PANDORA Monstertruck: A 4WS4WD Car-Like Robot for Autonomous Exploration in Unknown Environments

George Kouros
Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki, Greece
Email: gkourosg@yahoo.gr

Loukas Petrou
Department of Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki, Greece
Email: loukas@eng.auth.gr

***Abstract*—This paper presents a four-wheel-steering four-wheel-drive (4WS4WD) car-like robot, able to autonomously explore unknown environments with flat or uneven terrain. The robot uses a set of sensors and actuators and employs a number of algorithms to effectively map the environment, localize in it, plan and execute motions while utilizing the maneuverability of the four-wheel-steering kinematic model. For the task of autonomous navigation, a ROS-based 4-Tier architecture is proposed consisting of global path planning, dynamic local path deformation for collision avoidance, kinematically feasible local path synthesis and finally path tracking control using a fuzzy logic controller for 4WS car-like robots.**

I. Introduction

Autonomous Mobile Robots (AMR) are robots able to perform tasks without any human intervention or control. AMRs are increasingly used in commercial, industrial, military and lately even domestic settings and are extensively researched worldwide. The AMR presented in this paper, was specifically developed for research purposes in the field of Unmanned Search and Rescue (USAR).

An AMR must be able to perceive its environment, move safely and efficiently inside it and make decisions based on sensory input and artificial intelligence algorithms about how to proceed in order to complete a given task. Therefore, it utilizes a set of algorithms that can be separated into three basic categories, which are perception, decision making and motion planning & execution. In this particular case, those categories translate to simultaneous localization and mapping (SLAM), exploration (target selection) and autonomous navigation respectively.

This paper presents the architecture of an AMR and the algorithms required for its autonomous operation, focusing on motion planning and execution, while the problems of simultaneous localization and mapping (SLAM) and exploration are tackled, by utilizing a number of open source algorithm implementations. In particular, CRSM-SLAM [1] is used to construct a 2D map of the environment and localize in it, while exploration is achieved using a frontier exploration algorithm, which selects a frontier to explore based on an objective function that takes into account the distance and orientation difference to each frontier, as well as its size and reachability.

The developed robotic platform adheres to a four-wheel-steering kinematic model, which presents a set of nonholonomic constraints that restrict the feasible velocities of the robot in the plane and as a result render the problem of motion planning especially difficult in comparison with the more popular Differential, Skid-Steer and Omnidirectional kinematic models. As a result, the open source implementations of navigation algorithms using Dijkstra and A* graph search for path planning and the Dynamic Window Approach [2] or Elastic Band [3] for collision avoidance and path tracking, should be modified. The employed navigation system architecture follows the same structure as the aforementioned one, but features a collision avoidance and path tracking component that considers the kinematic constraints of the 4WS robot.

The proposed Reeds-Shepp Band method was based on the Nonholonomic Bubble Band by Khatib et al. [4], which dynamically deforms a global path, while satisfying the nonholonomic constraints of car-like robots. The proposed method uses a two-step approach to dynamically deform the global path for collision avoidance using the Elastic Band algorithm and then perform interpolation on the deformed path waypoints using collsion-free Reeds-Shepp paths [5] in order to achieve the kinematic feasibility of the path. The navigation system is complemented with a fuzzy path tracking controller, based on the methods presented in [6] for car-like robots and [7] for 4WS robots and aims to take advantage of both the counter steering and crab steering capabilities of the robot.

In Section II, the developed PANDORA Monstertruck robotic platform is presented. Section III focuses on the software architecture of the robot and the proposed autonomous navigation system. Section IV demonstrates the experimental results from simulation and physical robot operation, followed by the conclusions and future work in Section V.

II. The PANDORA Monstertruck Robot

The PANDORA Monstertruck robotic platform that is depicted in Fig. 1 was built using a Redcat Racing Groundpounder 1:10 RC monstertruck, that features simultaneous and independent front and rear wheel steering (4WS), as well as drive transfer to all four wheels (4WD).

