Eurathlon 2013

Scenario Application Paper (SAP) – Review Sheet

Team/Robot MuCAR
Scenario Autonomous navigation

For each of the following aspects, especially concerning the team's approach to scenario-specific challenges, please give a short comment whether they are covered adequately in the SAP.

Keep in mind that this evaluation, albeit anonymized, will be published online; private comments to the organizers should be sent separately.

Robot Hardware

The third generation of our Munich Cognitive Autonomous Robot Car (VW Touareg with a V6 TDI engine).

Processing

A multi-CPU system with dual hexacore Intel Xeon L5640, 12 GByte memory, shock resistant solid-state drives and Linux as operating systemThe core element of the framework is the real-time database KogMo-RTDB. It provides shared memory for data exchange between the applications.

Communication

MuCAR-3 works fully autonomously. They have one operator monitoring the program routines (a communication link not specified) and they have a safety driver.

Localization

OxTS RT3003 (Oxford Technical Solutions) inertial navigation system aided by GNSS (Global Navigation Satellite Systems) satellite receiver (GPS, GLONASS).

Sensing

Velodyne HDL-64E-S2 LiDAR with 64 single laser beams rotating around a common axis with 10 Hz. MarVEye-8 multifocal active/reactive vision system. It consists of three cameras (AVT-Guppy F-036C): Two are mounted on the camera platform, while one is integrated into the hollow shaft of the platform's yaw axis.

Vehicle Control

MuCAR-3 works fully autonomously. Feed-forward and feed-back algorithms to fulfill its goal - follow a lane. The output of the statemachine (lane information) is fed to the underlying vehicle control routine.

System Readiness

In my opintion Hw/sw 9/9.

Eurathlon 2013

Overall Adequacy to Scenario-Specific Challenges

The vehicle seems perfectly adequated to the scenario. From 0-5 the score is 5. Also the certifications reported and the achieved results show its effectiveness. However, presence of a safety driver will most likely result in disqualification.

Autonomous Navigation of Team MuCAR at euRathlon 2013

Sebastian F. X. Bayerl, Dennis Fassbender, Carsten Fries, Benjamin C. Heinrich, Michael Himmelsbach, Hanno Jaspers, Thorsten Luettel, Andre Mueller, Georg R. Mueller, Sebastian Schneider and Hans-Joachim Wuensche

***Abstract*— This paper briefly describes the hard- and software system of Team MuCAR's entry for the euRathlon 2013 robotic trial, participating at the autonomous navigation scenario. We show how the system uses high-level mission planning in combination with a local trajectory planner that incorporates data from a multi layer terrain map for goal-oriented navigation and local obstacle avoidance.**

I. Team MuCAR and the Autonomous Robot Car MuCAR-3

Team MuCAR is headed by Prof. Dr.-Ing. H.-J. Wuensche, chair of "Autonomous Systems Technology" institute at University of the Bundeswehr Munich. We participated in ELROB competitions from 2007 to 2010 and 2012. Further, together with TU Karlsruhe and TU München, we competed as part of Team AnnieWAY in the DARPA Urban Challenge 2007, where we were one of only 11 teams which made it into the finals.

Our robot is named "MuCAR-3" and is the third generation of our Munich Cognitive Autonomous Robot Cars. It is based on a stock VW Touareg with a V6 TDI engine, modified to allow computer control of steering, brake, throttle and automatic gearbox. Full body skid plates allow for testing in rough terrain. Since the hardware is mounted inside the driver's cab or protected with an appropriate housing, the vehicle is able to act in fog, rain or other humidity. A detailed description of the vehicle's hard- and software can be found in [1].

All authors are with department of Aerospace Engineering, Autonomous Systems Technology (TAS), University of the Bundeswehr Munich, Neubiberg, Germany. Contact author email: tas@unibw.de

Fig. 1: Munich Cognitive Autonomous Robot Car 3rd Generation (MuCAR-3) with Velodyne HDL-64E-S2, MarVEye-8 and OxTS RT3003 (based on stock VW Touareg)

A. Sensors

Basic Sensors for e.g. the steering-wheel angle or the wheel speed are available in the stock car and can be used for autonomous driving. The sensing of the vehicle was extended with following three main components:

1) LiDAR: MuCAR-3 is equipped with a Velodyne HDL-64E-S2 LiDAR, which is mounted on the roof of the vehicle. It consists of 64 single laser beams rotating around a common axis with 10 Hz. This commercially available sensor provides a horizontal field of view (FOV) of 360° and a vertical FOV of 26.8°. Its resolution is 0.09° and 0.4°, respectively.

