S. Diefenbach, N. Henze & M. Pielot (Hrsg.): Mensch und Computer 2015 Tagungsband,
Stuttgart: Oldenbourg Wissenschaftsverlag, 2015, S. 133-142.

Natural 3D Interaction Techniques for Locomotion with Modular Robots

Dennis Krupke[1,2], Paul Lubos[1], Gerd Bruder[1], Jianwei Zhang[2], Frank Steinicke[1]

HCI, Department Informatics, University of Hamburg[1]
TAMS, Department Informatics, University of Hamburg[2]

Abstract

Defining 3D movements of modular robots is a challenging task, which is usually addressed with computationally expensive algorithms that aim to create self-propelling locomotion. So far only few user interfaces exist which allow a user to naturally interact with a modular robot in real-time. In this paper we present two approaches for baseline research of 3D user interfaces for intuitive manipulation of 3D movements of a modular chain-like robot in the scope of an iterative design process. We present a comparative evaluation of the techniques, which shows that they can provide intuitive human-robot interaction via remote control for real-time guidance of modular robots to move through heavy terrains and pass obstacles. In particular, our results show that steering a robot's locomotion via rotational hand movements has benefits for challenging locomotion tasks compared to translational hand movements. We discuss the results and present lessons learned for steering user interfaces for modular robots.

1 Introduction

Designing real-time 3D user interfaces in the domain of modular robots' high-level control is a challenging problem. Modular snake-like or caterpillar-like modular robots have great kinematic capabilities (González-Gómez et al. 2006), but the drawbacks lie in the lack of flexible and easy-to-use control methods. In particular, due to the large number of degrees of freedom that have to be controlled continuously in parallel (see Fig. 1a), it is very difficult to control these in real-time by a human operator. Instead, the movements of robot modules are usually actuated by embedded control software with sophisticated sensor-driven control loops (Kamimura et al. 2004). Autonomous generation of displacements of the modules of hyper-redundant chain-like robots, i.e., travelling waves (Hirose 1993), is usually realized with sinusoidal generators (González-Gómez et al. 2006) or central pattern generators (CPGs) (Herrero-Carrón 2007).

Although, these embedded solutions present significant advances to autonomous locomotion of modular robots, their inflexibility often results in robots getting stuck in terrain that has not been anticipated or pre-programmed (Li 2013). Physics-based simulations are used to optimize locomotion patterns, and specialized settings can be learned to pass individual obstacles (Krupke 2013). However, because most of these optimizations rely on evolutionary algorithms and reinforcement learning techniques (Li 2013), they cannot generally be applied to real-time applications.

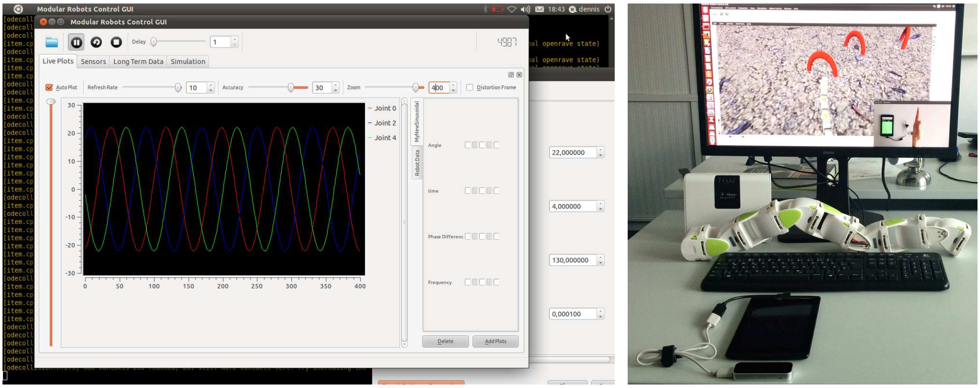
While it is inherently difficult to define the movements of modular robots in real-time, it is possible to define target poses or locomotion goals for semi-autonomous movements. Most existing approaches for human-robot interaction today make use of high-level interfaces, such as defining a locomotion goal by pointing at a 3D position which the robot then tries to propel itself toward (Yang et al. 2006; Park & Lee 2011; Nickel & Stiefelhagen 2007). Others tried to utilize a brain interface but the results were less efficient (Chae et al. 2011). Some researchers tried to imitate walking movements with the hand meanwhile wearing a special glove but this techniques cannot be easily applied to a modular robot with many joints (Komura & Lam 2006). It is a challenging question how 3D user interfaces should be designed to provide more direct and natural control over the movements of modular robots (Noeske et al. 2012).

