A Health and Usage Monitoring System for ROS-based Service Robots

Eduardo Pinto¹, Pedro Deusdado², Francisco Marques¹, André Lourenço¹, Ricardo Mendonça¹, Pedro Santana¹, Luis Flores² and José Barata¹

¹CTS-UNINOVA, Universidade Nova de Lisboa (UNL), Portugal
²INTROSYS SA, Portugal
³ISCTE - Instituto Universitário de Lisboa (ISCTE-IUL), Portugal
emp@uninova.pt

ABSTRACT

This paper presents a multi-core processing solution for ROS-based service robots. The power management together with the control and availability of the processing resources are supervised by a custom-made Power Management Board (PMB) based on a Digital Signal Processor (DSP) microcontroller, implementing a Health and Usage Monitoring System (HUMS). The proposed architecture also allows for the PMB to control the most critical robot functions in case of low battery conditions or impossibility of performing energy harvesting, thus extending the lifespan of the robot. All PMB data is recorded on a SD card so as to allow offline analyses of the robotic mission and, thus, support subsequent maintenance activities. Two different implementations of the proposed system have been fielded in two Multi-Robot Systems (MRS) for environmental monitoring, covering aerial, water surface, and wheeled ground vehicles. An additional implementation of the architecture is currently being deployed on an industrial autonomous logistics robot. These three implementations are presented and discussed.

1. INTRODUCTION

ROS - Robot Operating System[1] has started a new age on robotics, promoting a new software framework that is becoming immersed in the research community as a de facto standard. The code modularity, the solution’s portability, together with an eased expandability, lead to robotics systems with a larger number of functionalities, sensors and hardware. Also, new advancements on using ROS in different cores/processors allow for the implementation of real-time ROS architectures[2] that are essential for many robot control applications (e.g. motion control, obstacle avoidance, timely sensor acquisition, etc.). Common to all of these applications is the importance of power management, health management and self-diagnostics. Only through those a system can achieve the reliability, safety, maintainability, availability, supportability, and economic affordability requirements of their applications. These functionalities are all established under PHM – Prognostics, Health Management, although many other designations exist according to different manufacturing domains. PHM is now widely used in different areas such as electronic components, aerospace, defense and military¹, renewable energies, batteries, mechanical systems, civil infrastructures, consumer electronics, computers and robotic[3].

In the last few years have witnessed an enormous increase in computing power with the availability of a wide range of multi-core microprocessors and Systems On Chip (SOCs) with very low Thermal Design Power (TDP). This associated to the fact that PC architectures became extremely reliable with the utilization of solid states memory, acting as secondary system memory, and the availability of new and distributed computing model operating systems², many of them totally open-source lead to the design of robots equipped with multi PC systems / multi processors. We can find this approach in many well-known robots: ASIMO from Honda, with 2 PCS (control, planning) and additional Digital Signal Processor (DSP) for sound processing[8]; Robonat from NASA and General Motors, with a highly distributed architecture with two central PowerPC processors (Sensor perception and safety system, control law and kinematics) and “Superdrivers” (FPGA with an embedded PowerPC) on each joint[9] or in the Japanese Humanoids Robotic Project (HRP-1 to HRP-4) with two on board PCs[10], only to name a few.

However, this increase in computer power was not accompanied by similar advancement in energy sources (e.g. batteries), making it essential to have a PMB capable of controlling sensors/computers/actuators based on the actual power-level capacity. The system here proposed is not only capable of power and processing capacity management but also incorporates some typical Health and Usage Monitoring System (HUMS) functions such as monitoring and recording parameters as temperature, vibration and power consumption of critical components of the system. A system like this offers several advantages[3][11] when applied to robotic systems as the ones depicted in section 3:

- Increase of system availability and reliability with the reduction of intermittent failure and No-Fault Found (NFF) and downtime times;
- Capacity to operate in low-consumption modes or even “go to sleep” modes to avoid very low battery levels;
- Anticipation of possible errors by integrating a prognostic software in the PMB that would constantly monitor a series of parameters and detects abnormal conditions of operation;
- Increase of the maintenance effectiveness and reduce of repair-induced failures by providing an easy identification of the malfunctioning component and even the cause for the detected error;
- Simplified wiring and more compact solution;
- Easily adaptable and expandable by incorporating in the design extra IOs and bus voltages.
- Enable predictive maintenance, often called Condition-Based Maintenance (CBM);

¹ U.S. Department of Defense (DoD) 5000.02 policy document on defense acquisition makes PHM a requisite for systems acquisition.

