CONTROL OF A BIO-INSPIRED FOUR-LEGGED ROBOT FOR EXPLORATION OF UNEVEN TERRAIN

Dirk Spenneberg[1], Stefan Bosse[2], Jens Hilljegerdes[2], Frank Kirchner[1,2], Andreas Strack[2], and Heiko Zschenker[2]

(1) DFKI Robotics Lab
Robert-Hooke-Str. 5, D-28359 Bremen, Germany

(2) University of Bremen, Robotics Lab
Faculty of Mathematics and Computer Science
Robert-Hooke- Str. 5, D-28359 Bremen, Germany
dirk.spenneberg@dfki.de

Abstract

This paper describes the four-legged ARAMIES robot which was built on the experiences gained with a first integration study. It describes the revised mechanics. The new modular electronics low-level control concept is presented and the developed FPGA-based joint control is explained in detail. Furthermore an overview on the software control approach is given and the new ground contact detection module using an IR-sensor is explained.

1. INTRODUCTION

The ARAMIES project deals with developing a walking robot which is capable of autonomous operation in extremely difficult environments, especially very uneven and steep terrain, e.g., the slopes of Martian / Moon canyons or craters. These sites are of high scientific interest. It is expected that in-situ measurements in the different layers of sediments would give new insights for the exobiology and exogeology.
Furthermore it is likely to find frozen water in craters near the pole. For example, in February 2005 the HRSC on ESA's Mars Express obtained a picture during orbit 1343, which shows an impact crater located on the Vastitas Borealis plain at approximately 70.5 North and 103 East, which at the centre contains residual water ice. The problem with craters is that they are extremely difficult to access for in-depth scientific studies. Conventional wheeled rovers are not able to provide this access reliably. Therefore we started to build the ARAMIES robot. A first integration study has been build and tested in 2005 [2].
Based on the results of these tests, a first full robot system (see fig. 1) has been built which features actuated claws, an additional hip joint, a fully integrated head segment, and a custom designed body. In the following, we present the improved mechanics, the new electronics concept for low-level joint control and further improvements in the software control, namely the ground contact detection, which is crucial for stable walking on uneven ground and a general problem for walking robots.
In particular, we address a novel ground contact detection using a very simple infrared distance sensor. For the overall locomotion control we are using the bio-inspired PCR-control approach [5] which was already successfully implemented on the SCORPION robot [8].

2. MECHANICS OF THE ARAMIES ROBOT

The ARAMIES robot was build based on the integration study presented at the iSAIRAS 2005 [7] and at the CLAWAR 2005 [2] The main improvements in the mechanics are:
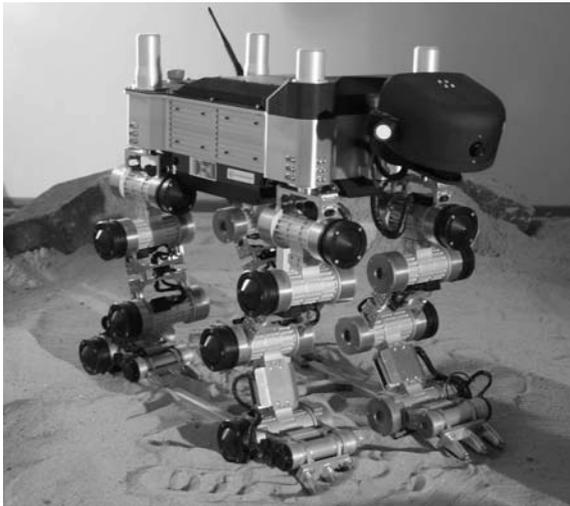
Figure 1. The ARAMIES Robot

Length	700 mm
Width	450 mm
Height	600 mm
Weight	28 kg
38 mm Joint:	
Torque (cont. / peak)	13 / 26 Nm
Rotational Frequency	1 Hz
22 mm Joint:	
Torque (cont. / peak)	2 / 3 Nm
Rotational Frequency	0.5 Hz
Accumulator:	
Capacity / Voltage	9 Ah / 24 V
Power Consumption:	
Standby	$\sim$ 20 W
Average	$\sim$ 600 W

Figure 2. Technical data of the ARAMIES prototype

1. Development of an integrated corpus to encapsulate the complete electronics (see fig. 1).

2. Development of additional hip joints, which are integrated into the corpus allowing to rotate the legs 360 deg.

3. Construction of an actuated sensor head, equipped with a laser scanner, a ccd camera and two ultrasound distance sensors.