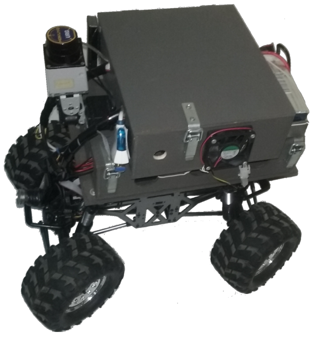
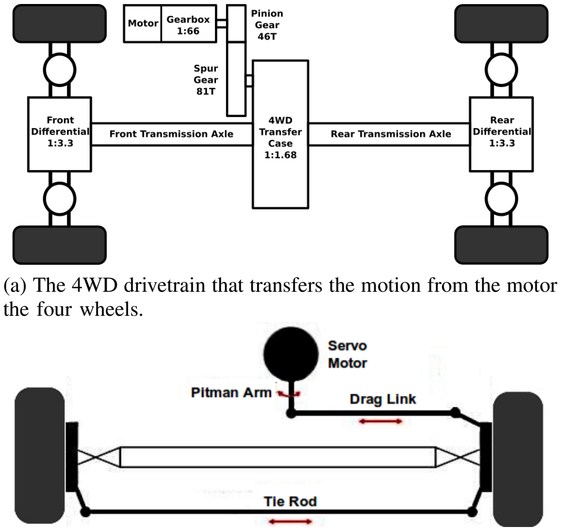


Fig. 1. PANDORA Monstertruck.

A. Actuators and Motion Transfer

1) Drive System: The driving motion of the wheels is generated by a single brushless $18000\,rpm$ maxon EC-max 283858 motor, controlled by a maxon EPOS 24/1 positioning controller. The turning motion of the motor is transferred to the wheels via a drivetrain mechanism, as shown in Fig. 2a, which consists of four basic transfer stages: the motor gearbox (1:66), the spur & pinion gears (1:1.76), the 4WD transfer case (1:1.68) and the front and rear differentials (1:3.3), resulting in a 1:644 transfer ratio and a maximum speed of $0.2m/s$.

2) Steering System: The steering motion of the wheels is generated by two Hitek HS-7954TH servo motors, controlled by a Pololu Micro Maestro 6-Channel USB servo controller. The servos steer the front and rear wheels via two drag link steering mechanisms, as depicted in Fig. 2b.



(a) The 4WD drivetrain that transfers the motion from the motor the four wheels.

(b) The drag-link-steering mechanism used for the 4WS system.

Fig. 2. The motion transfer systems of PANDORA Monstertruck.

B. Hardware Components

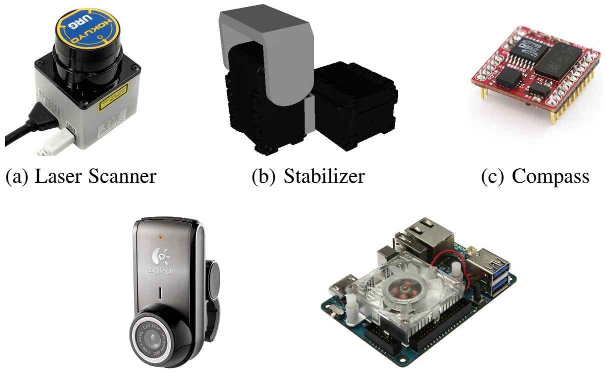
The main goal of the developed robotic platform is to autonomously explore unknown environments with flat or uneven terrain. In particular, the robot must be able to map its environment and detect obstacles. These tasks are accomplished using a Hokuyo URG-04LX Laser Scanner (Fig. 3a) that offers laser scans, with a 10Hz frequency, $60-4095mm$ range and $240°$ measurement area.

Reliable localization, mapping and obstacle detection demands the laser scanner to always be stabilized on the horizontal level. This is accomplished by employing a roll-pitch stabilization mechanism consisting of two Dynamixel AX-12A smart servos in a roll-pitch topology (Fig. 3b) that continuously correct the roll and pitch deviations of the sensor.

The laser scanner stabilization task requires the knowledge of the pitch and roll deviation angles in order to adjust the mechanism rotations using the inverse angles. The roll and pitch angles are measured using a compass and specifically an Ocean Server Compass OS4000 (Fig. 3c), which contains a 3-axis magnetometer and a 3-axis accelerometer and offers measurements of the $roll$, $pitch$ and yaw angles, as well as the linear accelerations $\alpha_x, \alpha_y, \alpha_z$.

In addition, the sensor equipment of the robot is complemented with a Logitech Portable Webcam C905 (Fig. 3d), used at the time, only for monitoring purposes during the teleoperated and autonomous operation of the robot.

Finally, the central node that acts as the brain of the robot and communicates with the sensors and actuators, consists of an ODROID-XU4 single board computer (Fig. 3e) that was chosen due to its small size, low power consumption and adequate processing power.



