(preprint)

Emergent Control

Peter Dittrich,Peter Kreyssig (2011)

Bio Systems Analysis Group, Department of Mathematics and Computer Science,
Friedrich Schiller University Jena, 07743 Jena, Germany,
{peter.kreyssig|peter.dittrich}@uni-jena.de

Published as:

P. Dittrich, P. Kreyssig. Emergent Control. In: C. Muller-Schloer, H. Schmeck, T. Ungerer (Eds.), Organic Computing A Paradigm Shift for Complex Systems, Autonomic Systems, Volume 1, Part 1, 67-78, Springer, Basel , 2011

doi: 10.1007/978-3-0348-0130-0_4

Emergent Control

Peter Kreyssig and Peter Dittrich

Bio Systems Analysis Group, Department of Mathematics and Computer Science, Friedrich Schiller University Jena, 07743 Jena, Germany,
{peter.kreyssig|peter.dittrich}@uni-jena.de

Summary. In order to control the dynamics of a system, feedback control (FC for short) is an extremely successful strategy, which is widely applied by engineers. Here we discuss a different strategy of control, called *emergent control* (EC for short), which can be found in large, distributed systems of components interacting only locally. For comparison we present a basic architecture for emergent control and two simple examples. In these examples, emergent control is achieved by a chemical computing approach. In the first example the number of objects of a particular type in a distributed system has to be kept constant. The example shows that on a macroscopic level EC and FC can display exactly the same behaviour. Hence for comparing their performance quantitatively a more refined model has to be taken into account. This model indicates a trade-off between cost and robustness. FC tends to operate at a lower cost than EC, however it also tends to instability when the system under control is large, decentralised, and/or heavily perturbed. In the second example the number of clusters in a distributed system should be controlled. The example shows how a user can "control", i.e., provide goals in EC even if the system is not tractable analytically due to highly non-linear effects.

1 Introduction

In order to exploit the increasing quantitative complexity of computational systems, various self-organisation principles are applied to *control* them. In general, the aim of control is to change the dynamics of a system in a desired way. This means to keep the system as close as possible to a desired attractor or, in particular, close to a stable state. The easiest way to achieve this is adding a feedback control loop.

Here, however, we will study a strategy based on the self-organisation metaphor. *Self-organisation* refers to a phenomenon where a system becomes organised by elements of that system. That is, the system's organisation can be explained by referring to the systems components rather than to external effects. Self-organising systems are usually thermodynamically open.

In a technological environment a user usually wants to keep control, which results in a paradoxical situation when this demand is combined with the need for self-organisation. Here we show, how this paradox can be resolved by a strategy termed *emergent control* (cf. [12, 4, 10]). In emergent control, the controlled behaviour emerges [2] at a macro-level from many microscopic local interactions [23].

The remaining paper is organised as follows. In the next section, we briefly describe the classical feedback control and sketch an architecture for emergent control. In Sec. 3 we present two examples and give an overview of techniques for achieving emergent control (Sec. 4). Finally, the difficulty of comparing EC with FC is illustrated in Sec. 5. The article concludes with a short discussion and an outlook in Sec. 6.

2 Feedback Control and Emergent Control

Feedback Control

In classical control theory a feedback controller takes measurements from the system to be controlled and uses this information to "decide" how to manipulate the system in order to achieve the desired behaviour (Fig. 1). This loop of measurement and manipulation is performed continuously. The user demands can be easily integrated by comparing them with the actual system state. For example in a *PID* (proportional integral derivative) approach the user demands, i.e., a numerical value, is simply subtracted from the measured output of the system. Then, the difference in demand and output is used to set the input of the system to be controlled. This control strategy can be found in virtually any technical system.

In passing we note that this feedback control loop is also the central explanation pattern of cybernetics [28]. Note further the relation to the *observer/controller (O/C) architecture*, which provides a regulatory feedback on the internal mechanism of the controlled system [20]. In contrast to a classical observer-controller approach [14] such an O/C loop can also modify how the controlled system works. Therefore, the system can still operate without the O/C on top, but looses its ability to adapt.

