

YaMoR and Bluemove – An Autonomous Modular Robot with Bluetooth Interface for Exploring Adaptive Locomotion

R. Moeckel, C. Jaquier, K. Drapel, E. Dittrich, A. Upegui, A. Ijspeert

Ecole Polytechnique Fédérale de Lausanne (EPFL), Logic System Laboratory (LSL), CH 1015 Lausanne, Switzerland
moeckel@ini.phys.ethz.ch, (andres.uegui, auke.ijspeert)@epfl.ch

Abstract

Modular robots offer a robust and flexible framework for exploring adaptive locomotion control. They allow assembling robots of different types e.g. snakelike robots, robots with limbs, and many other different shapes. In this paper we present a new cheap modular robot called YaMoR (for “Yet another Modular Robot”). Each YaMoR module contains an FPGA and a microcontroller supporting a wide range of control strategies and high computational power. The Bluetooth interface included in each YaMoR module allows wireless communication between the modules and controlling the robot from a PC. With the help of our control software called Bluemove, we tested different configurations of our YaMoR robots like a wheel, caterpillar or configurations with limbs and their capabilities for locomotion.

Keywords: Modular robot, Bluetooth, Locomotion, FPGA

1 Introduction

Locomotion with modular robots constitutes a great potential and at the same time a very difficult challenge [1-4]. In comparison to conventional monolithic robots, modular robots present the advantage of supporting a fast reconfiguration of their structure. To build a robot of the desired form a completely new robot does not need to be constructed but it can be reas-

sembled by simply disconnecting and reconnecting modules. Furthermore modular robots constitute a challenging framework for exploring distributed control when each module contains its own controller and sensors.

YaMoR is designed to act as a cheap platform for (1) testing different control algorithm for locomotion and their implementation in both software and hardware, (2) exploring the capabilities for locomotion of a large variety of different robot configurations and shapes as well as for (3) finding new applications for wireless networks.

The main characteristics of our modular robots are: (1) each module contains a Bluetooth interface for inter-module communication as well as for communication between the modules and a base station like a PC — most modular robots use direct electrical connections, which are less flexible — and (2) each module comprises an FPGA for reconfigurable computation — most modular robots use traditional microcontrollers, but see [5] for an example of a robot using a single FPGA for controlling all modules.

We designed and implemented a control software called Bluemove that allows to control the YaMoR modules from a PC via Bluetooth. Bluemove offers an easy way for exploring the capabilities for locomotion of different configurations of modules.

In section 2 we give an overview about the mechanics and electronics of our modular robot. Section 3 describes the Bluetooth interface each robot modules contains. Section 4 gives an introduction to Bluemove, the control software that we use for exploring locomotion. In section 5 we describe first examples of locomotion illustrating the capabilities of YaMoR. Finally section 6 concludes and gives an outlook about future work.

2 YaMoR – mechanics and electronics

YaMoR consists of mechanically homogeneous modules. One of the key features of YaMoR is its low cost: in contrast to the majority of modular robots, YaMoR is constructed with off-the-shelf components. Each module contains a powerful one degree of freedom servo motor (with a 73Ncm maximal torque). Its casing consists of cheap printed circuit boards (PCB) that can also serve as support for printed circuits (see **Fig. 1.**).

The casing of each module is covered with strong velcros. Velcros offers the advantage to connect robot modules together with no restriction on angles between the surfaces of the modules. Unfortunately, it does support self-reconfiguration and the modules can only be connected together by hand.

The modules are autonomous. They are powered by on-board Li-Ion batteries and include the necessary electronics for power management, motor control, communication and running algorithm. To achieve more flexibility and modularity in terms of control each YaMoR module contains three separated control boards: (1) one board including a Bluetooth-ARM microcontroller combination, (2) one board carrying a Spartan-3 FPGA, and (3) a service board containing power supply and battery management.

YaMoR was constructed as a framework for a variety of different projects. For instance, a user may choose between using a microcontroller, an FPGA or a combination of both for implementing the desired control algorithm. Configuring the FPGA to contain a MicroBlaze soft-processor [6], allows exploiting the hardware-software codesign capabilities offered by the platform, taking also advantage of the flexibility provided by the FPGAs partial reconfiguration feature [7].