In this paper we present and compare two high-level approaches for natural 3D user interfaces, which allow a user to steer a chain-like modular robot in real-time like a remote controlled toy car. Therefore, the user indicates the desired movement translation and rotation of the robot's head with his dominant hand in mid-air, which periodically applied to the robot. In order to reduce the complexity and increase the robustness of the locomotion we make use of a high-level control function. Parameters, including the phase difference between neighboring modules, which may cause unstable movement with the risk of undesired turning over, are restricted by trained thresholds. Trained fixed phase differences are efficient in sinusoidal pattern generators and CPGs (Ijspeert 2008) depending on the topological structure of the robot. In this way, 3D hand movements in mid-air can be converted to movements of modular robots in real-time at minimal computational cost. We present and evaluate two approaches that may provide interactive, intuitive control to guide a modular robot in an easy and efficient way. The system design focusses on a mobile implementation that allows the user to change his position while operating the robot. Thus, future implementations will enable users to convoy the robot while operating.

This paper is structured as follows. In Section 2 background information on modular robots and traveling waves are provided. In Section 3 our prototype is described and two user interfaces for altering locomotion of a modular robot are presented. In Section 4 we present a comparative usability evaluation of the approaches. In Section 5 we present the lessons learned, conclude the paper and finally give an overview of future work.

2 Background

Self-propelling movements of limbless animals with longitudinal bodies highly depend on the establishment of static frictional forces against the desired direction of travel. For example, locomotion of caterpillars is characterized by repetitive waves of slight movements of body parts that travel in waves from tail to head or vice versa, which can propel the whole body if the slippage against the locomotion direction is sufficiently low.



(a) *(b)*

Figure 1. Soft-/Hardware setup: (a) SiROS2 environment visualizes the angular positions of the robot's joints. Angular positions are calculated by a mobile device. (b) Leap Motion connected to an EVGA Tegra Note 7. The workstation runs a simulation framework (Krupke et al. 2012). Robot control commands are passed via UDP in the local network. Instead of a virtual robot the real robot can be addressed by the remote.

For actuating the joints of snake-like robots several methods are commonly used:

- Sinusoidal generators (González-Gómez et al. 2006)
- CPGs (Herrero-Carrón et al. 2007; Li et al. 2011)
- Pre-calculated patterns in combination with a transition function (Yamashina et al. 2011)

In this work, sinusoidal generators are used to generate smooth waves that are easy to modulate. They offer control parameters including frequency, amplitude, phase difference and offset, which are needed to influence the behavior of the locomotion regarding speed, direction and stability.

The most challenging issue with controlling modular robots is the need to optimize the control parameters for specific situations which sometimes rely on the properties of the terrain, e.g., height differences, obstacles, friction or external forces (Hirose 1993, Zhang et al. 2009). To the best of our knowledge no universal autonomous locomotion techniques exist that are applicable in arbitrary terrains. Reinforcement learning methods are able to train algorithms for passing special situations (Li 2013). However, these solutions cannot be applied to many other obstacles of different kinds. Direct interactive modulation of the control parameters via graphical user interfaces (GUIs) (Krupke et al. 2012) is challenging, since it requires special knowledge about the impact of certain parameters and their valid range. While such GUIs are often used in laboratory environments to initiate robot locomotion, these solutions usually suffer from low performance and unintuitive control. Alternative input to modulate the control parameters methods, such as hardware remote devices, have been found to raise similar issues (Noeske et al. 2012). So far, to our knowledge, no user interface exists, which provides intuitive teleoperation of modular robots in situations when it becomes necessary to switch to manual control to pass difficulties that cannot be traversed autonomously. The following sections describe our baseline research on mid-air control of snake-like robots. Further research will focus on locomotion synthesis based on 3D user interfaces in order to generate new locomotion patterns on-the-fly.

