Mobile Robots Design and Implementation: From Virtual Simulation to Real Robots

Fernando Osório, Denis Wolf, Kalinka Castelo Branco, Gustavo Pessin

USP – Universidade de São Paulo
ICMC - Instituto de Ciências Matemáticas e de Computação
SSC - Departamento de Sistemas de Computação
São Carlos, SP - Brazil
E-mail : { fosorio, denis, kalinka, pessin } @icmc.usp.br

Abstract: This paper presents an analysis of mobile robot designing focusing on its computational components. More specifically, we address: design methodology, system requirements, architectural components, and virtual simulation. Considering the recent growing in the use of mobile robots for domestic and industrial applications, it is important to review and to structure the main concepts related to the design and implementation of these complex systems. In order to meet the requirements of the consumer market, mobile robots should be adequately designed incorporating aspects such as: modularity, software reuse, safety and fault tolerance. We propose a general framework for designing, building and testing mobile robotic systems using a virtual simulation environment.

Key words: Autonomous Mobile Robots, Tele-Operated Mobile Robots, Virtual Design, Simulation, Robot Control System Architecture.

1- Introduction

Nowadays, mobile robots are becoming more and more usual in our domestic and professional environments. Starting from small robots like autonomous vacuum cleaners (e.g. iRobot Roomba) and lawn mowers (e.g. Husqvarna Auto-mower), up to more sophisticated autonomous security and service robots[1]. According to a new market research report, "Global Service Robotics Market (2009-2014)", published by MarketsandMarkets[2] , the total global service robotics market is expected to be worth US$21 billion by 2014. On the other hand, although the demand for autonomous robot applications and solutions keeps growing, the development of such systems still requires a long product development time. It is also considered a very complex task, due to the several requirements of hardware and software integration and even due to safety aspects (since mobile robots usually have direct contact with humans).

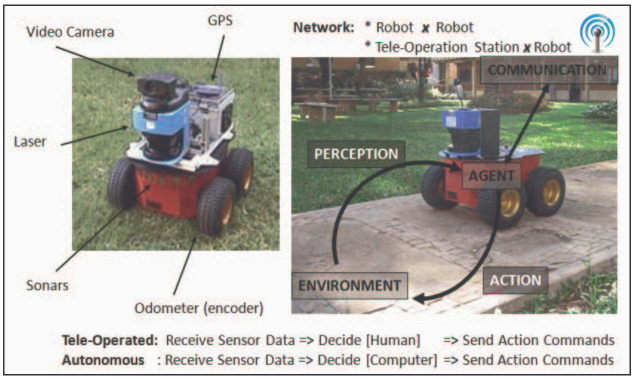

Figure 1: Mobile Robots – Sensors, Actuators and Control (Tele-Operation and Autonomous)

In order to design, build and test mobile robot systems, we should be able to select, integrate and configure the hardware, and then develop the software to operate different types of sensors, that range from sonars, infrared, lidars, cameras, and other devices. It is also necessary to consider the different types of locomotion systems, based on wheels, tracks, and even on legged walking systems [B1,DJ1,SN1]. Mobile robots are usually equipped with contact, distance and visual sensors that enable them to perceive the environment and to avoid collisions, allowing for safe autonomous navigation. Some more complex applications requires advanced capabilities like map building, auto-localization and trajectory planning [L1,TB1]. Robots can be tele-operated and/or have an autonomous control (Figure1), but even tele-operated robots are expected to avoid collisions and do not cause damages to objects or people that are around them [AS1].

Mobile robotic platforms and applications design requires the development of different hardware and software modules, and also interfaces that connect users, robot control systems (mobile agent) and the embedded hardware of the robot. In

[1] Technical Committee for Service Robots of the IEEE Robotics and Automation Society. http://www.service-robots.org/applications.htm
[2] MarkestandMarkets: Global market research and consulting - http://marketsandmarkets.com

the same way structured design techniques are nowadays adopted for embedded systems development mainly based on Hardware and Software Co-Design Paradigm [SW1,D1], we propose to adopt similar techniques to develop mobile robots (although those techniques have to be adapted to take into account some particularities of the mobile robot systems requirements). One key feature of Hardware-Software Co-Design techniques is the capability of integrated system simulation during design, allowing a better components integration (Hardware-Software) and adaptation of the system functionalities to the user-defined requirements.