4. A redesign foot with integrated pressure and distance sensors.
   The pressure sensors are integrated in each claw, in the center of the footprint area, and below the heel. The sensors are embedded in rubber elements to be protected against mechanical stress The sensor itself has a pressure range up to 5N. The embedding of the sensor results in a distribution of force into the surrounding material and the resulting effective pressure range is up to 50N. Furthermore, an infrared distance sensor is integrated. Its working range is 3cm-30cm. With this contactless integrated sensor the system gets information about the distance to the underground before the foot hits the ground.

5. Improved actuators for higher continuous torque. The new motor has the double torque. It increases the length of the housing to 15mm more. Technical details shown in tab. 2.

6. Integration of a battery system. Below the corpus the robot now holds NiMh batteries with a capacity of 9Ah

7. Integration of a tether system. At the front and the back of the robot are mounting points for a load cell to detect the traction of the tether to secure the robot. The main function of this sensor is to control an external active driven winch. The measurement of the force applied to the tether enables active control of the winch which allows using it as support during ambulation in steep terrain.

3.   AN AUTONOMOUS JOINT CONTROL BOARD

The low level control concept of the ARAMIES robot is centered around a newly¡ developed electronic control board named MOTCON6. This control board features autonomous position control of six independent joint actuators. The power electronics can drive DC motors within a voltage range of 24-35V and can deliver 8A current continuously as well as 80A peak current for each motor. The high efficiency around 99% of the developed H-bridge technology removes the necessity of heat sinks. The PCB is entirely assembled using SMD technology thus leading to a small size of 140x100 mm.
To control DC-motor driven joints, several functional blocks are required. These are 1) power electronics driving the electrical motors, 2) data acquisition of sensor signals (angle and temperature, drawn current among others), 3) position control for each motor, 4) limit protection (joint position, peak current values, average power consumption of each joint, temperature) and 5) communication between the low level actuator control,including data acquisition, and higher control levels using microcontrollers or embedded PCs (see also fig. 3).
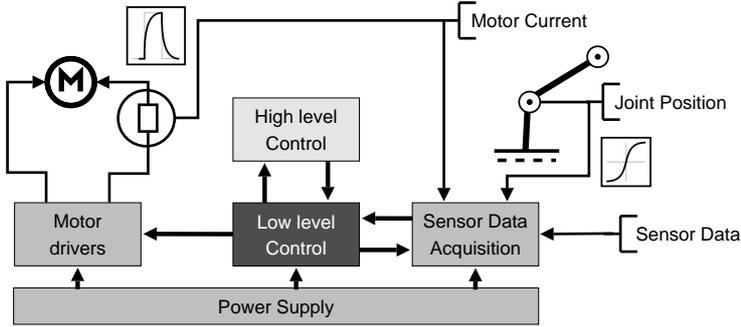
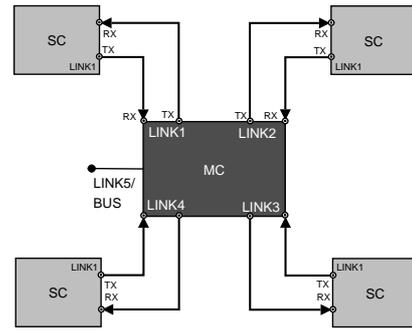Figure 3. Functional blocks required for autonomous robot actuator control.

Figure 4. Star network topology. MC: Master controller, SC: Slave controller.

Several special purpose peripheral circuits are required to interface the power electronics, the data acquisition blocks, and the communication blocks. Generic microcontrollers deliver some of the required peripherals, but not all. Therefore, customized digital logic featuring system-on-chip architecture is preferred. For rapid prototyping, Field Programmable Gate Arrays (FPGA) are best suited for this purpose, implementing the following functional blocks: a) the peripheral blocks, like PWM generators and ADC control, b) low level control of actuators, for example arithmetic position controllers, limit control, and c) communication control.

For reliability and for maintenance reasons, a motor control unit should be independent from a master controller, like a microcontroller or an embedded PC. That means, that all functional blocks consisting of analog and digital components must fit on one printed circuit board. Either one controller board for each actuator, or a bundled version for N actuators. Another important design aspect is power efficiency, regarding the power electronics and the power consumption of the analog and digital parts required for operation.

And, finally, the motor control board must provide a simple and reliable way of communication, either with a main controller, or as part of a distributed control network with other motor control boards. The communication must be managed by a simple but powerful communication protocol, independent of the used physical link device, for example generic serial RS232 or LVDS connection links. The communication protocol and the link technology must be independent of the target robot architecture and the main controller architecture, and must provide some kind of fault tolerance and reliability. Another import design goal is the size of the controller board.