3 Description of the Teleoperation System

In this section we describe our prototype setup and user interface of a modular robot teleoperation system. The setup consists of a mobile device connected to a Leap Motion controller (see Fig. 1b), a chain-like modular CUBO robot with wireless communication, as well as a workstation and a network router (see Fig. 1b). Movements of the robot modules are controlled remotely with hand movements tracked by the Leap Motion in mid-air.

3.1 Prototype Setup

The Leap Motion is connected directly to an Android device via an USB OTG adapter. In our experiments a Google Nexus 5 smartphone and EVGA Tegra Note 7 performed very well with processing framerates from 30 fps to 60 fps. As desired for a teleoperation system we get a fully mobile system in this way. An Android application installed on the mobile device works as a control unit for the robot and processes the sensor data from the Leap Motion. Sensor processing and robot control are performed in different threads in order to achieve a high responsiveness of the system with low latency. The Android SDK alpha of the Leap Motion is used to acquire data frames. In the preprocessing step valid sensor frames are filtered by the Android application to increase the reliability of the control system. Visual feedback for the user is implemented by using a colored widget. Green indicates good positioning of the hand. Red reminds the user to place the hand closer to the center of the interaction box spanned by the Leap Motion in order to maintain the tracking of the hand. Captured position and orientation of the captured hand's palm is transformed to control parameters that determine the output of the sinusoidal generators, running on the Android tablet.

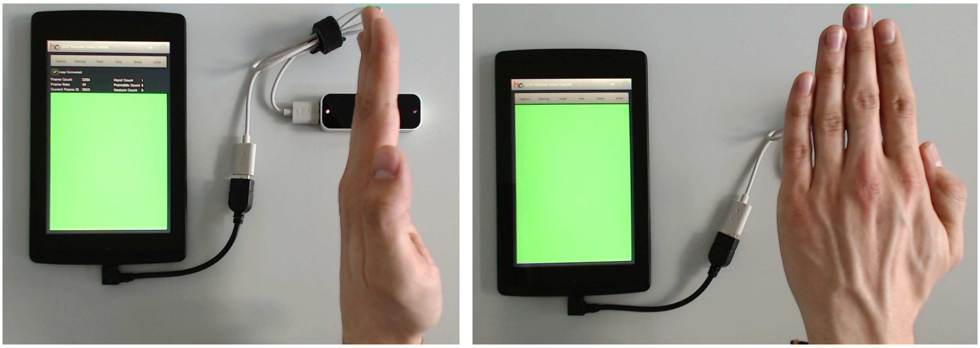
The control of the real CUBO robot (see Fig. 1a) is implemented with Bluetooth sockets that enable serial communication via the RFCOMM protocol. After the establishment of a connection between the mobile device and the robot commands for locomotion can be sent to the robot. A single command consists of the module's address and the desired angular position. In each step of the locomotion cycle every module's joint position needs to be updated. A high updating frequency and low latency are needed to achieve smooth locomotion patterns. These real-time constraints are the reason why smoothing algorithms do not fit the requirements and direct control is important. The GUI of the Android application is used to display data from the sensor and to manipulate the connection state of the robot using 2D touch interaction.

In order to focus on the evaluation of the 3D user interface without interference by technical limitations like runtime with batteries, we conducted experiments in a stationary setup with a simulated CUBO robot that runs in a physics based simulator. The control signals are generated by the mobile remote and sent via WLAN in UDP packages to the simulator that immediately executes the control commands.

3.2 3D User Interfaces

In this section we describe two direct 3D control methods based on hand movements in mid-air that are transferred to the robot's head movements. Continuously hand gestures and postures are analyzed to extract locomotion parameters. These parameters are fed forward to sinusoidal pattern generators in the mobile device, which are capable of generating 3D travelling waves on-the-fly. After calculating the angular positions these are applied directly to the joints of the robot. To create locomotion that results in self-propulsion of the robot the other modules are addressed subsequently with a short time shift of the dynamically calculated wave that accumulates with the increasing number of modules.

