Development of a control architecture for the ROBUDEM outdoor mobile robot platform

Daniela Doroftei, Eric Colon, Yvan Baudoin
Royal Military Academy (RMA)
Department of Mechanical Engineering (MSTA)
Avenue de la Renaissance 30, B-1000 Brussels, Belgium
{daniela.doroftei,eric.colon,yvan.baudoin}@rma.ac.be

Abstract

Humanitarian demining still is a highly labor-intensive and high-risk operation. Advanced sensors and mechanical aids can significantly reduce the demining time. In this context, it is the aim to develop a humanitarian demining mobile robot which is able to scan a minefield semi-automatically. This paper discusses the development of a control scheme for such a semi-autonomous mobile robot for humanitarian demining. This process requires the careful consideration and integration of multiple aspects: sensors and sensor data fusion, design of a control and software architecture, design of a path planning algorithm and robot control.

Introduction

The goal of this research project is to prepare the ROBUDEM, an outdoor mobile robot platform as shown on Figure 1, for a humanitarian demining application. In this setup, the robot navigates and searches for mines by moving and sensing with the metal detector for suspicious objects in the soil. Once a suspicious object is detected, the robot stops and invokes its Cartesian scanning mechanism. This scanning mechanism performs a 2D scan of the soil, allowing mine imaging tools to make a reliable classification of the suspicious object as a mine or not. This paper describes partial aspects of this research work and focuses mainly on the design of the control and software architecture. A goal for the future is to implement an existing cognitive approach for mobile robot navigation on the mobile robotic platform. This will allow the robot to scan a suspected minefield semi-autonomously and return a map with locations of suspected mines. The development of such an intelligent mobile robot requires consideration of different side-aspects.

Robots use sensors to perceive the environment. Sensors under consideration for this research work are ultrasonic sensors, a laser range scanner, a stereo camera system, an inertial measurement system, a GPS receiver and of course a metal detector. All but the last one of these sensors return positional and perceptual information about the surroundings. This sensor data has to be fused in a correct way to form a coherent “image” of the environment. If a robot needs to gain a more or less complete “image” of its environment, it cannot rely on only one type of sensor. Hence the need for an intelligent sensor fusion algorithm to combine the often erratic, incomplete and conflicting readings received by the different sensors, to form a reliable model of the surroundings. Sensor fusion has been subject to a lot of research [1][4], most of the proposed methods use Kalman Filtering [17] and Bayesian reasoning [15]. However, in recent years, there has been a tendency to make more and more use of soft computing techniques such as artificial neural networks [8] and fuzzy logic for dealing with sensor fusion. [3][6].

An autonomous mobile agent needs to reason with perceptual and positional data in order to navigate safely in a complex human-centered environment with multiple dynamic objects. This translation of sensory data into motor commands is handled by the robot navigation controller. Its design is closely related to the design of the control architecture which describes the general strategy for combining the different building blocks. The basis for this reasoning process is often a map, which represents a model of the environment. These maps can be simple grid maps, topological maps [7], or integrated methods [16]. The used path planning technique depends highly upon the type of map chosen before. A survey of different methods can be

found in [5].The goal of this research is to use a behaviour-based control architecture to navigate while modeling (mapping) the environment in 3 dimensions, using vision as a primary sensing modality.
The control architecture has to be translated into a software architecture which manages the building blocks on a software level. This software architecture has to provide the flexibility of modular design while retaining a thorough structure, enabling an easy design process. All the different processes (sensor measurements, measurement processing, sensor fusion, map building, path planning, task execution …) must be coordinated in an efficient way in order to allow accomplish a higher goal [2]. A number of control strategies can be set up, varying from simple serial sense-model-plan-act strategies to complex hybrid methods. A discussion of some of these control strategies can be found in [13]. An interesting approach here, is to use fuzzy behaviours, partially overriding each other, to build up complex navigation plans, as discussed in [9][10][11][12]. This research work aims at implementing such a hybrid control strategy.
During the design of all these sub-aspects, the outdoor nature of the robot has to be taken into account. Outdoor robots face special difficulties compared to their indoor counterparts. These include totally uncontrolled environments, changing illumination, thermal, wind and solar conditions, uneven and tough terrain, rain, …

The rest of this paper is organized as follows: The control strategy and architecture are described in section 2, the software architecture is summarized in section 3 and finally, conclusions are given in section 4.