² Some examples are: MiRPA- Middleware for robotic and process control application[4], MRPS - Microsoft Robotics Developer Studio[5] or CARMEN[6]. An excellent survey can be found on [7].
However, some disadvantages should be pointed:
- More complex to develop and to “get-ready”. Demands for a custom PCB board with compatibility demands for several types of electromagnetic noise typical of robot actuators.
- The extra weight added could be critical especially in aerial vehicles.
- Extra components and development costs are necessary, but compensated by reduction of maintenance costs.

This paper is organized as follows. Section 2 is dedicated to present and describe in detail the generic architecture (hardware and software) proposed. On section 3 we present the practical implementation in 3 different robotic projects of the PMB architecture described in this paper. Many of the solutions presented here are open-source and schematics, layout files, etc. can be found on the site of our group under the respective projects. Finally, some conclusions and future work directions are drawn in Section 4.

2. PMB architecture

The architecture here proposed is divided in 3 basic blocks:
1. A custom designed PMB for each application. To decrease the developing time, electronic blocks are developed around the same microcontroller, allowing for reuse of hardware and software.
2. A set of PCs based on commercial solutions powered by DC/DC intelligent power supplies. This allows a decrease of cost when compared with industrial PCs and use of the most advanced PC processors solutions in the market.
3. A set of ROS nodes and drivers that allow distributed computing and share computer processing resources.

1.1 Hardware

The PMB is designed around a 16 bits Digital Signal Controller (DSC) from Microchip™ dsPIC®33F/E line of products (Figure 1). The selection for this type of microcontroller was based on the high number of peripherals available, 70MIPS, the large pin (up to 144) packages highly configurable with Peripheral Pin Select, the very low power modes with several wake-up modes, allied to a large range of supporting libraries.

All DC voltages necessary for the components in the system are generated using isolated high efficiency controllable DC/DC converters with the critical controls opto-isolated. Parameters such as temperature, voltage, current for each converter and outputs (smart switches) are stored with the DSC real time clock timestamp together with data from the Inertial Measurement Unit (IMU) MPU-9150 IC, on a SD card. This allows the implementation of true PHM analysis on all the critical components of the system. The board also serves as an interface between the high-level software modules to all the actuators’ signals. This way the PMB can attend to all the critical system functions and become the ultimate control center, in the case of low battery levels. When the battery level reaches a dramatic low level, the PMB sets the system in a sleep mode to avoid additional problems, enabling an external signal to wake it back up at any time. To ease the system’s maintenance, the board incorporates a set of displays and leds for each voltage/output/input that allows an easy and immediate visualisation of a problem (e.g. voltage not correct or fuse broken). For possible expansions, a wide range of communications is available: CAN, RS485, RS232, USB, USB (serial FTDI), Serial 3.3V/5V, 16 bits PWM signals, PWM capture signals and several protected configurable digital and analog input and outputs. The battery management system is also connected to the PMB allowing for an easy read of the battery cells’ status.

![Figure 1. PMB hardware architecture.](image-url)
2.1 Software architecture

The PMB/HUMS can be used as a self-contained unit, managing all the necessary components by providing power supply and monitoring. However, the main objective of its development is to allow higher-level software modules the access to the necessary data and control of all the robots’ components. For the implementation of these high-level functions, ROS, designed to be used on multi-language distributed computing systems, was chosen. It is built upon the concept of a basic processing unit, called node, being as much independent from the hardware as possible, and holding no assumptions about where it will run in the network, allowing computation to be relocated to match the available resources and implement true distributed systems.

![PCs block architecture.](image)

Figure 2. PCs block architecture.