The interaction designs presented in the following are the results of focus studies with experts and novices in the domains of human-computer interaction and robotics. The two techniques are based on tracking hand postures and have in common that the palm position of the hand can easily be detected by the Leap Motion. This effectively increases the probability of successful hand recognition and minimizes tracking loss, i.e., problems caused by occlusion in skeleton tracking algorithms are avoided. Both control methods have a unique base-posture and use the same technique for altering locomotion speed. The control methods are tuned to reduce the possibility of unwanted turning over to minimum. An emergency stop gesture is always available by closing the hand to a fist.



(a) *(b)*

Figure 2. Hand interaction: Posture and movements of the hand in the (a) rotational control method, and (b) translational control method.

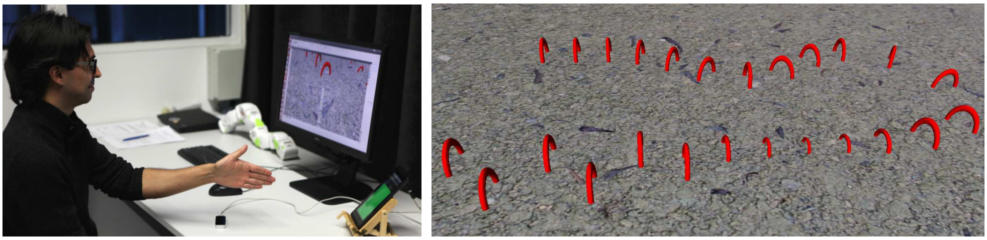
3.2.1 Type I: Rotational Control

The first design of hand-based robot locomotion is based on hand recognition and palm pose tracking. The basic static posture of the hand is shown in Figure 2a. In the control loop, values of the yaw-orientation and the sagittal axis of the currently tracked reference point of the palm are calculated and transformed to parameters of the sine functions that directly affect the locomotion of the robot. Steering to the left or right of the robot is initiated by turning the hand around yawing ax is. Absolute movement of the hand forth along the sagittal axis increases the forward speed, while translating the hand back causes the robot to move backwards with speed relative to the amount of the hand movement. The resulting forward or backward speed of the robot is computed using the deviation of the position from the reference point of the tracked hand relative to the zero position of the Leap Motion's sagittal axis. The maximum speed of the robot was set to 0.05 m/s.

3.2.2 Type II: Translational Control

The second method implements a different approach for steering the robot. When operating the robot the basic hand posture is always parallel to the Leap Motion sensor as shown in Figure 2b. Turning locomotion of the robot to the left or the right is initiated by a translational movement of the hand along the lateral axis of the Leap Motion. Hand movements along the sagittal axis forth or backwards are mapped to the backward or forward speed of the robot, respectively. We used the same maximum speed as for the rotational control technique.

4 Evaluation



(a) *(b)*

Figure 3. Experiment setup: (a) Participant seated in the laboratory during the experiment while steering the modular robot using his dominant hand. (b) Parcours for the usability comparison of the different control methods. Participants passed the gates with the modular robot in the given order.

In this section we describe the experiment, which we conducted to evaluate the steering techniques for locomotion of the modular robot. To account for variability in real-world modular robotics studies we conducted the experiment using the in-house SiROS2 simulation and robot training environment, which is a fine-tuned software framework for our tested modular robot and provides good ecological validity of simulated to physical movements.

4.1 Participants

We recruited 12 participants for our experiment, 7 male and 5 female (ages from 23 to 54, M=35). The participants were students or professionals in human-computer interaction or robotics. Participants were naive to the experimental conditions. 9 participants reported that they were right-handed and 3 reported that they were left-handed. In the experiment they completed the spatial tasks with their dominant hand. None of the participants reported known visual or motor disorders.

