Distributed Fault-Tolerant Robot Control Architecture Based on Organic Computing Principles

Adam El Sayed Auf, Marek Litza, Erik Maehle

University of Lübeck, Institute of Computer Engineering, Ratzeburger Alle 160, 23538 Lübeck, Germany
{elsayedauf, litza, maehle}@iti.uni-luebeck.de

Abstract   Walking animals like insects show a great repertoire of reactions and behaviours in interaction with their environment. Moreover, they are very adaptive to changes in their environment and to changes of their own body like injuries. Even after the loss of sensors like antennas or actuators like legs, insects show an amazing fault tolerance without any hint of great computational power or complex internal fault models. Our most complex robots in contrast lack the insect abilities although computational power is getting better and better. Understanding biological concepts and learning from nature could improve our approaches and help us to make our systems more "life-like" and therefore more fault tolerant. This article introduces a control architectural approach based on organic computing principles using concepts of decentralization and self-organization, which is demonstrated and tested on a six-legged robotic platform. Beside explaining the organic robot control architecture, this study presents a leg coordination architecture extension to improve the robustness and dependability towards structural body modifications like leg amputations and compares experimental results with previous studies.

Introduction

Legged locomotion encounters challenges like controlling the leg's different phases with unequal responsibilities, coordinating legs to achieve an effective walking pattern as well as adapting the achieved gait to the environmental circumstances and a given task. In contrast to a wheeled rotary motion, a leg passes through a stance phase, carrying part of the systems weight and stemming it to a given direction, and a swing phase, lifting up the leg and moving it in the opposite direction to put it down again. The precondition for a stance phase is the leg's ground contact and must be obtained at the end of each swing phase. All legs of a

walking system accomplishing alternating stance and swing phases must be coordinated to a stable walking pattern. In the animal kingdom, these tasks are solved by each single walking creature. Even insects that are alleged to have low "computational power" cope with these tasks. Animals use smooth transitions of their gaits instead of hard switches. Horses do not switch between standing on a feedlot, trotting and galloping but use a smooth changeover between different gaits depending on the animal's velocity, equally insects do. That leads to the assumption that controlling walking patterns does not base on a set of predefined gaits, but the use of general rules providing the ability of adaptation. Taking into account the mentioned challenges are passed by each walking animal, the question arose if a central or decentral control is underlying legged locomotion. The present work introduces an organic computing approach facing the challenges with decentralised and self-organizing mechanisms by the use of a six-legged walker.
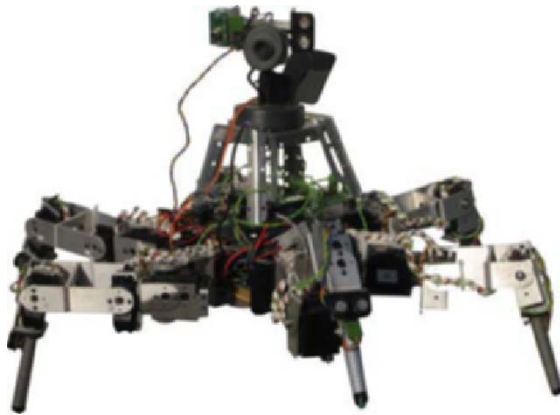

Figure 1: The Robot Platform OSCAR.

The Robot Platform

OSCAR (Organic Self Configuring and Adapting Robot) is a six-legged walking machine with 18 degrees of freedom (DOF). Its servo motors as well as its skeleton parts are commercially available. Set together, the parts result in a symmetric robot with a round body and a span length of 74.5 cm over completely stretched out legs (Fig. 1). Each leg has three degrees of freedom and is equipped with a binary contact sensor. Its head consists of a wireless camera, an ultra sonic sensor and a heat sensor, altogether movable with one joint. While the binary contact sensor signals if the leg is touching the ground or not, the camera sends a color picture to a personal computer, the ultrasonic sensor measures the distance between OSCAR and an obstacle and the heat sensors detects heat sources like

humans. The onboard control hardware, also commercially available, consists of a SD21 Servo Driver Module and a JControl/SmartDisplay, programmable in Java. All parts are connected by an I2C-Bus.
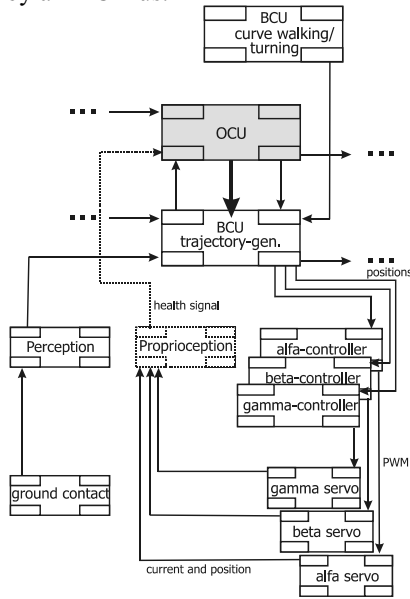
leg i

