accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

Combining Multiple Robot Behaviors for Complex Off-Road Missions

Thorsten Luettel, M. Himmelsbach, M. Manz, A. Mueller, F. v. Hundelshausen and H.-J. Wuensche

***Abstract*— This paper gives an overview of the autonomous off-road navigation approach developed for the ground robot vehicle MuCAR-3. It integrates our approaches to goal-directed and GIS-data supported autonomous navigation, autonomous person and vehicle following and shuttling on a taught track into a single system. The perceptual prerequisites necessary to realize the different robot behaviors are presented in detail, and we show how the individual behaviors available to our robot are coordinated to solve the complex off-road mission of the mule transport scenario at the European Land Robot Trials 2010. Finally, we give an impression of the system's performance by analyzing the results obtained at the trials.**

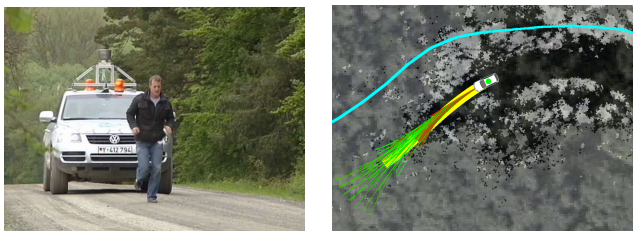
I. Introduction

The last years have shown an immense advancement in autonomous vehicle capabilities. To improve the comfort of driving a vehicle and to make it more efficient and safe, research groups and industry all over the world have worked on building technical systems that can perceive their environment and take appropriate action in order to realize a specific behavior. Examples for such behaviors include lane keeping [1], platooning [2] and obstacle avoidance or autonomous safety stop [3].

Single behaviors are often treated as independent functionality, and attempts to create complex robot behavior by chaining single behaviors are rare. A remarkable attempt to show the possibilities of complex robot behavior was the DARPA Urban Challenge (UC) in 2007, where several robot cars managed to autonomously drive together in urban traffic and obey the local traffic rules. Here, most of the successful teams relied on finite hierarchical state machines to coordinate the interplay of the behaviors available [4]–[6]. In contrast, so-called behavior-based systems like [7] try to merge or blend different behaviors in parallel to form a more complex behavior. The increasing complexity of such systems is hard to maintain, and constraints imposed by control theory are often neglected.

Last year, our robot vehicle MuCAR-3 participated at European Land Robot Trials (ELROB 2010), where it had to deal with a complex mule transport scenario. As in the Urban Challenge, an interplay of different behaviors was required to succeed in this scenario. In the mule scenario, the robot was to autonomously approach a point P_1 (see Fig. 6), follow a person through the woods (see Fig. 1(a)), passing P_2, to another point P_3 in convoy mode and then autonomously shuttle between these points as often as possible within a 60 min time limit. The scenario took place at a military training area characterized by hilly terrain with lots of woods, limiting the availability and quality of GPS signal measurements.

All authors are with the Institute for Autonomous Systems Technology (TAS), University of the Bundeswehr Munich, 85577 Neubiberg, Germany. Contact author email: thorsten.luettel@unibw.de



(a) (b)

Fig. 1. (a) The robot MuCAR-3 is autonomously following a running person in convoy mode during the Teach-In task of ELROB 2010 mule scenario. (b) Navigating with tentacles: Drivable tentacles colored according to the results of evaluation in an occupancy grid (*transparent overlay*), where green tentacles are preferred over red ones. The tentacle selected for motion execution is shown by the yellow clothoid. The large offset between the robot and a GPS target trajectory (*cyan*) is typical for difficult off-road scenarios.

The main differences between the mule scenario and the Urban Challenge thus are:

- Compared to the UC, the environment in the mule scenario was by far less structured. For example, it contained many different kinds of roads of different width and surface conditions, and different kinds of static and dynamic obstacles (other human-driven vehicles, bicycles, pedestrians). In addition, the UC took place under an open sky, thus promoting reliable GPS signals.
- Little information was given to the teams in advance. The only thing the robot was told was the GPS coordinates of the point P_1 it had to approach first. Only the person the vehicle was to follow from thereon knew about P_2 and P_3. This is in contrast to the very detailed route network and mission definition files the DARPA provided to all robots at the UC.
- At the UC, every decision about whether a major shift of behavior was necessary only required to map the robot to the route definition file and look-up the behavior desired for that particular location (*e. g.* at road crossings or parking areas). Hence, no perception other than that of the robot's GPS position was necessary for switching behaviors. In contrast, at the ELROB mule scenario, the decision to shift behavior could not be based solely on GPS measurements. Instead, perception of the immediate surrounding of the robot and its temporal aspects needed to be considered.