4.2 Materials

The experiment was conducted in a laboratory environment using the prototype setup described in Section 3. Participants were seated at a desk in front of a 24-inch screen as illustrated in Figure 3a. We used an Intel Core i7-4930K 3.4 GHz computer with 16GB RAM and Nvidia GeForce GTX 780 Ti graphics card for the simulation of the *SiROS2* environment. The environment was rendered with the Coin3D engine. A Leap Motion sensor was connected via USB OTG to an EVGA Tegra Note 7 tablet. Communication between the remote and the simulator is performed via WLAN in UDP packages by utilizing a router.

The task environment for the modular robot consisted of a ground surface with gates of 7cm to 10cm diameter in the environment at distances of 8cm to 12cm, which participants had to pass with the modular robot. We designed the test environment such that once a gate was passed by the modular robot it disappeared and its kinetic body was removed from the scene. We rendered the environment using a third person camera, positioned behind the robot's head, to provide always good visibility of the robot, independent of its position and orientation in the environment.

4.3 Protocol

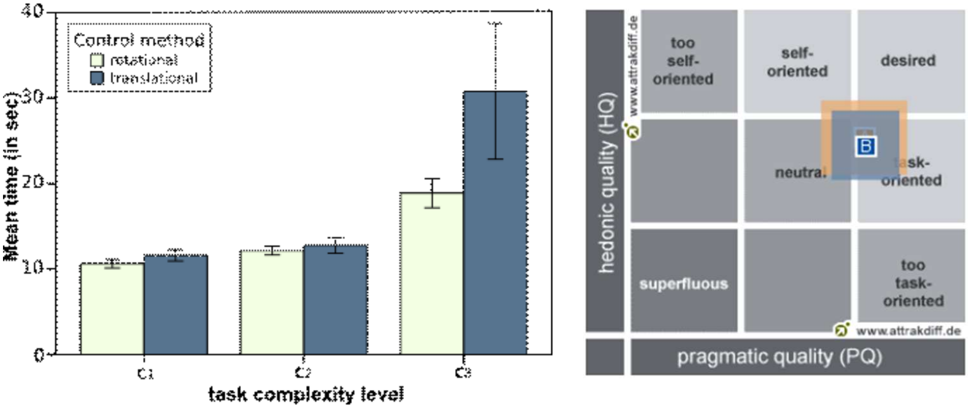
Participants were instructed to steer the modular robot through all gates as fast and as accurately as possible in the order that was shown in the 3D environment. If participants missed a gate they had to backtrack their path with the modular robot to complete the task. We considered three task complexities in the experiment: (C1) gate diameter 10cm with slight lateral displacement between gates, (C2) gate diameter 7cm with medium lateral and rotational displacements, and (C3) gate diameter 9cm with large lateral displacements. To account for the expected durations participants had to pass 10 gates in both conditions C1 and C2, and 5 gates in condition C3.

4.4 Methods

We used a 2x3 repeated measures within-subjects design. The independent variables were the control method (rotational vs. translational) and the three task complexities (C1, C2 and C3). The dependent variable was the time it took the participants to pass the gates during the experiment. Furthermore, we collected demographic information with a questionnaire before the experiment and measured the participants' task load with the NASA TLX questionnaire as well as the sense of attractiveness with the AttrakDiff questionnaire. After completion of the tasks we collected informal responses from the participants and asked them to provide qualitative feedback related to the two tested steering methods. Participants were allowed to take breaks between the conditions. The total time per participant including pre-questionnaires, instructions, experiment, breaks, post-questionnaires and debriefing was 30 minutes.

4.5 Results

Figure 4a shows the pooled results for the gate passing times for the two control methods and three task complexities. The vertical bars show the standard error of the mean. We removed one participant from the analysis due to a technical problem. We analyzed the results with a repeated-measures ANOVA and Tukey multiple comparisons at the 5% significance level. Degrees of freedom were corrected using Greenhouse-Geisser estimates of sphericity when Mauchly's test indicated that the assumption of sphericity had been violated.



