National Conference with International Participation
ENGINEERING MECHANICS 2007
Svratka, Czech Republic, May 14 – 17, 2007

HIGH LEVEL SOFTWARE FOR AUTONOMOUS RACING ROBOT

J. Krejsa[*], S. Věchet*, V. Ondroušek[+]

Summary: *The paper describes the high level software issues in the task of autonomous robot driving on the park pavement. The wheel robot is equipped with limited number of sensors - odometry IRC sensor on front wheels and vision sensor represented by digital camera. The task in question is to use the sensor data in order to autonomously navigate on the path given in predefined map. The map is given in RNDF (Road Network Definition File) format specified by DARPA. The path to be taken by the robot is given in the MDF file (Mission Definition file), which contains the sequence of check points in RNDF format. The paper gives detail information on the development of software tools for the task and describes our experience with testing different ways of sensors data acquisition.*

1. Introduction

Fully autonomous behavior still represents a challenge in mobile robots development. In order to test robots abilities various competitions gives a good opportunity to compare different approaches in more or less defined environment. DARPA grand challenge is a representative of such an event. Most of the tasks are independent on the size of the robot, therefore methods and software routines developed on smaller size robots (which are substantially cheaper) can be successfully transferred into full size robots. This paper describes high level software development of autonomous racing robot Bender, developed for Robotour 2006 competition held in Prague in 2006. The task to be solved was to drive autonomously on park pavements, following the defined path on predefined map. The robot is a four wheel mobile robot, equipped with IRC sensors on front wheels, digital camera and GPS sensor, carrying a HP subnotebook for the control. Technical details can be found in [1].

2. Outside world and robot representation

The outside world is defined as 2+D map (two dimensional map with information about the slope of certain road segments). The map is given in RNDF (Road Network Definition File) format specified by DARPA. In this format, the map network consists of several segments. Each segment consists of several lanes. Each lane is represented by the list of waypoints containing latitude and longitude in WGS84 system. Lanes are oriented and can contain

* Ing. Jiří Krejsa, PhD., Ing. Stanislav Věchet, PhD., Institute of Thermomechanics, Czech Academy of Science, Brno branch, Technická 2, 616 69, Brno, tel: +420-541142885, email: krejsa@fme.vutbr.cz
[+] Ing. Vít Ondroušek, University of Technology, Fakulty of Mechanical Engineering, Technická 2, 616 69, Brno, tel: +420-541143356

special waypoints – exits and check points. Exits define the interconnection between lanes of various segments by the pair of waypoints on two lanes belonging to different segments (therefore the way how certain crossroad can be passed is uniquely defined). The example of RNDF file format is shown on fig. 1a. As the slope of the segments can play a role in robot motion control, the slope of the lanes can be added into the map.

The path the robot is required to follow is defined using MDF file (Mission Definition file), also specified by DARPA. It is basically the sequence of checkpoints defined in corresponding RNDF file.

RNDF_file example num_segments 30 num_zones 0 creation_date 17.6.2006 segment 1 num_lanes 1 lane 1.1 num_waypoints 10 lane_width 5 check_point 1.1.2 1 exit 1.1.10 2.1.1 exit 1.1.10 3.1.1 1.1.1 50.063171 14.256622 1.1.2 50.063141 14.255622 ... end_lane end_segment segment 2 num_lanes 2 lane 2.1 ... end_file	MDF_file example_path RNDF example checkpoints num_checkpoints 3 checkpoint_id 3 checkpoint_id 143 checkpoint_id 32 end_checkpoints speed_limits num_speed_limits 0 end_speed_limits end_file
RNDF example	MDF example

Fig. 1. RNDF and MDF formats

The RNDF file contains basically the GPS coordinates of points on the road. It is advantageous to process the information to extract certain features of the map in the form which is more natural for the robot motion control. The MDF and RNDF files are therefore processed prior to the mission to create internal representation of the path to be taken.

The MDF&RNDF parser, called Mapedit, was created for this purpose (and implemented using Delphi 7). This software is not connected to the main control software of the robot. Mapedit can read and vizualize MDF file, as well as paths placed in the RNDF format. Furthermore, Mapedit can modify each waypoint, lane or segment. This is necessary in the case where input file doesn't match the real world or the input file is insufficient, or the number of waypoints is low. This software also enables user to set the path interactively based on loaded MDF file. The ability to create new paths in short time is the key feature, as has been shown mainly in the phase of testing the control algorithms in the real world (city park)

on different paths. The main purpose of Mapedit software is to convert paths from RNDF format to the internal representation used by the control software of the robot.