There are three different physical control architectures used in robot control: 1) a central architecture approach, with one main controller centered on one physical board, 2) a distributed architecture with several controllers on the same priority level forming a group, servicing both low and high level control, and 3) a solution between (1) and (2) using a master-slave concept, with one or more master controllers responsible for high level and communication control, and (autonomous) slaves providing low level actuator control.

The central approach is the most simple one concerning the design complexity, both soft- and hardware. But in case the main system fails, the robot is not operable anymore. Additionally, many connecting wires are required to supply the actuator and to record signal data, for example from sensors (position of each actuator is at least required).

A fully distributed architecture is much more complex, though all nodes of the distributed system can be built with equal sub-architectures and components, but the robustness of the system depends on the reliability and functionality of the communication between nodes. High-level control of robots demands much more computation power than low level actuator control.

The simplest network topology is implemented using a strict master-slave approach and a star connection network. One central master initiates a message transfer with or without a response message from a slave controller. Figure 4 shows an example architecture with one central master, in general a communication controller connected to a host computer, and four slave controllers, for example our MOTCON boards. Each controller board uses a register bank memory for data exchange between controller boards and a host computer. Messages can read or modify the register contents. All registers are mapped to controller parameters and measured sensor data. In this simple network topology with fixed point-to-point connections, there is no necessity of device network addressing. Therefore, a simple message-based communication protocol can be implemented.

The controller of the MOTCON 6 is implemented using a Field Programmable Gate Array (FPGA), a reconfigurable Application Specific Digital Circuit (ASIC). The control architecture is partitioned and modularized into these blocks supporting six independent joint actuators:
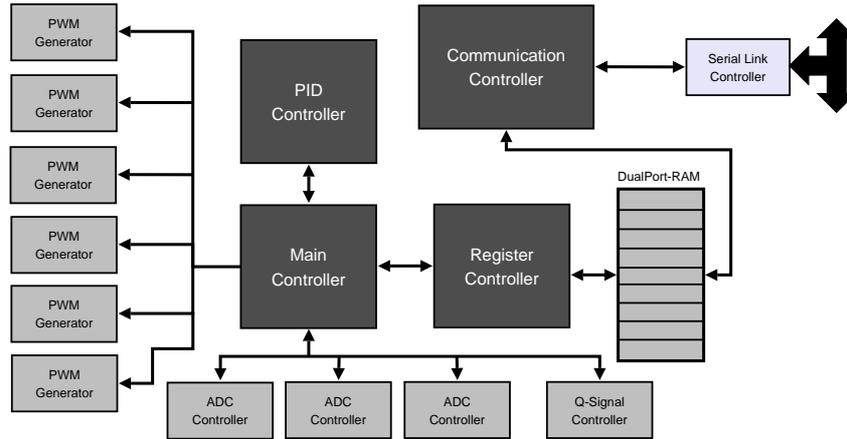
Figure 5. System partitioning inside the control FPGA.

1. Pulse-width-modulated (PWM) signal generators delivering drive signal for the H-motor-bridge power electronics with 8 bit pulse-width resolution,

2. ADC device drivers providing an interface to three external ADC devices with 12 bit data resolution, providing in total 24 channels,

3. Quadrature-signal decoder (rotational angle position measurement) with 12 bit resolution,

4. Register controller for managing access to 256 registers each 12 bit wide. The register bank contains all parameters required for actuator control and all acquired sensor data.

5. Communication controller implementing a simple and fault-tolerant communication protocol stack, following a connectionless text-message based approach, independent of the underlying serial link controller.

6. Serial link controller unit, attached to the communication controller. There are two different physical links available:

   - One RS232 port providing standard transfer rates in the range 9.6-118.2 kbit/s,
   - Two LVDS ports providing transfer rates upto 240 kbit/s.

7. The position controller for each actuator, implemented using an optimized and fully developed Proportional-Integral-Differential term controller (PID) with 12 bit data resolution and 1ms update time.

8. Limit observation and regulation, for example over-current detection.

All modules are connected using a System-On-Chip architecture and tristate-buses. There are different buses connecting different modules. The main controller is responsible for the processing flow and synchronisation between the different functional areas (see fig. 5).