Figure 2: Example for the Organic Robot Control Architecture ORCA for one leg.

The Software Architecture

The Organic Robot Control Architecture - ORCA – was designed to be modular and hierarchically organized in order to be easily manageable and to fit to the control architecture of a robot which is also assumed to be modular, hierarchically organized and behaviour-based [1,2]. An ORCA-based system is built using Basic Control Units (BCUs) to achieve the desired functionality. Each BCU encapsulates a specific functionality. It can for example implement generic signal filtering modules or PID-controllers, but may also encapsulate sensor or actor hardware of the robot at lower system levels. BCUs interact by interchanging data (signals). The connections between BCUs can also be used to trigger activity in the receiving BCU when new data are sent. Organic Control Units (OCUs) supplement these BCUs. An OCU uses the same unified interface that BCUs use to interchange data and trigger activities. In contrast to BCUs, OCUs do not realize a predefined custom function for the robot, but they monitor the signals generated by

one or more BCUs. When these monitored signals show a substantial change, the OCU can react by changing parameters of the BCUs. By defining "normal" or "good" ranges for some of the signals a BCU generates, an OCU can decide when to start changing parameters to bring the system back to a normal or "healthy" state (reasoner). Further, the BCUs shall be able to learn in order to improve their reactions in similar situations in the future by making use of a memory for a short-time history. So the BCUs manage single basic functions in close collaboration with other BCUs, which leads to an emergent collective functionality, while the OCUs ensure the system's robustness and dependability by monitoring single or several BCUs and reacting to stronger changes or failures. As well as BCUs, OCUs are modular and locally distributed with the ability to communicate with each other.

Self-Organizing Walking Patterns

Based on the ORCA approach and inspired by biological experiments and controllers like the Walknet [4] a decentralized controller is used to achieve organic walking patterns. Each of the six legs consists of three joints and has its own controller implemented as a separate BCU for trajectory generation (Fig.2). These BCUs generate position commands for the BCUs of the three leg joints (alpha, beta, and gamma-controller) which then send PWM (Pulse Width Modulation) signals to the respective servos.

Controlling one leg means managing an alternation composed of swing and stance phase. In the swing phase the leg is lifted up from the ground, moved to the front, and put down on the ground again. The ground contact signal triggers the stance phase when the leg moves backwards in respect to the body. In this phase it carries a part of the body's weight and pushes it to the front. To obtain an alternation between swing and stance phases, the trajectory controller uses the information of the ground contact signal and two extreme positions: the posterior extreme position (PEP) and the anterior extreme position (AEP). If a leg has reached the PEP in its stance phase, it switches into swing phase. So the leg will lift up and move along a fixed trajectory towards the AEP. After reaching the AEP and receiving the ground contact signal, the leg switches again into the stance phase and moves back towards the PEP. This principle leads to a swing-stance alternation. The frequency of this alternation can be changed by increasing or decreasing the duration of the stance phase only.

Each of the six trajectory BCUs uses the same flexible coordination rule to harmonize its movements with the neighbouring leg controllers' swing phases. The ground contact signals are processed by a perception BCU and then sent to the trajectory BCU which passes it to the trajectory BCUs of the two neighbouring legs. The leg controllers inhibit switching from the stance into the swing phase, when its two neighbouring BCUs are not sending a ground contact signal. This rule en-

sures that a maximum number of three legs are swinging at the same time. Thus, the robot's legs are always in a stable configuration. Combination of local alternating leg movements and the mentioned coordination rule lead to a global organic walking behaviour. Different walking patterns arise in a self-organizing way by increasing or decreasing the stance phase's duration without using a pre-programmed gait. In contrast to the stance phase the swing phase, as was observed in the stick insect Carausius morosus, has a fixed duration and can not be varied. So, the gait is adapting to the robot's walking velocity in a self-organizing way. In a very slow gait, achieved by usage of a stance phase about five times longer than the swing phase, a pentapod gait can be observed. So, five feet are on the ground while just one leg is swinging. Shortening the stance phase until it is as long as the swing phase, a tetrapod and a tripod result smoothly [5]. These gaits are observable in nature as well and biological experiments have shown that insects move in a very similar way and do not make use of a hard switching between their gaits.

