A HIERARCHICAL MODULAR ARCHITECTURE FOR RECONFIGURABLE MOBILE ROBOTS

Victor Andreev, Valerii Kim & Valentin Pryanichnikov

Abstract

In this paper we investigate a common issues associated with growth of computational complexity of mobile robots navigation algorithms. It is also shown that a realization of a particular robot application is usually slowed down because of lack of interchangeable and unified robot components. We propose a hierarchical modular architecture for reconfigurable mobile robots as a solution to these problems. In this architecture a mobile robot considered as a combination of modules, which in turn consist of more simple units - sub-modules. Each submodule incorporates a low performance microcontroller and is responsible for only basic functions. A set of sub-modules are then combined into a module – transport platform, robot leg, arm etc. Besides of this one of the main goals of the study is to provide a framework based on this architecture for a rapid robot prototyping with modules that can be assembled in a plug-and-play way. In this paper we mostly consider hardware modules and prototypes that have been already manufactured. We also briefly investigate a robot transport module structure for a motion on a rough terrain in an unstructured environment.

Keywords: mobile robot, modular architecture, reconfigurable robots, distributed computing systems.

1. Introduction

One of the common approach to technical devices design and their versatility increasing is a division of a technical system’s structure in separate functional units – modules. Modular approach is well-known in an automotive industry, machine tool industry and computer technology. In computers almost all components are standardized and interfaces for their communication are also well-developed so almost any experienced user can assemble own computer from compliant components-modules. Nowadays modular robotic systems actively expanding and they are not limited only to laboratory prototypes but start to appear on the market [1], [2].

Modular approach in mobile robotics is aimed on solving of a vital problem – integration and interoperability of a wide spectre of hardware and software components in a single robot application. Today specialists in robotics encounter a common problem of “reinventing the wheel” – researches struggling with the integration of hardware/software interfaces instead of dealing with new algorithms and models.

Another problem is that successful completion of the majority of robot’s tasks in real environment requires from an onboard computer execution of a great number of math computations with different data in the real-time. Computational
loads increases even more when robot must process vision data and perform navigational algorithms with map building. Modular approach to mobile robots design can address this problem if computations of one onboard computer will be distributed between modules.

In this paper we introduce concept of ModRob – hierarchical modular architecture for mobile robots which allows to rapidly building custom robot system using unified components (modules). This architecture is a distributed robotic system with computational nodes implemented on microcontrollers or onboard computers.

This paper organized as follows: section 2 discusses related work; section 3 covers modular architecture in a detail, section 4 provides information about some practical results and section 5 provides conclusions and thoughts about future work.

2. Related work

Robotics researchers that involved in a design of modular robot architectures try to achieve two main goals:

1. Simplification of other scientists and engineers projects by introducing unified robot components and/or universal program interfaces. As it was mentioned before today there are many incompatible hardware and software robot parts that must be integrated in one robotic system.
2. Development of robots for tasks in extreme conditions. Modular approach allows creating reconfigurable robots that can be used for many unpredictable tasks requiring different robotic systems.

The first goal is very close to development of hardware and software frameworks, middlewares and operational systems (OS) for robotics (for example ROS as the most popular OS for robot design). Frameworks, middlewares and OS chasing the same goal – introducing tools for rapid robot prototyping. But to best of our knowledge these systems work mostly with robots software. Comprehensive surveys of existing frameworks are already published by other authors, for example in [3]. So in this section we will mostly consider researches which involve not only software but also hardware of mobile robots.

Interesting approach was introduced in [4]. In this project the architecture of mobile robots (MRA) with standard set of hardware and software components was considered. Authors have described unified electrical and mechanical interfaces of robot modules and software libraries for distributed control and communication. Note that program structure of modular robot named MODBOT was built upon many levels which complicate the whole system. Inter modular communication was not considered in detail. The concept of MRA was perspective but proposed architecture was not implemented in real robotic solutions.