(a) *(b)*

Figure 4. Experiment results: (a) Gate passing times for the two control methods and three task complexities. The vertical bars show the standard error of the mean. (b) Results of the AttrakDiff questionnaire along the dimensions of pragmatic and hedonic quality. Method A corresponds to the rotational control and method B describes the results from the questionnaire about the translational control.

We found a trend for an interaction effect between control method and task complexity on gate passing time (F(1.03,10.35)=2.21, p=.17, η_p^2=.18). We found a significant main effect between the different task complexities on gate passing time (F(1.04,10.36)=9.59, p=.01, η_p^2=.49). We found a trend for a main effect between the different control methods on gate passing time (F(1,10)=2.96, p=.12, η_p^2=.23). Post-hoc tests revealed no significant differences in the gate passing times between the two control conditions for the different task complexities (p>.05). For rotational control, the gate passing times were significantly different between C1 and C2 (p<.05), between C1 and C3 (p=.001) and between C2 and C3 (p=.002). For translational control, the gate passing times were significantly different between C1 and C3 (p<.05) and between C2 and C3 (p<.05).

The NASA TLX data was analyzed for the different metrics. The mental demand was M=18.25 (SD=11.2). Physical demand was higher with M=56.75 (SD=21.8). Temporal demand is rated with M=24.2 (SD=18.7). Performance reached M=28.0 (SD=12.2) and effort is rated with M=32.1 (SD=17.6). The participants judged frustration with a low score of M=23.1 (15.1). The overall rating reached M=30.4 (11.8). The results of the AttrakDiff questionnaire are shown in Figure 4b. The results show the tendency to prefer the rotational control method. In general the hedonic and pragmatic quality indicate the suitability of our method to manual control of modular robots.

The collected qualitative feedback supports the quantitative results and generally indicates very positive judgments of the ability to steer the modular robot with both techniques. 10

participants preferred rotational control over translational control. None of them preferred the translational control method and 1 participants reported no preference.

4.6 Discussion

The results show that both control methods reach a similar performance for the C1 and C2 task complexities, but suggest a tendency towards a difference for C3. The qualitative feedback confirms this result, indicating that both techniques reached similar high attractiveness scores, but a tendency that participants preferred the rotational technique in more complex situations. Both methods have in common that the mental demand was rated very low in contrast to the physical demand. Evaluating the post-questionnaire revealed that the rotational control is the preferred method. Generally, participants gave positive feedback on the idea of a mid-air-based robot control but mentioned the physical afford. They asked for a faster moving robot, which emphasizes the easy controllability of the proposed methods.

5 Conclusion

Our results show that mid-air hand gestures provide a reasonable and intuitive way to steer a modular robot through a terrain with obstacles. However, optimally we would like to enable intuitive control over the entire body of such chain-like robots, not just the head. This may become possible by inducing traveling waves via wave-like hand gestures later on. Although such approaches may be leveraged as a natural extension of the described head-centered steering techniques, our initial results suggest that such approaches require very good motor skills and learning of a very limited set of wave gestures that can generate self-propelling locomotion of modular robots, whereas most wave gestures will not result in the robot moving from its current position even when the modules are moving. We believe that future 3D user interfaces for steering such modular robots will combine both head-based steering as introduced in this paper as well as direct control over the robot's body via more complex hand gestures. Such hybrid approaches can support users to intuitively steer robots over light terrain while being able to pass more complex obstacles in case the robot becomes stuck. We are considering these approaches in the next cycle of our iterative design process.

In this paper we presented two 3D user interfaces for intuitive control of the 3D movement patterns of chain-like modular robots. We performed an experiment which showed that the proposed techniques are effective in moving a modular robot and easy-to-use. Mid-air gestures proved to be suitable for modular robot control but dealing with fatigue of the user must be taken into account. All participants were able to steer a modular robot through heavy terrain with the techniques. The two proposed techniques reached similar task performance, with rotational hand gestures being subjectively rated higher than translational gestures in difficult steering situations. We believe that the presented approaches have the potential to be used as an effective, mobile solution to take over control of modular robots, as well as a versatile platform for teaching and testing of novel control loops in human-robot interaction. In future work, we plan to iterate on the design process to incorporate direct control methods for the body of the modular robot, i.e., providing a user interface that supports specification and testing of movement patterns that are not commonly possible with traditional locomotion algorithms. We plan to use the online control interfaces in courses on supervised learning and loop-generation by fast learning.