Turning and Curve Walking

One advantage of OSRAR's symmetric body is its ability to rotate around its centre without taking more space than in a standing posture. In contrast to walking the legs have not to be separated in left and right legs, moving in opposite direction in respect to the body centre, during rotating all legs move into the same direction. A rotating of the robot is achieved by one BCU per leg setting the AEP and the PEP of each leg on the same value. So changing two parameters while maintaining the coordination rule leads to a completely different behaviour.

Another challenging behaviour is curve walking. One simple approach to achieve curve walking is to shorten the stance trajectories of the three legs at the inner side of the curve and enlarging the stance trajectories of the three legs at the outer side. Diverging the PEP and AEP of the outer legs and converging the PEP and AEP of the inner legs lead to a curve walking. Converging the PEP and AEP until they change sides and diverge again, leads in the end to the above described turning behaviour. Thus, one BCU per leg is sufficient to produce a smooth transition from straight walking over curve walking to turning by changing the AEP and PEP values of the legs.

Another approach of curve walking, which can also be observed in experiments with the stick insect Carausius morosus [3], is to change the trajectory of each leg independently. Further biological experiments and computer simulations showed that decentralized control architecture combined with local rules lead to curve walking based on individual stance trajectories for each leg without explicit calculations [6]. In the ORCA project a mixture of both approaches is used. For walking in a curve, the front leg's PEP and AEP are shifted in the above described way. This shift of the extreme positions is detected by the next posterior legs and leads also in a shift of their extreme positions. So a change in the front leg extreme posi-

tions leads to a shift in the middle leg extreme positions and that leads to a shift in the hind leg extreme positions. Thus, a change in the front legs is spread by local rules through the whole system. The intensity of the middle and hind leg shifts depend on the intensity of their anterior leg shifts. In the following experiments front leg extreme position shifts are caused by a BCU that guides the robot to walk towards a detected stimulus. In the presented experiments this stimulus is a heat source, which can be detected by the heat sensor at the robot's head.

The System's Fault Tolerance

To show the system's robustness and dependability a strong structural body modification was realised by a leg deactivation, which was achieved by clicking the affected leg upwards and holding it above the robot's body. So a leg amputation is simulated, whereas in contrast to a real amputation the leg's weight is still being carried by the walking machine.

A deactivated leg is recognised by an OCU because of the missing feedbacks like ground contact or servo feedback. An OCU detecting a non-functional leg will modify the responsible trajectory BCU in a way that it is channelling its neighbouring legs the ground contact signals through. So the next leg waiting for the ground contact signal of the defect leg will receive the ground contact signal of the next functional leg. Thereby, the self-organized coordination of the remaining legs is continued and the walking pattern adapts to the system's new structure [5].

Experiments and Results

In previous experiments with the above mentioned software architecture the robot's walking abilities were shown. Beside the self-organising walking pattern and the robot's capability to handle leg damages during straight forward walking with an emergent gait [6], the walking machine's curve walking was tested in [7].

In [7] was shown that during a leg amputation the robot is still able to navigate towards a given goal position without any modifications in the robot's control software. But, without any changes of the machine's walking behaviour it is hindered in its stability during curve walking by the missing leg. The robot needs up to twice as long for reaching its goal position as in a "healthy" state with all six legs [7].