Emergent Control

A rather different approach to the problem of controlling complex systems can be found in large, distributed systems. By that we mean systems that consist of a large amount of elementary units that only interact with a few of their fellow units. Examples for this can be found in natural (e.g., gene expression in a cell), social (e.g., the economical system) and technical (e.g., the Internet) systems as well as in many others. In these systems some focal features of feedback control like the measurement of a global state or the feedback loop

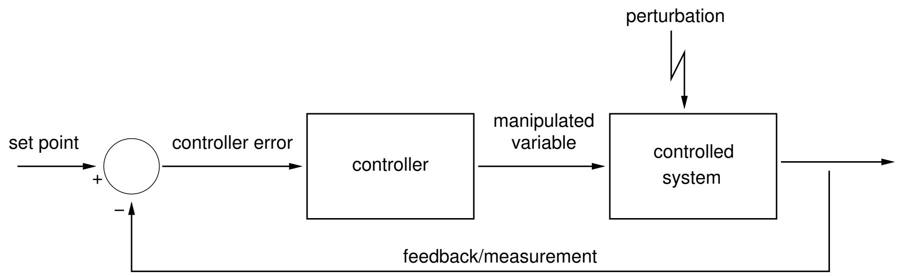

Fig. 1. Classical feedback architecture.

do not appear. Here we can find an approach where no explicit controller in a classical sense as sketched for the feedback control above exists. A set of (usually simple) local rules are executed by the different systems components, such that the desired behaviour appears at a global scale. We can say that the global behaviour is an emergent[1] property [2, 21]. Therefore we call, as a working definition, this kind of control strategy *emergent control*.

The term emergent control appears in the literature with a similar meaning as intended here. Doursat and Ulieru [8, 27] use the term *emergent engineering* to describe a framework to realise systems that exhibit controllable self-organisation. They also mention that performance, including the self-* properties, robustness etc., cannot be measured as a numerical value yet; a difficulty we will also meet here.

From a dynamical point of view, emergent control can be seen as *programming by equilibria*, a concept suggested by Tschudin and Meyer [25]. As an example they present a self-healing communication protocol. This protocol consists of rules implying a dynamics where the desired computational behaviour is an asymptotically stable equilibrium state with a large basin of attraction. The self-healing behaviour or the generation of a computational result, "emerges from the system's tendency to strive for an equilibrium" [25].

A more specialised, but also usable characterisation of emergent control systems is given in [23]: "We refer to distributed control systems based on local information as emergent control systems". Støy et al. [24] use role-based algorithms for a robot locomotion control systems. The modular robots are an example for the necessity of distributed control systems. Because of this, there is a lot of work done on emergent *control of robots*. We just mention a few and point out the idea behind the control used. Tsuchiya et al. [26] define a two-tier architecture for the movement of legged robots, one for the leg motion and one for the gait pattern. Steels [22] suggests evolutionary processed for control. Digney [5] employs Q-learning for a hierarchical control structure of

[1] By emergence we mean what Bedau and others call "weak emergence" [2]. In the first example in Sec. 3.1, however, we will be less strict concerning the non-linear property of the micro-macro relationship and take a linear system as a model to illustrate emergent control.

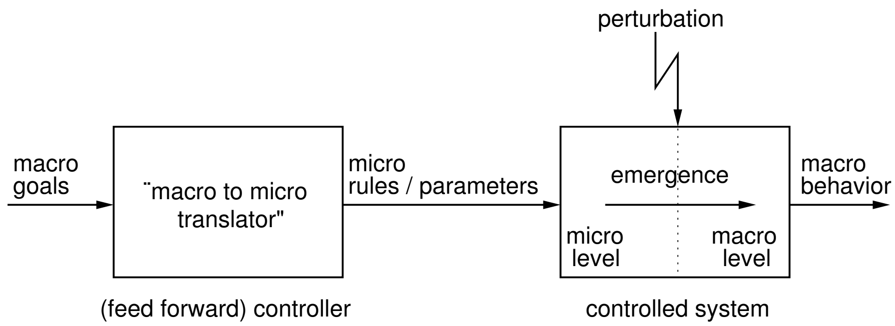

Fig. 2. Basic architecture for controlled self-organisation through emergent control. Note that there is no feedback at the macro level; except for potential user intervention as a result of the user's observation of the macro behaviour (not shown). There is especially no feedback of the macro quantity that should be controlled. There is however some kind of feedback at the micro level, which consists of many interacting elements.

robots. Meeden et al. [18] train an artificial neural network with reinforcement learning to control their robot "carbot". They also experimented with the "transplantation" of the learned network from a simulator to the real robot and the other way around.