Object-oriented approach to mobile robots design was proposed in [5]. Certain type of modules according to its function covered as a class and specific instances of this class are modules of mobile robot. This approach allows achieving functional flexibility, scalability and allows applying fundamental properties of object-oriented approach to modular robotics. This framework must simplify auto configuration of modular systems software architectures. Authors do not consider real prototypes of modular robots but approach interesting because of abstracting from modules hardware.

The OPEN-R system was introduced in [6]. The OPEN-R is an architecture that provide a set of unified interfaces for physical and program units and software framework. The architecture has served as a facility for rapid development of new mobile robots especially for entertainment (robot AIBO ERS-110). Robots based on the OPEN-R were modular but they lacked distributed computations, because only one central processor in “body module” was responsible for system-wide control.

International project CLAWAR [7] was initiated in 1997 year and was coordinated by University of Portsmouth. CLAWAR project developed concept of functional modules, demands and specifications for their design. In this concept object-oriented approach was used for methodology of modular components design for mobile robots. CLAWAR project describes 3 main types of modules: human-computer interface, actuator modules and sensor modules. In comparison with other similar architectures Virk G.S. introduces a hierarchical modular architecture: the whole system is divided in modules which in turn are divided in sub-modules. The CLAWAR concept is interesting one but according to our survey this architecture has not been implemented in any of the modern robot solutions.

In [8] a modular approach for autonomous mobile robot design is introduced. This architecture is based on continuous modularity of hardware and software. The hardware is consists of four layers: driving layer, sensor layer, actuator layer and control layer. Robot’s communication between modules is based on CAN-bus. The software architecture is realized on middleware MIRO. The effectiveness of proposed approach was approved on RoboCup 2003 competition for which the robot football team was built within less than a year. One of the main advantages of considered architecture is a presence of distributed computations but there is no detail description of communication process between hardware modules.

Modular reconfigurable humanoid robots are considered in [9], [10] and [11]. In [9] distributed computations are performed on functional modules: Head, Main Module, Right/Left Arm, Wheel Module. Authors have developed own real-time communication interface Responsive Link for inter module interaction. Besides Responsive Link electrical communication interface also includes two USB ports and two IEEE 1394 ports. Using proposed architecture authors manufactured prototype of reconfigurable humanoid robot. Disadvantage of proposed architecture consists in requirement of special embedded processor Responsive Processor that can exploit Responsive Link interface. In [10] a similar
approach to modular humanoid robot design is proposed. All modules of robot are full functional; one of the main advantages of this system consists in ability of modules to communicate even if they have different OS. This is done by software architecture and special connection architecture which defines 5 standards of information models. Inter modular communication is based on TCP/IP for sharing data and RS485 for sharing hardware resources. Hild M. Spranger M. et al. also developed a modular humanoid robot Myon with distributed control architecture DISTAL [11]. The main features of the robot: autonomous work of its separate components (modules) and possibility of reconfiguration even if the robot is powered-on. For modules interaction authors proposed two communication buses – Spinal Cord (SC) and Extended Spinal Cord (XSC). The disadvantages of the system include lack of program interface between modules; also distributed computing is performed only on a level of actuators control.

A concept of a modularized service robot DRP I is introduced in [12]. The goal of the project is to propose a robot architecture that allows rapid development of new personal mobile robot. DRP I consists of full functional modules based on onboard computers and some special units for control and testing. The each module’s software consists of an OS level, robot virtual machine level, application programming interface (API) and application software. Robot’s middleware is based on IEEE 1394; also the hot plugging of robot modules was realized along with automatic reconfiguration of navigation algorithms. Despite all advantages of the system there are no known modern robot solutions that have been implemented using this architecture.

In [13] Magnenat S., Retornaz P. et al. propose a design paradigm of a distributed architecture of mobile robots control system - ASEBA. ASEBA is one of few systems that is fully decentralized and consists of microcontrollers which control sensors and actuators. All modules (microcontrollers) communicate with each other using CAN-bus. In ASEBA an event-based control is implemented which is due to ASEBA virtual machines (VM) that run on microcontrollers.