2) Vision system: MuCAR-3 uses MarVEye-8 [2], a multifocal active/reactive vision system, which is mounted between the wind shield and the back mirror of the vehicle. It consists of three cameras (AVT-Guppy F-036C): Two are mounted on the camera platform, while one is integrated

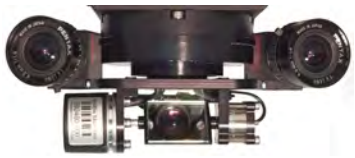

Fig. 2: MarVEye-8, vision system of MuCAR-3, with 2x AVT-Guppy F-036C on platform and 1x AVT-Guppy F-036C mounted into the hollow shaft

into the hollow shaft of the platform's yaw axis (see Figure 2). To allow for pitch stabilization, it perceives the environment through a mirror that is actuated according to the vehicle pitch motion. This composition allows to control the line of sight of all cameras $\pm 50°$ in yaw direction. Additionally the camera mounted inside the hollow shaft can be controlled $\pm 12°$ in pitch direction.

3) Inertial Sensors: Main sensor for localization and estimation of motion is the inertial navigation system OxTS RT3003 (Oxford Technical Solutions). Each three acceleration sensors and gyros allow to estimate vehicle motion in six degrees of freedom. The inertial measurement unit is coupled with a GNSS (Global Navigation Satellite Systems) satellite receiver (GPS, GLONASS).

By combining the information of the inertial navigation system and that of stock sensors like wheel-speed sensors, MuCAR-3 is able to perform robust ego-motion estimation, which compensates for short losses of the satellite link.

B. Hardware architecture

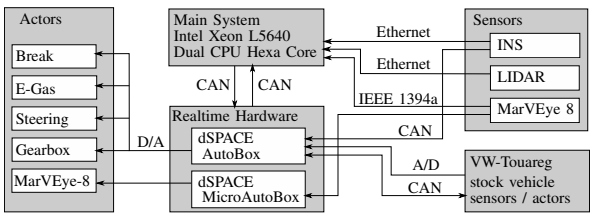

Fig. 3: Hardware Architecture of MuCAR-3

The main computing system of MuCAR-3 is a multi-CPU system with dual hexacore Intel Xeon L5640, 12 GByte memory, shock resistant solid-state drives and Linux as operating system. It has several interfaces for IEEE 1394a (FireWire), CAN, RS-232 and Ethernet to communicate with the sensors. It is connected with two real-time computer systems of type dSPACE AutoBox and MicroAutoBox. These are responsible for controlling the vehicle, the camera platform and all other actors. Additionally they provide access to the stock vehicle sensors. The overall hardware setup is shown in Figure 3.

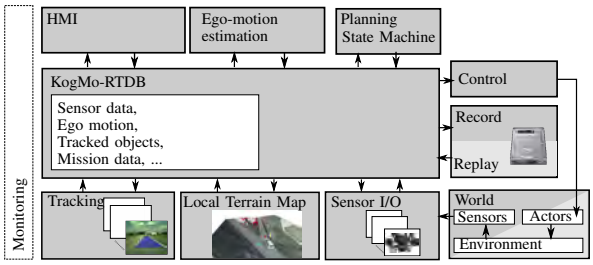

Fig. 4: Software Architecture of MuCAR-3

C. Software architecture

The software architecture consists of several independent modules/applications (see Figure 4). The core element of the framework is the real-time database KogMo-RTDB [3]. It provides shared memory for data exchange between the applications. A typical process cycle can be described as follows: Sensor I/O modules store their data in a RTDB data object comprising data timestamp and sensor values. Applications responsible for e.g. terrain mapping or object detection are able to use these raw information. The results can then be published to the database again. A statemachine in combination with a mission planner is responsible for optimal-state determination. More information about how the statemachine performs can be found in [4].