(a) Laser Scanner (b) Stabilizer (c) Compass

(d) Webcam (e) ODROID-XU4

Fig. 3. The sensors, laser scanner stabilizer and computer board of PANDORA Monstertruck.

C. Kinematics

In [8] the 4WS kinematic model is presented with the constraint that in order for a 4WS vehicle to perform slip-free motion in low speeds, it must comply with the 4WS Kinematic Condition. This condition states that the perpendicular lines to each wheel must intersect at the same point O, called the Instantaneous Center of Rotation (ICR) and is expressed as

$$\frac{1}{\cot\delta_{of}-\cot\delta_{if}}+\frac{1}{\cot\delta_{or}-\cot\delta_{ir}}=\frac{l}{w} \tag{1}$$

where δ: steering angle, i: inner, o: outer, f: front and r: rear, l: wheelbase and w: track.

However, the developed robotic platform uses steering mechanisms that do not comply with the above kinematic condition and define a steering relation between the left and right wheels that is approximated as equal ($\delta_{if}=\delta_{of}=\delta_f$, $\delta_{ir}=\delta_{or}=\delta_r$). As a result, the wheels are subjected to slipping by their corresponding side-slip angle α ($\alpha_{if},\alpha_{of},\alpha_{ir},\alpha_{or}$), even in low speeds. The slipping of the wheels is demonstrated in Fig. 4 and is expressed as

$$\frac{1}{\cot(\delta_{of}-\alpha_{of})-\cot(\delta_{if}+\alpha_{if})}-\frac{1}{\cot(\delta_{or}-\alpha_{or})-\cot(\delta_{ir}+\alpha_{ir})}=\frac{l}{w} \tag{2}$$

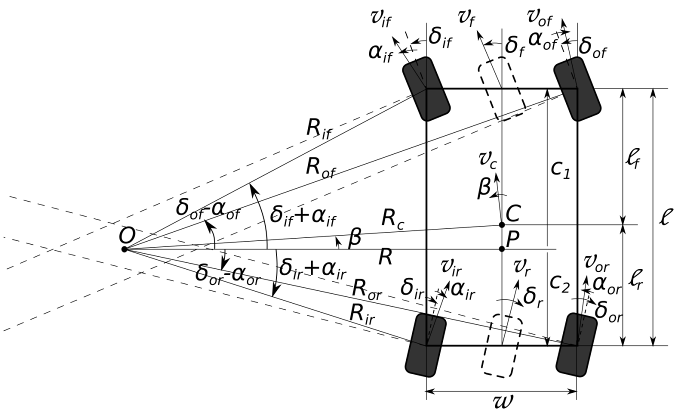


Fig. 4. The 4WS Kinematic Model with Slipping.

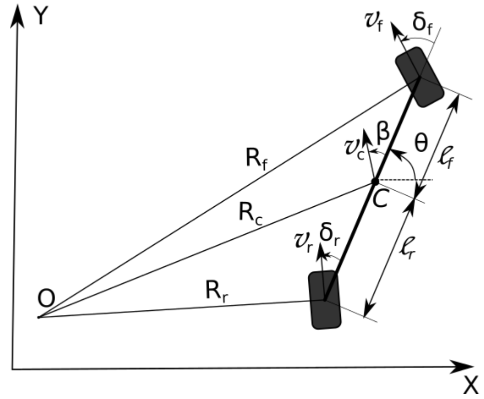


Fig. 5. Equivalent bicycle model of the 4WS model on the plane.

The motion of the robot in the plane, that is the linear and angular velocities $\dot{X},\dot{Y},\dot{\Theta}$, as shown in Fig. 5, using the equivalent bicycle model, can be expressed using the velocity equations from [9]:

$$\dot{X}=v_c\cdot\cos(\theta+\beta) \tag{3}$$

$$\dot{Y}=v_c\cdot\sin(\theta+\beta) \tag{4}$$

$$\dot{\Theta}=\omega_c=\frac{v_c\cdot\cos\beta\cdot(\tan\delta_f-\tan\delta_r)}{l} \tag{5}$$

where v_c is the resultant speed of the robot and β is the sideslip angle and are expressed by

$$v_c=\frac{v_f\cdot\cos\delta_f+v_r\cos\delta_r}{2\cdot\cos\beta} \tag{6}$$

$$\beta=\arctan\frac{l_r\cdot\tan\delta_f+l_f\cdot\tan\delta_r}{l} \tag{7}$$

III. Autonomous Navigation in Unknown Environments

A. Software Architecture

The software of PANDORA Monstertruck was based on the Robot Operating System (ROS) and as shown in Fig. 6 consists of six basic components, as described below.