In the mobile robotics design particular case, the modelling and simulation of the system can be successfully done using Interactive Virtual Reality Simulation Tools. Adopting 3D Realistic Virtual Environments to model and simulate mobile robots, testing and evaluating the different aspects of the system before implementing it physically is an efficient manner to decrease the development time. This type of approach is becoming more and more common not only in the robotics design but also in other industrial products design [OM1, OM2, F1, F2].

The main goal of this paper is to present an analysis of the design of mobile robots systems and applications, including the following items: design methodology, system requirements, architectural components and virtual simulation. The mobile robots design should also incorporate different aspects related to: modularity, software reuse, easy-of-use, rapid application development, safety and fault tolerance. We propose a general framework for designing, implementing and testing mobile robots using a virtual simulation environment, considering the above aspects. This paper is organized as follows. The next section (Virtual Simulation Tools) presents some widely used robot simulation tools, and describe the aspects that should be included in the virtual simulation tools in order to reach the main mobile robots system design requirements. In Section 3, we propose a framework with different components that should be included in a Virtual Reality Design and Simulation Tool for Tele-Operated and Autonomous Mobile Robots. Section 4 concludes the paper and discusses about further research directions in mobile robots design.

2- Virtual Simulation Tools

As robotics is invariably becoming part of our everyday life, real situations demand mobile systems capable of acquiring information of their own state and of the surroundings, interpret the data obtained, and make decisions to accomplish tasks. The more realistic are these tasks, the more complex are the designing and building of these machines. Although there are several mechanical challenges in creating robots that are capable of moving around and climbing stairs inside our house, most of the difficulty in the development of robots is related its intelligence and decision making capabilities. In this context, computer scientists have a major role in developing techniques that enable robots to think. Obviously, this is a major challenge in the field that is not going to be completely solved in the next decades. On the other hand, several solid advances have been obtained by the scientists in the last 20 years.
Differently from most computer science areas, robotics deals with physical environments and real situations. This certainly makes its development hard and slow as it is necessary to consider several aspects such as: battery charge, adequate connection of sensors, actuators, and computers, network accessibility, and even the weather conditions in outdoor experiments. Nowadays, the time to market is absolutely essential for a success of a commercial product. A single bug in the control software can lead to a robot colliding to an obstacle or falling from the stairs, which can result in hundreds or thousands of dollars in expenses due to a broken laser sensor in the experimental tests. In order to make the development process faster and easier, and to avoid such cases, simulation has a major role in mobile robots development. Of course, the simulation has to be as realistic as possible and its integration with real systems is essential to fulfil the requirements for its use in the robot development process.

Over the years, robot simulators have become more and more accurate in representing real environments, sensors and actuators. Nowadays there are several simulators available for virtual experiments with a large number of robotic platforms and sensors. Some of them are proprietary software and have to be purchased. Others are open source code, freely available for download in the internet. Probably the most used platform used by scientists and robotics developers is Player/Stage/ Gazebo[3] [VG1,GV1,GV2,CM1].

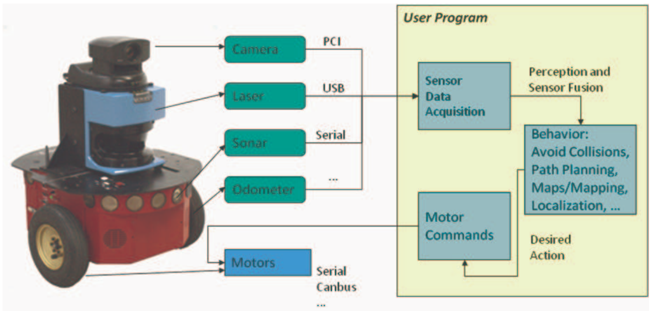

Figure 2: Mobile Robot based on a complete user control system development

2.1 – Player-Stage-Gazebo Software Architecture

In the earlier stages of mobile robots development, all the systems should be implemented by the user, including device drivers for different sensors and motors, as showed in Figure 2. As the development of mobile robot platforms become more complex, the reuse of software modules became indispensable, and software packages like Player appeared [VG1,GV2], allowing the user to reuse device driver components, and simplifying the task of creating a mobile robot control system (see Figure 3). The user has access to an API interface, with an easy configuration and use of different types of hardware devices. Using systems like Player, the user/programmer can focus his attention on the high-level control systems tasks development: environment mapping, planning trajectories using maps, auto-localize the robot in the environment, avoid collisions and obstacles, etc.