Ishiguro et al. [11] use a particular oscillator equation (inspired by slime mould behaviour) to arrange the morphology of a modular robot whose modules are connected by Velcro. In a more abstract setting, emergent control can be used to coordinate the processing of distributed knowledge within a multi-agent expert system [16]. From here we envision a fascinating application area of emergent control in hybrid social systems, which consists of software agents and social actors [15, 19, 9].

A basic diagram of a possible *general architecture* for systems using emergent control is shown in Fig. 2. Most significantly there are two layers or levels to distinct, the macro- and the micro-level. After handing the macro goals to the translator or compiler, it creates micro rules, which lead to the (emerging) macro-behaviour in the controlled productive system.

3 Examples

We present two examples. The first example is kept as simple as possible. It shows that when comparing EC with FC at macroscopic level as it is usually done in control theory by using an ODE (ordinary differential equation), their behaviour can be exactly the same. Thus refined models are required for comparison. Because the first example is linear and thus the relation between micro-level and macro-behaviour is simple, we also present a highly non-linear example. In this example, the macro-state (i.e., the number of clusters) cannot be inferred from a micro-element (i.e., a numerical value).

3.1 EC and FC result in the same macro-behaviour

Let us assume that we have a system containing a number of objects of type A. And let $x(t)$ denote the number of these objects in the system at time t. The control task is to keep the number of objects at a constant value x_r provided by the user. For feedback control we assume that the system can be manipulated by adding or removing objects at a rate y. Using a linear function as a controller (i.e., a P-controller) we obtain the following ODE for the system and the controller respectively:

$$\dot{x} = y(t) \qquad \text{and} \qquad y(t) = k(x_r - x(t)). \tag{1}$$

This yields the following total dynamics:

$$\dot{x} = k(x_r - x(t)) = kx_r - kx(t). \tag{2}$$

For $k > 0$ this is a linear system with one asymptotically stable fixed point at x_r, as desired. With k we can control the eigenvalues and thus the speed at which the fixed point is approached.

For emergent control there needs to be a micro-level. In our simple example, we obtain a micro-level by assuming that an object can vanish spontaneously at a certain rate k_2 (or, equivalently, with a certain probability). The micro-rules are given by the following simple chemical program[2]:

$$\xrightarrow{k_1} A \tag{3}$$

$$A \xrightarrow{k_2} \tag{4}$$

Assuming mass-action kinetics, this chemical program can be dynamically interpreted in the following way. Within in a sufficiently small time interval $d\tau$, an object of type A appears in the system with probability $k_1 \cdot d\tau$. Furthermore each object of type A disappears with probability $k_2 \cdot d\tau$. If the number of objects is large, we can safely model the system with an ODE:

$$\dot{x} = k_1 - k_2 x(t). \tag{5}$$

Note that with $k_1 = kx_r$ and $k_2 = k$ we obtain exactly the same ODE as in our feedback control model (Eq. (2)). Although the macroscopic ODE model is exactly the same, FC and EC differ as illustrated by Fig. 3.

In the example we see that the description of the system by ODEs only is not sufficient to describe the differences between EC and FC. Therefore we argue that a comparison is only possible with one (or several) additional refined models. Then a quantitative rather than a qualitative evaluation of properties like robustness becomes feasible. Often these refined models have to take the actual implementation into account and not only the abstracted goals. We extend this example by a toy model in Sec. 5.

[2]Here it is sufficient to see a chemical program just as a set of reaction rules or rewriting rules, which are equivalent to a Petri-net, see [6].