II. Mission Planning

By utilizing MuCAR-3's HMI (Human Machine Interface) module, the operator can load road network data from files, modify the network and generate mission files, which contain besides the map data other relevant parameters like target velocity or target point. The map data may contain commercially available maps like OpenStreetMap or information from land survey offices. If road data is not available for the car's current

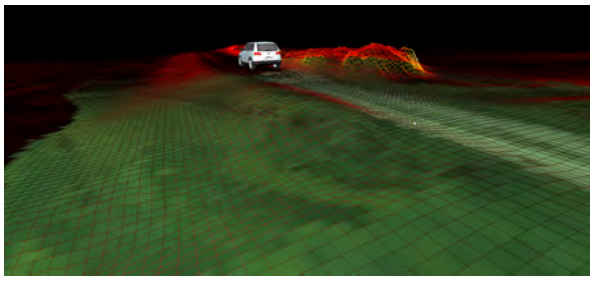

Fig. 5: A terrain map generated on-board MuCAR-3. Metric grid cells contain information about height slope, obstacle probability, color and road probability

environment, it is also possible to create new road networks from scratch.

When the operator selects a GPS position as (one of) MuCAR's new destination(s), the module determines the point on the road network being closest to and stores that point in its mission description. When the mission is started, the HMI module writes its description along with the road network data into the afore-mentioned KogMo-RTDB. The planning module then reads this data and performs a Dijkstra search on the road network to determine an optimal path that passes all GPS waypoints given in the mission description in the correct order. The path depends on its length, the sharpness of its turns, the steepness of its slopes, and its change in slope, amongst other things. The least-cost path is written into the database so it can be used by the control module as well as for visualization in the HMI. Since the GPS road network data may be inaccurate and the vehicle may encounter obstacles on its way, it can only use the planned path as a hint for navigation and needs to rely on its sensors to continually check which areas are actually passable.

III. Perception

A. Precise Local Terrain Maps

The major prerequisite enabling a mobile robot to autonomously navigate in unknown terrain is its ability to perceive the local environment. Creating some sort of environment map using its perceptual abilities thus has become a common task for nearly every robot.

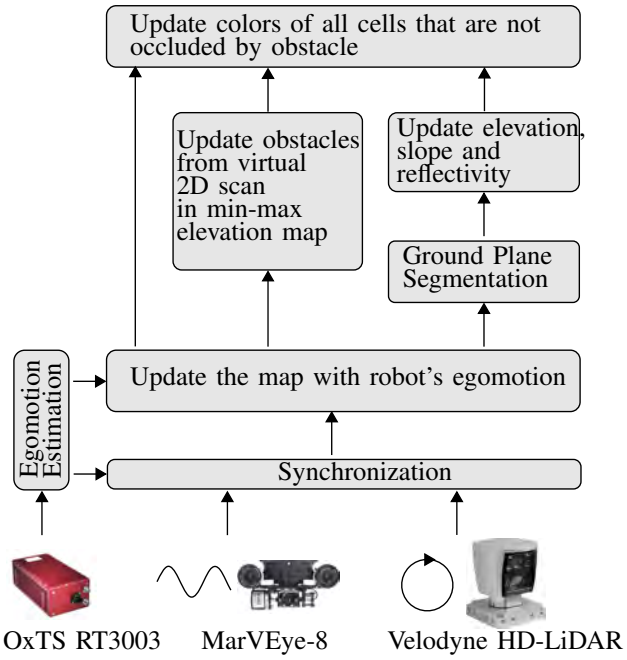

Fig. 6: Overview of the mapping architecture and the sensors involved.

For autonomous vehicles operating in outdoor environments, possibly off-road, the flat-world assumption underlying many mapping techniques is no longer valid. The approach used by MuCAR-3 in this context is the usage of elevation maps, which store the surface of the terrain over a regularly spaced metric grid. Such maps are commonly classified as $2\frac{1}{2}$D models as every cell of the grid only stores one height value and the third dimension is only partially modeled. The high degree of detail is reached by accumulating the data from multiple, complementary sensors in a single map as the vehicle moves. This way, a comprehensive, dense environment representation, including geo-referenced heights, obstacle probabilities, colors, infrared reflectivity and terrain slopes, is obtained.