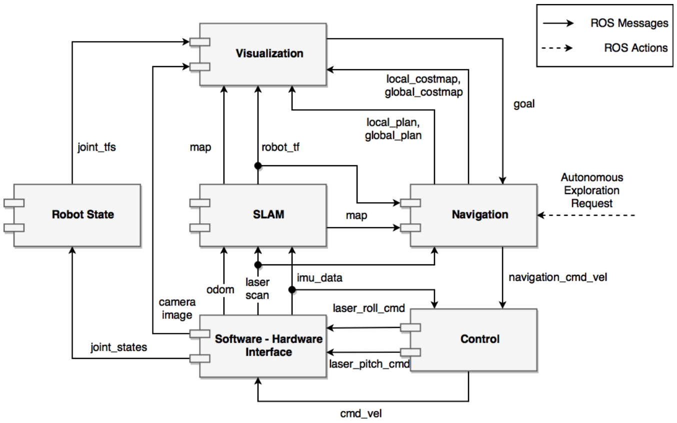


Fig. 6. A simplified component diagram of the software architecture of PANDORA Monstertruck.

1) Hardware-Software Interface: Component that contains drivers and ROS interfaces for sensors and actuator controllers.

2) Control: Component responsible for teleoperation, the laser scanner stabilization and velocity command priority multiplexing, between navigation and teleoperation commands.

3) SLAM: Component responsible for environment mapping and state estimation using the CRSM-SLAM algorithm.

4) Robot State: Component responsible for publishing the transformations between the reference frames of the joints of the robot.

5) Visualization: Component responsible for the visualization of the information provided by the sensors and algorithms of the robot (runs on the remote monitoring station).

6) Navigation: Component responsible for the exploration and autonomous navigation of the robot. It consists of two nodes, a frontier exploration node for autonomous target selection and the ROS node move_base, which, given a goal in the world, will try to reach it. The node move_base, as shown in Fig. 7, is used in combination with the global_planner plugin and the rsband_local_planner plugin, which implement the algorithms Dijkstra/A* for global path planning and the Reeds-Shepp Band algorithm for dynamic local path planning, respectively. The rsband_local_planner plugin, also, implements a fuzzy path tracking controller for 4WS car like robots that along with the Reeds-Shepp Band algorithm are presented in the following subsections.

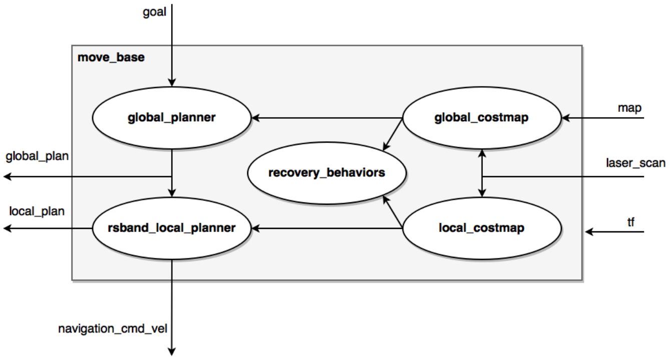


Fig. 7. The ROS move_base node with the global_planner and rsband_local_planner plugins.

B. Reeds-Shepp Band

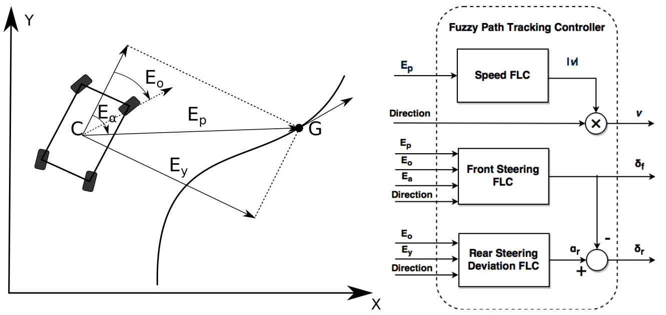
The proposed Reeds-Shepp Band algorithm is an approximation of the Nonholonomic Bubble Band algorithm by Khatib et al. [4], which is used for dynamic modification of a global path, in order to achieve dynamic collision avoidance, while at the same time transforming the path into a kinematically feasible one for car-like robots.

