Soft Robot Control with a Behaviour-Based Architecture

Christopher Armbrust, Lisa Kiekbusch, Thorsten Ropertz, and Karsten Berns

Robotics Research Lab, Department of Computer Science,
University of Kaiserslautern, P.O. Box 3049, 67653 Kaiserslautern, Germany
{armbrust,kiekbusch,ropertz,berns}@cs.uni-kl.de
http://rrlab.cs.uni-kl.de/

Abstract. In this paper, we explain how behaviour-based approaches can be used to control soft robots. Soft robotics is a strongly growing field generating innovative concepts and novel systems. The term "soft" can refer to the basic structure, the actuators, or the sensors of these systems. The soft aspect results in a number of challenges that can only be solved with new modelling, control, and analysis methods whose novelty matches those of the hardware. We will present prior achievements in the area of behaviour-based systems and suggest their application in soft robots with the aim to increase the fault tolerance while improving the reaction to unexpected disturbances.

1 Introduction

Soft robotics has recently received increasing attention by researchers. The aim to create systems that can interact in a more natural way with their environment has led to the development of robots with soft bodies [19], soft actuators [1,14], and even soft sensors [21] that hardly resemble classic, rigid robots. With these novel components, however, come new challenges concerning robot control.

Due to the complexity of the involved components, it is extremely difficult or even practically impossible to create accurate models describing the dynamics of the machines. Hence, the classic, deliberative control approaches are only applicable in a limited way. Purely reactive strategies, however, fail to grasp the full complexity of high-level control. We suggest the use of behaviour-based control systems [2,20], which combine fast reaction times with support for complex tasks. Soft robotic systems typically consist of many interconnected parts that are difficult to model and have to react fast to (unexpected) disturbances—which are also difficult (or impossible) to model. The distributed nature of behaviour-based systems (BBS) perfectly complies with the also distributed structure of soft robots. In typical BBSes there is a high degree of redundancy, which is essential in case of hardware failures, and a significant number of reactive elements. This allows for fast reactions to disturbances.

Using the example of the behaviour-based architecture iB2C[1], we will explain the advantages of behaviour-based approaches over purely reactive or purely

[1] iB2C: integrated Behaviour-based Control

deliberative ones in detail and will sketch our aim to apply them for controlling soft robotic systems. We have structured the remainder of this paper as follows: In Sec. 2, we will introduce the main features of the iB2C and the mechanisms we have invented for supporting the development and analysis of iB2C networks. We will then explain how the iB2C can be used to realise soft control systems in Sec. 3. Finally, in Section 4 we will summarise the main points of this paper and discuss our vision for future work.

2 The Behaviour-based Architecture iB2C

The behaviour-based architecture iB2C [22] has been implemented using the software frameworks MCA2-KL[2] and FINROC[3]. It is applied to different kinds of robots in our lab, e.g. a bipedal walking machine, a humanoid robot head, and several wheel-driven indoor and outdoor vehicles.

The basic component of the iB2C is the behaviour (see Fig. 1(a)), which is defined as $B = (f_a, f_r, F)$, where f_a calculates its *activity vector* $\boldsymbol{a}$ and f_r calculates its *target rating* r. The *output vector* $\boldsymbol{u}$ is transferred from the *input vector* $\boldsymbol{e}$ together with the *activation* $\iota = s \cdot (1 - i)$ using the transfer function $F : \boldsymbol{u} = F(\boldsymbol{e}, \iota)$. The activation indicates the effective relevance of a behaviour in the network. *Stimulation* s and *inhibition* $i = \|\boldsymbol{\imath}\|_\infty$ are signals coming from other behaviours to gradually enable or disable the behaviour.

The activity vector $\boldsymbol{a} = (a, \underline{\boldsymbol{a}})^T$ is composed of the behaviour's activity a and q so-called *derived activities* $\underline{a}_0, \underline{a}_1, \ldots, \underline{a}_{q-1}$ with $\underline{a}_i \leq a$, $\forall i \in \{0, 1, \ldots, q-1\}$. The activity indicates the degree of influence the behaviour wants to have in the network. It is also possible to transfer only a part of the activity to other behaviours using the derived activities. A behaviour's activity is limited by its activation, i.e. $a \leq \iota$. The target rating r describes the satisfaction of a behaviour with the current situation. The four *behaviour signals*, s, i, a, r and the internal behaviour value ι are limited to $[0, 1]$ and describe the interface of each behaviour. By contrast, there is no limitation of $\boldsymbol{e}$ and $\boldsymbol{u}$. They can differ among behaviours.