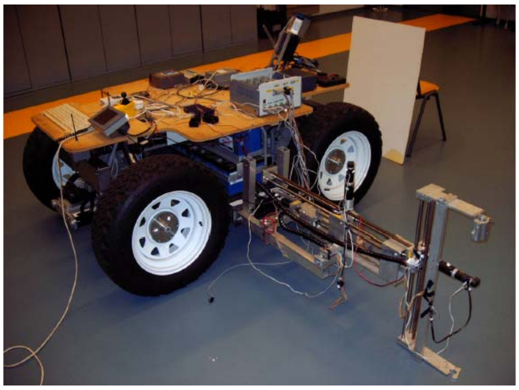

Figure 1: ROBUDEM robot with scanning mechanism

Control Architecture
The control architecture describes the strategy to combine the three main capabilities of an intelligent mobile agent: sensing, reasoning (intelligence) and actuation. These three capabilities have to be integrated in a coherent framework in order for the mobile agent to perform a certain task adequately.
The working principle of the proposed control architecture is sketched on Figure 2. There are three distinctive modules to be discriminated: Navigation (on the right side on Figure 2), Mine Detection - Scanning (in the middle on Figure 2) and Metal Detection (on the left side on Figure 2). These three processes are controlled by a watchdog, the robot motion scheduler, which manages the execution of each module and decides on the commands to be sent to the robot actuators. This robot motion scheduler is explained more in detail in Figure 3 and is discussed here more in detail for each of the three modules.

1. Navigation
Different Sensors provide input for a Simultaneous Localization and Mapping module
Sensors:
- GPS (Global Positionment System) gives absolute coordinates
- IMS (Inertial Measurement System) gives acceleration (and speed and position by integration)
- US (Ultrasonic sensors) give distance measurements to obstacles
- IR (Infrared sensors) give distance measurements to obstacles
- LASER gives line 3D data
- Mine Sensor: The mine imaging module will return locations of mines, which have to be represented on the map and which are obstacles themselves

As the SLAM module works with a global map, it doesn’t have to re-calculate the whole map from scratch every time, but the map can just be iterated to improve the different estimates, hence the loopback arrow. The SLAM module outputs a global map with obstacles and also with mines, thanks to the input from the mine imaging module. This map is used by the navigation module to calculate a safe path. The safe path is given as an input to the robot motion scheduler which will transform it into a motor command and execute it, unless another module has a higher priority task (and trajectory) to perform.