The PMB/HUMS can be used as a self-contained unit, managing all the necessary components by providing power supply and monitoring. However, the main objective of its development is to allow higher-level software modules the access to the necessary data and control of all the robots’ components. For the implementation of these high-level functions, ROS, designed to be used on multi-language distributed computing systems, was chosen. It is built upon the concept of a basic processing unit, called node, being as much independent from the hardware as possible, and holding no assumptions about where it will run in the network, allowing computation to be relocated to match the available resources and implement true distributed systems.

![An overview of the integration of the Power Management Board in the software architecture.](image)

Figure 3. An overview of the integration of the Power Management Board in the software architecture. Blue arrows depict each component status information relayed to the system via Power Management Board. Orange arrows depict the sensor data streams or actuator commands. Black arrow depicts the communication between the hardware interfaces and higher-level nodes.

The PMB is physically connected to high-level processing units via a serial port and has an interface node exposing its functionalities to the ROS network. The PMB not only provides status information about all the components that are connected to it but also allows their management. Each component, either sensor or actuator, has basic services and information that is provided by the PMB. On one hand, the information from those services is used to compile detailed diagnostics data like operation status, temperature and voltage. On the other, the services are used to manage the power supply from the top-down. Finally, it allows other ROS nodes to control actuators that are connected directly to the PMB, like, for instance, traction motors.

This simple interface allows the creation of power profiles to match the robot’s varying operation requirements, on a higher level reasoning stage of its decision making. There, the robot’s situational awareness comes into consideration to determine which hardware elements should be switched-off, and which are kept online for an optimum energetic consumption. That means the context that it perceives from the environment, its own energetic status as well as its operation energetic requirements come all into play on a supervisor layer, or meta-package. For instance, a water surface robot’s perception layer can provide data of the surrounding environment like water currents and swell and the PMB the current power consumption and battery state of charge. Information that can be combined to reason a more energetically efficient path prolonging, in the long term, the operation time. Furthermore, in certain situations the system might reason that is safer to stop operation altogether, and lay in a dormant state until the conditions become more agreeable.

A ROS-based architecture also fosters additional load sharing and energy saving. The possibility of having distributed modules extends techniques such as the one presented in [12]. Allowing not only the choice of the most efficient navigation mode for the environment but also the ability to turn on and off software modules and hardware components. In most practical implementations described in the next section the high-level processing is ensured by more than one computer. This, in addition to providing redundancy in case of malfunction, also makes it easier to escalate processing power if needed. The main computer might only contain essential functions with the others turned off as the PMB allows the system to turn on the additional units on demand. If a certain environment fosters the use of a dedicated software module the system will check the current processing load and available resources to decide if and where to run it.

3. PRACTICAL IMPLEMENTATIONS

In this section we present three different implementations of the approach described in section 2. More detailed information, including field results, can be found respectively in the following papers: ECHORD-RIVERWATCH[13][14], ROBOSampler[15], and will not be reproduced here due to space restrictions. The third application reported is, at the time of writing of this paper, still in development, namely the drawings are being simulated and validated. Since this is a very different application from the first two, we think it is justifiable to mention it on this paper.

3.1 Project ECHORD – RIVERWATCH

RIVERWATCH (Figure 4) is an innovative cooperative robotic system composed by two heterogeneous robots capable of carrying out missions in estuarine environments maintaining a symbiotic relation. This innovative approach makes the robotic system RIVERWATCH, as far as we know, the first aerial-
surface robotic team performing cooperative perception for autonomous safe navigation in riverine environments.

Figure 4. Main sub-systems RIVERWATCH (A) Communications. 2Wifi dual channel: 2.4Ghz, 5Ghz, 3G, 24Ghz RC, GPS RTK; (B) Pointgrey Ladybug 3 Spherical camera; (C) Airmar PB 200 Solid state Weather station; (D) Sick D-LRS2100 Lidar; (E) 2 x 2Hp Haswing Protractor 102lbs brushless motors; (F) UAV: Hexacopter; (G) Control box: BMS, PMB, PCs, Router & Switch, Proplex 800 GPS RTK; (H) Flir Quark 640 thermal camera; (I) Webcam and GoPro3 cameras; (J) Starboard and port battery banks: 25.6V 100Ah; (K) Airmar CS4500 Ultrasonic water speed; (L) Imagenex Delta-T 837B Multibeam sonar.