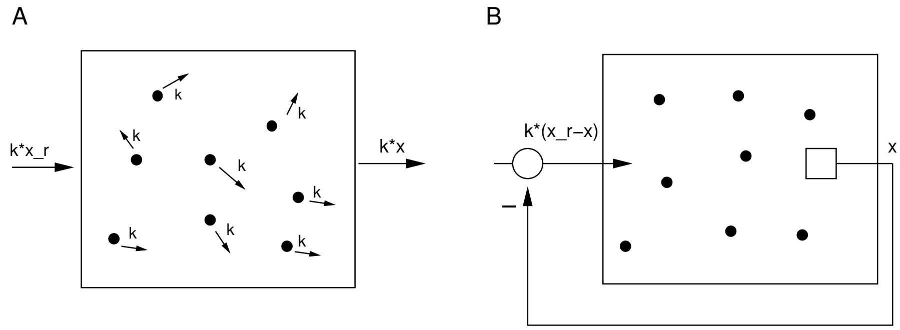

Fig. 3. Illustration of a concrete instance of emergent control (A) and feedback control (B) that have the same macro-behaviour expressed as an ordinary differential equation (cf. Eq. (5) and Eq. (2), respectively). This serves also as inspiration for the toy model shown in Sec. 5. In (A) every present molecule has the probability k to vanish (cumulative outflow of kx) and there is a global inflow of kx_r. In (B) x is measured and the inflow of molecules is regulated to $k(x_r - x)$.

3.2 Emergent control of the number of clusters

In the next example we assume that we have a population of objects, each characterised by a real number, which should form clusters. The aim of control in this example is to stabilise a certain amount of n clusters, while we do not care which object is part of which cluster.

To achieve this with emergent control, we assume a microscopic dynamics taken from the seceder model [7]. The seceder model is a simple individual based model that shows how a local advantage to be different gives rise to the formation of clusters. The model consists of a population of objects, which are real numbers, here. Objects reproduce and die. In a single reproduction event three objects are chosen randomly and the objects that possess the largest distance to their mean is reproduced by creating a mutated copy (offspring). Mutation is performed by adding a normally distributed random number with mean 0 and variance 1 denoted by $N(0, 1)$. The offspring replaces a randomly chosen object of the population.

Formally, the (basic) seceder model is defined as follows. The population of size M is represented by an array $P = \{P[1], \ldots, P[M]\}$ of objects $P[i] \in \mathbb{R}$. We write $P(t)$ for the population at time t, and $P(t)[i]$ for the i-th object of population $P(t)$. The population evolves over time according to the following algorithm:

while $\neg$terminate()do
  $s_1 := P[\mathsf{randomInt}(1, \mathsf{M})]$
  $s_2 := P[\mathsf{randomInt}(1, \mathsf{M})]$
  $s_3 := P[\mathsf{randomInt}(1, \mathsf{M})]$
  $P[\mathsf{randomInt}(1, \mathsf{M})] := f_{sel}(s_1, s_2, s_3) + N(0, 1)$
  $t := t + 1/M$
od

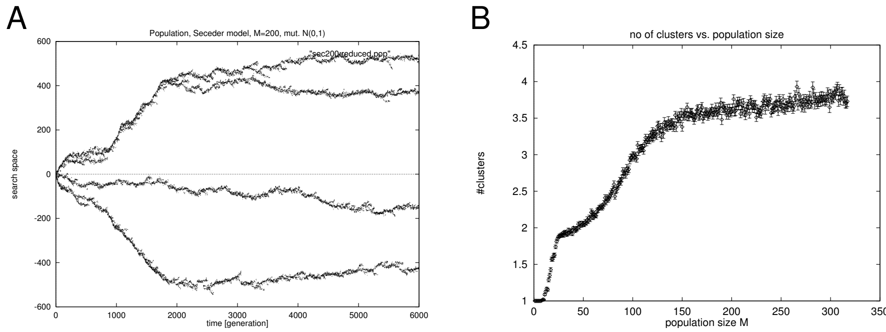

Fig. 4. Seceder model (second example). (A) Dynamics of the seceder model for a population size of $M = 200$. For each point in time t all individuals P[i](t) are plotted on the horizontal axis. (B) Relation between the number of clusters and the population size M.