Coordination Extension

To remain true to the decentralised control architecture on the one hand and improve the robots walking behaviour in a certain situation on the other hand, the leg observing OCU was modified. The OCU of an amputated leg gives additional information to its neighbouring legs' OCUs. A neighbour leg changes its swing start conditions to make the emerging gap caused by a leg amputation as small as possible. The leg is waiting for its neighbours to come as close as possible in their stance phase before starting its own swing phase. Its neighbours' positions are depending on their neighbouring legs' swing directions. To give an example (FIG. 3a), the robot's legs are numbered clockwise when regarded from above. The neighbour leg ready to swing is called leg i and its functional neighbours are called i-1 and i+1. In case that leg i-1 is swinging clockwise (cw) and leg i+1 is swinging anti-clockwise (acw) leg i will start its swing phase when leg i-1 and leg i+1 are near their AEPs. Another example with the opposite case is given in Fig. 3b. When the front left leg (FL) i+1 swings clockwise and the hind right leg (HR) i-1 swings anti-clockwise the hind left leg (HL) i has to wait until its neighbours are close to their PEPs. All four possible situations of the explained example are shown in Tab. 1.
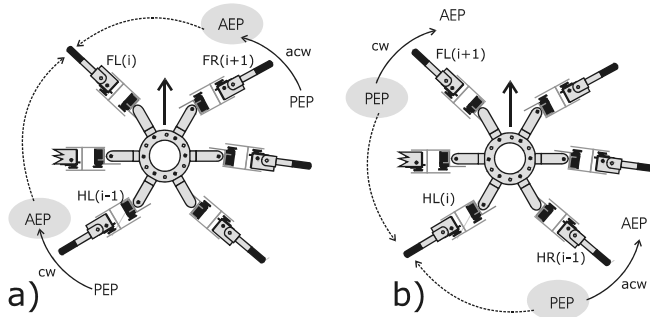

Figure 3: Example for the coordination extension. a) Front left leg (FL) as neighbour of a damaged leg is allowed to swing when its functional neighbours, front right leg (FR) and hind left leg (HL), are near their AEPs. b) Hind left leg (HL) as neighbour of a damaged leg is allowed to swing when its neighbours, front left leg (FL) and hind left leg (HL), are near their PEPs.

Table 1: Position areas in stance phase of the functional neighbours of leg i which are the preconditions for leg i to start its swing phase depending on the neighbouring leg's swing direction (anticlockwise acw or clockwise cw).

	i-1		i+1	
swing direction	acw	cw	acw	cw
area around	PEP	AEP	AEP	PEP

Experimental Setup

For being able to compare the experiments of this work to previous ones, the experimental setup is the same as in [7].

The setup is a 230x160 cm flat indoor-area in which the robot can freely walk. The Robot is equipped with two markers, one at its centre on top of his head and one at its front between the two front legs. The markers horizontal distance is 10 cm and the vertical 15 cm. A third marker is used for the heat source, also placed in the scene. The distance between the robot's centre marker and the position a foot touches the ground is varying depending on the robot's leg posture from 18 to 26 cm. The whole scenario is captured by a standard camera 310 cm above the ground. The results in [7] have shown the leg amputation's influence on the robot's curve walking behaviour. In two situations the robot is significantly hindered needs nearly twice as long for reaching its goal than in all other runs. These situations were a left curve walk with a right middle leg amputation and a right curve walk with a left middle leg amputation. In both cases the amputation causes a turning towards the hindered body side. The following experiments concentrate on these two critical situations. Two robot heat source configurations have been tested: one left middle leg amputation in a right curve walk and one right middle leg amputation in a left curve walk.

By rotating its head the robot scans its environment for heat sources. One of the results is shown in Fig.4. The coordination system in Figure 4 shows on the x and y-axes the robot's position in the indoor-area in cm. The robot's centre is marked as a circle with an arrow giving the robot's orientation, while the target heat source is represented as a diamond. Over time recorded positions of the robot result in the six-legged machine's walking path and are marked with additional timestamps. Time t=0 as well as the time passed until the robot reached its target and the time when the robot detects the heat source are given in Fig. 4. About the x-Position 120 the robot's right middle leg is amputated and at time 10 s the heat source is detected. Its orientation shifts towards the heat target while at about x-position 120 the robot's right middle leg was amputated to test the controller extension. Although the robot is still hindered by the missing leg that has to be carried with him, it reaches the heat source in a shorter time than in previous experiments. The time the hexapod reaches its target (63 s) is not comparable par to par because of the different distances that the robot walked in Fig.4 and Fig.5. Although the distance walked by the robot is longer in the previous runs (Fig.5), in three of five cases the walking machine reaches the heat source in a significant shorter time than in previous runs without the coordination extension. In the other two cases the robots does not find the heat source and misses its target.