By using two heterogeneous robots, an Autonomous Surface Vehicle (ASV) and a multi-rotor Unmanned Aerial Vehicle (UAV), we have attained an extremely flexible and reliable solution. The team combines the UAV’s higher vantage point with the ASV’s endurance and payload capacity, extending both vehicles’ capabilities for adaptation and self-organization to operate in demanding environments. Still, considerable attention has been given to each team member’s limitations, and how those may weaken the ensemble. For instance, aerial platforms have considerably lower energetic autonomy and robustness to demanding weather conditions than surface vehicles. The solution was to implement a symbiotic relationship between the two, where the ASV benefits from the augmented perception of the environment[16] afforded by the UAV’s aerial survey and by the cooperative perception approach, merging the field of views of the two robots. On the other hand, the UAV takes advantage of an extended energetic autonomy endowed by the possibility of recharging using the power management and energy harvesting mechanisms installed in the ASV. Furthermore, landing aboard the ASV fosters the UAV with the necessary protection in case of adverse weather conditions.

We can see in Figure 5, a direct implementation of the architecture described in section 2. All the control electronics including the PMB, routers and switches, three i7-3770 Ivy Bridge based PCs and programmable M4-ATX power supplies, were enclosed in a rugged waterproof hermetic Peli-Storm iM3220 case. This casing protected and waterproofed the electronics but brought up a delicate issue of how to dissipate all the generated heat inside the enclosed box. Our solution, was to use a water cooling with waterproof fans (Phobya Nano-G 12) and a radiator outside the box which is an innovative solution for mobile robots. The use of easily available commercial products from Aquacomputer offer the advantage of incorporating temperature sensors on all the water blocks and flux sensors, that is an essential requisite to implement PHM functionalities.

Figure 5. RIVERWATCH control center with PBM board and water cooling apparatus. (A) Peli-Storm waterproof box; (B) Proflex 800 GPS RTK; (C) 3x i7 Ivy Bridge computers (D) Router and switch; (E) PMB board; (F) Water-proof fans and radiator (G) commercial water cooler block for Dc/Dc converters (Xbox-360).

3.2 Project ROBOSampler

The approach that we introduced in RIVERWATCH design by SRS “Symbiotic Robotic Systems” is explored also in the project ROBOSampler. Here an aerial-ground field robotic team (Figure 6), was designed to collect and transport soil and biota samples in estuarine mud flats by using a custom made drilling tool. The system is composed of an Unmanned Ground Vehicle (UGV) and a UAV.

We can see in Figure 6, the main sub-systems of the Robot Sampler. All the control electronics including the PMB, routers and switches, three i7-3770 Ivy Bridge based PCs and programmable M4-ATX power supplies, were enclosed in a rugged waterproof hermetic Peli-Storm iM3220 case. This casing protected and waterproofed the electronics but brought up a delicate issue of how to dissipate all the generated heat inside the enclosed box. Our solution, was to use a water cooling with waterproof fans (Phobya Nano-G 12) and a radiator outside the box which is an innovative solution for mobile robots. The use of easily available commercial products from Aquacomputer offer the advantage of incorporating

---

2 http://introsys.eu/robosampler/
The UGV is equipped with individual 250W steer and 550W drive motors in each wheel, (motor drivers are part of PMB board), providing quasi-omnidirectional locomotion capabilities, with several possible modes. This multi-modal locomotion is especially important to deal with locomotion problems typically brought by a muddy environment, and to align the robot and the drilling tool with precision to a specific sampling spot. Several parameters from the drilling tool, as temperature, voltage, current, control signals are monitored by the PMB to detect and alert possible drilling problems. The same is done with the compliant 6 degrees of freedom robotic arm, Universal Robotics UR5 that is used to move metallic tubes and dredger between their storage sockets and their corresponding sampling tools. All of the controls signals (e.g. emergency and safeguard stop, remote on/off, safety signals, etc.) and power are monitored by the PMB board.