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

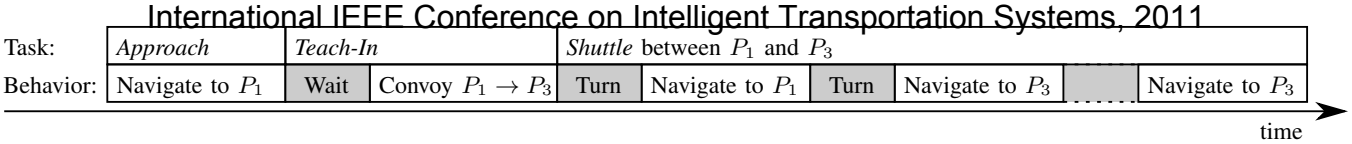


Fig. 2. Exemplary timeline explaining the complete mission of ELROB mule scenario in terms of a sequence of tasks.

In this paper, we describe how we combined a multitude of single basic behaviors to solve the complex off-road challenge posed by the ELROB 2010 mule scenario. Here, the focus is on the perceptual capabilities needed by the robot to both realize its behaviors and to monitor the status of its mission in order to decide which behavior applies best to the current situation.

The paper is organized as follows: Section II introduces the nomenclature of missions, tasks, behavior and conditions and how they are coordinated. Perception mechanisms for the different behaviors are presented in Section III, while Section IV deals with checking of conditions via perception. Finally, experimental results are presented in Section V, followed by a conclusion and a short discussion for future work in Section VI.

II. Coordination of Missions, Tasks and Behaviors

Before giving an overview on how the different tasks are coordinated in our system, we first introduce the nomenclature used in this paper: *missions*, *tasks*, *behavior* and *conditions*, all illustrated in the context of the ELROB mule scenario.

A. Missions

We call a complex sequence of different tasks a mission. Here, the complete mule scenario is one mission, which consists of different tasks. Fig. 2 shows a timeline of the mule scenario to better understand the series of tasks to be performed.

B. Tasks

A task specifies a concrete goal to be reached, utilizing an appropriate behavior while considering constraints, like the types of terrain that are allowed to pass or speed limits.

In the mule scenario, the first task is to *Approach* a target point. Its specification consists of the target point P_1 given in GPS coordinates, a maximum velocity and the designated behavior *Tentacle Navigation* to be explained in Sec. II-C. The robot has to determine a route from his starting point P_0 to target P_1 and is allowed to use map data to determine a target track it thinks suits the task.

The second task, *Teach-In*, requires the robot to wait for a leader person at P_1, follow this person and record the driven track until the person indicates that the task has been completed at an a priori unknown target point P_3. This corresponds to the *Convoy* behavior, following a person and recording the track followed. The *Convoy* behavior will be described in more detail in Sec. II-C.

The next task in the mule scenario is to *Shuttle* between the two points P_1 and P_3. The behavior is the same as in the first task, only that the target track is not generated from map data but from the track recorded in Teach-In. The task then is to loop between behaviors for reaching target point P_1 and reaching P_3.

Other possible tasks which are not involved in the mule scenario are vision-based *Road Following* [8], GPS-based waypoint following, or *Convoy Driving* with a leading vehicle [9], [10].

C. Behaviors

Behaviors describe long-term actions the robot can perform, e.g. follow an object or a lane. Each behavior depends on different perception modules, and each provides a clothoidal segment as output which is passed to the vehicle control modules at each cycle. In the mule scenario, different kinds of behaviors can be involved in each task, working together to make the robot fulfill its current task.

A first behavior is the *Tentacle Navigation* behavior that the robot uses for autonomous navigation in unstructured environments. While roughly following a target track leading to a given target point P_T, obstacle avoidance and generation of smooth vehicle movements are achieved by fusing information from LIDAR and vision in this behavior. The target track is generated either from GIS map information (used in the *Approach* task) or using a previously recorded track (used in the *Shuttle* task). The perception for *Tentacle Navigation* behavior is detailed in Sec. III-A.

Convoy is another often used and important behavior. It enables the robot to follow another vehicle or a person precisely on its track. Perception for object tracking and track generation is detailed in Sec. III-B. For smooth longitudinal control, the distance to the leader object taken along its track is incorporated. Such a convoy behavior is quite state of the art for forward driving. In our system, we extended the convoy behavior to also allow convoy backward driving when being pushed backwards by the leader. This is for instance used if the convoy leader took a wrong way or accidently reached a dead end.

