A TECHNICAL SOLUTION OF A ROBOTIC E-LEARNING SYSTEM IN THE SYROTEK PROJECT

Jan Chudoba
Center of Applied Cybernetics, Czech Technical University in Prague, Czech Republic
chudoba@labe.felk.cvut.cz

Jan Faigl, Miroslav Kulich, Tomáš Krajník, Karel Košnar, Libor Přeučil
Department of Cybernetics, Czech Technical University in Prague, Czech Republic
{xfaigl,kulich,tkrajnik,kosnar,preucil}@labe.felk.cvut.cz

Keywords: Mobile Robotics, E-learning.

Abstract: SyRoTek (a system for a robotic e-learning) is a robotic virtual laboratory being developed at Czech Technical University in Prague. SyRoTek provides access to real mobile robots placed in an arena with dynamically reconfigurable obstacles enabling variety of tasks in the field of mobile robotics and artificial intelligence. The robots are equipped with several sensors allowing students to realize how robots' perception works and how to deal with uncertainties of the real world. An insight to a technical solution of the SyRoTek project is presented in this paper. Detailed description of the newly developed educational robotic platform, motivated by identified requirements of SyRoTek, is presented together with a description of the arena and its functionalites. An overview of user access to SyRoTek, particularly how the robot is controlled and how sensory data are read is presented as well.

1 INTRODUCTION

Having real mobile robots as an indivisible part of teaching robotics or artificial intelligence is advantageous due to their attractiveness for students. On the other hand, robots need to be continuously maintained in order to be used flawlessly. To resolve this issue and to minimize the maintenance cost, so-called virtual laboratories allowing remote access to real hardware equipment have been built in nineties. In robotics, first systems have been focused on manual tele-operation (Telegarden, 2010; Telescope, 2010) or manual goal assignment in the case of autonomous mobile robots (Rhino, 2010; Xavier, 2010). The further progress allows remote access to robotic actuators and sensory data and robotic hardware have been integrated into e-learning frameworks (RobOnWeb, 2010; Siegwart and Sauc, 1999; RedRover, 2006; Guimarães et al., 2003; Masár et al., 2004). This is also the case of the project *SyRoTek - System for a robotic e-learning*, which is focused on developing a virtual laboratory allowing remote access to a set of real mobile robots that can be move inside dedicated space called arena (Faigl et al., 2010).

In this paper, we provide an insight to the technical solution of the SyRoTek (Kulich et al., 2009), describe details of our designed robots, and an arena where the robots are placed and which allows a dynamic reconfiguration of obstacles. Students control the robots by their applications, and therefore robots and the whole system have to be robust enough to be used in a long-term without necessity of human manual interventions. This means various parts of the system have to be monitored. In the case of undesired values of critical parameters, the robot has to be able to protect itself from damage caused by an error in student's application. Besides, a robot has to autonomously navigate to the recharging station when reaching low power state.

The SyRoTek project aims to provide support of teaching robotics and artificial intelligence courses mainly at the university level. It also aims to provide a hardware platform for experimental evaluation of new navigation and control techniques being developed in the postgraduate courses. Regarding this scope, users will have relatively open access to all necessary hardware parts of the system, and therefore SyRoTek sub-systems may not be much hidden by complicated hardware abstraction.

One of the main features of the SyRoTek project is a support of semi-automatic students' assignments evaluation in which students' program controlling the robot is autonomously executed. It is desired the mechanism will work as follows. A student submits its solution of a particular assignment. Then, a system schedules student's program execution to the time where none of robots are reserved to be used by other students. A video record of the robot operating in the arena together with logs of all sensory data and actuators' commands are recorded. The data files are then delivered to a teacher as supporting material of student's solution evaluation.

The paper is organized as follows. Identified requirements, which steer the robot design, are presented in the next section. An overview of SyRoTek is presented in Section 3, where basic subsystems are introduced. Detailed description of the robot hardware internals and used sensory equipment are described in Section 4. The realization of the arena with moving obstacles, robots' docking stations, global localization system, and concept of the visualization system are presented in Section 5. A remote user access to the robots is realized by the supporting computer that is directly accessible through the Internet, its role and concept of the user access to robots is briefly described in Section 6. Concluding summary of the presented solution is presented in Section 7.

2 SYSTEM REQUIREMENTS

This section summarizes requirements affecting the SyRoTek system design. At first, it is worth to mention the system is designed as a multi-robotic. This requires enough available robots for collective tasks, at least three or four at the same time. A robot can be available only until its battery is discharged, and therefore additional robots have to be prepared when the active robots need to return for recharge. Regarding to the expected scale of multi-robotic experiments, we have decided to create a system with 10–15 robots.

Two competing constraints to the robot size have been considered. The robot body has to be large enough to carry all the desired sensors. On the other hand, the robot should be as small as possible to maximize the robot working space as the arena dimensions are limited by local space conditions. The available space is about 10 m^2, and therefore the maximal robot diameter was fixed to 20 centimeters. With respect to various tasks to be solved within SyRoTek, the arena should provide maximal environment variability from free space to maze-like narrow corridor areas. The area where the robot operates is related to robot's kinematics and dynamics. With this respect a differential drive with two controlled wheels has been selected that allows the robot to turn at place and offering sufficient maneuverability in narrow space. The robot speed in not crucial in the expected students' tasks, and therefore velocity in range 0.2–0.5 m/s has been considered as sufficient.