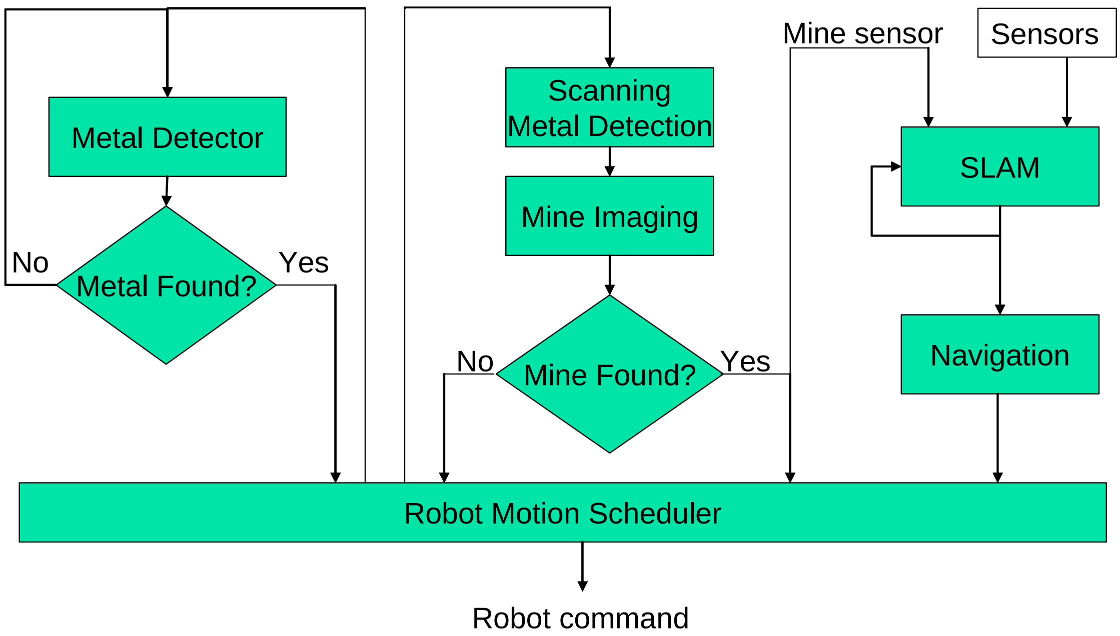

Figure 2: General Robot Control Architecture

2. Mine Detection

The Cartesian scanning mechanism makes a 2D scan with the metal detector. Mine imaging tools determine the likelihood of mine occurrence and the exact position of eventual mines. If a mine is found, this will be reported to the robot motion scheduler, which will take the appropriative actions. In addition to this the Mine detector acts as a sensor for the SLAM-algorithm, as it will return the locations of mines, which have to be represented on the map and which are of course obstacles themselves.

3. Metal Detection

The metal detector scans for metal in the soil. If no metal is found, it keeps on doing this and the robot keeps on moving. If a metal is found, this will be reported to the robot motion scheduler, which will take the appropriative actions.

Robot motion scheduler

The robot motion scheduler (Figure 3) needs to arbitrate which of the modules is executed and which of them can influence the robot actuators through robot commands.

Therefore, there are two main paths through the robot scheduler, one for the (normal) situation of exploring while avoiding obstacles and while detecting metals and one for the situation where a metal is found and more thorough investigation is needed (mine detection) while the robot is standing still.

In a normal situation, occurring e.g. in an initial situation (default inputs), or when the “no mine found” or “mine found” trigger are given, the scanning metal detection is turned off. The Navigation module gives at all time instances a safe path and trajectory, as this module loops infinitely without interaction with the other modules. This Trajectory is set as the trajectory to be executed, but with a low priority. The Metal detector module is activated.