The *Turn* behavior enables the robot to change its driving direction even in narrow space. For lateral control, a clothoid maximally curved with respect to the robot's kinematic constraints is generated. The steering angle is adjusted while standing to enhance maneuverability. Driving distance is limited on the one hand by rotating a maximum of $90°$ in one direction and on the other hand by checking the distance to the first obstacle along the turning curve by probing an obstacle map built from LIDAR data. Forward and backward turns are then executed in a loop until the desired direction is reached. Thus, in an environment totally free of obstacles, a $180°$ turn would be performed as a three-point-turn.

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

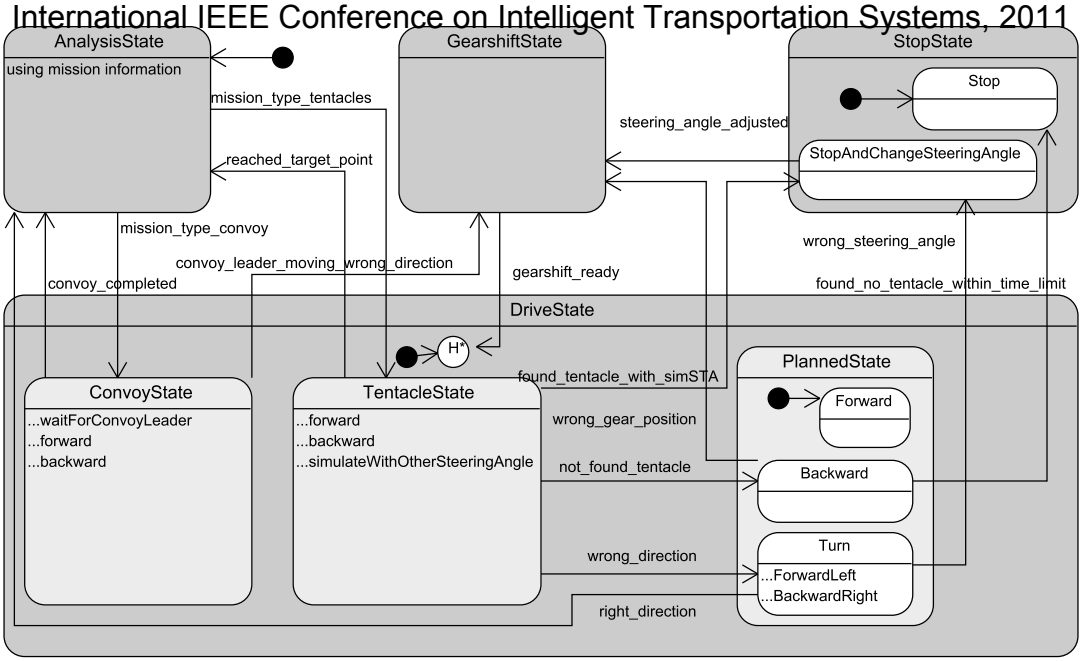


Fig. 3. UML diagram of the hierarchical state machine, reduced to show only the components used within the mule scenario

If there are no options to go for a forward driving behavior, a general *Back-Up* behavior gets triggered. By backing up a few meters, the robot usually will be able to identify new options for driving forwards. As in the turn behavior, obstacle detection while backing up is performed with the help of the obstacle map mentioned earlier.

The most simple behavior is *Stopping*, and it is only utilized if all perception mechanisms fail to provide any driving option, no matter what direction.

Additionally, all moving objects detected on the road are considered for longitudinal control, making all forward driving behaviors automatically behave in a way known from adaptive cruise control systems. Object detection and tracking is further described in Sec. III-B.1. There are more behaviors within our framework, but not all of these are used in the mule scenario. These include *Road Following* on marked urban roads as well as dirt roads. More information on perception for *Road Following* is given in [8].

D. Conditions

Conditions are used for switching between different tasks and behaviors. For checking these conditions different kinds of sensors like GPS, vision or LIDAR need to be considered.

Possible conditions for switching to the next task are reaching a GPS coordinate (in *Approach* and *Shuttle* task) or detecting that the convoy leader has reached its target (in *Teach-In* task). The latter is an example for a condition that cannot be directly measured, but must be inferred by observing the leader's long-term behavior, like the leader not moving for a longer time. Details on the perception mechanisms needed to check such kind of involved conditions are given in Sec. IV.

Conditions can also cause a change in active behavior within one task. An example is a road blockade, causing the robot to back-up or to just stop. By evaluation of the robot's orientation towards the target track, conditions like *Wrong Direction* can initiate a turn behavior, which would be active until the corresponding condition *Right Direction* was verified.

E. Coordination of Tasks and Behaviors

For coordination of tasks and involved behaviors, a hierarchical state machine (HSM) is used in our system, as in many of the successful teams participating in the Urban Challenge [4]–[6]. Fig. 3 gives an overview of the state machine used in our system, showing only the states and conditions involved in the mule scenario for clarity.