First three individuals are chosen randomly. Then a randomly chosen individual is replaced by selecting the individual with the largest distance to others and adding a random number. Finally the time counter is incremented. The procedure $\mathsf{randomInt}(a, b)$ returns a uniformly distributed random number out of $\{a, a+1, \dots, b\}$. The selection function

$$f_{sel}(g_1, g_2, g_3) = \begin{cases} g_1 & \text{if } F_1 \geq F_2 \wedge F_1 \geq F_3, \\ g_2 & \text{if } F_2 \geq F_1 \wedge F_2 \geq F_3, \\ g_3 & \text{otherwise,} \end{cases} \text{ where } F_i = \|g_i - \frac{1}{3}(g_1+g_2+g_3)\|, \tag{6}$$

returns the argument that possesses the largest distance to the mean of the three arguments. Note that the seceder model can be interpreted as a chemical program with third order catalytic reaction rules and dilution flow:

$$s_1 + s_2 + s_3 \to s_1 + s_2 + s_3 + s_\mu \quad \text{with} \quad \mu = f_{sel}(s_1, s_2, s_3), \tag{7}$$

$$s_i \to \quad \text{for all objects } i. \tag{8}$$

When running the system (e.g., initialised with, $P[i] = 0$ at $t = 0$) clusters appear spontaneously (Fig. 4). The number of clusters depends on the population size and can be also controlled by changing the tournament (e.g., using size four instead of size three).

For emergent control we need in addition of the micro-rules a *macro-micro feedforward controler*, which maps the user demands (a number of clusters) to manipulable parameters of the micro-structure of the system. To keep things simple, we assume that this feedforward controller maps a number of groups to a number of objects. Because analytical derivation of this mapping is difficult, and in general impossible, we obtain the mapping experimentally, here. That is, we simply perform simulations for different population sizes (Fig. 4, B) and invert the experimentally determined function.

Note that there is no straightforward way how the same control aim can be achieved by classical feedback control. Measuring the number of clusters is relatively easy. But it is unclear how a feedback controller should increase and decrease the number of clusters.

Finally it is interesting to note that in this example no microscopic entity has any clue about the macroscopic state. This could be different in systems where microscopic object have a memory, as for example in the social system.

4 How to Construct Macro-to-Micro Feed-forward Controller?

There are various ways to design the local or micro interactions of the system's components which achieve the emergent behaviour on the macro level.

As demonstrated for the simple chemical system in Sec. 3.1, there is sometimes the possibility to perform the deduction of local rules *manually*. The macro behaviour could then be proven mathematically. In particular for the construction of artificial chemical systems *chemical organisation theory* can be helpful (Cha. 2.6).

Another principle is looking for and copying solutions in *nature*. A mimicking of processes in nature is also possible as for example done in swarm robotics [3], artificial hormone systems (Cha. 4.4) or ant inspired algorithms (Cha. 1.6). The Publish/Subscribe architecture (Cha. 2.1) can also be seen as imitating human behaviour.

The design of the second example in Sec. 3.2 follows a different principle. By performing experiments on a particular system (in our case the seceder model) the relationship micro to macro behaviour is established. Several experiments, i.e., executing the micro-rules to produces the associated macro behaviour, are performed. From the acquired data a general relation is deduced. This can also be described as the *inversion of an experiment* or the extrapolation from data.

Another general principle is *evolution* or more general *optimisation*. Starting from an initial population of different micro-rules and by applying mutation and selection to them, it is searched for a better macro level behaviour. An example for this approach applied to a chemical implementation of a flip-flop given by Lenser et al. [13]. Astor and Adami [1] present a method for decentralised growth of artificial neural networks using evolution and development.

The use of *scouting* or *exploration* methods is discussed by Matsumaru et al. [17]. There systems are explored and searched for interesting behaviours. The basic idea is that an autonomous system is used, as a preliminary step, to explore the behaviour of the chemical reaction system. Then a specific aspect of the system's behaviour will be utilised for a particular computational purpose.

All the mentioned approaches are limited and, like manual programming, can only explore some regions of the space of all possible emergent control strategies.