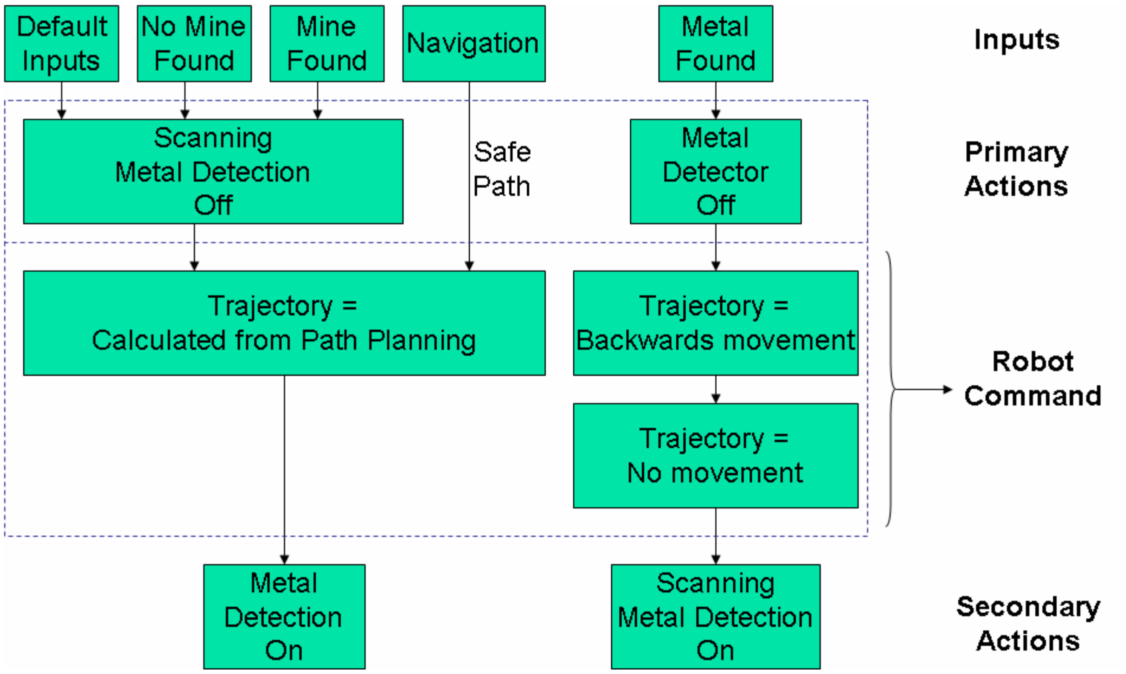

Figure 3: Control architecture for the robot motion scheduler

If the "metal found" trigger is given, the metal detector is switched off. The trajectory for the robot is set to a predefined movement, more specifically, to back off a little. This is done to be able to centre the scanning metal detection better around the suspicious object. This trajectory has a high priority. When this movement is completed, the robot is halted, by giving a "no movement" trajectory with a high priority. Finally, the scanning metal detection module is activated.

Software Architecture

As control architectures which aim to mimic human thinking risk of becoming highly complex, the choice of a flexible, extendable and real-time capable software architecture is very important. This software architecture has to ease the use of reusable and transferable software components. The chosen software architecture, MCA (Modular Controller Architecture) [14] achieves this by employing simple modules with standardized interfaces. They are connected via data transporting edges which is how the communication between the single parts of the entire controller architecture is managed. The main programs only consist of constructing modules that are connected via edges and pooled into a group. This results in an equal programming on all system levels. As modules can be integrated both on Windows, Linux and on RT-Linux without changes, they can be developed on Linux-side and then transferred later to RT-Linux. As errors in RT-Linux lead to system-hangs this development strategy prevents from many reboot cycles and results in faster software development.

The proposed MCA software architecture, as it is depicted on Figure 4, consists of three main groups: one for sensor-guided robot control (using a behavior based navigation method and SLAM), one for Scanning metal detection and one for metal detection.

The robot motion scheduler controls which of the three groups is executed and with which parameters. Each group consists of several modules and/or subgroups.

Each MCA module is determined by four connectors with the outside world: Sensor input (left below), Sensor output (left top), Control Input (right top), Control Output (right below). As a result sensor data streams up, control commands stream down. The Sensor input and output are connected through a Sense procedure which enables to process the sensor data and the Control input and output are connected through a Control procedure which enables to process the control commands. Sensor data flow is shown in yellow, control command flow in red.

