"Hands-On Mechatronics": Problem-Based Learning for Mechatronics

Yves Piguet, Francesco Mondada, Roland Siegwart
LSA - I2S, Swiss Federal Institute of Technology, Lausanne
yves.piguet@epfl.ch, francesco.mondada@epfl.ch, roland.siegwart@epfl.ch

Abstract

The project "Hands-On Mechatronics" aims to develop a problem-based learning environment for mechatronics. This environment is based on four specific aspects: a mobile robot competition as the motivation factor, a Web environment as a framework for all the students, the SmartEase robot as an experimental platform and a good software environment for design and fast prototyping. The actual status of this concept is applied at the Swiss Federal Institute of Technology in Lausanne (EPFL) with participants coming from several schools of different levels.

We present the actual implementation of the concept "Hands-On Mechatronics", the hardware and software tools used, some problems found in our approach, and their solutions.

1 Introduction

The learning style varies from person to person. Some might be very efficient in learning by only reading well-selected learning material, while others absolutely need to experiment. However, psychological investigations have shown that in general peoples remember only about 10% of the content they read, but 90% of what they try and realize.

In mechatronics in general and mobile robotics in particular, many robotics contests take place each year in many universities. This has shown to be a very motivating framework for interdisciplinary team projects. At EPFL, until 2000, the annual contest was completely disconnected from the official teaching activities; it was based on the teaching motivation of one single professor, Prof. Jean-Daniel Nicoud. On the opposite, the Swiss Federal Institute of Technology in Zürich (ETHZ) had a robotic contest strictly included in a classical (ex cathedra and practical) course organized by Prof. Roland Siegwart.

The goal of the project "Hands-On Mechatronics" is to use the existing Swiss robotic contests to promote an innovating approach to mechatronic teaching based on problem-based learning. This initiative wants to be open to participants coming from everywhere in Switzerland, from technical schools or universities, at graduate or post-graduate level. It has therefore to be based on an asynchronous teaching environment, open geographically, and accessible to everyone having some engineering background.

2 SmartRob Championship

2.1 Concept

Mobile robots have shown to provide a very good motivation to students. This aspect is used at EPFL in courses about automatic control, software and hardware design. A complete robot competition brings even more, needing teamwork and the full range of competences included in mobile robotics engineering. Figure 1 illustrates the feedback given by the students who have participated to the SmartROB competition of last year. It is clear that the participation to a competition is an important factor, but students appreciate also very much the interdisciplinary aspect and the fact of having a real problem where they can apply the theory they learn elsewhere.

2.2 Web environement

The Web environment offers to participants a central location where they can find information, request support from the remote teachers, and collaborate together. Documentation is provided as HTML and PDF documents as well as a searchable, cross-linked database of records which describes briefly the physical and technical principles of sensors and actuators. Documents cover case studies which illustrate how a multi-disciplinary project can be carried on and instructions about how to set up and use the hardware and the software (cf. Sections 2.3 and 2.4).

2.3 SmartEase PC104 robot platform

For this project we developed, together with the Swiss Center for Electronics and Microtechnology (CSEM), a

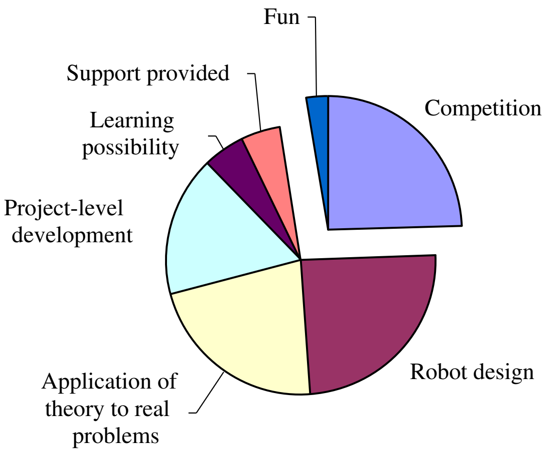

Figure 1: Motivation of the participants of the SmartRob competition in 2001.

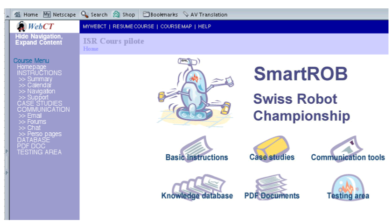

Figure 2: Web environment.

new robotic platform, called SmartEase PC104 (figure 1). This platform has the advantage of having an on-board standard PC running QNX as real-time operating system.

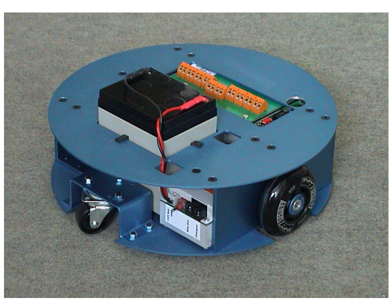

Figure 3: Robotic platform.

In addition to the PC104 CPU, daughter boards with axis controllers and digital I/O control the robot and the additional hardware the students can add. Sensor and actuator modules developed at EPFL are connected to the RS-232 interface of the PC104 the via an I2C serial bus. Modules proposed in the basic set-up include proximity triangulation sensors, a linear camera of 64 pixels, and servo-motors.

2.4 Software tools

Software cover two needs: experimenting with the sensor modules to lear more about their capabilities, their limitations, and how they could be used to extract information about the environment; and controlling the robot autonomously and in real time. Experimentation is made with SysQuake, scientific software with a Matlab-like language (named LME) and interactive graphics support. SysQuake has the advantage over competing products of permitting the fast development of user interfaces together with powerful data processing.

For real-time control, two solutions are proposed currently: cross-development in Java and on-board development directly in LME. The use of Java was proposed by our partners; it seems to have more drawbacks (cross development, no debugger, no QNX-based real-time) than advantages (high-level language, abstraction classes for the sensors and actuators). LME, the language of SysQuake, runs directly on the PC104 board. It has extensions for low-level hardware access and tasks. It is not real-time either, but its performances are sufficient for the medium-to-high-level control, while low-level control loops are performed on the wheel PID controller chips. Its main advantages are the interactivity which permits students to have access to sensors and actuators in two or three lines of code typed at the keyboard, and the compatibility with SysQuake. SysQuake has already been used at EPFL to control Khepera mobile robots over a serial line, and performances are adequate.

3 Conclusions

In addition to the new robot based on an industrial PC, the previous set-up, based on a M68k micro-controller board with a real-time embedded operating system, is used for the same competition. It will be interesting to get feedback from the students about how they weight the advantages of each platform: a more powerful computer with a traditional, desktop-like OS, and a lower power embedded system with low-level functions optimized for maximum I/O efficiency.

In the future, in addition to Java (requested by some of our partners) and LME, a third alternative for controlling the robot will certainly be added: development in C, probably on the robot itself with a source-level debugger, which will give a direct access to the POSIX real-time features of QNX. The performances of QNX will be compared to Linux, which would have the advantage of having drivers for a much larger choice of hardware (USB cameras come immediately to mind).