---

[3] Stage and Player are freely available under the GNU General Public License from http://playerstage.sourceforge.net.

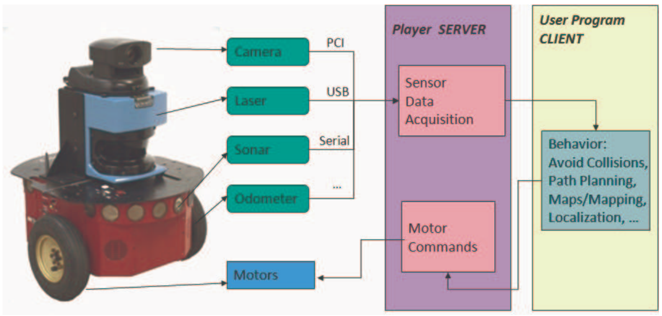

Figure 3: Mobile Robot based on the use of player device drivers and specific user control system development

Player is a server for robots and sensors control developed for Unix-like systems. It provides simple and efficient interfaces to robots and sensors over the network. Application programs (clients) can obtain data and send commands to robots and sensors using well defined and high level functions without to worry with hardware connection issues. All the hardware access is done by the server, which have specific drivers for each sensor or robot supported. Different devices of the same type have the same client interface, making the replacement of hardware parts a transparent task.

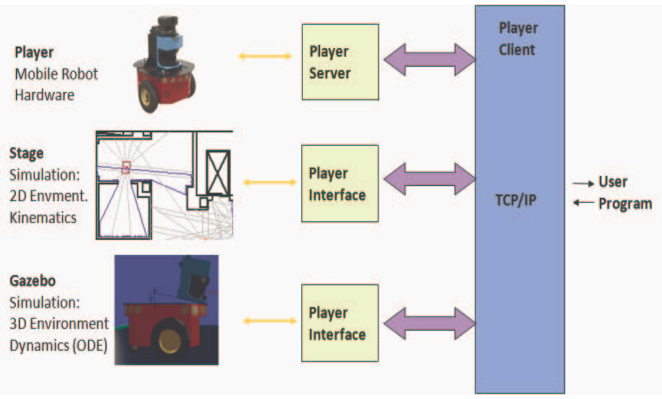

Figure 4: Player-Stage-Gazebo Architecture: Controlling real and virtual (simulated) mobile robots

Player was primary designed to interface to real hardware but it is also fully compatible to virtual robots and sensors through the use of the Stage and Gazebo simulators. As the client/server communication is performed over the network, it is completely transparent to the client (high level control software) whether it is connected to real or simulated devices (see Figure 4).

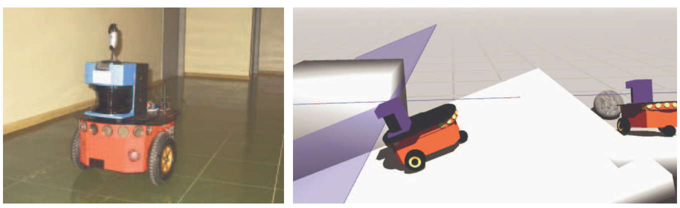

Figure 5: Pioneer DX robots: real and simulated

While Stage simulates robots in two-dimensional spaces, Gazebo is capable of recreating complex 3D environments. Through the use of the ODE library [OD1], Gazebo can simulate the physical interaction of several robots and sensors in realistic environments and situations. Its seamless integration to Player makes it a very interesting platform for robotics control software development. Figure 5 show real and simulated Pioneer DX robots using Gazebo. Figure 6 shows a robotic vehicle. It is possible to notice that fine details of the environment and represented in the virtual (simulated) world.

Figure 6: Robotic vehicle in an urban environment: real and simulated

The mobile robotic platforms presented in Figure 5 and 6, represent the research efforts of two working groups (WG) of the INCT-SEC[4] (Brazilian National Institute of Science and Technology in Critical Embedded Systems). The Working Group WG1 aims to develop indoors tactical robots (monitoring and security robots), tele-operated and/or autonomous, based on platforms similar to those presented in Figure 5. The WG2 intend to develop computer assisted driving systems and unmanned ground vehicles (autonomous), based on platforms like those presented in Figure 6 which is being developed at USP (Univ. of Sao Paulo) in collaboration with SENA Project[5].