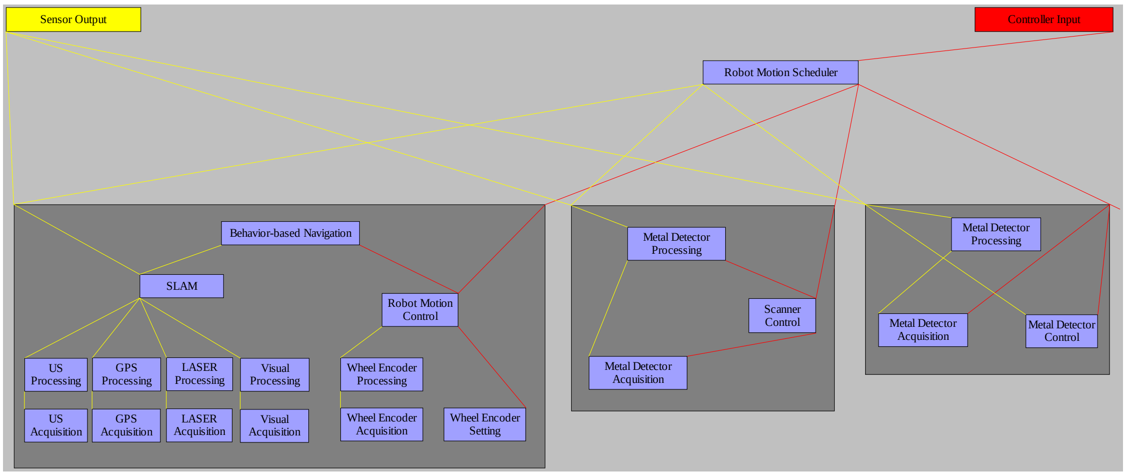

Figure 4: MCA Software Architecture

For now, the scanning and metal detection modules are implemented and operational. The X-axis of the scanner has been removed in the mean time, so scanning is only performed in the Y-direction. The whole architecture contains interfaces that can be used via TCP-IP (Ethernet). In this way all sensors values can textually or graphically be presented on a second PC. A common graphical user interface has been developed to simplify the procedure. Figure 5 shows the graphical interface which was developed for controlling the mine detection process. This computer interface enables the user to control the robot scanning mechanism or to order the robot to scan the suspected area for mines. It also shows the map of suspected mine locations, as detected by the robot. This map is shown here in an initial stage where all non-scanned, and therefore non-cleared, terrain is treated as suspected, and therefore indicated with a red led. As the SLAM and path planning modules are not implemented yet, the robot is currently still restricted to following predetermined trajectories.

Conclusions

In this paper, we have demonstrated our solution for the control problem of a mobile humanitarian demining robot. The results so far are encouraging: the robot is able to follow a predetermined trajectory and find mines along this path. Future research will enable the robot to find its way semi-autonomously, by the integration of extensive navigation, map-building and path-planning techniques. These will be integrated in a behaviour based reactive-reflexive framework, such that the robot can at the same time react quickly to dynamic changes in the environment, and perform high-level reasoning on a 3D model (map) of the environment.

References

[1] R. R. Brooks and S.S. Iyengar. Multi-Sensor Fusion. Prentice Hall, New Jersey, USA, 1998

[2] J. Budenske and M. Gini. Achieving Goals Through Interaction with Sensors and Actuators. Department of Computer Science, University of Minnesota, Minneapolis, USA, 1992

[3] G. De Cubber, H. Sahli, F. Decroos, Sensor integration on a mobile robot, ICMCR02, International Conference on Measurement and Control in Robotics, pp 89-92, Bourges, France, 2002

[4] T. D. Larsen. Optimal Fusion of Sensors, Technical University of Denmark, Department of Automation, Lyngby, September 1998

[5] J.C. Latombe, J.-C., Robot Motion Planning, Kluwer Academic Publishers, 1991.

[6] M. Lopez, F.J. Rodriguez, J.C. Corredra, Fuzzy Reasoning for Multisensor Management, 1995 IEEE International Conference on Systems, Man, and Cybernetics, Vol. 2, pp. 1398-1403, October 1995, New York, USA

[7] K. Nagatani, H. Choset, S. Thrun, Towards exact localization without explicit localization with the generalized Voronoi graph. Carnegie Mellon University, USA

[8] R. R. Murphy. Sensor Fusion. In Handbook of Brain Theory and Neural Networks, Colorado School of Mines, Dept. of Mathematical and Computer Sciences

[9] A. Saffotti Proceedings. Fuzzy Logic in Autonomous Robotics: behavior coordination, Proc. of