5 Quantitative Comparison of the Performance of Emergent Control vs. Feedback Control

There is currently no satisfying theory that allows to compare the performance of EC and FC theoretically. The reason lies in the difficulty of quantifying additional features of emergent control and of self-organising systems in general, like robustness, self-optimisation, self-configuration, etc. These features are not appropriately captured by simply measuring cost and comparing the performance on this basis. We argue that the mentioned properties depend on the particular implementation. It is clear that a macro-level analysis, e.g., by ODEs like shown in the examples, is not sufficient. Therefore we suggest to use a refined model that includes a macro-level and a micro-level in order to measure performance, which includes for example robustness. Already from the toy example we presented, we can see some important differences between FC and EC.

As an example for a refined model we implemented a discrete version of the reaction vessel described in the example in Sec. 3.1. We used a 100 times 100 grid of fields, which each carry a molecule of type A or not. We assume initially $k = 1.0$, $x_r = 0.25$ and an empty field. After 30 simulation steps we change to $x_r = 0.5$ and execute another 30 steps.

A single step in the EC mode of operation consists of the following. For each empty field a molecule of type A is produced with a probability of kx_r and each molecule of type A is erased with probability k.

The FC mode of operation is lead by the idea that there is a controlled inflow at the left hand side columns of the field and a measurement at the right hand side columns of the field. First we count the molecules in a predefined measurement region consisting of the last 83 columns of the field. This gives an estimate for the total amount of molecules in the field. The difference between the number of molecules required (given by x_r) and the measured amount is added (or removed respectively) at the left hand side columns of the field. The molecules are subject to a diffusion process. With fixed probability each molecule is moved to a neighbouring (Moore neighbourhood) field if it is empty.

The cost can be defined to be the amount of operations on the field we have to conduct, i.e., the cost is the amount of deletion and production operations performed on the field. We remark that neither the cost for the measurement nor for the diffusion are taken into account here.

Fig. 5 shows the trade-off between correctness and effort, i.e., how precise EC is in the example, but also the immense work necessary to achieve this. The EC possessed almost perfect control behaviour and respond time. Whereas the

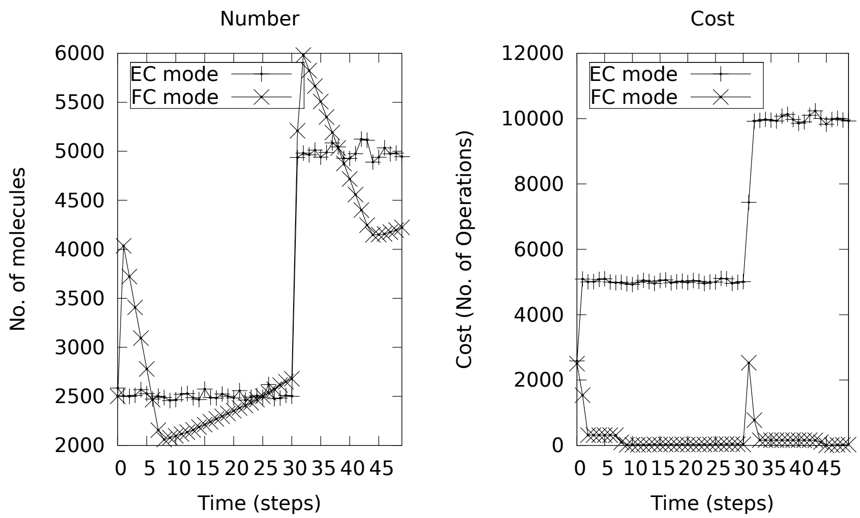

Fig. 5. Comparison EC vs FC in the toy model. The left figure shows the number of molecules present on the field for each step. The right figure shows the cost (number of modifications) needed at each step. We used the following parameter values: size of the vessel 10000, measurement on 8300 fields, $k = 1.0$, $x_r = 0.25$ (steps 0 to 30), $x_r = 0.5$ (steps 30 to 50).

FC in this case needs at least 20 steps to regulate the amount of molecules to an acceptable value, though still oscillating around the goal of 2500 molecules. After the user's interference, the setting of x_r to 0.5, the FC starts an even stronger oscillation around the goal of 5000 molecules. An interesting effect here is that the cost in the EC case depends on the amount of molecules present in the field. To summarise, FC tends to operate at a lower cost than EC, however it also tends to instability when the system under control is large and decentralised.