Maintaining the map is efficient enough to allow building the maps online on-board our autonomous vehicle. This is achieved by an efficient method to manage the map's memory in case the robot moves that does not need to reorganize or copy any data already stored in the map. Given all position, image and depth sensors, the aim of mapping is to produce a dense local representation of the environment, making use of all of the data the

sensors provide. Considering the limited FOV of a (rigid) camera and the limited angular resolution of even the most advanced LiDAR sensors, a dense representation can only be achieved by accumulating data as the robot moves. For maps of limited physical size, this necessitates managing the map's data, removing data that gets out of scope and adding free map space for areas that just entered the FOV of any sensor.

Currently, each cell of the maps we build contains information about obstacles, geospatial height (both from local LiDAR sensing and from publicly available GIS-data, making use of the high-grade GPS sensor on-board), infrared reflectivity, color (from vision) and slope. Due to the different nature of sensor data, we first update the obstacle information and heights, slopes and reflectivity in parallel before updating the colors. This is because we need up-to-date heights to decide which cells are visible to the camera before updating their colors.

B. Tracking road networks

For autonomous driving on road networks the simultaneous detection and tracking of roads and crossings is a fundamental requisite.

MuCAR-3 is configured by a mission tool, which has access to commercially available digital road maps (see section II). This means the vehicle has information about all crossings of the planned track by using its localization abilities. If no road map data is available, a crossing detection algorithm from section III-C can be used as backup. By default, MuCAR-3 tracks the road until a certain distance to a known crossing is reached. Then, a crossroad-tracking algorithm is applied until the crossing is passed.

In order to detect and track roads and crossings, MuCAR-3 utilizes the 4D-approach in combination with a particle filter as shown in [5]. The geometry and dynamics of both objects are modelled and used to project them into the terrain map. Thus a weighting of the projected objects can be carried out with all available sensor data such as color, occupancy, slope and heights of the terrain.

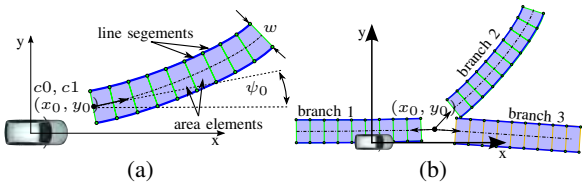

Fig. 7: Model of a road (a) with its line and area segments and a crossing (b) consistiang of three single road segemnts as branches

1) Road model: MuCAR-3 represents a road as a sequence of clothoid segments. Their shape can be characterized by an initial yaw angle ψ_0, an initial curvature c_0, the change in curvature c_1 and the road width w. Furthermore, a starting point x_0, y_0 is established. This model allows us to sample the road segment and build line segments describing left and right road boundaries and to generate area elements which represent the region in between (see Figure 7(a)). The dynamics of the road segment are modeled such that its longitudinal position is fixed to the vehicle, but rotates and moves lateral with inverse motion of the vehicle.

2) Crossing model: A crossing can be seen as a combination of several road segments, which share a common starting point $x_{SP} = [x_0, y_0]^T$. Figure 7(b) shows such a model with three branches. In contrast to the road model, a crossing is modeled as a static object that moves complete with inverse ego motion.

3) Weighting of hypotheses: During the tracking process the estimation has to be compared to the measurement. Since the branches of a crossing can be treated as road segments, roads and crossings show similar characteristics. Each line and area segment is projected into the terrain map and used for weighting the roadsegment or crossing. For example the roadsemgent is expexted to cover a obstacle free region. Similarly, it is assumed that the road boarders show edges in color data. More detail regarding the algorithm can be found in [6] or [7].

A fixed number of the best weighted particles is used to build mean and standard deviation, in order to build a single estimation from the high number of particle hypotheses.

The tracking routine itself provides a self-monitoring, which estimates the quality of the tracking result.