The main goal of the SyRoTek system is to support education with real robots that will help students to realize how robots sense the environment and how to deal with uncertainties that are inevitable part of real world. In order to provide such experiences, variety of sensors commonly used in mobile robotics are requested to be placed at the robot body. The range-measuring sensors are the most typical sensors for basic robotic tasks like collision avoidance. That is why simple types of these sensors are required to be in a basic configuration of each robot. Also robot navigation based on image processing is becoming common nowadays, so a color camera has been included in the basic set of robot's sensor equipment. Rotating laser scanners (LIDARs) are also frequently used in the mobile robot localization and mapping tasks. An improvement in miniaturization of this type of sensors reduced their costs and dimensions to be used also with small robot platforms, e.g. Hokuyo URG-04. However, the power consumption of such LIDARs is still restrictive for the robot design, thus we have considered it as an optional equipment.

The sensor equipment requires appropriate computational resources that will allow simple sensory data processing on-board. A computer running operating system is highly desired for such tasks, as it will allow a comfortable maintenance and reconfiguration. This requirement leads to use a PC-compatible computer module running at frequency 200 MHz or higher.

Wireless communication device is needed on-board for transmitting sensor data, control commands, and software updates. IEEE 802.11 (WiFi) network modules are widely available for various computer modules on the market. Event though they allow a wide bandwidth, a disadvantage of the regular WiFi transmission is absence of the latency definition and eventual drop-outs in noisy environments with many wireless networks running alongside. This issue motivate us to use another communication module (channel) beside the WiFi to control the robot, even at the cost of lower bandwidth.

The SyRoTek system is requested to run 24 hours a day with a minimal maintenance that is mainly related to robots charging. Fully autonomous charging is required. A minimal run-time for fully charged robot's battery is about two hours, which is derived from the duration of regular course lab at our university taking 90 minutes. In a stand-by mode, when robot does not move and all sensors are not activated, the operating time about 10 hours or higher is advantageous, because this situation is typical for testing student's application in the development phases. The power subsystem of the robot also requires additional requirements that will address safety issues. The most dangerous components are the robot batteries and the autonomous charging system, with respect to possible battery destruction going along with overheating or explosion. The accidental short circuit condition on the robot charging contacts could cause destruction of power supply or fire when not handled properly.

Additional safety issues have to be solved as well, guaranteeing any part of the system cannot be damaged as a result of unexpected user action or internal failure. It is advised for critical parts to double or triple-check dangerous conditions if possible.

3 SyRoTek OVERVIEW

An overview of the SyRoTek realization is depicted in Fig. 1 and (Kulich et al., 2009). The system consists from an arena where mobile robots are located. A user access to SyRoTek from his/her workstation is realized through the main control computer that is accessible from the Internet. A wireless communication infrastructure is used to connect robots with the main control computer. A visualization system with several cameras provides views to the real scene and an access to on-line streams or archived video files is realized by a video server. A localization system based on a camera placed above the arena is used to estimate real position of the robots. The particular subsystems are described in more detail in the nexts sections.

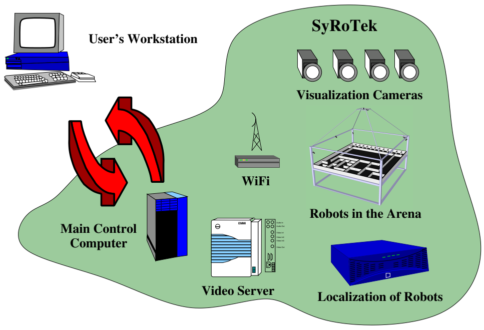


Figure 1: SyRoTek overview.

4 ROBOTS

At the beginning of the SyRoTek project, several available robotic platforms were discussed whether they are applicable and meet identified requirements of the project needs. Many robots were rejected due to their size over 20 centimeters in diameter. Smaller robots mostly did not meet desired sensor equipment or had very poor options for extensions. The Khepera III robot (K-Team, 2010) developed by K-Team Corporation has been evaluated as the most promising commercially available robotic platform with the closest parameters to the desired requirements. However, the robot equipment would have to be extended and the selected model was not available in the system design phase.

The most critical feature considered in the platforms evaluation is mechanism of the robot charging. All available robots considered are recharged with human attendance that would result in the need to adjust robot for automatic recharge. Considering the amount of necessary modifications of the considered robots in combination with its price, we have decided to design and manufacture a robot to serve our needs. The result is a new robotic platform for education. The robot is called S1R and it is shown in Fig. 2. Its design and properties that meet the aforementioned requirements are described in this section.

Figure 2: S1R robot with laser rangefinder installed.

The most important component of the robot, influencing its design, are batteries and motors. Lithium-Polymer rechargeable cells were selected because of the best trade of between capacity and size (weight). Required operation time on batteries results in need of about 50 Wh battery capacity. A battery of 6-cells, each 2400 mAh, has been selected with regard to keep maximal battery size under the required robot dimensions. Monitoring of each cell individually is necessary during the charging of the lithium based battery, therefore a charger circuit has been embedded into the robot body. The battery and motors represent the heaviest parts of the robot body. Thus, these components define the robot center of mass that should be as low as possible.

