A CONCEPT OF A ROBOT FOR THE ROBOTOUR COMPETITION

Jaroslav Rozman[1], František V. Zbořil[2]

[1]Brno University of Technology, Faculty of Information Technology,
Božetěchova 2, Czech Republic
[2] Brno University of Technology, Faculty of Information Technology,
Božetěchova 2, Czech Republic

rozmanj@fit.vutbr.cz (Jaroslav Rozman)

Abstract

This paper deals with design of an autonomous robot for the robotic competition Robotour. It describes short history of this competition and its rules with changes for this year. The second section describes hardware of the robot, its sensors and proposes changes and modifications suitable or necessary for participation in the competition. In the section Objectives and strategy the strategy for participation in the competition based on robot's sensors and appropriate algorithms is presented.

Keywords: Robotour Competition, Autonomous Robot, Navigation, Path Finding

Presenting Author's biography

Jaroslav Rozman was born on November 3, 1979 in Brno. He entered the Faculty of Electrical Engineering and Computer Science of the Brno University of Technology in 1998 where he studied cybernetics and automation. He received M.Sc. in 2003. He is currently a Ph.D. student at the Faculty of Information Technology of University of Technology, Brno. His main research interests are robotics, artificial intelligence and computer vision.

1 Introduction

Competitions like the famous DARPA Grand Challenge and Urban Challenge or European Eurobot [1] are a favorite way to test autonomous robots and new algorithms. It is also great chance to meet other people involved in robotics and share their experience. That is why there are new competitions appearing. Some of them are focused purely on software and no real robots are required. Others need robots constructed according to the goal of the competition and equipped with necessary sensors and actuators. A typical example of such competitions is Eurobot, where the goal is different every year, therefore, it is necessary to construct slightly different robot every year.

2 Robotour Competition

One of these competitions is the Robotour competition [2] which started in 2006 as an mini analogy to the DARPA Grand Challenge. The goal of the competition is similar; the robots should travel the path from the start to the finish and should not leave the path. To make the competition open for everybody, the race itself is held in a park (Fig. 1) and the robots are represented by small vehicles instead of real cars or big vehicles which are capable to overcome tenths of kilometers.. This makes the contest accessible not only for the people from the university, but also for other people, who have robotics as a hobby. This also contributes to the wide range of various robots from professional solutions to the homemade robots with simple sensors and even simpler algorithms.

The rules in the first year were quite simple. Robots had to carry a payload along the closed loop on the paths in the park and they gained points for every finished round. The only restrictions on the robots were the use of the electric power and the size of the robot (the robot should be small enough to move around the narrow paths in the park and ,at the same time, it should be able to carry the payload- a small keg filled with water). The payload was not obligatory, but the robots with the payload were awarded by the double of points. The important rule was that if the robot went off the road it could have been stopped and put back on the road with some penalty.

2.1 The Goal

The rules changed slightly every year and every year they were stricter; in the last year the robots which went off the road could not return and their race ended. The changes for the upcoming year are even bigger. The robots will not ride any circles in the park, their task will be to get to the goal placed somewhere in the radius of 1km within 30 minutes. They will start from the same place at the same time. Each robot will be placed in the 1,5x1,5m area at one side of the road.

Fig. 1. The winning robot from the LEE team during Robotour 2009.

2.2 Robots

The particular construction of the robot is up to every team. The only restriction is that the robot has to have a maximum speed of 2.5m/s and red emergency button to stop the robot immediately. The race for the robot is over after pushing this button. The robots must not be too big so that they are able to ride in all the paths in the parks. On the other side the robots have to be able to carry 5L beer keg, at least an empty one.

2.3 Map

The map will not be provided in RNDF [3] format as in the previous years and hence it will not be similar to the map of DARPA Urban Challenge. Instead , the idea of the Open Street Map will be used and the map will be given in JOSM format [4]. The teams and their robots have to use this map supplied by organizers of the competition. This arrangement should stop teams from spending days in creating their own maps and force them to use one standardized map - understandable even to humans- for all the robots. This solution also ensures that robots can go to the areas where they have never been before and use only a map created for humans.

2.4 Obstacles

The robots have to be able detect and avoid obstacles. It can be benches and other things in the park like flowerpots or bins. Obstacles can also be artificial like boxes placed in front of the robot for testing of the ability of avoidance. Other robots and moving people in the park are also considered as obstacles. If the robot hits an obstacle, its race will end for him.