Fig. 5 shows the result with the same setup and a robot using the not extended control architecture. In comparison Fig. 5 shows a stronger influenced walking path than it can be seen in Fig. 4. Using the control architecture without any additional information like it was done in previous works the robot reaches also its

goal, but the above mentioned decentral extension to the control system, based on local rules between the neighbouring legs can improve the robots abilities as well as its robustness.
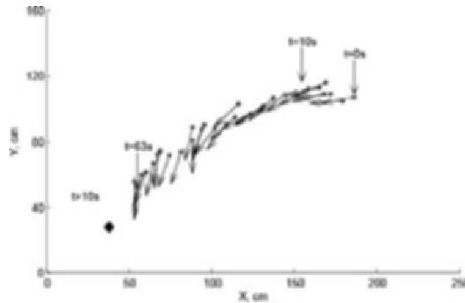

Figure 4: Robot's walking path shown as positions(XY). Robot's centre shown as circle, orientation shown as arrow, heat source as diamond. At time 10 s the robot detects the heat source and needs 63 s to reach the target. A right middle leg amputation [ra] was triggered at X-Position 120.
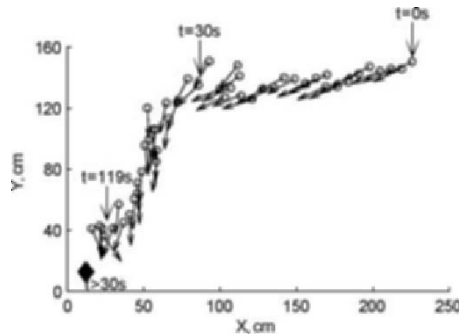

Figure 5: Result from a previous experiment when the robot uses the original control architecture without the extension. Robot's walking path shown as positions(XY). Robot's centre shown as circle, orientation shown as arrow, heat source as diamond. At time 30 s the robot detects the heat source and needs 119 s to reach the target. A right middle leg amputation [ra] was triggered at X-Position 120.

Summary

The introduced approach for controlling the six-legged walking machine OSCAR shows a complex curve walking behaviour with six legs handled by a decentralized control architecture based on local rules by changing a few parameters. Moreover, it is explained how continuous changing of parameters can induce a smooth transition between different walking behaviours and how a decentralized control architecture can be extended by additional local rules to improve the systems abilities. The presented experiment shows the improvement to the system's fault tolerance and robustness towards strong damages like leg amputations by a decentral extension of the control architecture and how the system is able to handle even strong damages while maintaining complex behaviours by adapting in a dependable way.

Beside the handling of strong damages it is open how to give the system the ability to detect smaller defects like loose screws or broken gears in its servo motors and if it is possible to compensate them.

Acknowledgments

References

1) Brockmann W, Großpietsch K.-E, Maehle E, Mösch F: ORCA - Eine Organic Computing-Architektur für Fehlertoleranz in autonomen mobilen Robotern. Mitteilungen der GI/ITG-Fachgruppe Fehlertolerierende Rechensysteme, Nr. 33, 3-17, St. Augustin 2006

2) Brockmann, W, Maehle E, Mösch F: Organic Fault-Tolerant Control Architecture for Robotic Applications. 4th IARP/IEEE-RAS/EURON Workshop on Dependable Robots in Human Environments, Nagoya University/Japan 2005

3) Dürr V, Ebeling W: The behavioural transition from straight to curve walking: kinetics of leg movement parameters and the initiation of turning. The Journal of Experimental Biology 208, 2237-2252, 2005

4) Dürr V, Schmitz J, Cruse H: Behaviourbased modelling of hexapod locomotion: Linking biology and technical application. Arthropod Structure and Development, 33 (3), 237-250, 2004

5) El Sayed Auf A, Mösch F, Litza M: How the Six-legged Walking Machine OSCAR Handles Leg Amputations. Proceedings of the Workshop on Bio-Inspired Cooperative and Adaptive Behaviours in Robots at the SAB IX, Rome 2006

6) Rosano H, Webb B: The control of turning in real and simulated stick insects. Proceedings of the Ninth International Conference on the Simulation of Adaptive Behaviour, Lecture Notes in Artificial Intelligence volume 4095, (2006)

7) El Sayed Auf, A; Larionova, S.; Litza, M.; Mösch, F.; Jakimovski, B.; Maehle, E.: Ein Organic Computing Ansatz zur Steuerung einer sechsbeinigen Laufmaschine. AMS, 233-239, Springer-Verlag, Berlin Heidelberg 2007