Another interesting project is a R2P [14] that is analogous to our work. R2P is the open source hardware and software framework that allows combining different modules (sensors, actuators) in a plug-and-play mode. R2P modules based on ARM Cortex-M3 microcontrollers that interacts through CAN-bus using RTCAN protocol developed by authors. Disadvantage of the RTCAN is a message fragmentation which can produce CAN frames colliding. Also it’s not clear on what principle the R2P program nodes must be deployed on hardware modules. Similar to R2P is a RoboCAN project [15] which considers a distributed modular CAN-based architecture for mobile robots. This architecture is integrated in the existing framework RIDE. RoboCAN modules use microcontrollers and support plug-and-play mode. Note that this architecture is based on custom protocol which is only slightly better than CANOpen (the RoboCAN bandwidth is about 25% less than the CANOpen when motors stopped).

Shmakov O. Korolev D. et al. developed the kit for rapid prototyping of modular mobile robots primary for education needs [16]. The robot kit includes different blocks-modules: control system module, power supply, power drive, communication, mechanical transmissions etc. Modules are connected using RS485 or directly to control system module. Kit’s modules have in-built electronic devices: for example instrumentation modules include sensors; power drive module includes gear motor etc. This approach allows easily assemble mobile robots like in LEGO kit but it will limit mobile robot designs (one may want to use more powerful motors).

An ambitious project is the Hardware Robot Operating Systen (H-ROS) [17]. The project goal is to develop the universal hardware-software framework for different robot components integration and interoperability for rapid development of mobile and industrial robots. H-ROS is based on ROS but uses functionality of ROS 2. Physical level of communication between modules is implemented using EtherCAT. Linux kernel with patch PREEMPT_RT is situated upon physical layer. The disadvantages of the system include requirement of special onboard computer in each module (for Linux and ROS 2 deployment) and use of expensive EtherCAT technology. Unfortunately H-ROS project now is halted.

Modular architecture of a mobile robot based on concept of the Operator-Controller Module (OCM) is described in [18]. The novelty of the work lies in a hierarchical structure of the robot and distributed computing which implemented on local part (robot itself) and cloud part. The cloud part allows adding a computing power for more complex algorithms meanwhile not using the robot’s limited power supply. Jahn U. Wolff C. et al. developed the prototype of the mobile robot DAEbot which distributed system includes two Raspberry Pi, Arduino Mega, STM32 Discovery board and FPGA ZynqBerry TE0726-03M. All devices except FPGA are communicates using CAN-bus with custom protocol. Note that this architecture doesn’t standardize robot’s components (modules) so it’s may be difficult to reconfigure mobile robot.

The brief analyzing of considered related works shows that despite presence of modular approach not all architectures and frameworks are mature enough to provide other researchers reliable tools for rapid robot development. Moreover some of the modular architectures have distributed computations only on a low level of actuator control – there are only few projects considering hierarchical structure. Also some systems require powerful onboard computers for modules.

3. Hierarchical modular architecture of mobile robots

A distribution of computational load of a mobile robot’s onboard computer may be performed by robot structure decomposition onto unified full-featured modules [19]. The research was supported by the RFBR: Grant 16-07-00811a. According to the proposed concept every module is functionally independent and solves strictly defined set of tasks. The full-featured modules united into one robotic system must interact between each other using special protocol [20] that standardizes modules command system, transmission data format, message priority etc. These modules are responsible
only for particular functions so they require less computational resources compared with a whole robot. Figure 1 shows general hierarchical structure of considered modular architecture.

Nevertheless some of the robot modules must perform significant amount of actions despite present function distribution between mobile robot units. For example the transport module is responsible for navigation in environment, obstacle avoidance, path following and low-level control of the actuators. The transport module performs all these tasks in cooperation with a sensor module that in turn is in charge of data fusion and complex algorithms implementation such as map building or particle filters.

![Diagram of modular architecture](image)

**Fig. 1.** The functional-modular architecture of a mobile robot consisting of full-featured modules: transport module, power module, remote sensors module etc.