The internal representation consists of list of path segments. Each segment contains list of points which hold both the GPS coordinates and Cartesian [x,y] coordinates in meters, originating in the path beginning. The Cartesian coordinate system is oriented with x axis to east and y axis to north. The points are generated depending on the curviness of the segment lane so that direct parts only contain first and last point of the path. Each segment starts and ends on the crossroad, which is described by the type of the crossroad (left, right, T, X, V), required turning angle (defined as the difference of slopes in the last pair of points in the pre-cross segment and the first pair of points in the post-cross segment). Such information is essential for the crossroad identification using image processing routines.

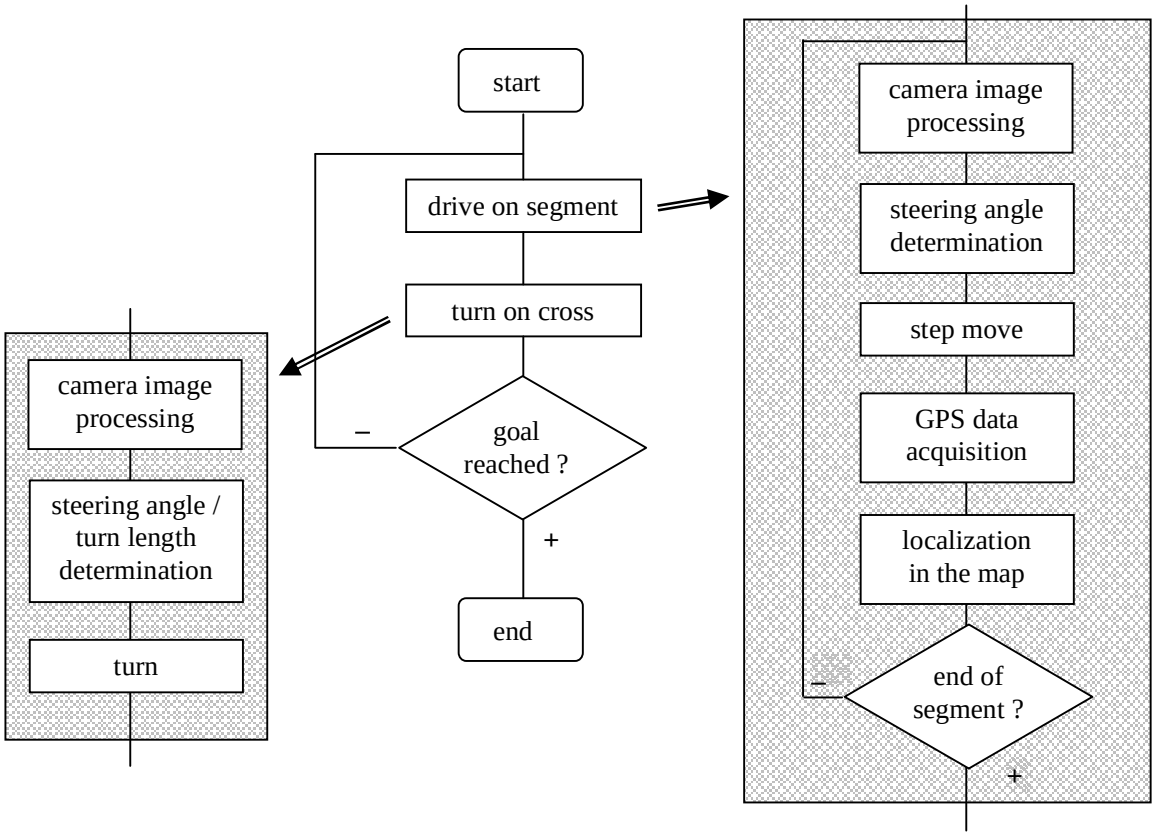

Fig. 2. Main control loop

3. Control algorithms

During the robot motion it's position must be identified on the map path [2,3]. Robot's position is calculated independently based on the odometry sensor readings (left and right front wheel IRC sensors) using the Ackerman steering model. The history of the steering together with traveled distance is logged and used for the corrections of the total traveled path on the map. GPS information is logged the same way, including the information on trustworthy of the data and during the segment travel and prior to the crossing the data are fused together to find the current robot position on the map. As the GPS information is often unreliable (depending on the actual number of satellites visible from current location) the

weight of the GPS data is reduced. Main control algorithm repeats the simple loop shown in Figure 2.