Two Faulhaber 2224 motors with 18 V nominal

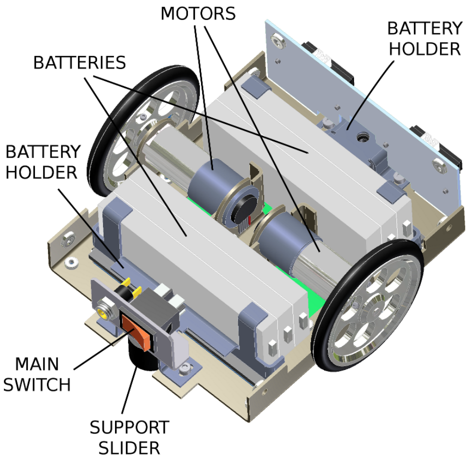


Figure 3: S1R robot chassis.

voltage have been selected to meet the required differential kinematics. The motor has an integrated 86:1 gearbox and 512 pulse-per-rotation encoder. The motors and battery are located at the bottom of the robot where they are placed in the S1R robot chassis, see Fig. 3. Two driven wheels are mounted directly to the motor gearbox axises. Outer casing of the chassis is a bumper of octagonal shape with rounded corners, as shown in Fig. 4.

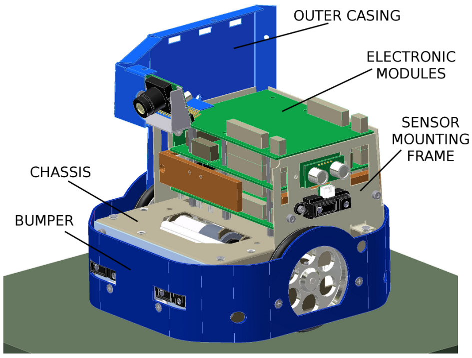


Figure 4: Main parts of the S1R robot.

On the top of the chassis, electronic modules are mounted together with a mounting frame for on-board sensors (chassis sensors). Three main electronic boards are part of the robot internal electronics

- power control and charger board (*power board*),
- motor control and sensor data acquisition board (*control board*) and
- on-board computer interface board with on-board computer (*OBC*) mounted on.

A space in front of the robot is available for a replaceable sensor module (*front module*). Currently two modules are designed and manufactured. Both modules provide range measurements. The first module uses infrared sensors while the second module is a laser rangefinder module with Hokuyo URG-04.

The top of the robot is covered by a lid with a unique pattern used for robot localization and identification from the camera placed above the robots working space.

4.1 Sensor Equipment

The encoders together with designed wheels with radius 70 mm provide odometry information with resolution about 200 units per 1 millimeter of distance traveled. Actual motor currents are monitored and used for a simple collision detection if the robot is stucked and the motor current is increased.

The exteroceptive sensors include two infrared sensors directing forwards, mounted on chassis, and three additional sensors mounted on the sensor mounting frame. The additional sensors are directing to left, right and backwards. Above the left, right and rear infrared sensors, three sonars are mounted with the same directions. The directional sensitivity characteristics of the sonar has to be adjusted by installing textile tubes on sonar transducers in order to reduce detection of objects far from an axis of measurement, including a ground. Also, the maximal range and gain of the sonar has to be reduced.

Aside the range measurement sensors, color camera, compass and 3-D accelerometer are installed in the robot. All these sensors, mounted as a basic equipment, are called *chassis sensors* in the rest of this paper.

Beside the chassis sensors, the robot can be equipped by the front module with additional sensors. A commonly equipped module consists of three infrared sensors and three sonars, directing front and 45° left and right. This module is referenced as the *front sensor*.

Two *floor sensors*, each containing four reflectivity detectors, are mounted on the bottom of the robot. These sensors allow to detect color patterns or lines drawn on the ground.

Several thermal sensors are placed on the battery cells, power components of the charger circuit and the power regulator for internal monitoring with regard to safety

All sensors available to SyRoTek robot end–users at this time are summarizes in Table 1.

4.2 Electronic Modules

The robot electronics comprises of four main electronic circuits: *power module*, *control board*, *on-board computer* (OBC), and a *radio module*. Besides, additional circuits are a part of the sensors. Each component is connected with the OBC that provides the

Table 1: Robot sensors.

Sensor	**Type**
Chassis sensors:	
IR range sensors	5× Sharp GP2D120
Sonars	3× Devantech SRF10
Compass	Devantech CMPS03
Accelerometer	Freescale MMA7260Q
Camera	CmuCam3
Other sensors:	
Floor sensor	2× 6-detector lines
Front sensor:	
IR range sensors	3× Sharp GP2Y0A21Y
Sonars	3× Devantech SRF10
Laser rangefinder:	Hokuyo URG-04LX

main access to the robot through wireless communication channels. The used communication interfaces are depicted in Fig. 5. The main purposes of the electronic modules are described in the following paragraphs.