6 Discussion, Conclusion and Outlook

The examples discussed here are probably not realistic. The emergent phenomenon shown in Sec. 3.1 is a simple linear accumulation of the state on the micro level, i.e., it can be observed just by counting. Nevertheless Sec. 5 shows that a refined model taking into account a more realistic implementation can exhibit properties like robustness and makes it possible to quantitatively measure them. This linear relation between micro and macro level cannot be found in the seceder example (Sec. 3.2). There is not an analytical result on the relation between local rules and number of clusters known.

In our example, micro rules are changed immediately. However in an asynchronous, spatially distributed system there is almost never a direct instantaneous manipulation of the micro level. Reconfiguration can then lead to unwanted behaviour, e.g., oscillation or chaos.

Some of the methods for the construction of macro to micro translators shown in Sec. 4 are very specialised and therefore only usable in particular situations, e.g., manual design or mimicking of nature. Others, e.g., evolution or inversion, are applicable to many problems. In general there is a lack of theoretical foundation to the methods.

FC is an extremely successful strategy in a plethora of technical systems. This is also due to relatively simple architecture which has a lot of advantages. In many cases, however, EC seems to be preferable, because of additional qualitative properties that are missing in the FC. To make it equally successful a more abstract and reduced architecture should be beneficial. One approach is shown in Fig. 2.

EC systems are fundamentally different from FC systems. Since they cannot be described by classical feedback control loops, they require a different architectural perspective. In particular a micro-level description is necessary, since macro-level models of the dynamics are not enough for quantitative evaluation (Sec. 5). Therefore a powerful abstraction (including micro- and macro-level) of self-organisation and emergent control is needed.

One extension or addition to the presented concept is the combination of EC with FC. This can for example be simply achieved by controlling some parts of the system by FC while controlling other parts by EC. A combination with the Observer/Controller architecture (Cha. 4.1), learning algorithms, etc. is also possible.

An interesting aspect which needs to be discussed further is the inclusion of user demands, which is more difficult in EC than in FC systems. The quantitative analysis shown in Sec. 5 is not convincing yet and needs further investigation. These shortcomings relate to the observation that there is a lot more theory on emergence and self-organisation needed to understand and engineer systems and controllers using emergent phenomena resulting from the local-to-global problem.

Acknowledgement

We acknowledge financial support by the German Research Foundation (DFG) under grant Di 852/4-2/3.

References

[1] J. C. Astor and C. Adami. A developmental model for the evolution of artificial neural networks. *Artif. Life*, 6(3):189–218, 2000.

[2] M. A. Bedau. Downward causation and autonomy in weak emergence. *Principia*, 6:5–50, 2003.

[3] E. Bonabeau, M. Dorigo, and G. Theraulaz. Inspiration for optimization from social insect behaviour. *Nature*, 406(6791):39–42, 2000. ISSN 0028-0836.

[4] J. L. Deneubourg and N. R. Franks. Collective control without explicit coding: The case of communal nest excavation. *J. Insect Behavior*, 8(4): 417–432, 1995.

[5] B. Digney. Learning and shaping in emergent hierarchical control systems. *Proceedings of Space96 and Robots for Challenging Environments II*, 1996.

[6] P. Dittrich. Chemical computing. In J.-P. Banâtre, J.-L. Giavitto, P. Fradet, and O. Michel, editors, *Unconventional Programming Paradigms (UPP 2004)*, volume 3566 of *LNCS*, pages 19–32. Springer, Berlin, 2005.

[7] P. Dittrich, F. Liljeros, A. Soulier, and W. Banzhaf. Spontanous group formation in the seceder model. *Phys. Rev. Lett.*, 84:3205–8, 2000.

[8] R. Doursat and M. Ulieru. Emergent engineering for the management of complex situations. In *Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems (Autonomics 2008)*, pages 1–10. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[9] J. Dron. Social software and the emergence of control. In *Advanced Learning Technologies, 2006. Sixth International Conference on*, pages 904–908. IEEE, 2006. ISBN 0769526322.