Contact

https://www.inf.uni-hamburg.de/en/inst/ab/hci/people/krupke.html

References

Chae, Y., Jo, S. & Jeong, J. (2011). Brain-actuated humanoid robot navigation control using asynchronous Brain-Computer Interface. In (Hrsg.): IEEE/EMBS NER: S. 519–524.

Diankov, R. (2010). Automated Construction of Robotic Manipulation Programs. PhD thesis, Robotics Institute, Carnegie Mellon University.

González-Gómez, J., Zhang, H., Boemo, E. & Zhang, J. (2006). Locomotion Capabilities of a Modular Robot with Eight Pitch-Yaw-Connecting Modules. In (Hrsg.): CLAWAR ‘06.

Herrero-Carrón, F., Rodriguez, F. B. & Varona, P. (2007), Study and application of Central Pattern Generator circuits to the control of a modular robot, Master's thesis, Escuela Politécnica Superior, Universidad Autónoma de Madrid.

Hirose, S. (1993). Biologically inspired robot: snake-like locomotors and manipulators. Oxford, New York: Oxford University Press.

Ijspeert, A. J. (2008). Central pattern generators for locomotion control in animals and robots: A review. Neural Networks 21 4, 642 - 653.

Kamimura, A., Kurokawa, H., Yoshida, E., Tomita, K., Kokaji, S. & Murata, S. (2004). Distributed adaptive locomotion by a modular robotic system, M-TRAN II. In (Hrsg.): IEEE/RSJ IROS ‘04, 3. S. 2370–2377.

Komura, T. and Lam, W.-C. (2006). Real-time locomotion control by sensing gloves. Comp. Anim. Virtual Worlds 17, 513–525.

Krupke, D. (2013). Development of Bio-inspired Locomotion Using Modular Robotic Simulation and Control System. Diploma thesis, Universität Hamburg.

Krupke, D., Li, G., Zhang, J., Zhang, H. & Hildre, H. P. (2012). Flexible Modular Robotic Environment for Research and Education. In (Hrsg.): ECMS ‘12.

Li, G. (2013). Hierarchical Control of Limbless Locomotion Using a Bio-inspired CPG Model. PhD thesis, Universität Hamburg.

Li, G., Zhang, H., Herrero-Carrón, F. & Zhang, J. (2011). Locomotion of Limbless Robots using Novel Biomimetic Neural Circuits. In (Hrsg.): IEEE/RSJ IROS ‘11.

Nickel, K. & Stiefelhagen, R. (2007). Visual recognition of pointing gestures for human–robot interaction. Image and Vision Computing 25 12, 1875–1884.

Noeske, M., Krupke, D., Hendrich, N., Zhang, J. & Zhang, H. (2012). Interactive control parameter investigation of modular robotic simulation environment based on Wiimote-HCI's multi sensor fusion. In (Hrsg.): IEEE MFI ‘12, S. 478–483.

Park, C. & Lee, S. (2011). Real-time 3D pointing gesture recognition for mobile robots with cascade HMM and particle filter. Image and Vision Computing, January 2011, 51–63.

Yamashina, R., Kuroda, M. & Yabuta, T. (2011). Caterpillar robot locomotion based on q-learning using objective/subjective reward. In (Hrsg.): IEEE/SICE International Symposium on System Integration (SII) ‘11, S. 1311–1316.

Yang, H., Park, A. & Lee, S. (2006). Human-Robot Interaction by Whole Body Gesture Spotting and Recognition. In (Hrsg.): 18th International Conference on Pattern Recognition ‘06, S. 774–777.

Zhang, H., González-Gómez, J. & Zhang, J (2009). A new application of modular robots on analysis of caterpillar-like locomotion. In (Hrsg.): IEEE International Conference on ICM ‘09, S. 1–6.