The power module has two main purposes: battery maintenance and generation of on-board voltages in two levels 5 and 3.3 volts. Embedded battery charger is able to charge the robot Li-Pol battery, when an external voltage is present. Voltage of each cell and the battery pack temperature is monitored permanently to avoid dangerous states and possible destruction of the pack. A speaker is mounted on the power board to inform or warn nearby personnel about important events, like robot re-start, battery discharged under safe treshold, battery cell overvoltage due to a charger failure, or an overheating of any monitored components. Lithium based batteries are very susceptible to discharge under allowed limit, which often results in its permanent destruction. Permanent monitoring of the battery voltage is therefore an important function of the power module. The robot main power supply is cutoff in the case of battery critical condition.

The main function of the control board is to control robot motors and collect data from the chassis sensors. The control board is based on the Hitachi H8S/2639 micro controller with embedded hardware counters for the quadratic encoders. The computational power of the controller is used for odometry based estimation of the robot position within its local coordinate frame. Besides, the range measurements are used to avoid collision if this functionality is requested.

The on-board computer (OBC) represents the main computational power of the robot. The Gumstix Overo Fire with OMAP 3530 at 600 MHz computer module has been selected. Beside UART, SPI, I^{2}C, and USB communication interfaces, it provides on-board 802.11g wireless network module. The Overo computer module is interfaced with other robot modules by the so-called *OBC interface board*. The board serves as an interface between OBC and other components. In addition, the 3-D accelerometer sensor with own processing microcontroller is placed on the board.

The radio module is dedicated to transmit real-time control commands and low-data-rate sensor data between the robot and an external control computer providing user access to the robot through the Internet. It is based on the Nordic nRF24L01 chip, allowing full-duplex communication at speed over 100 kbps.

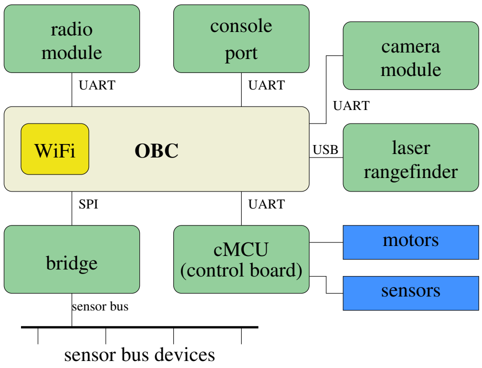


Figure 5: OBC connections to modules.

4.3 Inter-module Communication

Having several modules and variety of used sensors, a unified approach to connect as much sensors as possible has been highly desirable. The I^{2}C bus has been selected as the primary communication bus between the most electronics modules and sensors modules in the robot. We created so called *sensor bus* by adding 2 lines to the standard I^{2}C. The additional lines allow to reset modules and enter a firmware update mode. Specific common communication protocol has been designed in order to unify the microcontroller settings, sensor readings and firmware updates. Even though such unification requires placing additional microcontroller to read sensors that can be directly connected to the OBC, it is advantageous, because all such sensor modules behave in same way, making the software configuration much easier. Devices connected via the sensor bus include the power module, front sensor module, floor sensor modules, and the accelerometer module.

Although the OBC has a hardware I^{2}C interface embedded, we have decided to connect it to the sensor bus using a serial interface, like UART or SPI and an additional translating module called *bridge* (see Fig. 5). The original reason for using the bridge was absence of usable implementation of I^{2}C slave driver for the OBC. During a design and testing of sensor bus devices, we found bridge very useful, as it allows replacement of the OBC by a regular PC without I^{2}C interface. The bridge module is based on the Atmel ATmega microcontroller mounted on the OBC interface board.

A sensor bus communication protocol is datagram-based and it basically follows common I^{2}C communication based protocols. A fixed length datagram header that may be eventually followed by a variable size data message is used. The structure of the data part may vary device to device, yet a data-part header format is unified for most devices. A publisher/subscriber schema is also used for sensors readings to avoid polling based on the request/response schema. Although the schemes are commonly used, the main advantage of the proposed protocol is that the data may be transmitted to the user application without significant additional processing through WiFi or additional radio channel. To clarify this advantage it is necessary to realize the complete transmission path of the sensory data to the user that can be used in the SyRoTek system. The user application can run at a user's workstation. Such application is connected to the SyRoTek control computer accessible through the Internet. At this computer the robot OBC is accessible by one of the used wireless communication channels. The sensor can be connected to the sensor bus, thus there is the bridge module between the OBC and the sensor. The designed protocol allows such transmission without necessary interpretation or even translation of the transmitted data, thus it tries to minimize possible transport delays. Moreover, at a particular computer, additional software modules can be used to pass data from the robot hardware to the users application, more details about the user access are described in Section 6.

A microcontroller of each sensor-bus compliant module has a boot-loader code implemented, allowing remote firmware update over the sensor bus. This feature has proven to significantly speed up development process. For a firmware update the two additional bus lines are needed. A *Reset* line is used to reset all devices on the bus. Second line *Flash* is pulled down when enter into bootloader-mode is requested after reset. When resetting all devices is not convenient, a particular device can be reset using reset command while holding *Flash* line low, if the device supports it. In the bootloader mode the device accepts specific commands to update firmware in the flash memory.

4.4 Power Consumption

As we require maximal operating time while robot is powered from batteries, minimization of the power consumption of individual components is crucial.

The power consumption of motors is about 1–2 watts depending on actual load. Considering the robot is not moving all the time and the load under common conditions is relatively small, except occasions of collisions, the average consumption may be under 500 mW.

