α should be taken

 q_{α} : probability that pushing direction α should be avoided

Behavior selection and FBR_{BS} : After the behavior field is established, a *mCell* needs a mechanism for behavior selection. In this case study, we assume that the *mCell* will change their location when the probability is below a threshold instead of choosing the "best" locations.

 FBR_{BS} = [Select any action, randomly from the actions that has a bigger than threshold probability in the *bField*]

In this case study, we will show how the above mentioned behavior field can be useful and effective for not only a single mCell case but an emergent system of multiple mCells.



Figure 8. Simulation for Design Case 2, CSO Object Mover

Simulation Results

Figure 8 shows the time sequence of simulation screen dumps with time steps indicated at the bottom of each block. All each *mCell* chooses a location to push the blue square Object. Each *mCell* attempts to choose a "highly" recommended zone and move into it when the zone of its current location has the probability below the threshold. There is no explicit communication between the *mCells*, reducing the need for more design efforts. However, the *mCells* interact indirectly by avoiding overlapping with each other. Our simulation results showed that with the setup of this simulation, in almost all simulated test runs, the *mCells* were successful in pushing the square object into its destination.

One advantage of this behavior based design is that the shape of the Object and therefore the shape of the overall system are not predefined and limited in any way. The *mCells* observe the world and decide on their behaviors locally, as the global behavior and result emerge. Based on Kolmogorov complexity measure (Li and Vitanyi 2008), our CSO system of multiple *mCells* can be considered highly complex since the states of each *mCell* changes dynamically without certainty and it takes a rather long description to capture the whole system. However, using FBR makes it possible to regulate *mCells*' behaviors and to lead the emergence process to a productive direction.

Figure 9 illustrates the dynamically changing behavior field (*bField*), and how *mCells* choose their behaviors (i.e., locations) through FBR. As shown in Figure 8 the different current situations introduce two different

bFields. Depending on the relative locations of the destination, obstacles and the object, the field changes, as shown as color changes in Figure 9. Different colors in Figure 9 correspond to different probabilities, as indicated in the figure. The *mCells* try to choose the "green" or "yellow" zone to occupy. Through the use of field driven behavior regulations (FBR), the system dynamically adapts to its new situations even for the simple designed *mCells* of limited capability (can only push from its center to the object's center). The system can move the object in an unknown environment by *mCells* using the fields as their dynamic vision of the world. It is conceivable that the *bFields* and FBR concepts can be applied to those task situations where physical fields and chemical fields exist. We plan to expand our application example domains to assess the effectiveness of our field and FBR concepts.



Figure 9: Illustration of the Dynamic *bField* of the CSO Mover in the Simulated Field of Obstacles

In our simulation test runs, we also examined how the system might perform if some *mCells* become inactive. Figure 10 shows the resilience of the overall system when some of the *mCells* become "dead" during the simulation. There are four *mCells* that were deactivated at the step 400, since the system is fully decentralized, deactivated *mCells* had little influence to the rest of the *mCells* in the system. This way, although the system losses its performance due to the loss of *mCells*, it could still successfully accomplish the task of moving the object to its destination, showing the system resilience.

Because CSO systems are decentralized and have redundancies maintained among its *mCells*, they are more resilient than the systems with specified local functional components. When one part of the system fails, other nearby *mCells* can modify its functionality and redistribute their functions. This way, the system can not only adapt to the environmental change but also to the system change.



Figure 10: Resilience Test by Deactivating 4 of 12 mCells at Step 400

Conclusions

This paper presents a field based behavior regulation approach to designing cellular self-organizing (CSO) complex mechanical systems. The concept of CSO systems is developed based on the observations of 1) that current engineered systems are inherently incapable of dealing with variable functional requirements, changing environment situations, and possible system failures, and 2) that natural systems including dynamical systems and biological systems are formed in a bottom-up fashion and inherently equipped with capabilities to deal with uncertainties and unknowns. By combining the current engineering concepts of functions with the fundamental mechanisms of stability, self-organizing and DNA, our proposed CSO systems framework promises a different approach to developing engineered systems.