2.5 Points

The points will be calculated as a difference between airline distance form start to the end and the end point of the robot and the end. If the robots carry the full keg they will get double points. If the robot hits an obstacle or if it goes off the road the race will be over for it and this place will be considered as its final point. The competition will have four rounds, each with different start and finish.

3 Robot Hardware

Our robot (see Fig. 2) consists of a small chassis based on the Lynxmotion 4WD1 robot. The chassis itself is made from lexan and it has five decks. There are four motors with encoders in the space above the bottom deck. On the deck above there are batteries and the board for motor controlling. Next deck holds sonars, a compass, an accelerometer and necessary wiring. The deck above contains the controlling board called FITkit [5]. FITkit has replaced the original board supplied with the Lynxmotion robot.

The computing power of the FITkit is not sufficient to control the robot and so the robot is controlled by an external PC. The rules of the competition require the robot to be autonomous and so it will be necessary to equip the robot with a laptop. Because of the small dimensions of the robot (approx. 20x23cm) the only suitable laptops are 10” or smaller. The small size of the robot is its biggest disadvantage and therefore it will be the most challenging problem. The keg will be attached at the back of the robot on the aluminum holder. The keg has to be empty which means points loss, but the robot is too small to carry extra load with weight of more than 5kg.

Fig. 2. The complete robot as a result of the masters work of one of our students [6] .

The sensors on the robot today are sonars, accelerometer, compass and four encoders. Sonars and the compass are connected via the I2C bus to the FITkit. Encoders and accelerometer are connected to the FITkit directly. The sonars are SRF08 and they create a ring around the robot. The angle of the sonar beam is around 55°, therefore the ring will be created by 5 or 6 sonars. The ranging distance is denoted from 6cm to 6m, but in reality it could be about 3 or 4 meters. The measurement of the one sonar takes 65ms so we get the information about distances around the robot every 65ms times number of sonars. In case we need information about distances more often, we can start the measuring from all the sonars at one time, but then errors may occur because some of the sonars can detect the sound from another sonar. The sonars are the only robot’s sensors for distance measurement and for obstacle avoidance, so their correct function is crucial and maybe some of the sonars will be doubled.

Sensors for determining of the distance moved are encoders and accelerometer. The function of the encoders is quite simple; they send pulses based on the rotation of the wheels, where one rotation corresponds to 120 pulses. We can determine the rotation and the distance traveled by the robot from this information. The information about motion and heading of the robot can also be given by the 3-axis accelerometer ACC7260. The last sensor of the robot is a CMPS03 compass. The disadvantage of the measurements from the encoders and from the accelerometer is that the accumulated error grows with the distance moved. We need to eliminate this error, so we have to compound both measurements together and correct it by the other sensors whose error does not depend on the distance moved from the start. This means we have to correct the position of the robot by measurements from the compass and try to correct it also by the measurements form the sonars.

Because we won’t know the exact position of the start point, it would be better to supply the robot with a GPS module. There are two ways to connect it to the robot. The first is to connect the GPS directly to the robot and send the measured positions via FITkit to the laptop. It would be harder to implement but it is a somewhat “cleaner” solution. Second, we can connect GPS module to the laptop. This solution is easier to implement, but it is not as good as the previous one.

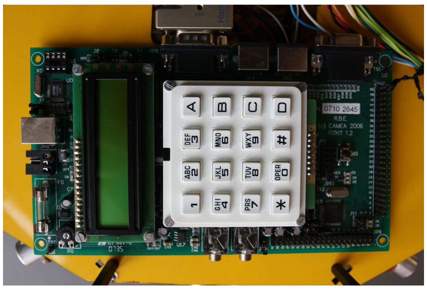

Fig. 3. Educational board FITkit with MCU and FPGA units. All the robot sensors are connected to the bus in the down-right side.

The heart of the robot is the FITkit as we mentioned before. The FITkit (Fig. 3) is a board developed by our university for students as an educational platform, which helps them to familiarize with hardware programming and to handle with various peripherals. It is widely used in courses and students can use it in their bachelor or master theses. It contains low-power microcontroller and a set of peripherals like keyboard, LCD display, audio and video interface and expansion connectors. It also contains FPGA. The robot, as in the case of FITkit, helps the students to learn about hardware and embedded systems and to be able to test them on a real system. Besides, the robots are en excellent tool for education on the technical universities. Because of their complexity, they can be used for learning things like wireless communication, regulation and controlling, modeling, computer vision, artificial intelligence, hardware and many others.