The most power consuming component is the OBC module, with consumption about 2 watts with the WiFi module running. When the WiFi module is disabled, the consumption falls to approximately 1.1 watt. It's obvious that OBC is most power demanding device in the robot.

The consumption of most remaining electronic components is about 10–20 mA at 5 V per processor or module. Some sensors, like infrared range sensors, or floor reflectivity sensors have approximately double consumption than other. Total consumption of all these devices and sensors is between 2–3 watts in full operation mode. The total power consumption of all these components is comparable to the consumption of OBC, so proper power management is advisable, considering that all devices are not necessary to run all the time. Several mechanisms were implemented, allowing switching these devices off in cases they are not needed.

When a laser rangefinder front module is used, the power consumption rise significantly, by approximately 3 watts. Usage of the laser module results in operation time drop by 30–50%. The necessity of the laser module power switching is obvious.

4.5 Robot Overall Parameters

Overall parameters of the robot are summarized in Table 2. The robot operation time has been experimentaly measured when the robot was performing a IR-sensor based obstacle avoidance. An on-board computer with the WiFi module enabled was turned on during the experiment.

Figure 6: The SyRoTek arena and robots

Table 2: Robot parameters.

Robot parameter	**Value**
Length × Width × Height	174 × 163 × 180 mm
Weigth	about 2 kg
Maximal velocity	0.34 m/s
Odometry resolution	200 samples/mm
Battery type	Li-Pol (6 cells)
Battery voltage (nominal)	22.2 V
Battery capacity	2400 mAh (53 Wh)
Total power consumption	about 5 Watts
Robot operation time	about 8 hours
Robot charging current	2 A
Computation power	ARM CPU @ 600 MHz

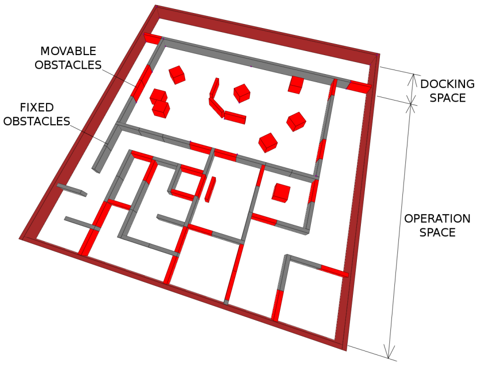


Figure 7: Obstacle configuration in arena.

5 ARENA

The SyRoTek arena is an enclosed space dedicated for robots. It is not only the space itself, but it also consists of necessary supporting subsystems, e.g. charging, lighting, visualization etc. The size of the arena is constrained by an available space of a computer lab where it is located (see Fig. 6), resulting in dimensions 3.5 m× 3.8 m.

The robot working space is a flat area with an outer barrier 18 cm tall. Additional 13 cm tall obstacles are placed inside. Some obstacles can be remotely retracted, while the rest of them is fixed, however all obstacles can be manually removed in order to create various configurations. The obstacle placement is depicted in Fig. 7.

5.1 Robot Charging Docks

For each robot in the arena a dedicated docking space is allocated. In the docking place a recharging mechanism is available. Thirteen recharging docks are placed at one side of the arena that is well accessible for a human personnel. A space of docks is separated from the working space and robots are exclusively controlled automatically in this part of arena.

Several technical solutions of charging connector were discussed during the system design phase, with respect to contact resistance, durability and maintenance-free. Among other solutions a wireless power transfer was tested and rejected mainly due to necessity of very precise docking with an error under 1 mm.

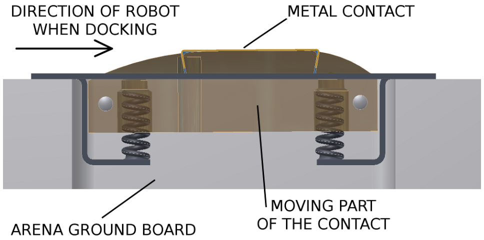


Figure 8: Charging contact detail, side view.

The final solution uses two flexible bronze contacts, pushed by springs against gilded metal pads on the bottom of the robot, as shown in Figures 8 and 9. Two contact pads on each contact are used to measure the contact resistance. When high resistance is detected during docking maneuver, the robot may adjust its position to reach better conditions. Although the bottom of the robot is isolated to prevent short circuit on the charging contacts, the short circuit cannot be avoided completely. The charging circuit is constructed with ability to withstand short circuit for a short time and disconnecting the power supply when it persists longer. Each dock provides 2-Ampere power supply, allowing the robot to fully recharge in about 1–2 hours.

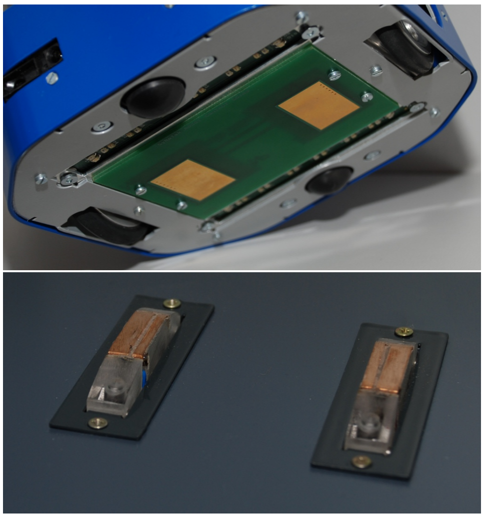


Figure 9: Charging contacts on robot and in arena.