The YaMoR architecture with distributed electronic components gives a flexible solution for connecting the electronic boards: the FPGA board can be left out if it is not needed to save energy; or it can be replaced by a board with specific sensors if useful. The new sensor board can still take advantage of the electronics mounted on the remaining boards. So a designer for an additional sensor board does not have to worry about power supply or battery management.

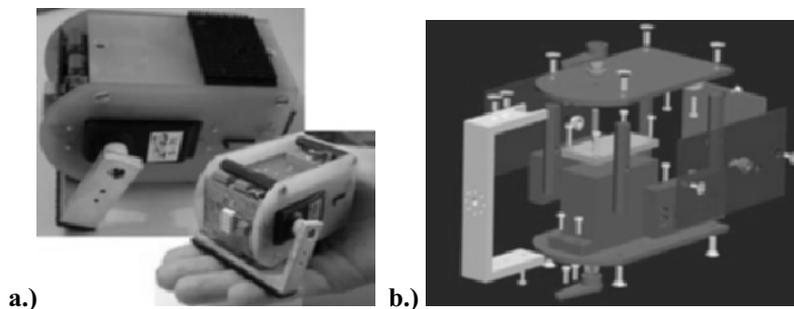


Fig. 1. YaMoR module
a.) Module closed and with open casing. b.) Assembly of YaMoR mechanics.

3 Bluetooth – the wireless interface to YaMoR

We chose Bluetooth for wireless communication given its flexibility and energy efficiency. Wireless communication between modules allows creating a new robot configuration by simply disconnecting and reconnecting the mechanical modules without the need for reconnecting cables or

changing the control infrastructure. In comparison to a communication based on wires, Bluetooth has the constraint that a module normally does not know its physically connected neighbours just by communicating with them. However, this constraint can be overcome with the addition of touch or distance sensors.

The ARM on the Bluetooth board is running both a real time operating system and the embedded Bluetooth stack. It can also be used for customized software e.g. a control algorithm or for reconfiguring the FPGA via Bluetooth. The ARM program code and FPGA configuration bitstream are stored inside a Flash memory on the microcontroller board.

The Bluetooth-ARM board was designed to provide a wireless interface that can be easily controlled. The embedded Bluetooth stack allows taking advantage of wireless communication by sending simple commands via UART. For instance, a researcher concentrating on FPGA based algorithms may implement a simple UART module on the FPGA and is able to communicate wirelessly with a PC or other modules.

4 Bluemove – controlling YaMoR via Bluetooth

For easily exploring new configurations of modules and their capabilities for locomotion, an interactive Java based control software called Bluemove has been developed. Using a graphical user interface (GUI) on a PC, a user can quickly start a new project, register all modules used for the current robot configuration and implement a controller. To control the modules Bluemove allows both (1) writing trajectories that can be continuously sent to the modules via Bluetooth and interactively modified without any resetting, (2) the use of plugins for controlling the modules from a PC, and (3) programming the FPGAs as well as the ARMs for autonomous control in the modules without needing a PC. Plugins can act as inputs (hand-drawn trajectories, generators, oscillators, etc.), filters (signal processors, multiplexers, etc.) and outputs (data sent to the modules, files, streams, etc.). Plugins support the generation of controllers with feedback from sensors. The whole project including the trajectories and plugins can be saved in XML. The main parts of the graphical user interface are:

1. The Module Manager serves to manage all modules belonging to the current robot configuration including module names and Bluetooth addresses of the modules.
2. The Timelines Manager (see **Fig. 2.**) allows generating trajectories for each actuator of the modules registered in the Module Manager by set-

ting key points on the GUI with a mouse. Linear and spline interpolation can be chosen to draw the trajectories and connect the key points set before. Trajectories even can be changed “online” while transmitted to the modules e.g. by changing the position of the key points.

3. The Real-Time Module (see **Fig. 3.**) supports an easy use of Bluemove with the help of plugins. New plugins can be created with a script editor. The relations between different plugins are visualised with the help of a graph.

Bluemove is implemented in Java, taking advantage of its standard and consistent interface for Bluetooth. Given the popularity of Bluetooth and Java, it would be possible to create Bluetooth applications for the modular robot on mobile phones, PDAs or other small systems that support Java and Bluetooth.