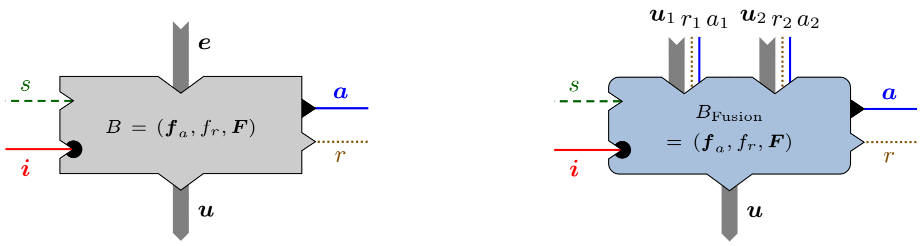
iB2C behaviours can be connected in various ways. The most common types are stimulating and inhibiting connections, in which the activity output of one behaviour is connected either to the stimulation or the inhibition input of another behaviour. Other connection types include the combination of the outputs of a number of competing behaviours using a fusion behaviour (see below) or the sequencing of behaviours using the special coordination behaviour CBS [4].

The iB2C fusion behaviour (see Fig. 1(b)) combines the outputs of several behaviours connected to it according to one of three possible fusion modes (maximum, weighted average, and weighted sum). For example, if p competing behaviours B_{Input_c} with activities a_c, target ratings r_c, and output vectors $\boldsymbol{u}_c$ $(c \in \{0, \ldots, p-1\})$ are connected to a weighted average fusion behaviour B_{Fusion}, then the outputs of B_{Fusion} are calculated as follows:

---

[2] MCA2-KL: Modular Controller Architecture Version 2 - Kaiserslautern Branch

[3] FINROC is the successor of MCA2-KL. See `http://finroc.org/` for more information.



(a) The general symbol of a behaviour (see text for description of ports).

(b) The symbol of a weighted average fusion behaviour for $n_c = 2$ competing behaviours.

Fig. 1. The symbols of two basic iB2C behaviours.

$$\boldsymbol{u}_{\text{Fusion}} = \frac{\sum_{j=0}^{p-1} a_j \cdot \boldsymbol{u}_j}{\sum_{k=0}^{p-1} a_k}, \quad a_{\text{Fusion}} = \frac{\sum_{j=0}^{p-1} a_j^2}{\sum_{k=0}^{p-1} a_k} \cdot \iota_{\text{Fusion}}, \quad r_{\text{Fusion}} = \frac{\sum_{j=0}^{p-1} a_j \cdot r_j}{\sum_{k=0}^{p-1} a_k}$$

The handling of a large number of interconnected behaviours is facilitated by behavioural groups, which encapsulate a number of behaviours or further groups and act as new behaviours in a network. To fulfil complex tasks, networks of simple behaviours are constructed and possibly combined. The challenge lies in the connection of these behaviours. As example, the control system of the biped (see Sec. 3) consists of over 350 behaviours, while the one of RAVON (see also Sec. 3) contains even more than 500 behaviours. To build up such huge networks, sound guidelines for the development and implementation are needed together with verification techniques to prove a system's correctness.

Fig. 2 gives an overview of the concept developed in our lab for the development and verification of behaviour networks. Soft robotic systems are typically complex and consist of many interconnected components. This makes them perfect for the application of BBSes, but this in turn results in complex software. In Sec. 2.1, we propose the use of a strict design concept for the development of behaviour networks that realise high-level, complex tasks. In such strongly connected systems, oscillations can easily occur. We therefore suggest a solution for oscillation detection in Sec. 2.2. Distributed systems like the ones found in soft robotics call for verification techniques that specifically take into account their high degree of distribution. We will present two such techniques in Sec. 2.3.