5.2 Reconfigurable Obstacles

The arena was designed to be reconfigurable without need of human attendance. This feature was achieved by installing several moving obstacles, allowed to be retracted under the surface, which is schematically shown in Fig. 10. The presence of moving obstacles allows to prepare more environment configurations in the arena before a user starts solving his/her task, as well to define tasks with dynamically changing environment. Moreover, the arena workspace may be divided into several mutually separated closed areas to create robot working spaces for several users without affecting each other.

When a mechanism of moving obstacles was designed, an operation noise and budget limitations were the main criteria. The final solution is a trade-off between using low-cost components and keeping noise on tolerable level. Each obstacle needs about 10 seconds to extrude or retract. Due to a power supply current limit, all obstacles cannot be displaced simultaneously and therefore additional obstacle management is required. A noise generated by the obstacles when displacing is nonneglible, so frequency of the arena reconfiguration should be minimized.

5.3 Robot Localization and Visualization

A global localization system for robots is an important functionality of the arena. A robot identity together with its position and orientation is estimated using an image processing method. A grayscale camera mounted above the arena working space provides an image in which patterns on the top of robots are recognized. These patterns (an example is shown in Fig. 11) consist of outer ring used for position and orientation estimation and inner code–circle for identification of a single robot. Figure 12 shows result of a convolution-based localization algorithm on a sample image. The top plane in the figure shows an image from the camera transformed to the arena coordinates. The graph below represents a result of the convolution function. It is obvious that function maxima representing robot positions are very distinctive, alowing robust localization under various light conditions.

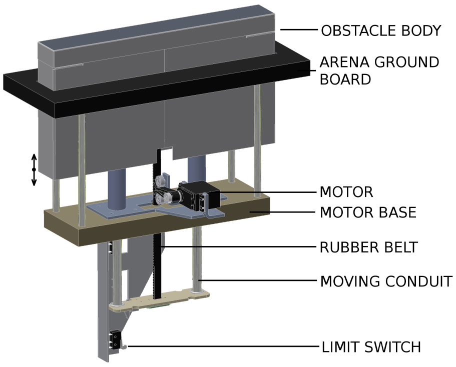


Figure 10: A designed moving obstacle.

An Unibrain Fire-i 820b camera connected by the IEEE 1394 (FireWire) interface is used with 1600×1200 pixels resolution at 12 frames per second. The localization algorithm is executed on a dedicated computer, however we intend to transfer localization algorithm to an embedded FPGA device. An achieved accuracy of the localization is about 3 mm in position estimation and 5° in robot orientation.

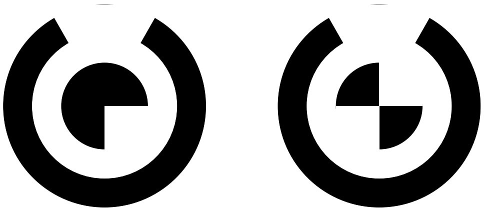


Figure 11: An example of robot identification/localization patterns.

Several other color cameras are mounted above the arena ground, providing view of the arena. An user may observe robots in the arena thru visualization components of an user interface, or he can download video streams of the performed experiments later. The visualization cameras are connected via an Ethernet interface, which provides also a power

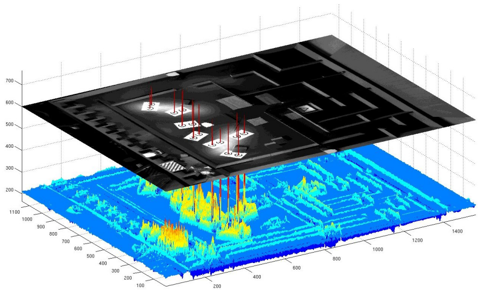


Figure 12: Result of a robot localization algorithm.

supply.

5.4 Construction

A construction of the arena was designed robust enough to support wooden surface board with obstacles, lights, outer glass barriers and other equipment. Beside the fixed equipment, the construction also has to be able to withstand additional weight of a human body in case of maintenance. To minimize a dust settling on the surface, the top of the arena is covered.

Four fluorescent tube lamps are mounted above the surface, providing necessary arena lighting. An automatic control of the lights is necessary to turn them on only when experiments are executed with respect to energy saving and less disturbance of people in lab when the system is not used.

6 USER ACCESS

The user access to SyRoTek is realized through *control computer* accessible from the Internet. The access can be divided into three categories. The first category represents web pages with basic information about system, courses and supporting materials. Two next categories are related to the robot control application being developed by SyRoTek users, which are a remote session at the control computer and a remote access to the robot. The remote session allows users to use pre-configured development environment and tools while the remote access to the robot provides direct robot control and sensory data readings. From the users' applications point of view, the most important part is the application programming interface (API). Instead of developing a new API, well-known and widely used the Player framework (Gerkey et al., 2003; The Player Project, 2010) has been selected as the main user application interface.