The INCT-SEC research projects have some constraints that are not entirely satisfied by the use of simulation tools like Player-Stage-Gazebo. These constraints are mainly associated to critical aspects related to communication, safety, security and fault tolerance.

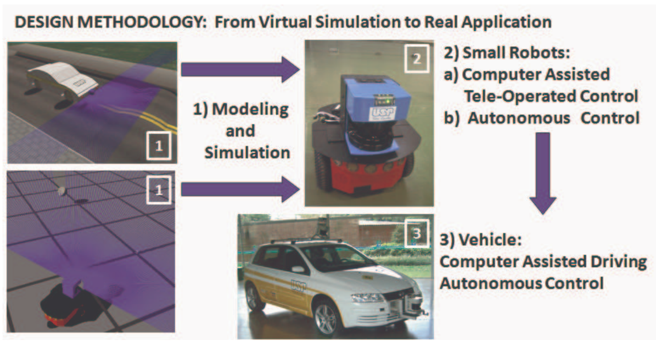

Figure 7: Mobile Robots Design Methodology

2.2 – **Virtual Simulation**

In order to develop the tele-operated and autonomous mobile robotic systems (like those under development at INCT-SEC presented in the previous section) initially a virtual model should be created, and the automated control systems should be implemented and tested using a virtual simulation tool.

[4] INCT-SEC: http://www.inct-sec.org/actrep/
[5] SENA Project:
http://www.eesc.usp.br/sena/url/en/index.php

The mobile robots design methodology is presented in Figure 7, consisting of an initial virtual simulation, followed by the adaptation to the real robot/vehicle. Based on the previous experience with Player-Stage-Gazebo and considering the system constraints and functional specifications, a list of requirements was defined to a virtual simulation robotic design tool:

• **3D Virtual Environment**: the simulator should be able to simulate 3D scenarios, since the mobile robots will operate in a real environment. The Virtual Reality Simulation Tool should be realistic, including the positioning and simulation of sensors. The sensors should be able to detect 3D objects present in the environment, as in a real system. Note that the laser positioning and attachment in the robot is also very important, once the laser beam is directed according to its spatial orientation. Different types of objects and terrains should be also easily modelled;

• **Sensors and Actuators Simulation**: the simulation of sensors and actuators should be realistic, which means that they should be imprecise/noisy as the real devices are (include an error model in the device simulation);

• **Physics Simulation**: the real world is governed by physical rules, so the virtual environment should also respect similar rules. The mobile robots should move respecting the kinematic model of different robot configurations (e.g. differential holonomic robots, steering wheel robots/vehicles). Also, the robots should behave according to the system dynamics, including the application of forces, like: accelerate, decelerate, inertia, gravity, friction and collisions/reaction to collisions. The ODE engine (Open Dynamics Engine) [OD1] is a well known tool included into robotic simulation tools in order to simulate rigid bodies' dynamics;

• **Hardware Device Abstractions**: the hardware device abstractions are a key concept for modular, portable and reusable robot code. The software implementation of hardware device drivers is a complex task, then the adoption of complete and previously tested software libraries that provide a hardware device abstraction through a simple program interface, is something really desirable in a robotic development tool. The client-server architecture adopted by Player-Stage-Gazebo presents a good solution to this problem, since it is possible to access the hardware devices through high-level functions provided by this tool. It is also possible to access virtual simulated hardware devices (simulated sensors and actuators), simplifying the development task: we can easily switch from using virtual devices to real devices, using the very same mobile robot control system code;

• **Robot Control Basic Functions**: some basic behaviours and robot functionalities could be provided by a robotic design tool, including: avoid collisions, avoid obstacles, move towards a specific direction, use of maps (e.g. geometric/grid maps, topological maps), environment robot mapping, path planning algorithms, auto-localization and pose estimation (e.g. EKF, Markov Localization, MCL), SLAM, etc.

• **Supervisory Level for Safety (safe behaviours)**: in the tele-operation control mode, it is very usual that a remote human operator make mistakes when interpreting the available sensorial data and deciding which commands to send to the robot. The robot should have a low level supervisory control system that avoids collisions or any other type of action that conducts to physical damages. Collisions and other improper actions can be caused by an incorrect action command but sometimes the robot has sufficient local sensorial information to detect and take counter-measures against potentially harming situations and actions. This supervisory level for safe behaviours should be included into the "kernel of the robot operating system";

• **Communication Facilities**: the robotic design tool should also provide means to simulate and also to access different real network devices and communication services, allowing to connect the tele-operation base to the robot and also to connect the robots into a client-server or peer-to-peer network. These questions will be discussed in more details in Section 2.3;

• **Security**: what are the major problems posed if someone not authorized gain access to the robot data (e.g. position, video stream, status)? What can happens if someone not authorized gain permission to send commands to the robot? These questions are very important, since as indicated in the beginning of this paper, mobile robots usually have direct contact with humans and can cause damages/hurt to objects/people. These questions will be also discussed in section 2.3.