Here, *AnalysisState* has access to the sequence of tasks as part of the mission description, *i. e.* the sequence of incorporated tasks like *Approach* or *Convoy* in this case. It reacts on task-changing conditions emitted from the other states to toggle the required next task in sequence.

Main functionality is concentrated in *DriveState*, where the different behaviors like *Tentacle Navigation* or *Turn* are contained as sub-states. Consequently, most behavior-changing conditions are checked here.

Additionally, there are some helper states like *Gearshift-State*, changing the current gear position according to desired driving direction, or *StopState*. On the one hand, *StopState* can stop the vehicle on a clothoid within a given distance. On the other hand, changing the steering angle of the stationary robot is also performed in this state. Clothoids, our standard interface to the lateral control module, cannot be used for changing the steering angle of the stationary robot as this would require a discontinuity in the clothoid's curvature.

III. Perception necessary to realize Behavior

In this section the perceptual capabilities are presented which are utilized to realize the different behaviors mentioned above. They are based on different sensory input like LIDAR, vision images or GPS measurements. First, we present *Tentacle Navigation* and then detail different perception methods for *vehicle and person following*. Afterwards, we conclude with GPS-based localization, mostly used as a supplement to perception only.

A. Tentacles Approach

Forest environments like the one encountered at ELROB mule scenario produce challenging GPS conditions with major disturbances and outages. To achieve a robust navigation system under these circumstances, the significance of GPS information needs to be lowered in favor of a stronger focus on the reactive parts of navigation, without neglecting the difficulties in choosing the right of possibly many driving options at crossings [11].

These issues are addressed by the tentacles approach to robot navigation [12]. Its basic ingredients are integral structures for sensing and motion, termed tentacles. These are a set of possible driving options of predefined clothoidal shape that probe an ego-centered occupancy grid constructed from LIDAR data for drivability and other ground surface characteristics. Commanding an action then corresponds to selecting one of the feasible drivable tentacles for motion execution. Fig. 1(b) shows a snapshot of the robot navigating in a forest environment using the tentacles approach.

Most recent details on the tentacle approach can be found in [13]. Here, we only give a short overview of the main concepts involved. Most important, a tentacle is only drivable if it is feasible and the robot would not hit an obstacle when executing the tentacle, taking into account the distance needed for the robot to stop at its current velocity. Obstacles are looked up in the LIDAR occupancy grid, which accumulates point cloud data as the robot moves to get a dense representation of the environment. For every drivable tentacle, we then combine a multitude of properties in a linearly weighted sum. These properties include ground flatness, distance to obstacle and distance to a GPS target track, adding a goal-leading behavior to the otherwise reactive nature of the tentacles approach.

A last property is derived by projecting all drivable tentacles into the colored camera images, evaluating if the surface beneath a tentacle "looks" like a road. To understand the benefits of fusing camera information into the tentacles approach imagine the terrain around the vehicle to be equally flat. Then, we cannot tell which tentacle to prefer over others, resulting in a shaky behavior of robot motion, as theres no preferred direction to go. With fusion of LIDAR and vision information the robot thus is able to follow a curved road in an otherwise equally flat terrain.

B. Vehicle and Person Following

In order to further increase the mobility of our autonomous robot MuCAR-3 we have implemented an object following behavior into the system. In our architecture the subdivision of object detection and tracking on the one hand and track generation on the other hand is important to the generality of the convoy behavior. While object detection and tracking can be done with any sensor, generating a track from the estimated relative object positions can be done independently of the sensor used. The generated track together with the estimated object position and velocity is used for lateral and longitudinal control units.

Before we explain track generation, we detail the tracking by both LIDAR and monocular camera.

1) LIDAR-based Object Tracking: The first step for object tracking in 3D LIDAR point cloud data is segmentation of the data into small subsets representing possible objects. Especially in off-road scenarios, it is important that the segmentation method can also cope with sloped terrain. Also, time resources for segmentation are limited as the actual estimation process is yet to follow. The method presented in [14] provides these properties, allowing real-time segmentation of large point clouds also in difficult non-flat environments. A result of segmentation is shown in Fig. 4(a).

Given a segmented point cloud, a generic bounding box model is then fit to each segment. The oriented bounding box is a reasonable abstraction for most kinds of object geometries. Thus, by using the object bounding boxes as measurements in a multi-target Kalman-Filter-based tracker, no prior knowledge of the geometry of an object to track is needed. However, for this tracker to be used in a convoy scenario, a stronger commitment to a certain object, *i. e.* the convoy leader, is required. This is achieved by applying the classification framework presented in [15] to each point cloud segment before tracking. Fig. 4(b) shows a snaphot of LIDAR-based object tracking in a convoy scenario.