During the drive on the segment the steering angle of front wheels is determined by the analysis of the CCD camera image. Image processing determines the relative distance of road curbs (or the road obstacle). Only blue channel of the signal is used due to the high contrast between road and its surrounding. Portion of the image (smaller for segment drive, higher for cross roads) is summed up, filtered with floating average filter and thresholded. The widest portion of the thresholded signal corresponds to obstacle free road.

4. Tests

Number of tests was carried out during developing the control software of the robot.

4.1 First set of tests - straight pavement

The test was carried out on two straight hard pavements. One of them was wide and concrete and the other one was made out of interlocking pavers, each of them with different light reflection characteristics. There were curbstones on both sides of the pavements separating the pavement surface from light-green grass. We had designed several ways of data interlock (data from vision and IRC sensors). The tests showed which of the designed ways would be used for the motion control of the robot and allowed for the calibration of IRC sensors.

First set of tests revealed several critical problems:

- change in robot´s geometry after the robot hit a curbstone

solution: stiffer springs in the axle

- accuracy on IRC sensors depending on light strenght

solution: sensor encasement

- sensitivity of the vision sensor to sunlight

partly removed by adding a protection light shield to the vision sensor

- small field of vision of the camera

solution: use of a divergent lens placed in front of the sensor

4.2 Second set of tests - small circuit

The test was carried out on a hard circle pavement with diameter, d = 7 m, and one crossroad, see fig. 3. The aim of the selection of this pavement was to test the ability of the robot to take a path that turns, and robot's behavior on crossroads as well as the ability of control algorithms to cope with unclear edges of the pavements. The tests resulted in changes in the control algorithm defining robot's behavior on crossroads.