• **Fault Tolerance**: using virtual simulation tools are a very efficient manner to investigate the system behaviour in presence of hardware and software faults. The virtual design and simulation tool should provide means to introduce errors in the system in order to test and validate the system fault tolerance.

Unfortunately none of the robotic simulation tools and systems available nowadays[6] do not provide a complete solution, considering all the above aspects.

2.3 – **Aspects Related to Robot Communication**

In general, robotic software tools available do not offer all the features needed to ensure safety in the communication. Although these are well known concepts in the networking area, authenticity, confidentiality and integrity of information are usually not considered in other fields like robotics. In this sense, Figure 8 illustrates the inclusion of secure communication facilities in the Player-Stage based model.

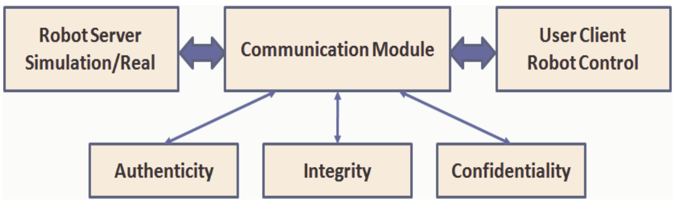

Figure 8: Safety in network communication

Authenticity: is the assurance of the correct identity of both parties involved in the communication.

[6] We are considering here the "open" robotic systems and tools available to be used in research and development. Commercial products sold as a "black-box" were not considered.

Integrity: is the assurance that the data being accessed has neither been tampered with, nor been altered or damaged through a system error, since the time of the last authorized access.

Confidentiality: is the assurance that only the authorized parties have access to the information exchanged. The information cannot be intercepted by others or at least cannot be understood when intercepted. The authenticity of the entities is essential to achieve confidentiality. It is imperative to identify the other party involved in the communication to establish a common cryptographic key that allows the use of secure encryption.

Providing safety for this type of system is one of the main goals of the virtual design environment proposed here. To illustrate the importance of this subject, we can analyse three different scenarios:

• *Multi-robotic Squads* - when you have multiple robots connected by a communication network, usually ad hoc, the information exchanged by these robots is usually confidential and may not have its content modified or even accessed by non authorized persons. Supposing the case where a squadron of robots is used for monitoring a security area. The mission information of the robots cannot simply be changed by unauthorized persons, which certainly leads to authentication needs. Therefore, a middleware network layer should provide that authentication making it transparent to the end users. Similarly, the information exchanged by the robots should be integral and reliable, as well as the missions' details sent to them, which suggests the existence of an encryption system based on keys that guarantee both the integrity and confidentiality of information transmitted from the robots to the base station.

• *Convoy* - the convoy of autonomous vehicles is something that is being increasingly used to transport cargo in order to save money and improve safety. This has caused automobile manufacturers to invest in proprietary communication protocols for this type of application. These protocols should take into consideration security issues and have been studied and exploited. This example illustrates the needs and bases presented here. The network framework should be simple enough to be transparent to the user and allow him to choose which level of security will engage in its application. In the case of convoy, integrity is needed to ensure that the information exchanged between the elements of the convoy is reliable and do not allow it to be intercepted by an unauthorized entity. The unauthorized access could cause an incident, for example, an element of the convoy can change the direction or even be sent to another destination. Moreover, imposing security usually degrade (adds overhead information) the overall system performance. In the case of a convoy, this creates a delay in communication resulting from the processes of encryption and decryption. Due to this fact, if information does not arrive on time, it can result in a system crash. If those issues are transparent to the user, it is not necessary to understand encryption, authentication, or any other process that ensures safety. It should transparently add these features to the applications in the same way that currently a user of player-stage does not need to know exactly how to start a laser or service to access it.