2) Visual Vehicle Tracking: Our approach to model-based monocular vehicle tracking with a single camera mounted on a moving vehicle was presented in [10]. The algorithm can cope with cluttered color images, complex lighting conditions as well as partial occlusion of the leading vehicle, and it is able to detect and track a vehicle even within unstructured off-road environments (see Fig. 4(c)). Thanks to the vehicle-specific 3D model used, which describes the characteristic geometry and appearance of the vehicle in terms of vertices, edges and colored surfaces, no special visual markers need to be attached to the vehicle. The knowledge of the vehicle's geometry and appearance gained from the 3D model are used within a particle filter framework to estimate the 6-DOF position relative to the ego vehicle, fusing edge as well as color information.

3) Fused Vehicle Tracking: The purely vision-based approach [10] has been extended to incorporate measurements from a LIDAR-generated occupancy grid. With this, a higher stability especially under conditions typically known to cause problems for visual perception (*e. g.* dark woods), could be achieved. In addition, as our Velodyne LIDAR sensor provides a 360 degrees horizontal field of view, the vehicle can still be tracked even if it is not visible in the camera image in the very sharp turns encountered at ELROB 2010

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011

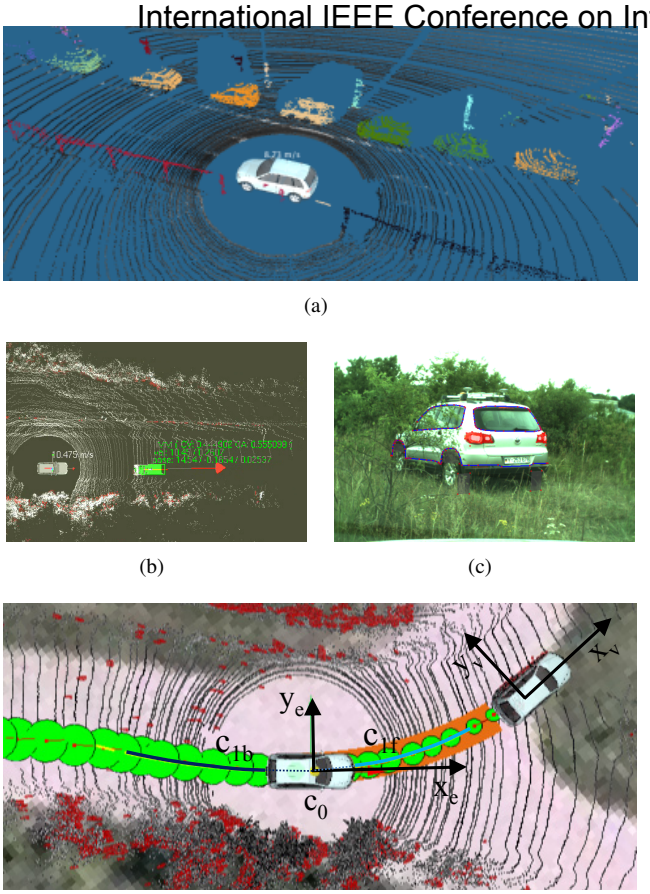


Fig. 4. Following: (a) LIDAR point cloud segmentation (b) Tracking an arbitrary bounding box in a LIDAR-based occupancy grid (c) Vision-based 3D vehicle tracking (d) Trajectory generation using two clothoid segments: light blue for forward driving, dark blue for backward driving

convoy scenario.

Summarizing, object perception in our system is able to detect and track arbitrary objects using a LIDAR-based approach as well as special objects with a known 3D model using vision. In a 12 minute test run we compared pure visual-based vehicle tracking to RTK-DGPS ground truth. Distance has a moderate root mean square error (RMSE) of $\approx 0.41\,\mathrm{m}$, the RMSE of the direction estimate is as small as $\approx 0.52°$. More details can be found in [10]. Comparing LIDAR-based vehicle tracking to ground truth, we showed a position error way less than $\approx 0.5\,\mathrm{m}$ in [15].

In case both sensors are available, the two sensors are fused, achieving a more robust tracking performance. Some results of fused tracking were gained at ELROB 2010 convoy scenario in a track of approx. $6350\,\mathrm{m}$ through the woods and a muddy driving-school parcours for trucks and tanks. A video providing more details can be found at `http://www.youtube.com/watch?v=aWzQdgsWrWE`.

Unlike most other approaches we do not use any kind of special markers placed upon the tracked object, no matter if it is a person or a vehicle or something else.

4) Track Generation: Being able to track an object does not mean to be able to autonomously follow the object. For the robot to do so, a drivable track is needed.