According to our concept of ModRob every full-featured module itself must be built as a modular architecture with distributed nodes – sub-modules. Because of that modular robot can be represented as a hierarchical structure in which simple sub-modules form more complex modules as it shown on fig. 1.

This architecture should allow distributing the computational load of different functional significance between sub-modules, especially in cases where the number of actuators and sensors may increase or their type may change. This may be true for mobile robots off-road type. In addition, it is possible to combine sub-modules on a first level of hierarchy and modules on a second level of hierarchy.

Consider the distributed architecture of the transport module (fig. 2).

In this hierarchical structure the interaction of the transport module with the other full-featured nodes of the robot takes place on the bus network at the second level of the hierarchy (CAN, Ethernet, USB, etc.). It is assumed that this interaction can be carried out in a “soft” real-time (the sampling rate of the modules is not less than 10 Hz, but in general is determined by the required speed of the robot itself).

The transport module structure is based on three main types of sub-modules:

1. The Motherboard sub-module – responsible for control and setup of other sub-modules.
2. The Actuator sub-module – control attached electromotor (fig. 2 shows partial case – sub-module for DC control).
3. The General Sensor sub-module – allows connecting a variety of distance sensors, encoders, gyroscopes, etc.

The presence of a generalized sensor submodule in the transport module is optional; it can be used in situations where the sensor module (at the top level of the hierarchy) is not present in the mobile robot and there is a risk of collision with unexpected obstacles.

Note that set of sub-modules is not limited by these 3 devices – this list can be extended by special sub-modules – for example, camera sub-module.
Figure 2 also shows the Power Module that provides power to all connected devices. The power module is not a part of the transport module and is considered as a more specific node of the system, since it communicates with the modules (the second level of the hierarchy), but provides power supply to sub-modules.

![Image of the distributed architecture of the mobile robot’s transport module]

The interaction between all sub-modules must be performed in a “hard” real-time (the sampling rate is not less than 100 Hz) on a bus network at the first level of the hierarchy (CAN, Ethernet).

Let’s take a closer look at the structure of sub-modules.

4. ModRob hardware sub-modules

The Motherboard – is a computing unit which main functions are programmed by user. In a general case the Motherboard may be responsible for calculation of a transport module linear and angular velocities according to motion algorithm, getting data about robot’s real position and its comparison with a goal position, determination of the task feasibility etc. Besides of mentioned functions the motherboard performs a task of a mediator (bridge) between developer’s PC and other sub-modules. This is necessary for these sub-modules setup: setting PID controller coefficients of actuator submodule if needed, setting filtration and interpolation parameters for sensor, etc. The connection of the Motherboard with PC is performed using USB. The Motherboard also includes a mini Wi-Fi board for linking with remote server for wireless control.

The Actuator – is an electronic controller designed to control various electric motors: brushed and brushless DC motors, stepper motors, servomotors. In a transport module the actuators actually regulate various locomotors mounted on motors shafts: wheels, legs, tracks, etc. The sub-module main functions: motor’s shaft speed and position control, torque control. Also in certain cases some additional functions may be required: fixing the fact of a wheel/leg slipping, determination of a force and/or torque vectors applied at the locomotors. According to the proposed concept actuator sub-modules depending on a controlled electromotor type may be of four main kinds: a brushed DC motor controller, a brushless DC motor controller, a stepper motor controller, a servomotors controller.

The General Sensor sub-module (GSS) - is an electronic device designed for attaching of different sensors, mainly range finders, encoders, switches and inertial measuring devices (gyrosopes, accelerometers). It’s supposed that submodule must be compliant with the most common typical sensors. The main sub-module functions: collecting sensor data, sensor data processing and fusion. An important feature of the general sensor submodule is the ability to connect to it different models of sensors of the same type, for example, ultrasonic rangefinders. It is assumed that the user must configure the sub-module before installation: select the model of the connected sensor, the interpolation method, filtering, etc.; this configuration should be done using the Motherboard sub-module that connects to the user’s computer.