• *Autonomous/assisted* Systems – A vehicle could be autonomous or assisted, and in both cases we need that the information exchanged between the base station-car and car-road is transmitted securely. The misinformation and interferences of unauthorized entities could cause accidents (collisions, pedestrian deaths, among others). Therefore it is a requirement that the autonomous or assisted drive is guaranteed by cooperative safety systems, using a safety communication network. In addition to that, the information used both in assistance and autonomous systems can be categorized by its degree of dynamism: Permanent – mission information; semi-dynamic – road work zones (in this case we can use WLAN to provide the information); Dynamic – hazard notification provided by DAB transmissions; and more than Dynamic – emergency information provided by consumer to consumer (direct link between vehicle and base station).

The in-vehicle wireless network considered is composed of wireless sensors located in the vehicle and nomadic devices of the driver and/or passengers. Our framework considers radio interfaces for in-vehicle communications, positioning (Satellite), inter-vehicle and roadside communications, Wi-fi and Wlan communication, and communications with cellular networks. Mature technologies are thought, such as Bluetooth, Zigbee, GPS, RFID, WiFi or UMTS, and WiMAX.

3- Virtual Design Framework

Considering the previous discussion related to simulation facilities, robot control modules, security issues, and also fault tolerance aspects, we propose a Framework for Virtual Design of Mobile Robots. The proposed framework is presented in Figure 9.

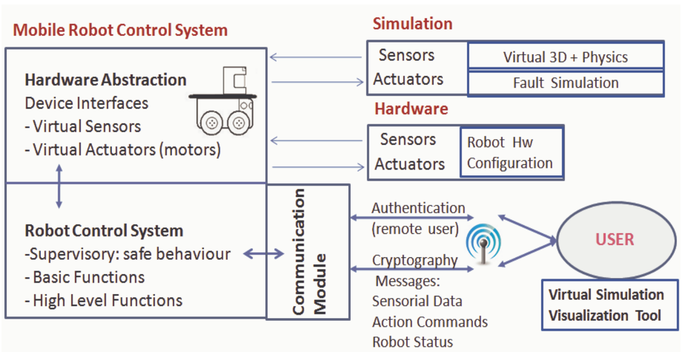

Figure 9: Framework for Virtual Design of Mobile Robots

The main components of this framework are:

- **Simulation of Sensors and Actuators**: responsible for reproducing the behaviour of sensors and actuators, based on a physical simulation tool. This simulator can also be responsible for generating simulated faults in order to test the system fault tolerance;
- **Hardware Drivers**: allows to configure the robot hardware and provide device drivers to directly access the robot hardware components (sensors/actuators);
- **Hardware abstraction device interface**: provides access to virtual devices, so the application has access to a simpler interfaces to connect to physical hardware

devices. This software layer also allows to switch from accessing real hardware or virtual simulated hardware transparently to the end user;

- **Robot control system**: provides three levels of control functions – I) Supervisory function: enables the user to have access to safe behaviour in tele-operation and/or autonomous control mode (avoid collisions); II) Basic Functions: offer simple robot behaviours (usually reactive behaviours, like, wander, wall following, move towards a specific direction); III) High Level Functions: offer more sophisticated robot behaviours and function (as for example, environment mapping, robot localization and pose estimation, path planning);
- **Communication module**: provides secure network communication services, providing mechanisms that guarantees authenticity, integrity and confidentiality;
- **Virtual Simulation Visualization Tool**: provides a 3D visualization tool allowing to the user to follow the real-time virtual simulation of mobile robots.

4- Conclusion

This paper presented a discussion about the design of mobile robots applications, from the design methodology to the analysis of system requirements; from the description of architectural components to the required elements of virtual reality simulation tools. Important aspects related to communication, safety, security and fault tolerance were also discussed. From these aspects and requirements emerges our proposition of a Framework for Virtual Design of Mobile Robots. This presented proposal contributes to improve existing simulation models and tools (e.g. Player-Stage-Gazebo platform), including new components focused in providing safety, security, and fault tolerance.

Acknowledgments. The authors acknowledge the support granted by CNPq and FAPESP to the INCT-SEC (National Institute of Science and Technology - Critical Embedded Systems - Brazil), processes 573963/2008-9 and 08/57870-9.

5- References

[AS1] Alonso, Diego; Sánchez, Pedro; Álvarez , Bárbara; Pastor, Juan A. "A Systematic Approach to Developing Safe Tele-operated Robots" In: Reliable Software Technologies – Ada-Europe 2006. Luís Miguel Pinho and Michael González Harbour (Editors). LNCS - Lecture Notes in Computer Science 4006. Springer Verlag. pp. 119-130. (2006).