The attached register controller provides access to the register-bank memory using a dual-port RAM. Each register of the register-bank is related to a controller parameter, for example parameters of the PID controller, or a measurement result, for example joint positions. The PID controller is serviced by the main controller using a handshake protocol, i.e. the position controller acts like a slave block. All other blocks are slaves and serviced by the main controller, too. The PWM generators are executed concurrently, but data transfer is managed by the main controller. The communication controller is independent from the main controller and is executed concurrently. The communication controller implements a simple message-based communication protocol.

The joint position controller actually implemented is a compact version of a traditional PID-controller. The input data is derived from the actual and past angular joint position sensor signal, and the scaled output signal U from the controller is fed directly into the PWM generators driving the motor voltage using highly efficient H-Bridge technology. The control function delivers three terms: an proportional, integral, and differential function term.

The actual measured joint position is P_A, the target joint position is P_D. The actual error value is the difference of these two values. An enhanced recursive and compact version of the controller algorithm was derived for

efficient synthesis into digital logic concerning hardware resources and latency. Therefore, there is only one equation built from a linear superposition of the three terms. The control parameters K_P for the proportional term, K_I for the integral term, and K_D for the differential term must be transformed in the parameters K_0, K_1, and K_2 using the equation shown below, derived from [3].

$$U(n) = U(n-1) + \Delta U(n) \tag{1}$$
$$\Delta U(n) = K_0 E(n) - K_1 E(n-1) + K_2 E(n-2) \tag{2}$$
$$E(n) = P_D - P_A \tag{3}$$
$$K_0 = K_P + K_I + K_D \tag{4}$$
$$K_1 = K_P + 2K_D \tag{5}$$
$$K_2 = K_D \tag{6}$$

The MOTCON6 board can control six independent joint units. But to save hardware resources, all joint controllers share one central PID controller. Digital logic synthesis of this PID controller results in an equivalent nand-gate count of 10825 gates (=43300 transistors) derived from standard cell synthesis (Alliance/LIP6 SXLIB library) and requires 500 D-FLIP-FLOPs. The target technology used on the MOTCON6 board is a Xilinx Spartan-II FPGA with an estimated gate count of approx. 100 k gates, and the highly optimized PID controller fits well into the FPGA resources.

4. LOCOMOTION SOFTWARE APPROACH

We are using the bio-inspired PCR-control approach which allows to produce CPG-like rhythmic motion patterns on the basis of Bezier-splines, resulting in a very simple description of complex walking trajectories. In addition, it allows easy programming of reflex processes and posture control behaviors. The software of the ARAMIES robot is implemented using the M.O.N.S.T.E.R. RT-microkernel for programming behavior-based robots [6].

The motoric layer of the ARAMIES robot consists of processes realizing basic rhythmic motion patterns and reflexes. For the rhythmic motion production, a simplified model of biological CPGs is used which consists of a unit to produce rhythmic trajectories in the joint angle space based on third-order Bezier splines, which can be scled in the phase, frequency, and amplitude.

More details have been found in [5]. In the current ARAMIES software, the CPG processes generate rhythmic motion patterns of the hip forward joint while a process ("Leg Extension") controlling the extension of the legs causes them to be lifted in the swing phase and be kept at a constant height in the stance phase. The CPG processes are activated by the walking process which takes commands from an external GUI (simulating a higher control center) connected via a wireless communication link. The pulse process provides a synchronizing signal for the CPGs according to the walking speed set by the walking process.

The reactive processes use tilt, joint current, infrared distance, and foot pressure sensory information to counteract disturbances in the motion. The ground contact process is responsible to keep ground contact with the ankle joint during stance. The stumbling correction reflex gets activated in the early swing phase when a leg is blocked in its swing motion, resulting in lifting the leg. In the late swing phase the step control reflex is active to prevent overstretching of the leg if ground is hit earlier than expected. All three mechanisms are inspired by biological analysis of cat walking [1].

The stability control process compensates tilt of the robot by adapting the leg extensions. In the swing phase, the foot control process keeps the foot of the corresponding leg parallel to the body of the robot. In the stance phase, the ankle is controlled by this process in such a way that the ball of the foot and the claws are likely to keep ground contact increasing grip on slippy terrain (e.g. sand).

The deliberative control can take over control of single or all joints and is used as a bypass for special higher level control, e.g. to manipulate objects.

More details on the motoric layer architecture, the achieved walking in different postures, and speed and about the stumbling correction reflex can be already found in [9].

Therefore, in this article we focus on the step control reaction which is described in the following. Step control means to control the extension of a leg when a foot steps on an obstacle or another kind of extra elevation of the ground. In comparison to more-legged systems like, for example, the Scorpion robot [4] it is important to maintain a stable stance position with every leg.