The Bubble Band algorithm uses bubbles, which represent the locally reachable space around a given configuration, while considering the obstacles around it via an appropriate distance metric. A nonholonomic bubble is a bubble produced using the shortest nonholonomic distance [10] or the shortest collision causing Reeds-Shepp Path around a configuration, where Reeds-Shepp paths are defined in [5] as the optimal paths for a car that goes both forwards and backwards. The Bubble Band is produced by generating nonholonomic bubbles across the given global path and dynamically applying to them artificial internal forces that tend to straighten the path and external forces that tend to move the path away from obstacles.

The Reeds-Shepp Band algorithm is comparable to the Nonholonomic Bubble Band algorithm of Khatib et al. [4], but uses the computationally more efficient euclidean distance metric, resulting in a holonomic elastic band [3] algorithm that dynamically deforms a static global path to a collision free local path, which, however, is not kinematically feasible. Therefore, the deformation is followed by an interpolation using collision-free Reeds-Shepp paths, resulting in a both collision-free and kinematically feasible local path that the robot can follow using the fuzzy path tracking controller presented in the following subsection.

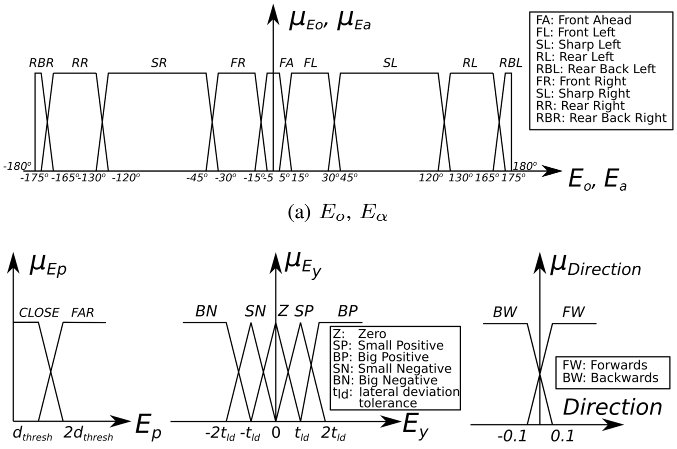
C. Fuzzy Path Tracking Controller

In order to track/follow the Reeds-Shepp Band path, the information about the path and the robot pose must be fed to a controller in order to produce suitable velocity commands. This role is fulfilled by a fuzzy-logic-based path tracking controller, shown in Fig. 8 and which was developed with the goal to utilize both the counter steering and the crab steering capabilities of the robot.



(a) (b)

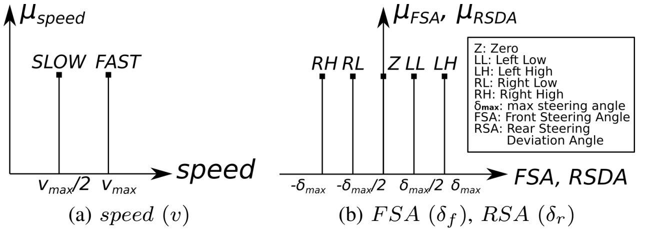
Fig. 8. (a) The deviation errors between robot pose and goal pose and (b) the fuzzy path tracking controller.



(a) E_o, E_α

(b) E_p (c) E_y (d) $Direction$

Fig. 9. The membership functions of the input variables of the controller.



(a) $speed$ (v) (b) FSA (δ_f), RSA (δ_r)

Fig. 10. The membership functions of the output variables of the controller.

As shown in Fig. 8, the controller considers four error variables and in particular the orientation error E_o, the angular deviation error E_α, the position error E_p and the lateral deviation error E_y, as well as the direction of motion defined as forwards for $|E_\alpha| \leq 120°$ and backwards for $|E_\alpha| > 120°$, whose membership functions are presented in Fig. 9. The path tracking controller, similarly to the controller presented in [7], consists of three individual fuzzy logic controllers (FLCs), responsible for the control of the speed v, the front steering angle δ_f and the rear steering deviation angle α_r, the membership functions of which are shown in Fig. 10.

The rule set of the FLCs (Table I) was designed such that the robot performs position correction when it is far from the goal and orientation correction when it is close to the goal so that it reaches the final goal pose similarly to the approach in [6]. In addition, using the rear steering deviation controller, the robot uses counter steering when the position and angular deviation errors are big and crab steering in order to correct slight lateral deviations without significant increase in the orientation error.