[B1] Bekey, George A. (2005) "Autonomous Robots: From Biological Inspiration to Implemen-tation and Control". The MIT Press: Cambrigde, London. 563p.

[CM1] Collett, Toby H.J.; MacDonald, Bruce A.; and Gerkey, Brian P. "Player 2.0: Toward a Practical Robot Programming Framework". In Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005), Sydney, Australia, December. (2005)

[D1] Damstra, A. "Virtual prototyping through co-simulation in hardware/software and mechatronics co-design", Technical Report 005CE2008, University of Twente. (2008).

[DJ1] Dudek, G.; Jenkin, M. "Computational Principles of Mobile Robotics". Cambridge, London, UK: The MIT Press, 280 p. (2000)

[F1] Fischer, X. "Characteristics, Development and Use of Models in Interactive Design". In: Virtual Concept (Proceedings). 1 ed. Biarritz, França: ESTIA - IEEE - Springer-Verlag, 2005, vol.1. (2005)

[F2] Fischer, X.; Bennis, F. "An Overview of Interactive Methods in Design and Manufacture". In: Virtual Concept (Proceedings). 1 ed. Biarritz, França: ESTIA - IEEE - Springer-Verlag, 2005, vol.1. (2005)

[GV1] Gerkey, Brian; Vaughan, Richard T.; Howard, Andrew. "The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems". In Proceedings of the 11th Internatl. Conf. on Advanced Robotics (ICAR 2003), pp. 317-323, Coimbra, Portugal, June. (2003)

[GV2] Gerkey, Brian P.; Vaughan, Richard T.; Støy, Kasper; Howard, Andrew; Sukhatme,Gaurav S. and Mataric, Maja J. "Most Valuable Player: A Robot Device Server for Distributed Control". In Proceedings of the IEEE/RSJ Internatl. Conference on Intelligent Robots and Systems (IROS 2001), pp. 1226-1231, Wailea, Hawaii, October 29 - November 3. (2001)

[L1] Latombe, J. "Robot Motion Planning". Kluwer Academic Publisher, Boston, MA. (1991)

[MF1] Michael, Nathan; Fink, Jonathan R.; Kumar, Vijay. "Experimental Testbed for Large Multirobot Teams". IEEE Robots and Automation Magazine, Volume 15, Issue 1, March 2008, pp. 53-61. (2008)

[OD1] Open Dynamics Engine, http://www.ode.org/

[OM1] Osório, F. S.; Musse, S. R.; Vieira, R.; Heinen, M. R.; Paiva, D. C. "Increasing Reality in Virtual Reality Applications through Physical and Behavioural Simulation". In: X. Fischer. (Org.). Proceedings of the Virtual Concept Conference 2006. 1 ed. Berlim: ESTIA - Virtual Concept - Springer Verlag, v. 1, p. 1-45. (2006)

[OM2] Osório, F.S. ; Musse, S.R.; Santos, C.T. dos ; Heinen, F. ; Braun, A.; Silva, A.T. da. "Intelligent Virtual Reality Environments (IVRE): Principles, Implementation, Interaction, Examples and Practical Applications". In: Xavier Fischer. (Org.). Virtual Concept (Proceedings - Tutorials). 1 ed. Biarritz, França: ESTIA - IEEE - Springer-Verlag, v. 1, p. 1-64. (2005)

[SN1] Siegwart, Roland and Nourbakhsh, Illah R. "Introduction to Autonomous Mobile Robots". A Bradford Book, The MIT Press: Cambridge, London. 317p. (2004)

[SW1] Staunstrup, Jørgen; Wolf, Wayne. "Hardware-Software Co-Design: Principles and Practice". Springer. 1st edition (October 31, 1997). 416pages. (1997)

[TB1] Thrun, S.; Burgard, W.; Fox, D. "Probabilistic Robotics". Cambridge: The MIT Press. 667p. (2005)

[VG1] Vaughan, Richard T.; Gerkey, Brian and Howard, Andrew "On Device Abstractions For Portable, Resuable Robot Code". In Proceedings of the IEEE/RSJ Internatl. Conf. on Intelligent Robot Systems (IROS 2003), pp. 2121-2427, Las Vegas, USA, October. (2003)