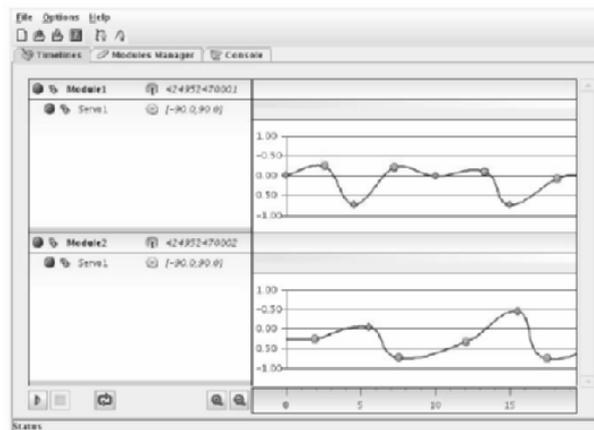


Fig. 2. Timelines Manager

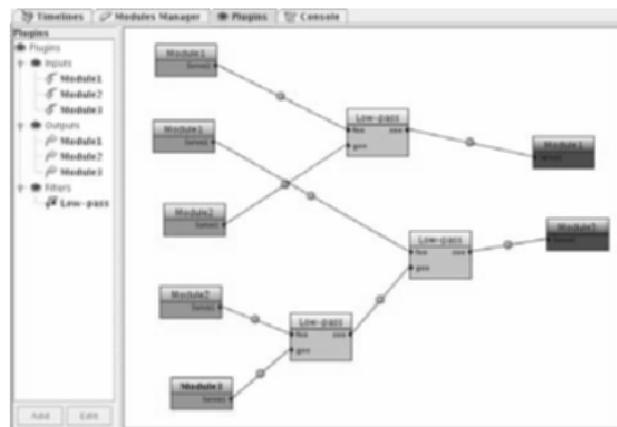


Fig. 3. Real-Time Modules

5 Exploring locomotion

We explored the locomotion capabilities of different YaMoR configurations with up to six modules, using Bluemove to generate the joint angle trajectories for the servo motors. By trial and error, interesting gaits could be generated for a variety of robot structures, such as travelling waves for worm and “wheel” structures (see Fig. 4. for four snapshots of a moving wheel), crawling gaits for limbed structures, and other peculiar modes of locomotion (see Fig. 5. for different examples of configurations of YaMoR modules). For videos, the reader is kindly requested to visit the project website [8].

These gaits are only a first step towards adaptive locomotion (see next section), but already represent in our opinion an interesting example of control with a “human in the loop”. The user can indeed interactively adjust the gaits in real time, in order to optimize the speed of locomotion for instance, as well as modulate the gaits in terms of speed and direction, by modifying frequency and amplitude parameters.

For exploring more complex shapes of a robot, tests with more than six modules are needed. For example, most of the configurations tested so far do not allow changing the direction of the movement, and would therefore not be capable of avoiding or overcoming obstacles.

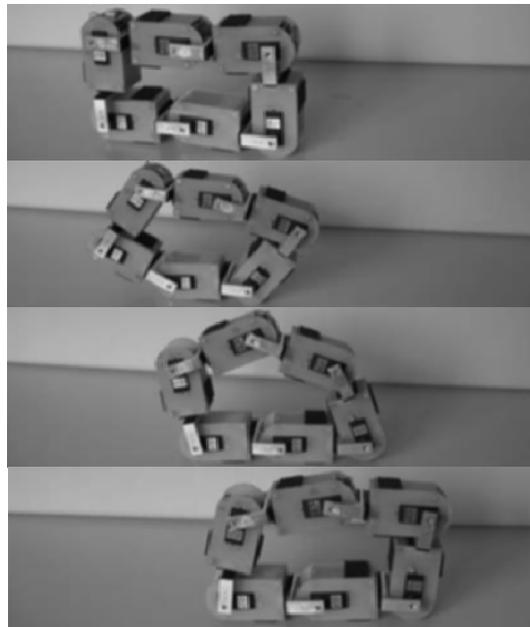


Fig. 4. Rolling wheel.