TABLE I
FUZZY PATH TRACKING CONTROLLER RULE SET

Speed	if Ep is CLOSE then Speed is SLOW
	if Ep is FAR then Speed is FAST

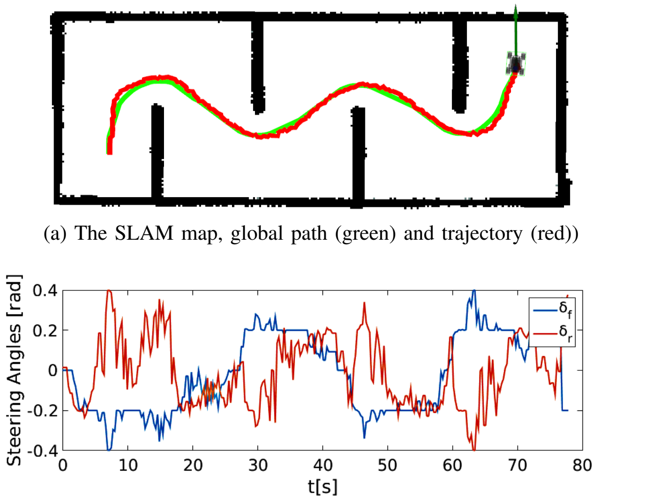
FSA	if Ep is FAR and Ea is RBL then FSA is Z
	if Ep is FAR and Ea is RL then FSA is LH
	if Ep is FAR and Ea is SL then FSA is LH
	if Ep is FAR and Ea is FL then FSA is LL
	if Ep is FAR and Ea is FA then FSA is Z
	if Ep is FAR and Ea is FR then FSA is RL
	if Ep is FAR and Ea is SR then FSA is RH
	if Ep is FAR and Ea is RR then FSA is RH
	if Ep is FAR and Ea is RBR then FSA is Z
	if Ep is CLOSE and Direction is FW and Eo is SL then FSA is LH
	if Ep is CLOSE and Direction is FW and Eo is FL then FSA is LL
	if Ep is CLOSE and Direction is FW and Eo is FA then FSA is Z
	if Ep is CLOSE and Direction is FW and Eo is FR then FSA is RL
	if Ep is CLOSE and Direction is FW and Eo is SR then FSA is RH
	if Ep is CLOSE and Direction is BW and Eo is SL then FSA is RH
	if Ep is CLOSE and Direction is BW and Eo is FL then FSA is RL
	if Ep is CLOSE and Direction is BW and Eo is FR then FSA is LL
	if Ep is CLOSE and Direction is BW and Eo is SR then FSA is LH

RSDA	if Eo is not FA and Eo is not FR and Eo is not FL then RSDA is Z
	if Direction is FW and Eo is FA and Ey is BP then RSDA is Z
	if Direction is FW and Eo is FL and Ey is BP then RSDA is Z
	if Direction is FW and Eo is FR and Ey is BP then RSDA is Z
	if Direction is FW and Eo is FA and Ey is SP then RSDA is LH
	if Direction is FW and Eo is FL and Ey is SP then RSDA is LH
	if Direction is FW and Eo is FR and Ey is SP then RSDA is LH
	if Direction is FW and Eo is FA and Ey is Z then RSDA is Z
	if Direction is FW and Eo is FL and Ey is Z then RSDA is Z
	if Direction is FW and Eo is FR and Ey is Z then RSDA is Z
	if Direction is FW and Eo is FA and Ey is SN then RSDA is RH
	if Direction is FW and Eo is FL and Ey is SN then RSDA is RH
	if Direction is FW and Eo is FR and Ey is SN then RSDA is RH
	if Direction is FW and Eo is FA and Ey is BN then RSDA is Z
	if Direction is FW and Eo is FL and Ey is BN then RSDA is Z
	if Direction is FW and Eo is FR and Ey is BN then RSDA is Z
	if Direction is BW and Eo is FA and Ey is BP then RSDA is Z
	if Direction is BW and Eo is FL and Ey is BP then RSDA is Z
	if Direction is BW and Eo is FR and Ey is BP then RSDA is Z
	if Direction is BW and Eo is FA and Ey is SP then RSDA is LH
	if Direction is BW and Eo is FL and Ey is SP then RSDA is LH
	if Direction is BW and Eo is FR and Ey is SP then RSDA is LH
	if Direction is BW and Eo is FA and Ey is Z then RSDA is Z
	if Direction is BW and Eo is FL and Ey is Z then RSDA is Z
	if Direction is BW and Eo is FR and Ey is Z then RSDA is Z
	if Direction is BW and Eo is FA and Ey is SN then RSDA is RH
	if Direction is BW and Eo is FL and Ey is SN then RSDA is RH
	if Direction is BW and Eo is FR and Ey is SN then RSDA is RH
	if Direction is BW and Eo is FA and Ey is BN then RSDA is Z
	if Direction is BW and Eo is FL and Ey is BN then RSDA is Z
	if Direction is BW and Eo is FR and Ey is BN then RSDA is Z