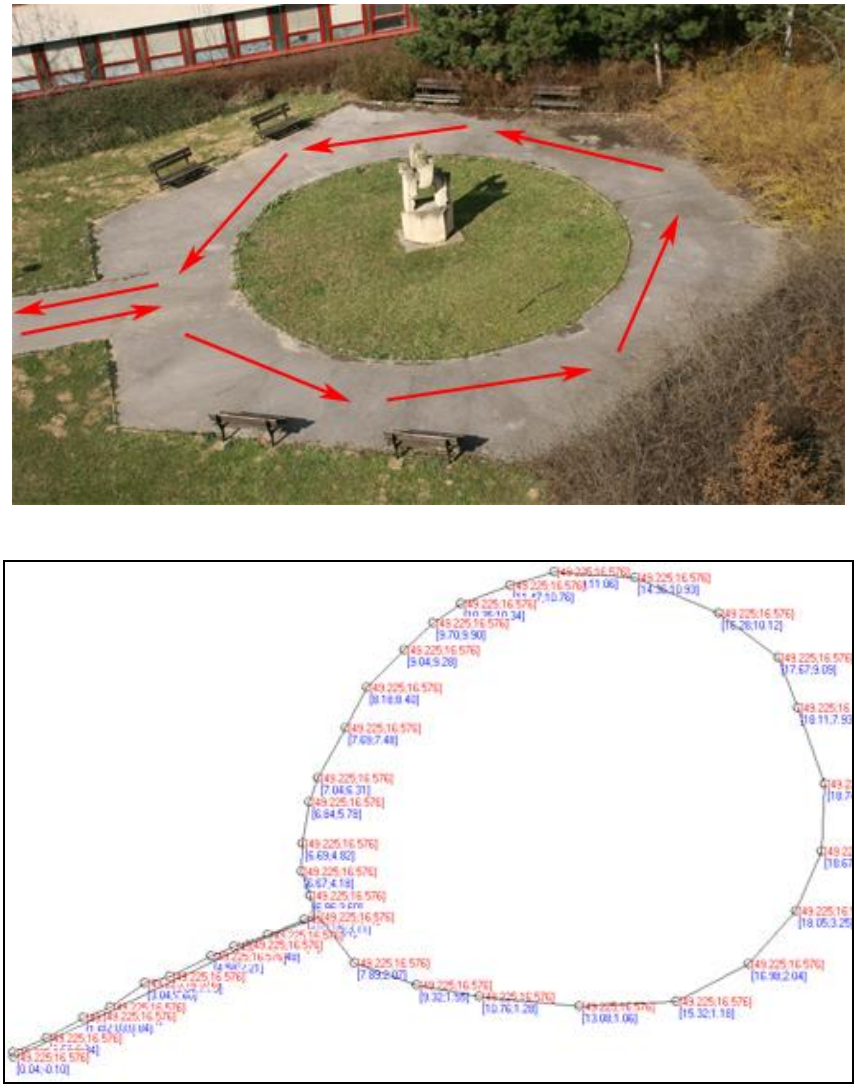

Fig. 3. Test drive (a) on the small circuit (b) internal representation of it's path

4.3. Third set of tests – athletic field

So as to achieve a more precise localization of the robot in the map and to acquire additional data to IRC-sensors data, a GPS sensor was placed on the robot. The tests were carried out on a 200-meter long straight pavement with a crossroads at its end. They revealed low accuracy of GPS-provided data so successful interlock with IRS data was impossible. The accuracy was influenced by trees along the pavement and low speed of the robot and that is why the GPS use was rejected.

4.4 Fourth set of tests - large city park Lužánky in Brno downtown

The aim was to test the robot in conditions similar to those of the final competition. For this reason a large park had been chosen, with many crossroads of different shapes, slight changes in the altitude, different types of pavements surfaces and a high number of obstacles (people and dogs). Before testing we had to design a map of the park in MDF format using Mapedit software. Big number of paths in RNDF format with lengths ranging from 50 m to 800 m was designed while testing.

On Figure 4 one can see the example of the progress of image signal during the obstacle avoidance from this set of tests. The values on the left graph are unfiltered means of the blue channel, values on the right graph are filtered and normed to 0-255 as single byte is used for signal value content.

The tests have shown the ability of the control software to cope with dynamic obstacles (walking people on the park pavements). The tests have proved that the approach used in the control software design was correct. It turned out that it was necessary to integrate information on robot's speeds into the internal representation of the path. Some parameters used in the localization of the robot in the map were precised. Long-lasting tests confirmed a very good battery life (both robot and notebook batteries - approximately 6 hours). The test was satisfying and showed that the robot was well-prepared for the final competition.

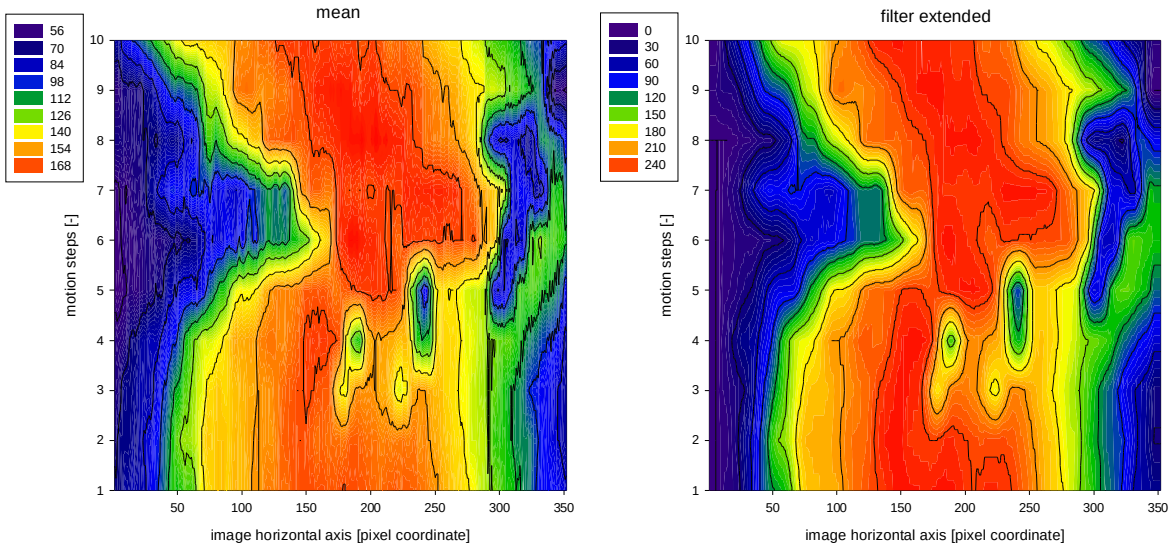

Fig. 4. Obstacle avoidance – filtered image progress

4.5. Final Competition - Robotour 2006 held in Prague

The Robotour 2006 competition track is shown on Fig. 5. The track consisted of mixture of interlocking pavers and asphalt roads with unclear edges (layers of leaves). The total length of the track was about 800 meters. The light conditions for vision sensor were strongly effected by the direct sunlight.