2.1 Design of Complex Behaviour Networks

As we have alluded in Sec. 1, soft robotic systems can be complex in many respects. The complexity of their hardware is reflected by the complexity of

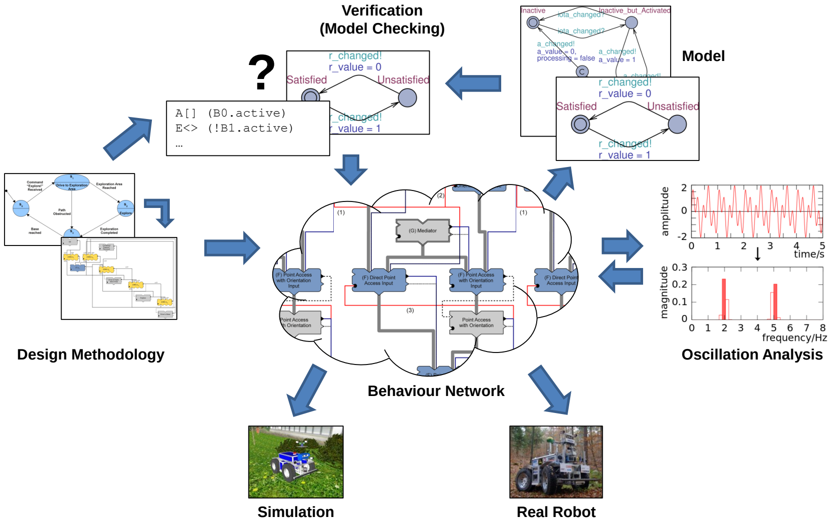


Fig. 2. Concept of the development and verification process for BBSes developed in our lab.

the software necessary to control a soft robot. A drawback that can often be seen in behaviour architectures is the lack of support for building large networks that are able to execute sophisticated tasks. For example, Brook's subsumption architecture [12] has been criticised for being unable to support large networks.

The ability to combine iB2C behaviours into a group (see above) provides significant help in dealing with the complexity of large networks. However, research conducted at our group has shown that the quality of a behaviour network often heavily depends on the experience of the developers and their personal preferences. Guidelines [22] can mitigate this problem, but target at developers with a certain experience in designing iB2C networks. Therefore, we have invented a three-step method for supporting completely inexperienced developers during the process of designing an iB2C network for a complex task [9].

The first step is to define a complex task as a finite-state machine (FSM). Two persons are usually needed for this step: an *end-user* with detailed knowledge about the task the system shall perform and the *main developer* with detailed knowledge about existing software components and the specification of the available hardware. As soft robotics is a new field compared to industrial robotics, it is highly likely that an end-user does not have much knowledge about the capabilities of a soft robot. Hence, he shall be able to define the task in a common, nearly intuitive way while being assisted by the main developer.

The second step consists of an automatic transformation of said FSM into the skeleton of a behaviour network, i.e. a behaviour network that only contains fusion and CBS behaviours (see above), empty behaviours without functionality, and the interconnections between the behaviours. The advantage of having an automatic process here is that no person is involved. Hence, the resulting behaviour network will correspond to the FSM defining the robot's task, while a manual mapping process could easily lead to errors in the network structure.

In the third step, so-called *system specialists* manually add the core functionalities of the behaviours in the previously created skeleton network. They have detailed knowledge about specific parts of the soft robotic system and are therefore able to implement sub-components. Due to the distributed nature of BBSes, this work can easily be done in parallel by several system specialists who are supervised by the main developer. An interesting aspect here is that none of the system specialists has to be an expert for behaviour networks as the network structure has already been created automatically in the second step.

2.2 Oscillation Detection in Behaviour Networks

Oscillations are a typical effect in soft robotics applications. They can be enforced by the control system to perform a special movement, e.g. the walking of a biped robot or fin movements of fish robots. But they can also be highly unwanted, for example if they appear in flexible joints and complicate a correct adjustment. The detection of oscillations is therefore a very important task. In [25] we presented an approach to detect oscillations inside a behaviour-based control system during run-time, which facilitates an early reaction in case of a wrong behaviour. The oscillation detection method is based on the analysis of the frequency spectrum of an arbitrary Fourier-transformed signal. In our approach, we used the activity data of the behaviours. In short, the data is buffered, transformed, and analysed for peaks in the power spectrum indicating an oscillation. In a second step, we traced oscillations through the network to gain an overview of the path the oscillation takes through the network and to find its root cause. Future work includes the definition of desired and undesired oscillations and possible reactions based on the underlying application.