In our proposed CSO systems, self-organization is the key concept and mechanism. To make mechanical cells self-organize in a bottom-up fashion, a field concept is introduced that allows mechanical cells to sense the tasks and environment and formulate a task field as a model of the task world. By following the field based behavior regulation mechanism that we devised, mechanical cells can transform the sensed task field into their behavioral field in which their possible behaviors/actions are profiled and ready to be selected. The final behavior section is carried out by the FBR behavior selection operator. It is worth mentioning that our field-based behavior regulation framework is composed of distinguishable stages of cellular operations, including sensing, field formation, field transformation, and behavior selection. These operators together with their associated variables provide a rich design space for us to explore and design CSO systems. The case studies discussed in the paper demonstrated how different FBR behavior selection strategies may yield different performances, and how transformation from task field to behavioral field determines the system behaviors and capabilities. It is expected that different domain tasks require different designs of FBR mechanisms. Future research is needed to classify the domain tasks and explore various possible FBR designs.

Our current work on this research includes expanding the case study into more sophisticated problem domains, examining trade-offs of having various combinations of *mCells* including heterogeneous ones and between swarm *mCell* structures as we presented in this paper and more structured organizations that require more tight connections, e.g., physical dockings, among *mCells*.

This paper is based on the work supported in part by the National Science Foundation under Grant No. CMMI-0943997. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

- Alon, U. (2007) "Network motifs: theory and experimental approaches", Nature Reviews Genetics Volume 8, 450-461.
- Ashby W.R. (1958). "Requisite variety and its implications for the control of complex systems." *Cybernetica* 1:2, p. 83-99.
- Audesirk, G., Audesirk, T., Byers, B.E., (2007). "Biology: Life on Earth with Physiology (8th Ed.)." Benjamin Cummings, San Francisco.
- Bentley, P.J. (1999). Evolutionary Design by Computers, Morgan Kaufmann Publishers Inc.
- Bonnie R.M. and Malaga, R. (2000), "A Co-Evolutionary Approach to Strategy Design for Decision Makers in Complex Negotiation Situation," *Proceedings of the 33rd Hawaii* International Conference on System Sciences
- H. Bojinov, A. Casal, and T. Hogg, (2000) "Multiagent control of self-reconfigurable robots," in Proc. IEEE ICMAS Int. Conf. Multiagent Systems, 2000, pp. 441–455
- Bongard, J., Zykov, V. and Lipson, H. (2006), "Resilient Machines through Continuous Self-Modeling", *Science*, 314. no. 5802, pp. 1118-1121.
- Butler, Z., Kotay, K., Rus, D., and Tomita, K. (2001). "Cellular automata for decentralized control of self-reconfigurable robots". In *Proceedings of the ICRA 2001 Workshop on Modular Robots*.
- Dickinson, M.H. (1999). "Bionics: Biological Insight into Mechanical Design." Proc. of the National Academy of Sciences of the USA, 96/25:14208-14209.
- Fan, Z., Seo, K., Hu, J., Rosenberg, R., and Goodman, E.D. (2003), "System-Level Synthesis of MEMS via Genetic Programming and Bond Graphs," Proc. 2003 Genetic and Evolutionary Computing Conference, Chicago, Springer, Lecture Notes in Computer Science, July, 2003, pp. 2058-2071.

- Ferguson, S., and Lewis, K., 2006, "Effective Development of Reconfigurable Systems Using Linear State-Feedback Control," *AIAA Journal*, Vol. 44, No. 4, pp. 868-878.
- Fogel, L.J., Owens, A.J. and Walsh, M.J. (1996), Artificial Intelligence through Simulated Evolution, Wiley, New York.
- Fukuda, T., Kawauchi, Y., (1990) "Cellular Robotic System (CEBOT) as One of the Realization of Self-organizing Intelligent Universal Manipulator", in Proc. of IEEE Int'l Conf. on Robotics and Automation'90 (R&A90), pp. 662-667
- Fukuda, T. and Nakagawa, S., (1988), "Approach to the dynamically reconfigurable robotic system." Journal of Intelligent and Robotic Systems, Volume 1, No. 1, pp. 55-72.