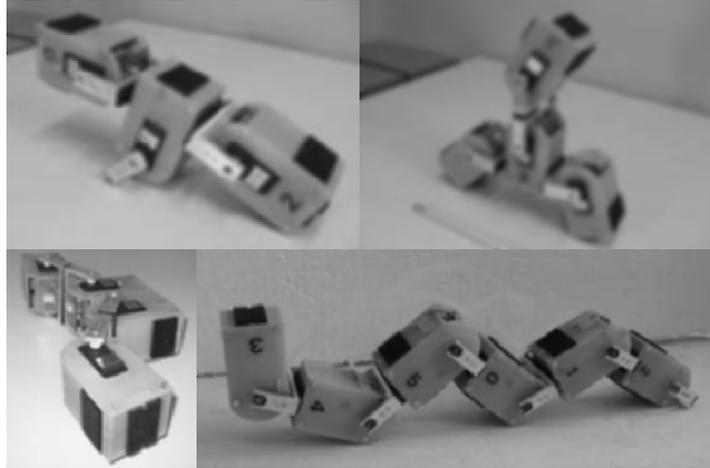


Fig. 5. Different configurations of YaMoR modules

6 Future work

To achieve adaptive locomotion, “human in the loop” control that we were using for the first experiments is clearly not sufficient. In a complex environment a robot has to react and adapt in real time. That is why we designed Bluemove to also support feedback signals from sensors with the help of plugins. Distributed algorithms can be implemented in the ARM and FPGA of each YaMoR module. Both the parameter of these algorithms and the shape of the modular robot can be optimized under different constraints like energy efficiency or speed.

We are currently extending the presented work along two main axes: (1) the design of the next generation of YaMoR modules with sensors capabilities — e.g. IR sensors, inertial sensors, and load sensors — and (2) the implementation of distributed locomotion controllers based on central pattern generators. Central pattern generators (CPGs) are biological neural networks capable of producing coordinated patterns of rhythmic activity while being initiated and modulated by simple input signals. We have extensively used models of CPGs for the control of locomotion in other projects [9, 10]. In particular, simulation experiments have demonstrated they are ideally suited for implementing distributed control of locomotion in simulated YaMoR units [11].

Acknowledgements

We would like to acknowledge André Badertscher for the technical support, Alessandro Crespi, Fabien Vannel and René Beuchat for many useful discussions and hardware debugging. We acknowledge also the Swiss National Science Foundation for the Young Professorship Award granted to Auke Ijspeert.

References

1. Duff D., Yim M., Roufas K. (2001) Evolution of PolyBot: A Modular Reconfigurable Robot. *Proc. of the Harmonic Drive Intl. Symposium, Nagano*, Japan, November.
2. Yim M., Zhang Y., Roufas K., Duff D., and Eldershaw C. (2002) Connecting and disconnecting for chain self-reconfiguration with polybot. *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 4, pp. 442–451.
3. Kurokawa H., Kamimura A., Murata S., Yoshida E., Tomita K., and Kokaji S. (2003) M-tran II: Metamorphosis from a four-legged walker to a caterpillar. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 3, pp. 2454–2459, October.
4. Castano A., Behar A., and Will P. (2002) The conro modules for reconfigurable robots. *IEEE Transactions on Mechatronics*, vol. 20, pp.100–106.
5. González-Gómez J., Aguayo E., and Boemo E. (2004) Locomotion of a Modular Worm-like Robot using a FPGA-based embedded MicroBlaze Soft-processor, *Proceedings of International Conference on Climbing and Walking Robots (CLAWAR 2004)*.
6. Xilinx Corp., MicroBlaze™, URL <http://www.xilinx.com> [June 2005]
7. Upegui A., Moeckel R., Dittrich E., Ijspeert A., and Sanchez E. (2005) An FPGA dynamically reconfigurable framework for modular robotics. *Workshop Proceedings of the 18th International Conference on Architecture of Computing Systems 2005 (ARCS 2005)*, VDE Verlag, Berlin, Germany, pp. 83-89.
8. YaMoR project website, URL <http://birg2.epfl.ch/yamor> [June 2005]
9. Ijspeert A.J. (2001) A connectionist central pattern generator for the aquatic and terrestrial gaits of a simulated salamander. *Biological Cybernetics*, vol. 84, no 5, pp. 331-348.
10. Crespi A., Badertscher A., Guignard A., and Ijspeert A.J. (2005) Amphibot I: An amphibious snake-like robot. *Robotics and Autonomous Systems*, vol. 50, no. 4, pp.163-175.
11. Marbach D. and Ijspeert A.J. (2005) Online Optimization of Modular Robot Locomotion, *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA2005)*, to appear.