Although the Player provides an usable interface to control mobile robots and is a flexible in multi-computer environment, it does not support user authorization to particular sensors. In an e-learning system, the authorization is mandatory because in a certain task, particular sensor is not allowed to be used by the user. For example an user access to the global localization may not be allowed in a task aiming to implement localization algorithm based on range measurements. However, to support evaluation of such localization algorithm a data log from the global localization system is useful to compare real performance of the algorithm. The authorization is the main reason why a *robot access module* called *robacem* was developed, representing a robot at the control computer. The player server executed by an user, e.g. in the remote session, connects to the robacem using specific player driver called *syrdriv*. Then, user's application can be connected to the player locally within user's session at the control computer or remotely through the Internet connection from user's workstation.

The advantage of robacem is a possibility to monitor user's commands to the robot and additional sensory data that are not read from user's application. Monitoring is implemented using the observer pattern (Gamma et al., 1995) that enables a flexible way to share date between several applications, i.e. user's one, monitoring and eventually evaluating application. The stand-alone monitoring application is responsible for monitoring all the robots, by collecting and evaluating data provided by the monitors in robacem modules of all robots. When a dangerous state is recognized, or when the robot does not respond for a defined period, an alert is generated. As a result of the alert, an administrator or maintenance personnel may be notified by an e-mail or short text message to cell phone, depending on the seriousness of the situation. Less dangerous situations may be solved autonomously, e.g. in low battery state of the operating robot the system takes control of the robot and navigates it to a charging dock. In all cases, a log record is generated for further system performance analysis.

For each robot one robacem instance is running at OBC (on-board computer) and another instance is running at the control computer. These two modules are connected by two independent channels. WiFi is used for TCP/UDP based communications, while the dedicated low bandwidth radio channel is used for regular communication like status and velocity commands. The player server may connect to the robacem instance running at the OBC, however this configuration is not expected to by used by students, mainly because an inexperienced user can accidentally saturate WiFi channel leading to degradation of WiFi connections to other robots. Such degradation has been observed during experimental evaluation of system performance in multi-robot exploration task, and therefore we introduce this limitation to SyRoTek. However, a configuration with player server or even with user's application running at the OBC, is possible for experienced users. Particular options of the user's application access to the robot are shown in Fig. 13.

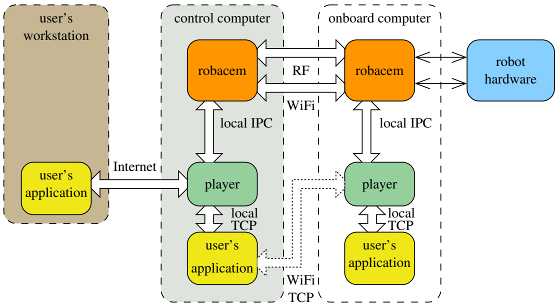
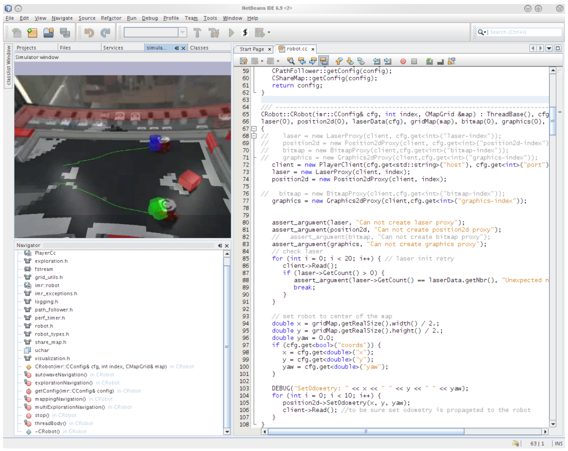


Figure 13: User's application access to robot.

The SyRoTek project aims to provide support for teaching course with real mobile robots, and therefore development process is the essential part of the courses. The player server provides an abstract layer hiding particular hardware details. Moreover, the *Stage* simulator (Vaughan, 2008) may be used, allowing application development without real hardware. The advantage of the abstraction consists in straightforward deployment of the application to the real robot. A network address of the player server is typically the only change needed to be made.

To provide students better understanding of the principles of robot perception of the real world a visualization of the sensory data is advantageous. Also to provide better overview of the real scenes the sensory data can be visualized in an image of the real scene using the Mixed Reality concepts (Milgram and Kishino, 1994).