Zouein, G., Chen, C., and Jin, Y. (2010): Create Adaptive Systems through "DNA" Guided Cellular Formation, Proceedings of First International Conference on Design Creativity (ICDC2010), 2010, Kobe, Japan

Gardner, M. (1970) "Mathematical Games: The Fantastic Combinations of John Conway's New Solitaire Game 'Life'", in Scientific American, Vol.223, pp.120123.

Gell-Mann, Murray (1994). *The quark and the jaguar: adventures in the simple and the complex*. W.H. Freeman, San Francisco, CA, USA.

Gershenson, J.K., Prasad, G.J. and S. Allamneni, S. (1999). Modular Product Design : A Life-Cycle View. *Journal of Integrated Design and Process Science*. 3/4, 13-26.

- Gilpin K, Kotay K, Rus D and Vasilescu I (2008) Miche: modular shape formation by selfdisassembly. *International Journal of Robotics Research* 27: 345–372.
- Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley Longman, Inc.
- Gu, P.,Hashemian, M., Sosale, S., Rivin, E. (1997) "An Integrated Modular Design Methodology for Life-Cycle Engineering," in CIRP Annals - Manufacturing Technology, Volume 46, Issue 1, 1997, Pages 71-74
- Holland, John H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. Cambridge, Mass: MIT Press.
- Jin, Y., Li, W., Lu, C-Y.S., (2005). "A Hierarchical Co-Evolutionary Approach to Conceptual Design," Annals of the CIRP, 54/1:155-158.
- Jin, Y. G. Zouein, S. Lu, (2010) "A Synthetic DNA based Approach to Design of Adaptive Systems", CIRP Annals – Manufac-turing Technology, Vol. 58/1, pp.153-156, 2009
- Kitano, Hiroaki (2002), "Systems Biology: A Brief Overview", Science 1 March 2002: Vol. 295 no. 5560 pp. 1662-1664
- Kopetz, H. (1998) "Component-based design of large distributed real-time systems," Control Engineering Practice, Volume 6, Issue 1, 1 January 1998, Pages 53-60.
- Koza, J.R., (1992), Genetic Programming: On the Programming of Computers by Means of Natural Selection. The MIT Press.
- Koza, J.R., Bennett, F.H., Andre, D. and Keane, M.A. (1999). Automated Synthesis of Analog Electrical Circuits by Means of Genetic Programming, in *IEEE Transactions* of Evolutionary Computing, Vol.1(2), pp.109-128.
- Kumar, S. and Bentley, P.J., (2000) "Implicit evolvability: An investigation into the evolvability of an embryogeny," in Proceedings of the Genetic and Evolutionary Computation Conference. Las Vegas, NV: Morgan Kaufmann, 2000.
- Lee, C-Y, Ma, L. and Antonsson, E.K. 2001, "Evolutionary and Adaptive Synthesis Methods," Formal engineering design synthesis, Pages: 270 – 320, Cambridge University Press, USA
- Li, M. and Vitanyi, P. (2008) "An introduction to Kolmogorov complexity and its applications", Springer
- Lipson,H., (2007) "Evolutionary Robotics: Emergence of Communication," *Current Biology*, Volume 17, Issue 9, pp R330-R332.

Maher, M.L., 2001, "A Model of Co-Evolutionary Design," Eng. with Comp., 16:195-208.