Given the known object position estimates, successive static way-points are generated. All way-points and their uncertainty are predicted with respect to egomotion to generate a chain of static way-points expressed in the current coordinate system of the robot (green circles in Fig. 4(d)). Together, these way-points represent the rough trajectory of the leading object. In order to get smooth feedback values for the lateral control unit, we utilize a sequential UDU-factorized Extended Kalman Filter. The filter estimates the relative position, curvature c_0 and change of curvature c_{1f}, c_{1b} of two local clothoid segments for forward and backward driving (see light and dark blue lines in Fig. 4(d)). For more details we refer to [8], [9].

C. Map Tools for GPS-based Localization

To enable the robot to follow an instruction like "go to GPS coordinate P_T", we developed MapTools, a GPS-based module performing localization on global map data. Mostly, our map information is taken from commercially available GIS data provided by the Land Survey Administration, consisting of a road network and some area information like residential areas, woods or farmland.

The shortest path in global coordinates is computed using the A^* algorithm [16], which is further smoothed and approximated by a cubic spline fit, in a way similar to [17]. On this path the robot is periodically localized, and the target trajectory is transformed into ego coordinates, mostly for supplement of the other perception modules.

Additionally, a clothoid assured the robot can follow is generated that is used in case no other track can be perceived by other sensors. Note that LIDAR obstacle avoidance will still prevent the robot from driving into obstacles, and high precision map data as well as good GPS conditions are assumed. That's why purely map- and GPS-based driving is not possible in the ELROB mule scenario (see Fig. 1(b)), and the robot has to rely on other perception.

IV. Checking Conditions via Perception and GPS

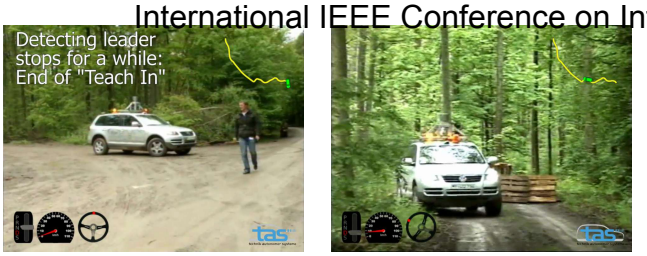
Perception is not only needed to enable particular behaviors, but also for checking conditions.

The most simple condition to be checked is if a given GPS coordinate is reached. For *Distance Condition* to become true the Euclidean distance between current robot position P_e and given target coordinate P_T is checked to be within a given limit.

To check if the robot is going the *Wrong Direction*, information from MapTools is analyzed to determine the robot's angle $\Psi_{e,T}$ to the target trajectory. For $\Psi_{e,T} \in 180° \pm 30°$ the condition will be true and the need for a turn will be indicated. Similarly, for the robot to go the *Right Direction* at the end of a turn, $\Psi_{e,T} \in 0° \pm 5°$ must hold.

To check if a *Teach-In is completed*, the history of the leader object's velocity estimates, reported by LIDAR-based object tracking or visual vehicle tracking, is searched for a longer period of inactivity.

accepted for publication at
International IEEE Conference on Intelligent Transportation Systems, 2011



(a) (b)

Fig. 5. Scenes during ELROB 2010 mule scenario: (a) The robot MuCAR-3 detects the *Teach-In* to be completed by using its perception to check leading person's behavior to fulfill a given condition. (b) Navigating with tentacles during *Shuttle*: this road barrier was placed on the track after *Teach-In* to evaluate the robot's reaction to changes of environment after being taught the track. MuCAR-3 detected this blockade and found another way to pass this obstacle.

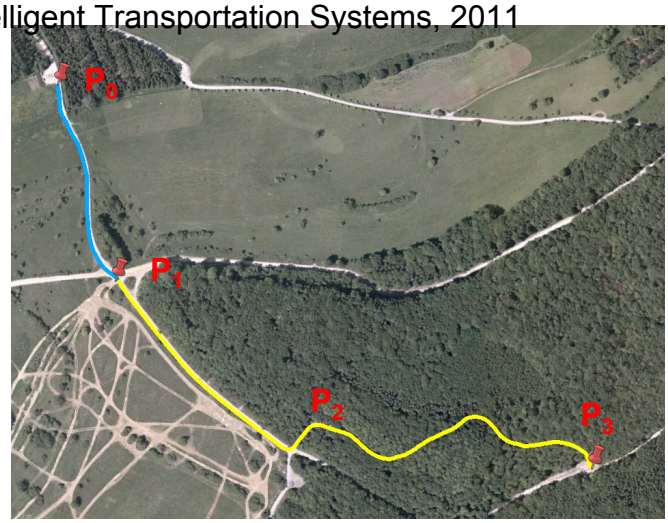


Fig. 6. Aerial view of ELROB 2010 mule scenario. Blue: *Approach* to target point P_1, Yellow: *Teach-In* by person following to a priori unknown P_3 and then *Shuttle* between P_1 and P_3

There are two different types of *Blockade* conditions. In tentacle navigation, the condition for a *Blockade* will be true only if the number of drivable tentacles equals zero. A second *Blockade* condition can occur during turn or backing-up maneuvers, where the clothoid segment used for backing-up is checked for obstacles with the LIDAR grid. If the distance to the first obstacle is below a fixed minimum distance, the *Blockade* condition will also be true for the back-up maneuver.