4 Objectives and Strategy

The objectives the robot has to accomplish results from its sensors and from the rules of the competition can be divided into five parts:

- Start
- Finding of the starting position
- Path planning
- Obstacle avoidance
- Ride on the path

The start will be collective. Each robot will start from its own square with dimensions 1,5m x 1,5m. To prevent collisions among many robots and people during the start, the robots must not be touched one minute before the start. This means that some delay that postpones the start about at last one minute has to be implemented.

The path planning is the simplest task of all. If we know the start position and the position of the goal, we can use one of the well-known algorithms e. g. Uniform Cost Search for the state space searching.

Finding the starting point will be much more difficult. It can be divided into two cases. First, we have GPS sensor, so we can determine position of the robot. We can assume that the position will not be precise because of the errors in the GPS. If the error is not too big, approximately 10m or a bit more, it should not be a problem. We know that the robot will start from the road, so we can specify the position of the robot to the road lying inside the circle given by the GPS. Moreover, as we can place robot anywhere inside its starting box and set him in a direction we want, we can point him in a direction of the road. This procedure will determine the position of the robot with sufficient precision.

The problem will arise if we do not have a GPS sensor. Then, we will have to determine the position of the robot by some software. Probably the only possible way is to use some of the localization algorithms. The most suitable is the particle filter, so called Monte Carlo Localization (MCL) [7]. The error of the position determined by this algorithm would be probably a few meters which should be sufficient to control the robot on its way towards the goal. The disadvantage of the localization algorithms is that they require movement of the robot. It is rather obvious, but it means that robot has to cover some distance before it will be localized enough. It means more obstacle avoidance, and more controlling to keep the robot on the road. It increases the probability of a failure and it costs some time, because there is no guarantee that the robot during its localization will take the right path directly to the goal.

```
$MonteCarloLocalization(\chi_{t-1}, u_t, z_t, m):$

$\overline{\chi}_t = \chi_t = 0$

for m = 1 to M do
    $x_t^{[m]} = sample\_motion(u_t, x_{t-1}^{[m]})$
    $\omega_t^{[m]} = measurement(z_t, x_t^{[m]}, m)$
    $\overline{\chi}_t = \overline{\chi}_t + \langle x_x^{[m]}, \omega_t^{[m]} \rangle$
endfor

for m = 1 to M do
    draw i with probability $\approx \omega_t^{[i]}$
    add $x_t^{[i]}$ to $\chi_t$
endfor

return $\chi_t$
```

Algorithm 1: The pseudoalgorithm for the Monte Carlo Localization with functions for the sampling of the motion and for the measurement.

As we can see, it is much more suitable to use the GPS. But not even using of the GPS can help the robot to find its starting position. The common problem is that GPS is often not able to localize itself because of the buildings or the trees. As a result, relying only on the GPS could have fatal consequences. That is why probably the best solution would be to use both GPS and MCL and merge information from both systems together to get the best knowledge about robot position. Moreover, the particle filter used in MCL can also be used to control the movements of the robot.

As for the obstacles, there can be two kinds; static, like benches or tree trunks and dynamic, like people or other robots. It is necessary for the robot to be able to drive around them. But the first task is to detect them correctly. The robot is equipped with sonar with quite a large beam, which means it has to get to the right distance before the obstacle. Let us assume that the robot will detect obstacle with its front sonar on the place, where should be free road. Because of the wide sonar beam, it is necessary to get close enough to make sure that it is really an obstacle and not a narrow but rideable, passage. If the front sonar returns shorter distance than front left and front right sonar, this will mean that the obstacle is right before the robot and it will try to drive around to the left, which corresponds to the by-law. In the case of the obstacle detection, it is also possible to start with creating of the occupancy grid-map and searching the way around the obstacle in this map.

The most challenging task will be the ride on the path. As the robot is not equipped with any camera, it is not able to detect the road and has to rely only on its sonars and the odometry. The odometry is not usable for longer distances. The biggest errors in odometry arise when turning, so its only usefu on the straight paths with minimum or no curves. The sonars can be used on the paths with clearly marked edges, like the curbs, bushes, walls and so on. If the path changes into lawn gradually, the sonars will have no chance to detect that it is not a path any more. There is one possible way how to detect this road edge. Because the robot has accelerometer, that is possible to detect the edge of the road as the increased vibrations or increased tilt of the robot. That is we can assume that the road will be smoother than its edge with grass. So the outcome is to detect the edges of the road by the sonars and where it is not possible because of the lack of clearly marked edges to rely on the pure odometry and try to detect the edge with the accelerometer.