- Martin, M. V. and Ishii, K. (2002). "Design for Variety: Developing standardised and modularized product platform architectures." *Research in Engineering Design* 13/4, 213-235.
- Parmee, I.C., 1997, "Evolutionary Computing for Conceptual and Detailed Design," in Genetic Algorithms in Engineering and Computer Science, John Wiley & Sons Ltd.
- Rus, D. and Vona, M., (2001). "Crystalline robots: self-reconfiguration with compressible unit modules." *Kluwer Autonomous Robots 10*, pp. 107–124.
- Salemi, B., Shen, W. M. and Will, P., (2001). "Hormone controlled metamorphic robots." Proceedings of the IEEE Intl Conf. on Robotics and Automation, pp 4194-4200.
- Shen W. M.; Will, P., Galstyan, A., Chuong, C. M.; (2004) "Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms," *Autonomous Robots*, Vol. 17, No. 1., pp. 93-105.
- Shen, W. M., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M. and Venkatesh, J., (2006). "Multimode locomotion for reconfigurable robots." *Autonomous Robots*, 20(2), pp.165–177.
- Shen, W. M., Salemi, B., and Will, P., (2002). "Hormone inspired adaptive communication and distributed control for CONRO self-reconfigurable robots." *IEEE Transactions on Robotics and Automation, Volume 18, Number 5*, pp. 700-713.
- Shu, L.H. Lenau, T.A., Hansen, H.N., Alting, L. (2003). "Biomimetics Applied to Centering inMicro-assembly," Annals of theCIRP, 52/1:101-104.
- Subramanian, L. and Katz R.H.(2000) "An Architecture for Building Self-Configurable Systems", IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC 2000), Boston, August 2000
- Suh, N. P., (1990). "The principles of design." Oxford University Press, New York, NY.

Suh, N. P., (2001). "Axiomatic design." Oxford University Press, New York, NY.

- Unsal, C., Kilic, H., and Khosla, P., (2001). "A modular self-reconfigurable bipartite robotic system: implementation and motion planning." *Kluwer Autonom. Robots 10*, pp.23-40.
 Vajna, S., Clement, S., 2002, "Autogenetic Design Theory: An Approach to Optimise Both
- Vajna, S., Clement, S., 2002, "Autogenetic Design Theory: An Approach to Optimise Both the Design Process and the Product," Proceedings of DETC02, ASME 2002 Design Engineering Technical Conferences, DTEC2002/DAC-34038, Montreal, CA.
- Vincent, J.F.V., Bogatyreva, O., Bogatyrev, N., (2006). "Biology Doesn't Waste Energy: That's Really Smart." Proceedings of the SPIE, 6168:1-9.
- von Foerster, Heinz, 1977."Objects: tokens for (eigen-)behaviors." In: *Hommage a Jean Piaget: Epistemologie Genetique et Equilibration*. B. Inhelder, R. Gracia, and J. Voneche (Eds). Delachaux et Niestel.
- von Neumann, J. (1966). The Theory of Self-Reproducing Automata. Arthur Burks (Ed.) University of Illinois Press
- Weisbuch, G. (1991) Complex systems dynamics: An introduction to automata networks. Santa Fe Inst. studies in the sci. of complexity: Lecture notes, Vol. 2.; Ryckebusch, S. (Trans) Addison-Wesley/Addison Wesley Longman. Reading, MA, US.
- Wolfram, S. (2002) A New Kind of Science, Wolfram Media, Inc. Campaign, IL, USA
- Yim, M., Duff, D. G. and Roufas, K. D., (2000). "PolyBot: a modular reconfigurable robot." *Proceedings of the ICRA IEEE International Conference on Robotics and Automation*, pp. 514-520.

Yim, M., Zhang, Y. Duff, D., (2002), "Modular Robots" *IEEE Spectrum 39/2*, pp.30-34. Yogev, O., and Antonsson, E. K. A Novel Synthesis Design Approach for Continuous

Inhomogeneous Structures, In 19th International Conference on Design Theory and Methodology (DTM) (Sept. 2007), ASME, Paper Number: DETC2007/DTM-35662.

Field Based Behavior Regulation for Self-Organization

- Yu, C. H.; Haller, K.; Ingber, D.; Nagpal, R.; , (2008) "Morpho: A self-deformable modular robot inspired by cellular structure," *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference*, vol., no., pp.3571-3578, 22-26.
- Zouein, G., Jin, Y. (2009) "A biologically inspired DNA-Based Approach to Developing Cellular Adaptive Systems", Ph.D. Thesis, USC, Los Angeles, CA.
- Zykov, V., Mytilinaios, E., Adams, B. and Lipson, H. (2005). "Self Reproducing Machines." *Nature*, **435**, pp. 163-164