[10] O. Holland and C. Melhuish. Stigmergy, self-organization, and sorting in collective robotics. *Artif. Life*, 5:173–202, 1999.

[11] A. Ishiguro, M. Shimizu, and T. Kawakatsu. Don't try to control everything!: an emergent morphology control of a modular robot. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 981–985. IEEE, 2005. ISBN 0780384636.

[12] M. King. Hieracical structure in emergent control. *Sociometry*, 27(1): 19–24, 1964.

[13] T. Lenser, N. Matsumaru, T. Hinze, and P. Dittrich. Tracking the evolution of chemical computing networks. In S. Bullock, J. Noble, R. A. Watson, and M. A. Bedau, editors, *Proceedings of the Eleventh International Conference on Artificial Life*, pages 343–350. MIT Press, Cambridge, MA, 2008.

[14] D. G. Luenberger. Observers for multivariable systems. *IEEE Trans. Automatic Controll*, 11(2):190–197, 1966.

[15] T. Malsch. Naming the unnamable: Socionics or the sociological turn of/to distributed artificial intelligence. *Artificial Intelligence. Autonomous Agents and Multi-Agent Systems*, 4:200–1, 2001.

[16] J. Martínez and P. González Pérez. Net of multi-agent expert systems with emergent control. *Expert Systems with Applications*, 14(1-2):109–116, 1998. ISSN 0957-4174.

[17] N. Matsumaru, F. Centler, K.-P. Zauner, and P. Dittrich. Self-adaptive scouting - autonomous experimentation for systems biology. In G. R. Raidl, S. Cagnoni, J. Branke, D. W. Corne, R. Drechsler, Y. Jin, C. Johnson, P. Machado, E. Marchiori, F. Rothlauf, G. D. Smith, and G. Squillero, editors, *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, volume 3005 of *LNCS*, pages 52–61, Coimbra, Portugal, 5-7 Apr. 2004. Springer Verlag.

[18] L. Meeden, G. McGraw, and D. Blank. Emergent control and planning in an autonomous vehicle. In *Proceedings of the Fifteenth Annual Meeting of the Cognitive Science Society*, pages 735–740. Citeseer, 1993.

[19] M. Meister, K. Schröter, D. Urbig, E. Lettkemann, H.-D. Burkhard, and W. Rammert. Construction and evaluation of social agents in hybrid settings: Approach and experimental results of the inka project. *Journal of Artificial Societies and Social Simulation*, 10, 2007.

[20] C. Müller-Schloer. Organic computing – on the feasibility of controlled emergence. In *Proceedings of the 2nd IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, pages 2–5. ACM, 2004. ISBN 1581139373.

[21] C. Müller-Schloer and B. Sick. Controlled emergence and self-organization. In R. Würtz, editor, *Organic Computing – Understanding Complex Systems*, pages 81–103. Springer, Berlin, 2008. ISBN 3540776567.

[22] L. Steels. Emergent functionality in robotic agents through on-line evolution. In R. A. Brooks and P. Maes, editors, *Artificial Life IV: Proceedings of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pages 8–16, 1994.

[23] K. Støy. *Emergent control of self-reconfigurable robots*. PhD thesis, University of Southern Denmark, 2003.

[24] K. Støy, W. Shen, and P. Will. Using role-based control to produce locomotion in chain-type self-reconfigurable robots. *Mechatronics, IEEE/ASME Transactions on*, 7(4):410–417, 2002. ISSN 1083-4435.

[25] C. Tschudin and T. Meyer. Programming by Equilibria. In M. Taferl, editor, *15. Kolloquium Programmiersprachen und Grundlagen der Programmierung (KPS'09)*, pages 37–46, 2009.

[26] K. Tsuchiya, K. Tsujita, K. Manabu, and S. Aoi. An emergent control of gait patterns of legged locomotion robots. In *Proc. of the Symposium on Intelligent Autonomous Vehicles*, pages 271–276, 2001.

[27] M. Ulieru and R. Doursat. Emergent engineering: A radical paradigm shift. *Int. J. Autonomous and Adaptive Communications Systems*, 2009.

[28] N. Wiener. *Cybernetics*. The Technology Press; New York, Wiley; Paris, Hermann, 1948.