5 Robot Software

The software of the robot consists of two parts. First part is the firmware in the FITkit. It provides the communication between the robot and the controlling computer and it also provides the functionality of all robot sensors. This is done via the I2C bus which connects all sonars and the compass. The other peripherals are connected directly to the pins of the FITkit. The FITkit receives data from the sensors, sends them via the serial port to the computer and the computer sends back the controlling signals for the motors. The communication between FITkit and the computer can be wireless with bluetooth devices or by the serial cable. The program in the FITkit is written in C/C++, except some parts in the VHDL.

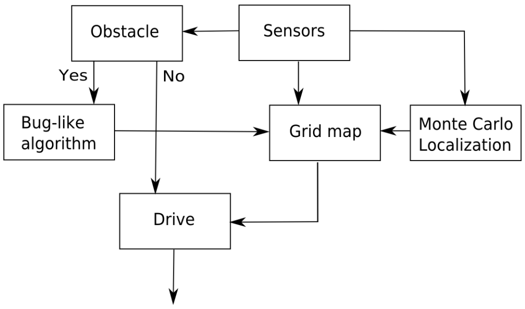

Fig. 4. The scheme of the local path planning – the obstacle avoidance.

The second part of the software is in the controlling computer, which is now not in the robot but on the competition it will be placed on the top deck. Considering the dimensions of the robot, probably the only suitable controlling computer is a small netbook. The software in the computer depends on the robot sensors and on the chosen strategy. There is a part that receives the data directly from the GPS module. Another part will keep two structures for the maps. First is a geometrical global map in the JOSM format and the second is a local grid-based map. The global map will serve for the path finding from the start to the goal. The output of this part will be the sequence of the coordinates from the robot starting point to the goal. The data from the sensors will be mapped in the local map, which will be used for the obstacle avoidance and the local path planning. The obstacle avoidance will use implementation of the Bug algorithm [8] which will prefer avoiding to the left side, because the rules of the competition demand the robots should keep the by-law.

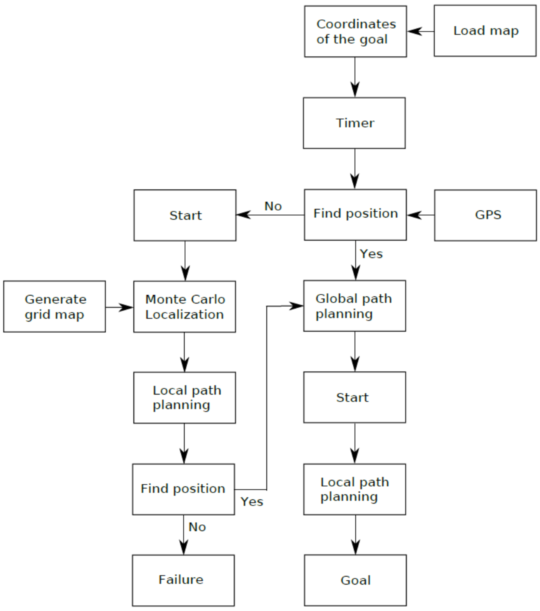

Fig. 5. The block-scheme of the function of the robot. The local path planning is on the Fig. 4.

6 Conclusion

This paper describes a concept of a robot for the Robotour 2010 competition which will held in one of the three parks in Bratislava, Slovakia. It describes the rules of the competition and the hardware and the software of the robot which were created as the results of BSc. and MSc. projects. The paper also suggests the strategies for successful participation in the competition.

Acknowledgement:

This project was supported by the Ministry of Education, Youth and Sport research project No MSM 0021630528.

7 References

[1] Eurobot competition. [cited 15.1.2010] [online] Available http://www.eurobot.org/

[2] Robotour competition. [cited 15.1.2010] [online] Available http://robotika.cz/competitions/robotour2010/cs

[3] DARPA, “Route Network Definition File (RNDF) and Mission Data File (MDF) Formats,” [cited 15.1.2010] [online] Available http://www.darpa.mil/GrandChallenge/docs/RNDF_MDF_Formats_031407.pdf

[4] Java OpenStreetMap Editor, [cited 15.1.2010] [online] Available http://wiki.openstreetmap.org/wiki/JOSM

[5] FITkit, [cited 15.1.2010] [online] Available http://merlin.fit.vutbr.cz/FITkit/

[6] D. Kubát. Robot pro Robotour 2010, Masters Work, Brno, FIT VUT v Brně, 2010

[7] S. Thrun, W. Burgard, D. Fox. Probabilistic Robotics, MIT Press, Cambridge, 2005

[8] H. Choset et al. Principles of Robot Motion, MIT Press, Cambridge, 2005