2.3 Verification of Behaviour Networks

We have already mentioned several advantages of using BBSes for soft robots and will go into detail about that in Sec. 3. A key aspect of BBSes is the distribution of the overall functionality over several components of the system. Unfortunately, this advantage comes with a downside: Determining whether a system really does what it is supposed to do can be hard as a considerable part of the intelligence of a BBS lies in its network structure, i.e. in the interaction of its behaviours. Determining whether a system operates as specified is done by methods of formal verification, e.g. deductive reasoning [15,16] as well as model checking [13,23]. With regard to the fact that a lot of intelligence of a BBS lies in its network, we have decided to pursue a top-down analysis approach by developing a verification technique that is especially tailored to the mostly neglected analysis of the network structure [5,7,6]. We have based our approach on model checking as this offers a high degree of automation and generates witnesses and counter-examples, respectively.

The idea underlying our verification concept is the modelling of iB2C networks as networks of synchronised timed automata using the model checking toolbox Uppaal [11]. Each behaviour is represented by five automata, one for each behaviour signal and one for the activation, as shown in Fig. 3. If there is a connection between the signals of two behaviours, a synchronisation channel is used to connect and synchronise the corresponding automata.

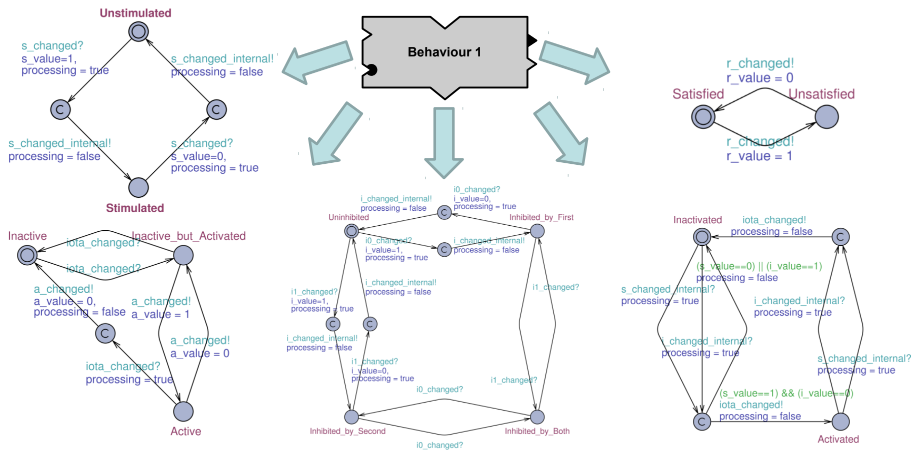


Fig. 3. Modelling of a behaviour interface using five automata for the behaviour signals stimulation, inhibition, activity, and target rating as well as for the activation.

Using a graphical interface, we can define properties that shall hold for the behaviour signals, e.g. that one behaviour shall get active before another one does. The graphical definition of properties can be automatically transformed to a corresponding set of observer automata and queries, which can be sent to Uppaal's verifier together with the system model. With the help of the verifier, we can then easily check whether our BBS holds the property in question and thus fulfils requirements like "the anti-collision behaviour has precedence over the driving behaviour, i.e. it can stop the robot". In recent work, we also take into account hardware failures during the verification process [17].

The use of timed automata for modelling BBSes offers several advantages like the ability of checking temporal properties (e.g. reachability) with the downside of requiring a strong abstraction in order to keep the verification process computationally feasible. By focusing on discrete system states and abstracting from state transitions, satisfiability modulo theories (SMTs) can be used to generate a more fine-grained model, which allows for checking richer properties concerning the discrete state as described in [24]. SMTs study practical methods to solve first-order logic formulae with equality in which sets of variables are replaced by predicates of the underlying theories. Examples for these background theories are the theory of real numbers, of integers, and of various data structures. The *SMT-LIB* [10] initiative has defined a standard for descriptions of background theories used in SMT systems. Due to the support of limited non-linear real arithmetic, SMT perfectly matches the expressiveness required for modelling the behaviour interaction and coordination. The rather simple modelling process allows for automation like the modelling based on timed automata, with the advantage of having a weaker abstraction concerning the behaviour signals.