In Figure 7 we can see the practical implementation of the architecture described in section 2 for the ROBOSampler project. The design of the PMB had to take into account that all the electronics will be in closed boxes that are part of the chassis robot. For power dissipation, we used custom made heat-pipes to transfer the heat from the PCs components to the chassis. During the development of the PMB we had to take special care to locate critical components (e.g. motor drivers) on the bottom face of the PCB, using a complete manual drawing process. This allows a direct heat transfer to the robot chassis that is made of an aluminum alloy and that simultaneously reduces the robot weight and increases thermal conductivity, making all the robot chassis a large heat sink. Due to limited space in each of the two compartments of the robot, it was necessary to divide the PMB board in two, with one of them only acting as a SLAVE passing information by a serial channel (converters status, current, temperatures, etc.) to the MASTER, that is in charge of ROS interface and system control. For design simplicity both use exactly the same architecture described in section 2.

Figure 7. ROBOSampler control centre and PMB board. (A) i7 Ivy Bridge computer and Switch, (B) PMB; (C) Motor drivers (part of PMB); (D) Manual remote control; (E) Detail of use of chassis robot as a big heat-sink.

### 3.3 Project ROBO – PARTNER3

FP7 ROBO-PARTNER is focused on the integration of the latest robotics solutions into industrial automation systems, namely for assembly operations. By combining robot strength, velocity, predictability, repeatability and precision with human intelligence, skill and dexterity, an increase in productivity is expected, leading to a new fenceless paradigm of robotics in industry. The robot here presented is part of the application scenario in automotive manufacturing that will involve the assembly of a vehicle’s rear axle where autonomous robots are to assist in the shop floor’s logistics, sharing the workspace and cooperating with human workers.

The design of this robotic platform is yet in the development phase with the validation of the 3D models for the mechanical part and their robustness using standard finite element analysis in SolidWorks. This innovative design will be object of a future publication. The electronics architecture will be based on the solutions adopted for the two previous projects, and centered on a custom developed PMB board and two commercial PCs platforms with the following components already selected: MSI-Z97-AC motherboard, i7 4770S processor with a Haswell architecture, GSkill 16Gb TridentX DDR3, SSD SanDisk Extreme Pro 240Gb and the proved M4-ATX intelligent power supplies.

### 4. CONCLUSIONS

A ROS-based hardware architecture for service robots was presented, as well as its instantiation to three heterogeneous multi-robot systems. These robotic systems range from aquatic surface vehicles to ground vehicles for natural outdoors environments (e.g., lakes, estuaries), as well as for industrial indoor facilities (e.g., car factories). The ability to handle the diversity imposed by these disparate robotic systems and corresponding operational environments highlights the robustness and flexibility of the proposed architecture.

As future work, we pretend to change the microcontroller used on the PMB board to an ARM® Cortex®-M from STM32F7 line of products from STMicroelectronics for more computing power: 462MIPS allowing the use of RTOS (e.g.) for the use of several operating systems (e.g. TI-RTOS, FreeRTOS™) that will speed the development time. We are at the moment starting the development of a custom PMB board.

---

3 http://www.robo-partner.eu/
that interface the main power source (Generator or a Li-Po battery) and aircraft electronic devices, providing redundancy for the essential systems (Auto pilot, Flying Actuators, etc.), being one of the first application for a Watertight UAV [17]. For UAV applications is essential to have some kind of redundancy and watchdog system, where the STM32F7 will be of great importance thanks to the hardware mechanisms incorporated and processing power available.

ACKNOWLEDGMENTS

Project ECHORD-RIVERWATCH was co-funded by EU FP7 ECHORD project (grant number 231143) and by the CTS multi-annual funding, through the PIDDAC Program funds. ROBOSAMPLER project was co-funded by Community Support Framework Program (LISBOA-01-0202-FEDER-024961).

5. REFERENCES