Consequently, the detection of ground contact of each leg is crucial for the stability of the ARAMIES robot. In order achieve this, we employ inexpensive infrared distance sensors which are mounted within the feet pointing

(a) Knee angles.

(b) IR Distances and Velocities.

(c) Pitch and Roll Angles.

(d) Knee angles.

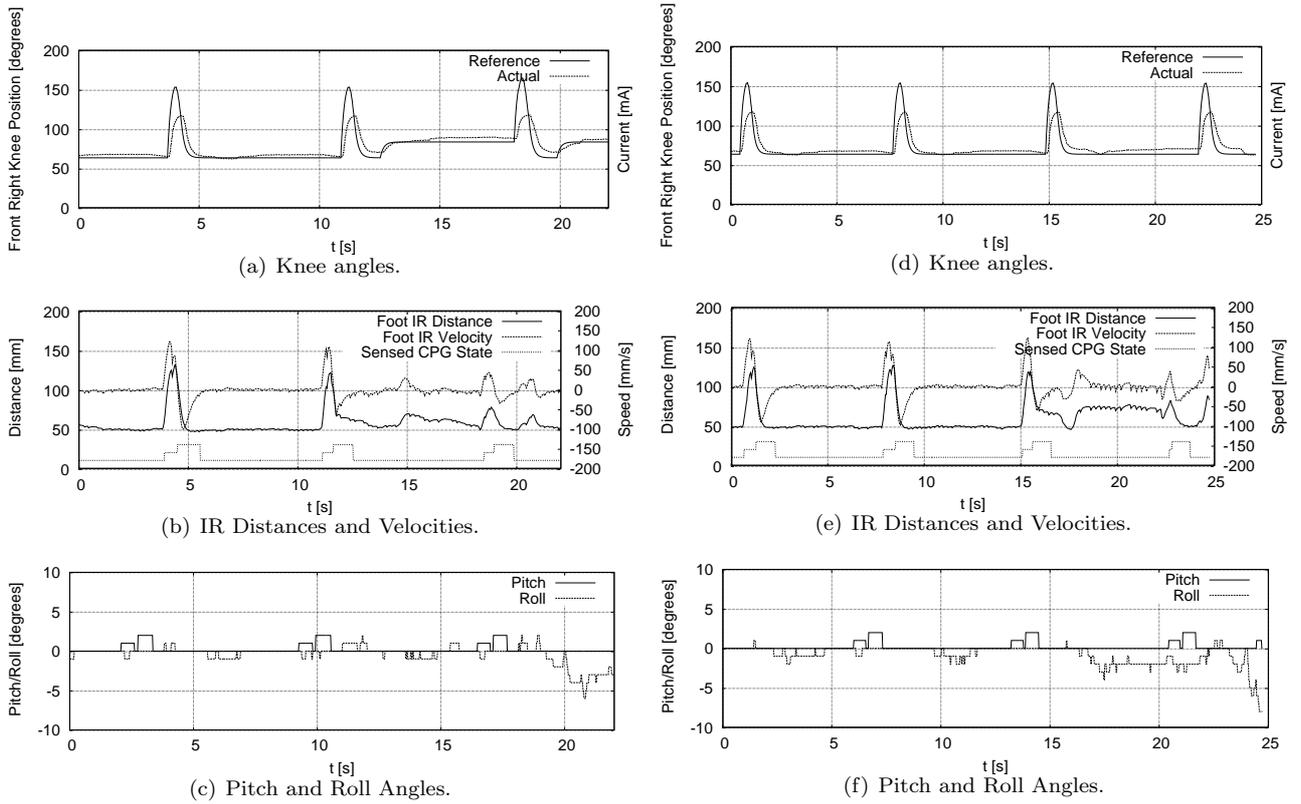(e) IR Distances and Velocities.

(f) Pitch and Roll Angles.

Figure 6. Illustration of two runs with (left) and without (right) use of the Step Control reflex. The infrared plots illustrates how the estimation of the walking phases stance, early swing, *and* late swing. *work*

downwards. It is not enough to use sparse pressure sensors because tests have shown that these fail too often to detect ground contact, especially on uneven terrain, where the foot is only partly connected to the ground. The ground contact detection via infrared distance information is accomplished using the derivative of the distance. Using absolute distance is not sufficient, e.g. the sensor might look into a ground depression.