As we have explained in Sec. 1, soft robotic systems are very complex and it is extremely difficult to create accurate models, which makes it hard to develop suitable control systems and find appropriate control parameters. The latter verification approach is also suitable for reducing the search space for parameter identification. Therefore, properties that the overall system shall be revealing can be used to determine value ranges of parameters which guarantee them.

3 Soft Control with the iB2C

In the following, we will illustrate how fusion behaviours executing a weighted average fusion (see Sec. 2) can be used to realise a seamless, hence soft transition from one controlling behaviour to another.

In the control system of the autonomous off-road vehicle RAVON [3], a large number of connected behaviours controls the robot's movement. The control system also handles the operator's steering commands, which influence the robot in different ways depending on which degree of autonomy (pure tele-operation, assisted tele-operation, full autonomy) the operator has chosen. While during pure tele-operation the operator's commands completely bypass the robot's anti-collision system, in the other two modes they are combined with the outputs of the safety system. What distinguishes RAVON's control system from others is that during assisted tele-operation or full autonomy, the operator can choose his level of influence on the vehicle's motion in a seamless fashion.

Figure 4 depicts a part of the behaviour network. The figure illustrates how an operator command is combined with outputs of p high-level navigation behaviours. The behaviour receiving the operator command (from any kind of input device) is connected via two streams with the remainder of the network: It sends the operator command along with activity and target rating down to a fusion behaviour of the lower layer and uses its activity to inhibit the fusion behaviour of the higher layer, which combines the outputs of high-level navigation behaviours. As a result, there is no binary selection of which commands are sent to the lower layers. Instead, the network can perform a soft transition from using only the high-level components over using a combination of them and the operator command to using only the latter. In [8], we provide details.

For complex soft robotic systems that shall be able to interact with their environment in a novel fashion, the ability to perform gradual, seamless transitions between two sources of control is essential. The soft integration of operator commands is only one example. Another one is a soft reaction of a robot's anti-collision system to nearby hazards: Instead of simply stopping the robot's

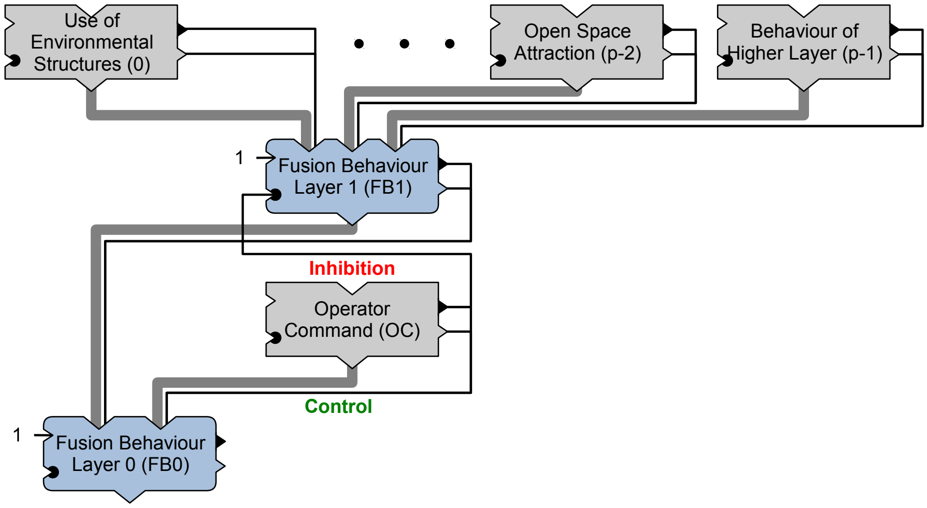
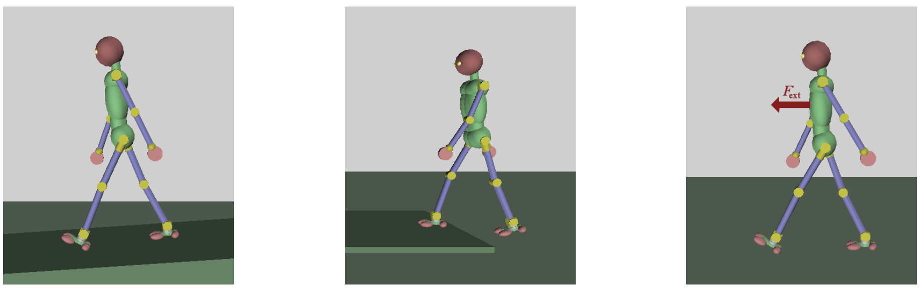