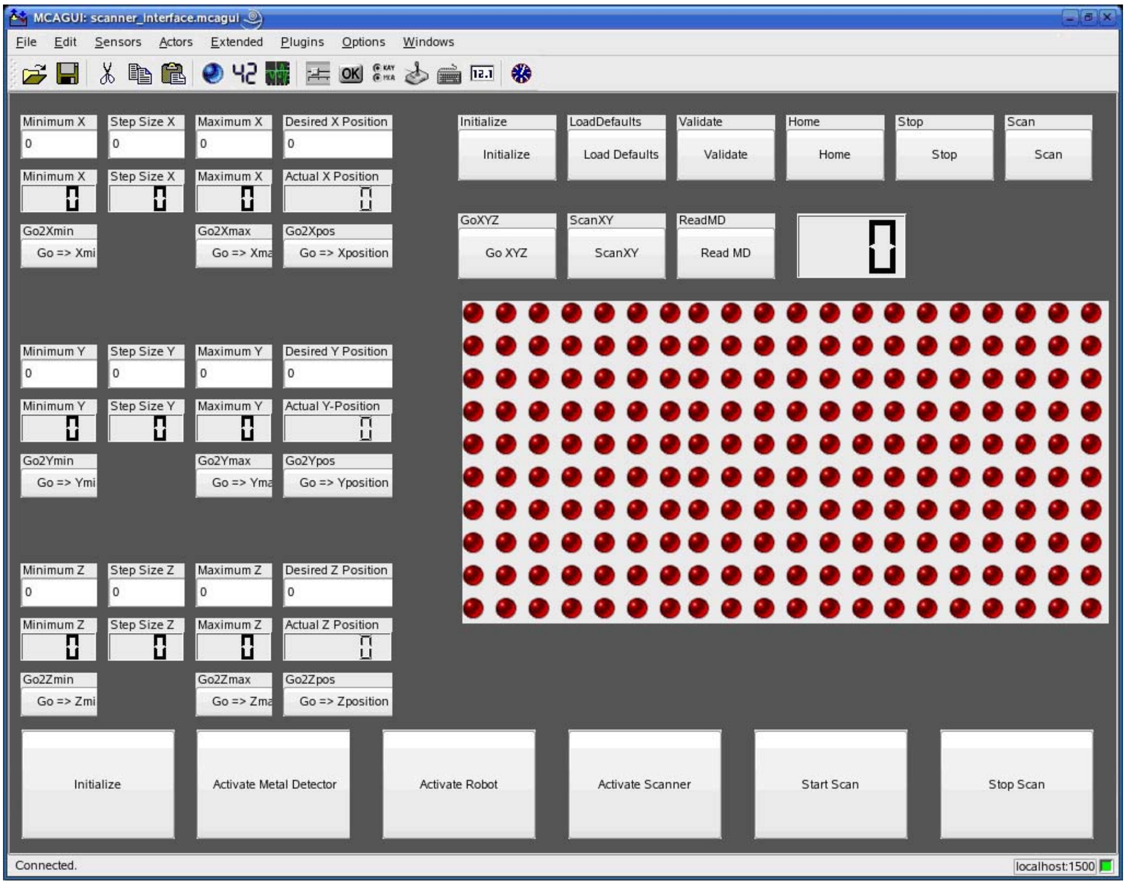

Figure 5: Graphical interface of the control program

the 6th IEEE International Conference on Fuzzy Systems, Barcelona, Spain, July 1997

[10] A. Saffiotti, K. Konolige, E. H. Ruspini, A Multivalued Logic Approach to Integrating Planning and Control, Artificial Intelligence Center, Menlo Park, Califonia, USA, 1995

[11] A. Saffiotti, E. H. Ruspini, K. Konolige. Robust Execution of Robot Plans Using Fuzzy Logic. In Fuzzy Logic in Artificial Intelligence – IJCAI '93 Workshop, LNAI 847, Springer-Verlag, Berlin, DE (1994), pages 24-37

[12] H. Schäfer, K. Berns, RAVON - An autonomous Vehicle for Risky Intervention and Surveillance, International Workshop on Robotics for risky intervention and environmental surveillance – RISE, 2006

[13] G.G. Schenkel, Memory Management, Navigation and Map Building. Universität Stuttgart, Stuttgart, Germany, 1994

[14] K.U. School, J. Albiez, B. Gassmann, MCA – An Expandable Modular Controller Architecture, Forschungszentrum Informatik an der Universitaet Karlsruhe (FZI), Interactive Diagnosis and Service Systems

[15] P. Sykacek, I. Rezek. Markov chain Monte Carlo methods for Bayesian sensor fusion. Kluwer Academic Publishers, 2000

[16] S. Thrun and A. Bücken. Learning Maps for Indoor Mobile Robot Navigation. School of Computer Science, Carnegie Mellon University, Pittsburgh, USA, April 1996

[17] G. Welch, G. Bishop. An Introduction to the Kalman Filter. University of North Carolina, Department of Computer Science, Chapel Hill, USA, 1996