4) Determination of a driveable lane: MuCAR-3 is interfacing its control with clothoids, which can be used during road tracking directly from its tracking result. In [6] it is shown that a seperation of perception and generation of lane to drive yields a smoother handling of the vehicle. Thus, a similar approch to [8] is used. While tracking a crossing or a road, a "point to follow" with a fixed distance to MuCAR-3 is taken from the centerline of the crossing or road. Those points are kept in memory and used by an Extended Kalman Filter (EKF) estimation process to fit a new clothoid into them. The driveability or quality of the clothoid will be taken from the tracking result. MuCAR-3 uses a statemachine, which analyzes amongst other things this value. If the drivebility is good enough, the fitted clothoid is directly used to control the vehicle, otherwise it activates obstacle ovoiding processes shown in section V. They calculate obstacle-free alternatives to reach the desired target.

C. Crossroad Detection without a-priori Information

The crossroad detection algorithm resorts on data from the multi-layer environment representation (see section III-A), especially on information regarding obstacles, height-differences and texture. By exploiting the results of multiple ray-tracing processes that incorporate these map informations, crossroad hypotheses are generated at a distance of up to 20m ahead of the vehicle. Figure 8 shows as example the detection of two crossings. These hypotheses are then tracked over time for further validation, utilizing an EKF that comprises the robot's ego-motion.

IV. Mapping

Environment maps are built up in a topological manner. Instead of mapping the whole environment, our method incorporates navigation-relevant landmarks only. In case of the euRathlon competition these are first and foremost valid crossroad hypotheses. I.e. only landmarks that support the navigation of our autonomous robot and their relative spatial relations are stored in a map. By recognizing a once-driven crossroad or a sequence of crossroads as well as a goal position in the map, robot localization and global navigation is improved.

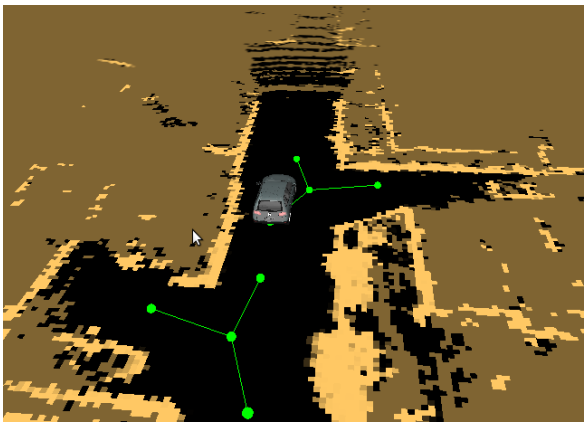

Fig. 8: Two valid crossroad hypotheses (*green structures*) in the vicinity of MuCAR-3 (*grey vehicle model*).

V. Obstacle Avoidance

As mentioned in Section III-A, MuCAR uses LiDAR data to build a local obstacle grid map of size 40 × 40m. Each 15 × 15cm grid cell contains a value between 0.0 and 1.0, reflecting the probability of that cell being occupied by an obstacle.

A. Incorporating obstacle grid data

In order to navigate in the grid, an A*-based algorithm is used to construct drivable paths built from clothoid segments. Apart from the obstacle grid itself, the input to the algorithm includes data such as the vehicle's current pose, velocity and steering angle. It then tries to find an obstacle-free path that will take it reasonably close to a target point on the global path (i.e. the path returned by the lane net planning algorithm) that is 20m ahead of the vehicle. The cost of a candidate path is affected by its distance to obstacles, the curvatures of the clothoid segments from which it is built, and the distance of its end point to the target point on the global path, amongst other things.

Once a path is found, the vehicle keeps following it until it gets (a) too close to an obstacle that had not yet been detected when

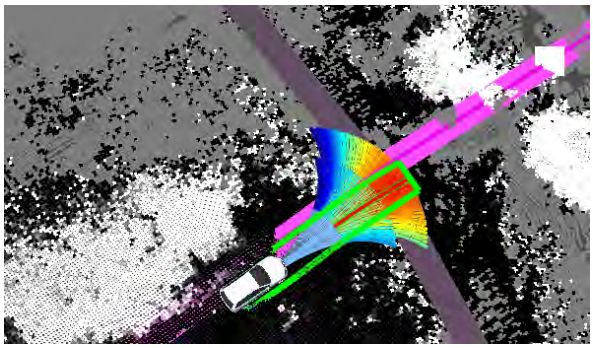

Fig. 9: Obstacle avoidance with tentacles: control clothoid (green), road net (purple), high weighted tentacles (red), low weighted tentacles (blue)

the path was planned, or (b) reaches a certain distance ahead of the end of the path. In either case, the grid planning algorithm performs its operations again to obtain a path that (a) avoids the obstacle, or (b) starts exactly where the old path ends in a way that MuCAR-3 makes a smooth transition from the old path to the new one.