Fig. 4. The operator's commands inhibit commands of higher layers and are sent to lower layers.

motion, a sophisticated system will first try to move the robot around the hazard in a soft fashion. Such a functionality is also realised in RAVON's control system and allows for fast and smooth reactions to environmental disturbances.

With regard to its hardware, RAVON is a typical representative of classic (i.e. stiff) robots. The only soft hardware components are its spring-mounted bumpers. But the iB2C is also used to control a robotic system with compliant actuators, namely a simulated bipedal robot [18]. Its control system is inspired by the human locomotion system. The robot is able to perform human-like walking and can properly react to environmental disturbances (see Fig. 5).



(a) (b) (c)

Fig. 5. The simulated biped reacting to different types of disturbances: downhill slope ((a)), step ((b)), external force acting on its torso ((c)) (source: [18]).

4 Conclusion and Future Work

In this paper, we have motivated the application of BBSes in soft robotics and explained which techniques offer help in developing or analysing a behaviour network. As we have shown in Sec. 3, the behaviour architecture iB2C is perfectly suited for supporting soft interactions of different sources of control like operator commands or outputs of an anti-collision system. With this, we have demonstrated the applicability of the iB2C in novel control systems for soft robots like a bipedal robot with compliant actuators. In the context of future work, our research group is going to improve the development and analysis techniques described above, invent new ones in order to face the challenges of soft robotics, and finally apply them to soft robots.

Acknowledgement Parts of the research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 285417.

References

1. Albu-Schaeffer, A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimboeck, T., Wolf, S., Hirzinger, G.: Soft robotics. IEEE Robotics and Automation Magazine 15(3), 20–30 (September 2008)
2. Arkin, R.: Behaviour-Based Robotics. MIT Press (1998), ISBN-10: 0-262-01165-4; ISBN-13: 978-0-262-01165-5
3. Armbrust, C., Braun, T., Föhst, T., Proetzsch, M., Renner, A., Schäfer, B.H., Berns, K.: RAVON – the robust autonomous vehicle for off-road navigation. In: Baudoin, Y., Habib, M.K. (eds.) Using robots in hazardous environments: Landmine detection, de-mining and other applications, chap. RAVON – The Robust Autonomous Vehicle for Off-road Navigation. Woodhead Publishing Limited (2010), ISBN: 1 84569 786 3; ISBN-13: 978 1 84569 786 0
4. Armbrust, C., Kiekbusch, L., Berns, K.: Using behaviour activity sequences for motion generation and situation recognition. In: Proceedings of the International Conference on Informatics in Control, Automation and Robotics (ICINCO). pp. 120–127. Noordwijkerhout, The Netherlands (July 28-31 2011)
5. Armbrust, C., Kiekbusch, L., Ropertz, T., Berns, K.: Verification of behaviour networks using finite-state automata. In: Glimm, B., Krüger, A. (eds.) KI 2012: Advances in Artificial Intelligence. Springer, Saarbrücken, Germany (September 24-27 2012)
6. Armbrust, C., Kiekbusch, L., Ropertz, T., Berns, K.: Quantitative aspects of behaviour network verification. In: Zaiane, O., Zilles, S. (eds.) Proceedings of the 26th Canadian Conference on Artificial Intelligence. Lecture Notes in Computer Science, vol. 7884. Springer, Regina, Saskatchewan, Canada (May 28-31 2013)
7. Armbrust, C., Kiekbusch, L., Ropertz, T., Berns, K.: Tool-assisted verification of behaviour networks. In: Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA 2013). Karlsruhe, Germany (May 6-10 2013)
8. Armbrust, C., Proetzsch, M., Schäfer, B.H., Berns, K.: A behaviour-based integration of fully autonomous, semi-autonomous and tele-operated control modes for