Figure 3 shows the computer models of the sub-modules considered above. Each submodule is a complete electronic device to which different controlled peripherals (motors, sensors, buttons, etc.) must be connected. All sub-modules are equipped with connectors for connection to the CAN network (various network topology options are possible).

Most of the different versions of a transport module require the Motherboard and the General Sensor sub-module, as well as a several Actuator sub-modules, a number of which depends on a number of controlled links of the mobile platform.
Fig. 3. Computer models of sub-modules: DC Controller, General Sensor Sub-module (GSS) and the Motherboard

Figure 4 shows the block diagram of the architecture of a four-wheel transport module, which consists of four Actuator submodules, that control geared DC motors, the Motherboard submodule and the GSS. This transport module can be used for example for a motion on a rough terrain in an unstructured environment.

Fig. 4. The transport module architecture for motion on a rough terrain

Depending on the specific application for which the transport module is assembled, the chassis design and the suspension mechanism will vary, but the hardware of the control system (sub-modules) must remain unchanged. The use of sub-modules should ensure the reconfigurability of the transport module: instead of wheels, one can use walking mechanisms. Also if the type of controlled electromotor is changed then the Actuator submodule must be replaced with an appropriate one.

5. Practical results

All ModRob sub-modules feature an STM32 ARM-Cortex-M3 with 72 MHz clock, 20 KB of RAM and 64 KB of Flash memory, a CAN driver TJA1050 and a voltage regulator (DC-DC step-down).

The Motherboard: the prototype of sub-module also include Wemos D1 mini platform for creating wireless connection between modular robot and user PC. For debugging and programming of the Motherboard and modules present at the CAN bus the UART is used.

The DC Controller: the prototype can drive DC motors up to 35 V, delivering a continuous 4 A current. Closed loop control with position and/or speed feedback is realized if an encoder is connected to the sub-module. The DC Controller publishes speed and position measurements on the bus; using the Motherboard the PID coefficients of the sub-module can be adjusted using the PC which can send messages to the CAN-bus.

The Power Module: the current prototype of this sub-module includes lithium ion batteries, two microcontrollers ATTiny85, DC-DC stabilizer etc. Output voltage lies in range of 10 to 16.8 V, continuous current is 5 A, short-time current is 10 A.

The GSS: the present prototype of GSS allows connecting various sensors: limit stop switches, buttons, encoders, sonars, IR sensors. It is possible to change type of connected sensor during the sub-module operation by sending special variable across CAN-bus. All digital sensors can be connected using groups of universal pins on GSS board, for analog sensors there is another groups. Also we added several pins for I2C connections. Currently the MPU6050 and MPU9250 inertial sensors can be connected to GSS via I2C. In the next iteration we will add support for other popular inertial sensors.
Figure 5 shows the several photos of manufactured sub-modules and modules.

![Figure 5. From left to right: DC Controller, the Motherboard, the Power Module](image)

Currently we work on an upgrade of present sub-modules (for example DC Controller with current feedback) and design new ones (BLDC Controller).

6. Conclusions and future work

In this paper we have shown that modular approach is used today for solving two main problems in robotics: integration of various hardware and software parts for rapid robot prototyping and reconfiguration and distribution of computation load between robot’s units for real-time operation of complex algorithms. Many researches and projects consider various architectures which are have some advantages and flaws. In this paper we propose a hierarchical modular structure which should provide unified components for robot development along with distributed computations for complex systems. In this architecture basic robot’s functions are distributed between modules which in turn consist of “smaller” sub-modules responsible for simple tasks with hard real-time constraints. Our modules and sub-modules don’t have mechanical links with devices which they control; this is important for a wider range of robot applications. We have briefly described structure of the transport module and its sub-modules which based on microcontrollers and are already implemented as laboratory prototypes.

Future work consists of a further developing of a hierarchical modular architecture, models and methods of task distribution on different control levels. Another important work is a testing of our protocol produced for the CAN-bus. It is important to conduct experiments which will prove that inter modular communications compliant with real-time constraints on both levels of hierarchy.

7. Acknowledgments

The Russian Foundation for basic research supports this research: Grant 19-07-00892a.

8. References