IV. EXPERIMENTAL RESULTS

This section presents a series of experiments that demonstrate the behaviour of the autonomous navigation and exploration capabilities of the developed robotic platform.

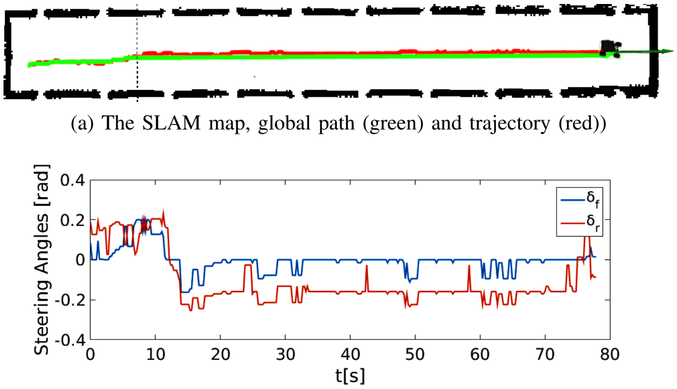
The path tracking experiments presented in Fig. 11-12 were conducted in the Gazebo 3D simulator, with a view to demonstrate the operation of the fuzzy path tracking controller. For this experiment, a feasible static global path was constructed, which the robot managed to track successfully in both cases by using both counter and crab steering in order to offset angular and distance errors so as to follow the path as close as possible.



(a) The SLAM map, global path (green) and trajectory (red))

(b) Front and rear steering angles $\delta_f(blue), \delta_r(red)$

Fig. 11. Path tracking experiment in Gazebo.



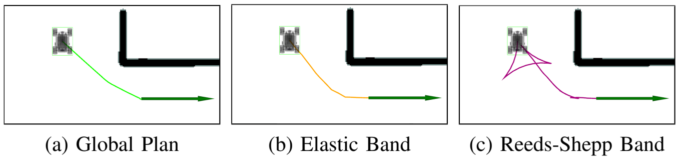
(a) The SLAM map, global path (green) and trajectory (red))

(b) Front and rear steering angles $\delta_f(blue), \delta_r(red)$

Fig. 12. Path tracking experiment under high slippage in a perpendicularly inclined corridor.

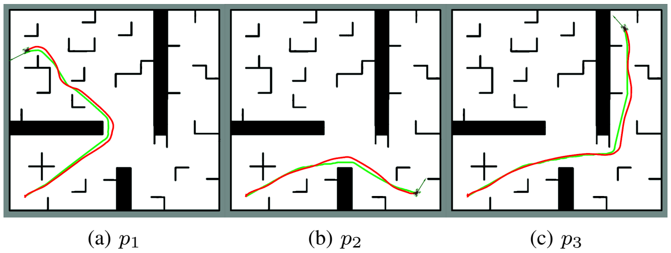
The experiments presented in Fig. 13- 14 were conducted in the STDR 2D simulator and demonstrate the planning stages of the proposed autonomous navigation system, as well as its successful application for three distinct goal poses.

Finally, an autonomous exploration experiment with the actual PANDORA Monstertruck robot was conducted in the Computer Systems and Architecture Lab (CSAL) of the School of Electrical and Computer Engineering, AUTH, as presented in Fig. 15.



(a) Global Plan (b) Elastic Band (c) Reeds-Shepp Band

Fig. 13. Demonstration of the planning stages of the proposed 4-Tier autonomous navigation system.



(a) p_1 (b) p_2 (c) p_3

Fig. 14. Autonomous navigation experiments using the proposed 4-Tier autonomous navigation system for three distinct goal poses in the map sparse_obstacles of the STDR simulator. The green line denotes the global path and the red line, the trajectory of the robot.