an off-road robot. In: Proceedings of the 2nd IFAC Symposium on Telematics Applications. IFAC, Politehnica University, Timisoara, Romania (October 5-8 2010), invited paper
9. Armbrust, C., Schmidt, D., Berns, K.: Generating behaviour networks from finite-state machines. In: Proceedings of the German Conference on Robotics (Robotik) (May 22–25 2012)
10. Barrett, C., Stump, A., Tinelli, C.: The satisfiability modulo theories library (smt-lib). `www.SMT-LIB.org` (2010)
11. Behrmann, G., David, A., Larsen, K.G.: A tutorial on uppaal 4.0 (November 28 2006), revised and extended version of "A Tutorial on Uppaal"
12. Brooks, R.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation RA-2(1), 14–23 (April 1986)
13. Clarke, E.M., Emerson, E.A.: Design and synthesis of synchronization skeletons using branching time temporal logic. In: Kozen, D. (ed.) Logics of Programs - Workshop. Lecture Notes in Computer Science (LNCS), vol. 131, pp. 52–71. Springer Berlin Heidelberg (1982)
14. Deimel, R., Brock, O.: A compliant hand based on a novel pneumatic actuator. In: Proceedings of the IEEE International Conference on Robotics and Automation 2013 (ICRA 2013). pp. 2039–2045 (May 6-10 2013)
15. Floyd, R.W.: Assigning meanings to programs. In: Schwartz, J.T. (ed.) Mathematical Aspects of Computer Science. Proceedings of Symposia in Applied Mathematics, vol. 19, pp. 19–32. American Mathematical Society, Providence, Rhode Island, USA (1967)
16. Hoare, C.A.R.: An axiomatic basis for computer programming. Communications of the ACM 12(10), 576–583 (October 1969)
17. Kiekbusch, L., Armbrust, C., Berns, K.: Formal verification of behaviour networks including hardware failures. In: Proceedings of the 13the International Conference on Intelligent Autonomous Systems (IAS-13). Padova, Italy (July 15-19 2014)
18. Luksch, T.: Human-like Control of Dynamically Walking Bipedal Robots. RRLab Dissertations, Verlag Dr. Hut (2010), ISBN: 978-3-86853-607-2
19. Marchese, A.D., Onal, C.D., Rus, D.: Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators. Soft Robotics 1(1), 75–87 (March 2014)
20. Matarić, M.J., Michaud, F.: Behaviour-based systems. In: Siciliano, B., Khatib, O. (eds.) Springer Handbook of Robotics, chap. 38, pp. 891–910. Springer Berlin Heidelberg (2008)
21. Park, Y.L., rong Chen, B., Wood, R.J.: Soft artificial skin with multi-modal sensing capability using embedded liquid conductors. In: Proceedings of the IEEE Sensors 2011 Conference. pp. 81–84. IEEE, Limerick, Ireland (October 28-31 2011)
22. Proetzsch, M.: Development Process for Complex Behavior-Based Robot Control Systems. RRLab Dissertations, Verlag Dr. Hut (2010), ISBN: 978-3-86853-626-3
23. Queille, J.P., Sifakis, J.: Specification and verification of concurrent systems in CESAR. In: Dezani-Ciancaglini, M., Montanari, U. (eds.) International Symposium on Programming - Proceedings of the 5th Colloquium. Lecture Notes in Computer Science (LNCS), vol. 137, pp. 337–351. Springer-Verlag, London, UK (1982)
24. Ropertz, T., Berns, K.: Verification of behavior-based networks - using satisfiability modulo theories. In: Proceedings for the joint conference of ISR 2014 and ROBOTIK 2014. pp. 669–674. VDE VERLAG GMBH (2014)
25. Wilhelm, L., Proetzsch, M., Berns, K.: Oscillation analysis in behavior-based robot architectures. In: Dillmann, R., Beyerer, J., Stiller, C., Zöllner, J., Gindele, T. (eds.) Autonome Mobile Systeme. pp. 121–128. Informatik aktuell, Springer (2009)