B. Tentacles

Instead of V-A the obstacle avoidance method described in [9] can be used. This method uses a set of "tentacles" that represent precalculated trajectories with different curvatures discretizing the basic driving options of the vehicle. Each tentacle is weighted according to its driveability with use of the terrain map. Also, the road network data of the mission planning module is used for weighting the tentacles. A example is shown in Figure 9.In order to control the vehicle, the best tentacle is selected and used to build a clothoid for interfacing with vehicle control.

VI. Control

The output of the statemachine (lane information) is fed to the underlying vehicle control routine. It then reacts accordingly using both feed-forward and feed-back algorithms to fulfill its goal e.g. follow a lane. Infeasible maneuvers (like excessively sharp turns) are not generated from the higher layers and thus do not happen by design. MuCAR-3 works fully autonomously. However, for safety reasons, we usually have one operator monitoring the program routines and an additional safety driver monitoring the actual physical robot.

Acknowledgment

The authors gratefully acknowledge funding by the Federal Office of Bundeswehr Equipment, Information Technology and In-Service Support (BAAINBw).

References

[1] T. Lüttel, "MuCAR-3: Aufbau und Struktur von Fahrzeughard- und -software," Universität der Bundeswehr München, Neubiberg, Interner Bericht UniBwM / LRT / TAS / IB 2007:3, Sep. 2007.

[2] A. Unterholzner and H.-J. Wuensche, "Hybrid Adaptive Control of a Multi-Focal Vision System," in *Proceedings of IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, Jun. 2010, pp. 534–539.

[3] M. Goebl and G. Färber, "A real-time-capable hard-and software architecture for joint image and knowledge processing in cognitive automobiles," 2007, pp. 734–740.

[4] T. Luettel, M. Himmelsbach, M. Manz, A. Mueller, F. von Hundelshausen, and H.-J. Wuensche, "Combining Multiple Robot Behaviors for Complex Off-Road Missions," in *Proceedings of International IEEE Conference on Intelligent Transportation Systems*, Washington, DC, USA, Oct. 2011, pp. 674–680.

[5] M. Manz, M. Himmelsbach, T. Luettel, and H.-J. Wuensche, "Detection and Tracking of Road Networks in Rural Terrain By Fusing Vision and LIDAR," in *Proceedings of IEEE International Conference on Intelligent Robots and Systems*, San Francisco, CA, USA, Sep. 2011, pp. 4562–4568.

[6] M. Manz, F. von Hundelshausen, and H.-J. Wuensche, "A Hybrid Estimation Approach for Autonomous Dirt Road Following using Multiple Clothoid Segments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Anchorage, AK, USA, May 2010, pp. 2410–2415.

[7] M. Manz, M. Himmelsbach, T. Luettel, and H.-J. Wuensche, "Fusing LIDAR and Vision for Autonomous Dirt Road Following – Incorporating a visual feature into the tentacles approach," in *Proceedings of 21. Fachgespraech Autonome Mobile Systeme (AMS)*, Karlsruhe, Germany, Dec. 2009.

[8] A. Mueller, M. Manz, M. Himmelsbach, and H.-J. Wuensche, "A Model-Based Object Following System," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Xi'an, Shaanxi, China, Jun. 2009.

[9] M. Himmelsbach, T. Luettel, F. Hecker, F. von Hundelshausen, and H.-J. Wuensche, "Autonomous Off-Road Navigation for MuCAR-3 – Improving the Tentacles Approach: Integral Structures for Sensing and Motion," *Kuenstliche Intelligenz*, vol. 25, no. 2, pp. 145–149, 2011, special Issue on Off-Road-Robotics.