V. Experimental Results

A. System Overview

The experimental results during ELROB 2010 mule scenario were achieved using our robot vehicle MuCAR-3 (Munich Cognitive Autonomous Robot Car 3rd Generation), a VW Touareg with full drive-by-wire modifications and appropriate sensors to allow for autonomous driving.

Our first main sensor in this scenario is the Velodyne LIDAR HD64-SE2, a high-resolution laser range scanner mounted at the vehicle's roof. For visual perception, we use our camera platform MarVEye 8 (Multi-focal active / reactive Vehicle Eye), which is build up similar to the human eye, with two wide angle cameras and a high resolution tele camera, inertially stabilized by an additional gyro moving the mirror. It is equipped with Firewire color cameras. The image recording of the three cameras is synchronized by a hardware trigger signal, which is synchronized to the LIDAR measurements and the current gaze direction [18]. Additionally, we use measurements obtained from the vehicle's CAN bus like odometry as well as position and motion estimates of an OxTS RT3003 Inertial Navigation system.

Sensor data processing and all other high-level work is performed on a multi processor system (Intel Xeon L5640 Dual CPU Hexa Core) on board of MuCAR-3. Two hard real-time capable dSPACE computers run the low-level controllers and access the vehicle and camera platform I/O.

B. Results

We successfully demonstrated the new combination of multiple robot behaviors in ELROB 2010 to cope with the complex mission of mule scenario within a 60 min time limit. Because of the safety driver requirements of our vehicle, MuCAR-3 participated non-competitive only.

MuCAR-3 executed the first task *Approach* fully autonomously within 3:40 min (327 m). Map data was incorporated to determine the shortest way from starting location to target point P_1 and for generation of a rough target track for Tentacle navigation. Reaching P_1, the robot realized this task to be completed.

There, the robot successfully waited for the leading person and initialized the LIDAR-based perception module for person tracking. Once detected, it followed this person fully autonomously on a primarily unknown track to P_3, utilizing our well-proven convoy for track generation. A hilly road, turning left at a crossing P_2 and along a curved track through the woods led to an open space after approx. 850 m. Here, the perception module that supervised the leader's behavior indicated that the *Teach-In* was completed, as shown in Fig. 5(a).

The track recorded during *Teach-In* was utilized as target track for driving back to P_1. First, a three-point-turn was performed by the robot to get into the right direction, then Tentacle navigation was used for driving back through the woods. On the first passage, the organization team placed a mobile road barrier onto the path, causing the robot to maneuver around it (see Fig. 5(b)). Utilizing its backing-up behavior to get new potential driving options, this was no problem for MuCAR-3. Reaching the loading point P_1, the *Shuttle* task selected P_3 as new target, so again a three-point-turn was required to adjust the robot's direction. This shuttle behavior with alternating targets was repeated for two complete loops, reaching the time limit after performing a last turn at P_3.

For Teach-In and Shuttle tasks, 97.1% of time and 99.3% of distance (48:25 min, 4637 m) were driven fully autonomously, including four three-point-turns at P_1 and P_3. All other robots participating accomplished much less distance.

At some very narrow passages the robot had to back-up to find a clear path around obstacles like the road barrier mentioned before or some branches of a bush reaching into the designated driving corridor. These flexible obstacles still pose a common problem in environment perception: it is very hard to differentiate between a soft branch of a bush and a solid metal pole of similar dimension. High grass can cause similar problems. In the mule scenario, our robot could cope with soft obstacles by using its capability for backing-up.

Unfortunately, a few interventions were necessary during the mule scenario: The second turn to perform at P_3 failed, because the area around P_3 was occupied by other vehicles driving around. The point could not be approached close enough for the turn condition to become true. The other problem occurred at a crossing in the middle of the track where the vehicle had to take a sharp turn into one direction. The place was not modeled as a crossing in our taught-in track but as an ordinary curve, thus automatically raising the influence of the target track on tentacle selection at crossings, as it had proven beneficial at ELROB 2009 [11], was not done. This caused the tentacle algorithm to weight the straight tentacles better than turning ones, and manual interventions were needed.