The track was circled repeatedly 5 times, competition lasted the whole day with light conditions changing. Due to the high number of visitors the dynamic obstacles were present at all times, however the control algorithms tuned in previous tests proved to be robust enough and the robot was the only one which finished the whole track successfully.

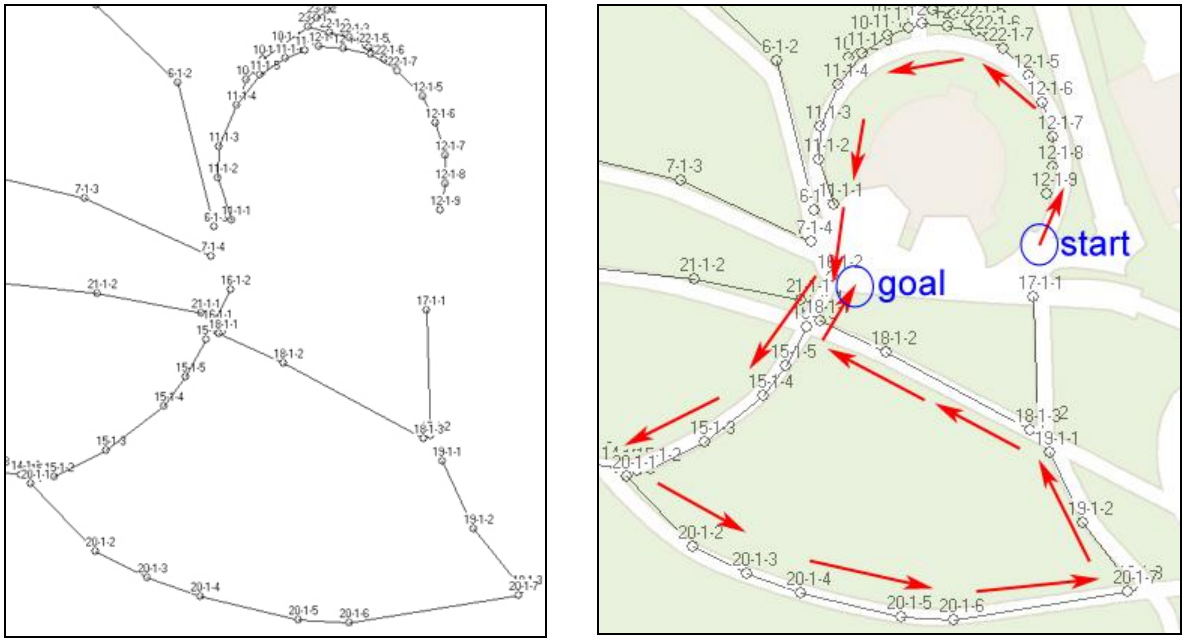

Fig. 5. Map of the final competition: (a) graphical representation of the input RNDF file (b) combination with a geographical map

5. Conclusion

This paper summarizes our experiences with the development of high level software for the autonomous mobile robot control, including the necessary software tools for map and path editing. Sensor data fusion is the main issue for successful performance of the robot. Presented approach proved to be usable, however there is still a number of problems, mainly in open spaces crossing and robustness of the motion planner to variable light conditions.

Acknowledgement

This paper was written with support of the pilot project no.: 920034 of Czech Academy of Science as a part of research project AV0Z20760514.

References

[1] Věchet S., Krejsa J., Ondroušek V..: The Development of Autonomous Racing Robot Bender, Engineering Mechanics, to be published

[2] Hu H., Gu G., Brady M.: Navigation and guidance of an intelligent mobile robot, Proceedings in Second Euromicro Workshop on Advanced Mobile Robots (EUROBOT '97), 1997.

[3] Šolc F., Žalud L.: Robotika, Brno Faculty of Mechanical Engeneering, 2002.