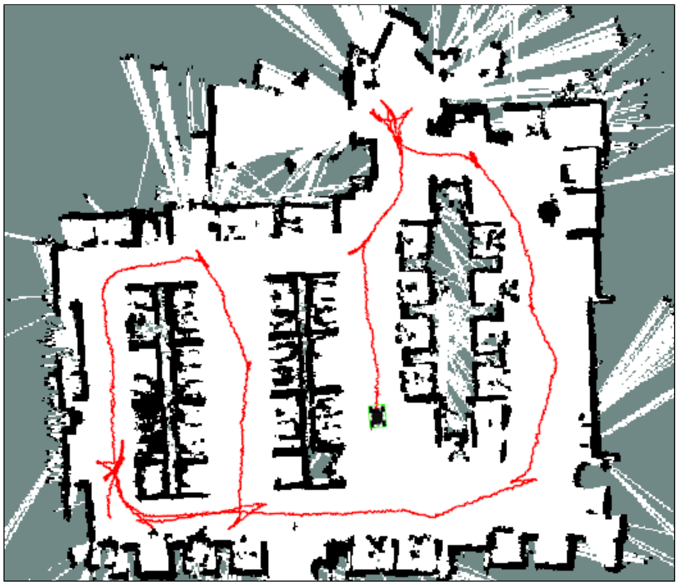


Fig. 15. Autonomous exploration experiment of PANDORA Monstertruck in CSAL. The map was constructed using the CRSM SLAM algorithm and the executed trajectory is denoted by the red continuous line.

V. Conclusions

In this paper, an autonomous 4WS4WD car-like robot was presented with regards to both its hardware and software architecture. As demonstrated in the experimental results, autonomous navigation is accomplished using the proposed 4-Tier autonomous navigation system that plans a holonomic global path, deforms it dynamically using the proposed Reeds-Shepp Band algorithm so that it can be collision free and kinematically feasible and finally tracks it robustly using the proposed fuzzy path tracking controller utilizing both counter and crab steering. It was also demonstrated that the robot can autonomously explore and map a previously unknown environment by employing the proposed software architecture.

Further work could be done to improve the robustness, stability and safety of the navigation system by using a more deterministic Reeds-Shepp planner and by adding collision detection capabilities in the path tracking controller. In addition, through the employment of additional sensors and perception algorithms, the capabilities of the robot could be significantly increased regarding localization, mapping, rough terrain traversal and point of interest search.

Acknowledgment

The authors would like to thank PANDORA Robotics of the Aristotle University of Thessaloniki for the equipment, source code and know-how provided for the development of the PANDORA Monstertruck robotic platform.

References

[1] E. Tsardoulias and L. Petrou, “Critical rays scan match slam,” *Journal of Intelligent and Robotic Systems*, vol. 72, no. 3-4, pp. 441–462, 2013.

[2] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” *IEEE Robotics Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar 1997.

[3] S. Quinlan and O. Khatib, “Elastic bands: connecting path planning and control,” in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*, May 1993, pp. 802–807 vol.2.

[4] M. Khatib, H. Jaouni, R. Chatila, and J. P. Laumond, “Dynamic path modification for car-like nonholonomic mobile robots,” in *Robotics and Automation, 1997. Proceedings., 1997 IEEE International Conference on*, vol. 4, Apr 1997, pp. 2920–2925 vol.4.

[5] J. A. Reeds and L. A. Shepp, “Optimal paths for a car that goes both forwards and backwards.” *Pacific J. Math.*, vol. 145, no. 2, pp. 367–393, 1990. [Online]. Available: http://projecteuclid.org/euclid.pjm/1102645450

[6] W. Lin, “Path following for a car-like mobile robot based on fuzzy-logic,” Master’s thesis, Orebro University, Department of Technology, 2012.

[7] A. Mendes, L. C. Bento, and U. Nunes, “Path-tracking controller with an anti-collision behaviour of a bi-steerable cybernetic car,” in *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA ’03. IEEE Conference*, vol. 1, Sept 2003, pp. 613–619 vol.1.

[8] R. N. Jazar, *Vehicle Dynamics: Theory and Applications*. Springer Science and Business Media, LLC, 2009.

[9] D. Wang and F. Qi, “Trajectory planning for a four-wheel-steering vehicle,” in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 4, 2001, pp. 3320–3325 vol.4.

[10] J. P. Laumond and P. Soueres, “Metric induced by the shortest paths for a car-like mobile robot,” in *Intelligent Robots and Systems ’93, IROS ’93. Proceedings of the 1993 IEEE/RSJ International Conference on*, vol. 2, Jul 1993, pp. 1299–1304 vol.2.