Visual impressions of the trials and more detailed results are shown in videos online at `http://www.unibw.de/lrt8/medien/audio_video` and `http://www.youtube.com/watch?v=uBZRA6wf2Qc` and also available on conference CD/DVD.

VI. Conclusions and Future Work

In this paper we showed how to manage complex autonomous off-road missions by combining multiple robot behaviors. Our system can incorporate different kinds of behaviors to deal with special tasks. These behaviors can be very different, *e. g.* convoy driving with a vehicle or person as leading object to an a priori unknown target location (*Teach-In*), or navigation in unstructured environments to a given target point incorporating map data or taught-in tracks for generation of an rough target track (*Approach*, *Shuttle*). A variety of perceptual capabilities important for both enabling behaviors and checking conditions for behavior transition was presented. The good results obtained at the ELROB 2010 mule scenario justify the design of the system.

Some aspects should be considered for future work. For a better handling of crossings, perception should be coupled more tightly to a priori map knowledge, *e. g.* perform active gaze control when approaching a crossing to have a look at all of its branches instead of measuring only the one branch that is accidentally in view. This will make it less difficult to detect and track the crossing via visual perception. Another idea to improve crossing behavior is a LIDAR-based crossing detection mechanism already in development at our institute.

Another aspect for future work is to enable perception to discriminate between nonhazardous vegetation and dangerous solid obstacles.

VII. Acknowledgements

The authors gratefully acknowledge funding by the Federal Office of Defense Technology and Procurement (BWB).

References

[1] E. D. Dickmanns and A. Zapp, "A curvature-based scheme for improved road vehicle guidance by computer vision," *Proceedings of the SPIE Conference on Mobile Robots*, pp. 161–168, 1986.

[2] A. Broggi, M. Bertozzi, A. Facioli, C. G. L. Bianco, and A. Piazzi, "Visual Perception of Obstacles and Vehicles for Platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 164–176, 2000.

[3] B. Ulmer, "VITA – an autonomous road vehicle (ARV) for collision avoidance in traffic," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, Detroit, MI, USA, 1992, pp. 36–41.

[4] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, *et al.*, "Team AnnieWAY's autonomous system for the DARPA Urban Challenge 2007," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.

[5] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, *et al.*, "Autonomous Driving In Urban Environments: Boss and the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 1, pp. 425–466, June 2008.

[6] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, *et al.*, "Junior: The Stanford Entry in the Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[7] C. Armbrust, M. Proetzsch, B.-H. Schäfer, and K. Berns, "A Behaviour-based Integration of Fully Autonomous, Semi-autonomous and Tele-operated Control Modes for an Off-road Robot," in *Proceedings of the 2nd IFAC Symposium on Telematics Applications*, Timisoara, Romania, Oct. 2010, invited paper.

[8] 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.

[9] 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, June 2009.

[10] M. Manz, T. Luettel, F. von Hundelshausen, and H.-J. Wuensche, "Monocular Model-Based 3D Vehicle Tracking for Autonomous Vehicles in Unstructured Environment," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 2011.

[11] T. Luettel, M. Himmelsbach, F. von Hundelshausen, M. Manz, A. Mueller, and H.-J. Wuensche., "Autonomous Offroad Navigation Under Poor GPS Conditions," in *Proceedings of 3rd Workshop On Planning, Perception and Navigation for Intelligent Vehicles (PPNIV), IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, Oct. 2009.

[12] F. von Hundelshausen, M. Himmelsbach, F. Hecker, A. Mueller, and H.-J. Wuensche, "Driving with tentacles: Integral structures for sensing and motion," *Journal of Field Robotics*, vol. 25, no. 9, pp. 640–673, 2008.

[13] 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.

[14] M. Himmelsbach, F. von Hundelshausen, and H.-J. Wuensche, "Fast Segmentation of 3D Point Clouds for Ground Vehicles," in *Proceedings of IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, June 2010.

[15] M. Himmelsbach, T. Luettel, and H.-J. Wuensche, "Real-Time Object Classification in 3D Point Clouds Using Point Feature Histograms," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, MO, USA, October 2009.

[16] N. J. Nilsson, *Principles of Artificial Intelligence*. Springer, 1982.

[17] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, *et al.*, "Stanley: The robot that won the DARPA Grand Challenge: Research Articles," *Journal of Robotic Systems*, vol. 23, no. 9, pp. 661–692, 2006.

[18] S. Schneider, M. Himmelsbach, T. Luettel, and H.-J. Wuensche, "Fusing Vision and LIDAR – Synchronization, Correction and Occlusion Reasoning," in *Proceedings of IEEE Intelligent Vehicles Symposium*, San Diego, CA, USA, June 2010.