In SyRoTek, a dedicated visualization component is being developed to provide a visualization tool. The component is based on modified Stage simulator (version 3.x) that is enhanced to support showing video streams on-line or from recorded files. The intrinsic and extrinsic parameters of the used cameras are identified and used in the transformation matrices for sensory data visualization in an OpenGL context. The component can be used in several ways. At first it can be used as an independent visualization application at user's workstation or as a plug-in for some of Integrated Developing Environment (IDE), e.g. see integration with Netbeans (Netbeans, 2010) in Fig. 14. The application allows on-line visualization using current sensory data transmitted from the SyRoTek control computer and live streams from visualization cameras. Alternatively the application can be used as data log player allowing detailed analysis of the robot performance. Finally the visualization component is planned to be used for off-line creation of documented video files to support evaluation of the users tasks.



```
   CPathFollower::getConfig(config);
   CShareMap::getConfig(config);
   return config;
}

/// -----------------------------------------------------------------------
CRobot::CRobot(imr::CConfig& cfg, int index, CMapGrid &map) : ThreadBase(), cfg
laser(0), position2d(0), laserData(cfg), gridMap(map), bitmap(0), graphics(0),
{
//    laser = new LaserProxy(client, cfg.get<int>("laser-index"));
//    position2d = new Position2dProxy(client, cfg.get<int>("position2d-index")
//    bitmap = new BitmapProxy(client,cfg.get<int>("bitmap-index"));
//    graphics = new Graphics2dProxy(client,cfg.get<int>("graphics-index"));
   client = new PlayerClient(cfg.get<std::string>("host"), cfg.get<int>("port")
   laser = new LaserProxy(client, index);
   position2d = new Position2dProxy(client, index);

//   bitmap = new BitmapProxy(client,cfg.get<int>("bitmap-index"));
   graphics = new Graphics2dProxy(client,cfg.get<int>("graphics-index"));


   assert_argument(laser, "Can not create laser proxy");
   assert_argument(position2d, "Can not create position2d proxy");
   //   assert_argument(bitmap, "Can not create bitmap proxy");
   assert_argument(graphics, "Can not create graphics proxy");
   // check laser
   for (int i = 0; i < 20; i++) { // laser init retry
      client->Read();
      if (laser->GetCount() > 0) {
         assert_argument(laser->GetCount() == laserData.getNbr(), "Unexpected n
         break;
      }
   }

   // set robot to center of the map
   double x = gridMap.getRealSize().width() / 2.;
   double y = gridMap.getRealSize().height() / 2.;
   double yaw = 0.0;
   if (cfg.get<bool>("coords")) {
      x = cfg.get<double>("x");
      y = cfg.get<double>("y");
      yaw = cfg.get<double>("yaw");
   }

   DEBUG("SetOdometry: " << x << " " << y << " " << yaw);
   for (int i = 0; i < 10; i++) {
      position2d->SetOdometry(x, y, yaw);
      client->Read(); //to be sure set odometry is propageted to the robot
   }
}
```

Figure 14: The SyRoTek visualization component within the integrated developement environment.

7 CONCLUSION

The development of the SyRoTek project is still in progress, as the project is not finished yet. However, the presented educational robotic platform called S1R is already finished and robots are now manufactured to be used in the desired number. Several real experiments with collaborating robots in the arena have been performed to verify the concepts and to test a developed hardware and firmware.

An other important part of the SyRoTek project consists of web pages with supporting materials and courses that will guide students (users) how to use the system and how to create an application to control a real mobile robot. Even though this part is still under development, it is expected that a trial application of SyRoTek for users will be opened from July 2011.

ACKNOWLEDGEMENTS

The work presented in this paper has been supported by the Ministry of Education of the Czech Republic under program ”National research program II” by the project 2C06005. Also, the support of the Ministry of Education of the Czech Republic under the Project No. 1M0567 to Jan Chudoba and Karel Košnar is gratefully acknowledged.

REFERENCES

Faigl, J., Chudoba, J., Košnar, K., Saska, M., Kulich, M., Saska, M., and Přeučil, L. (2010). SyRoTek - A Robotic System for Education. *AT&P journal*, 2:31–36. to appear.

Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns*. Addison-Wesley Professional.

Gerkey, B. P., Vaughan, R. T., and Howard, A. (2003). The player/stage project: Tools for multi-robot and distributed sensor systems. In *In Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323.

Guimarães, E., Maffeis, A., Pereira, J., and et. al (2003). Real: A virtual laboratory for mobile robot experiments. *IEEE Transaction on Education*, 46(1).

K-Team (2010, (accessed 23 October 2010)). `http://http://www.k-team.com/mobile-robotics-products/khepera-iii`.

Kulich, M., Faigl, J., Košnar, K., Přeučil, L., and Chudoba, J. (2009). SyRoTek - On an e-Learning System for Mobile Robotics and Artificial Intelligence. In *ICAART 2009*, volume 1, pages 275–280, Setúbal. INSTICC Press.

Masár, I., Bischoff, A., and Gerke, M. (2004). Remote experimentation in distance education for control engineers. In *Proceedings of Virtual University 2004, Bratislava, Slovakia*, pages 16–17.

Milgram, P. and Kishino, F. (1994). A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D(12).

Netbeans (2010, (accessed 30 September 2010)). `http://netbeans.org`.

RedRover (2006, (accessed 26 September 2006)). `http://www.redrover.reading.ac.uk/RedRover/index.html`.

Rhino (1997, (accessed 27 July, 2010)). `http://www.iai.uni-bonn.de/~rhino/tourguide`.

RobOnWeb (2002, (accessed 27 July 2010)). `http://asl.epfl.ch/research/projects/RobOnWeb/robOnWeb.php`.

Siegwart, R. and Sauc, P. (May 1999). Interacting mobile robots on the web. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*.

Telegarden (1997, (accessed 19 October, 2010)). `http://www.telegarden.org`.

Telescope (2010, (accessed 27 July, 2010)). `http://www.telescope.org`.

The Player Project (2010, (accessed 27 July 2010)). `http://playerstage.sf.net`.

Vaughan, R. (2008). Massively multi-robot simulation in stage. *Swarm Intelligence*, 2(2):189–208.

Xavier (2001, (accessed 27 July, 2010)). `http://www.cs.cmu.edu/afs/cs.cmu.edu/Web/People/Xavier`.