By using the difference between consecutive distance measurements, we measure the movement speed of the foot relative to the ground. Exploiting typical velocity curves (cf. Fig. 6(b) and 6(e)) we estimate the walking state of one leg. The estimation is done with a simple state machine with three states q_0-q_2 with the following semantics:

- q_0: *Stance Phase*

- q_1: *Early Swing Phase*

- q_2: *Late Swing Phase*

The transitions between states are triggered using the threshold values $v_{TH1} = 30\,\text{mm/s}$ and $v_{TH2} = 10\,\text{mm/s}$ on the foot velocity v:

- $q_0 \longrightarrow q_1$ iff. $v > v_{TH1}$

- $q_1 \longrightarrow q_2$ iff. $v < -v_{TH2}$

- $q_2 \longrightarrow q_3$ iff. $v > -v_{TH2}$

Figure 6 demonstrates the benefit of this estimation in conjunction with the step control reflex. If it encounters a premature transition to state q_0 (*stance*), it knows that the leg has made contact with the ground and prevents further extension of the leg by the central pattern generator. This can be seen starting from $t = 12\,\text{s}$ in the left side of Figure 6. The robot steps on an obstacle, and the knee angle is adapted to this disturbance of the

walking. The right side of the figure illustrates the behavior of the robot without the use of this reflex. There occurs no adaptation of the leg extension which leads to a high peak roll at $t = 24.5$ s because of which the experiment had to be stopped. Otherwise, the robot would have fallen over. The plot of the roll values on the left side show that the reflex manages to keep the values in a non-critical range.

5. CONCLUSION

Tests have shown that the achieved mechanics and electronics are very stable and robust. In comparison to the integration study, the new mechanics and electronics have significantly increased the robustness of the overall system. Furthermore, the modularity has been increased, resulting in a construction set which allows to change the morphology of the system very quickly.
At the moment we are developing a new six-legged robot using this construction set.
Due to the new feet with the integrated pressure and infrared distance sensors and after carefully analysing the received sensor signals on various terrain we are now able to detect ground contact on uneven ground. This enables us to focus now on rougher terrain. Without a reliable ground contact estimation controlling a four-legged robot with non-elastic legs is extremely difficult, especially the design of reflexes to overcome obstacles and the design of balancing behavior. Nevertheless, the ARAMIES robot is able to produce stable walking and turning at different speed, and different postures on even ground, and can overcome singular obstacles.
In the next phase of the work, we will focus mainly on enhancing the existing reflexes for steep terrain and expect that by the end of the year we will be able to walk on uneven inclinations up to at least 50%.

ACKNOWLEDGMENTS

REFERENCES

[1] H. Forssberg. Stumbling corrective reaction: A phase-dependant compensatory reaction during locomotion. *Journal of Neurophysiology*, 42(4), July 1979.

[2] J. Hilljegerdes, D. Spenneberg, and F. Kirchner. The construction of the four legged prototype robot aramies. In *Proc. of CLAWAR 2005*, London, UK, September 2005.

[3] Rolf Isermann. *Digital Control Systems: Fundamentals, Deterministic Control*. Secaucus, New Jersey, U.S.A.: Springer Verlag, 1989.

[4] F. Kirchner, D. Spenneberg, and R. Linnemann. A biologically inspired approach towards robust real world locomotion in an 8-legged robot. In J. Ayers, J. Davis, and A. Rudolph, editors, *Neurotechnology for Biomimetic Robots*. MIT-Press, Cambridge, MA, USA, 2002.

[5] D. Spenneberg. A hybrid locomotion control approach. In *Proceedings of the CLAWAR 2005 Conference*, September 2005.

[6] D. Spenneberg, M. Albrecht, and T. Backhaus. M.o.n.s.t.e.r.: A new behavior based microkernel for mobile robots. In *Proceedings of the ECMR 2005*, September 2005.

[7] D. Spenneberg, M. Albrecht, T. Backhaus, J. Hilljegerdes, F. Kirchner, A. Strack, and H. Zschenker. Aramies: A four-legged climbing and walking robot. In *Proceedings of 8th International Symposium iSAIRAS*, Munich, September 2005.

[8] D. Spenneberg, K. McCullough, and F. Kirchner. Stability of walking in a multilegged robot suffering leg loss. In *Proceeding of ICRA 04*, volume 3, 2004.

[9] D. Spenneberg, A. Strack, H. Zschenker, and F. Kirchner. Reactive leg control for four-legged walking based on cpg and reflex. In *Proceedings of the Clawar 2006 Conference*, 2006.