Design and Implementation of Autonomous Ground Vehicle for constrained environments

*G Rahul Kranti Kiran† krantikiran.68@iitkgp.ac.in
*Indu Kant Deo‡ indukantdeo@iitkgp.ac.in
*Sanskar Agrawal§ sanskaragrawal@iitkgp.ac.in
*Siddhant Haldar§ siddhanthaldar14@iitkgp.ac.in
*Het Shah¶ hetshah369@iitkgp.ac.in
*Sohan Rudra** sohanrudra@iitkgp.ac.in
*Harsh Maheswari§ harshmasheshwari135@iitkgp.ac.in
*Aditya Rathore§ arathore98@iitkgp.ac.in
*Poojan Shah¶ poojan@iitkgp.ac.in
*Ashwin Nehete¶ ashwinnehete@iitkgp.ac.in
Debashish Chakravarty[xii] dc@mining.iitkgp.ac.in

***Abstract*—The design, development and validation of the Eklavya 6.0 have been presented in this paper. Eklavya 6.0, a three-wheeled differential drive autonomous robot which can navigate to prespecified GPS coordinates as well as through lanes was developed to participate in the Autonomous Navigation Challenge of the 26th Intelligent Ground Vehicle Competition(IGVC). The mechanical, electronic, software architecture sub-modules including localization, lane detection navigation and motion planning have been discussed.**

***Index Terms*—Autonomous Vehicle, Computer Vision, Localization, Mapping, Motion Planning, Obstacle Avoidance, Lane Detection**

I. Introduction

Autonomous Vehicles form the rage of our times and are arguably the fastest progressing technologies. The DARPA Grand Challenge was the first ever autonomous vehicle competition which drew the attention of the researchers towards this budding field. The Intelligent Ground Vehicle Competition (IGVC) is one of the four unmanned systems competitions founded by the Association for Unmanned Vehicle Systems International (AUVSI). The IGVC comprises of a multitude of tasks for multiple categories of autonomous vehicles. Eklavya 6.0 was designed to participate in the autonomous navigation challenge of IGVC, which involves development of an autonomous bot having lane-following and GPS-tracking abilites. It is capable of traversing a rough grassland environment having scattered obstacles. The bot has been designed as an improvement over the previous iterations of Eklavya series of robots, having improved mobility with a more modular and robust system.

We propose simple prototype, having a differential drive system with a front wheel drive. The proposed system uses an array of high precision sensors for capturing a snapshot of the external world model, which then passes the information of to the processing systems for information extraction and decision making. The proposed system is relatively lightweight, and can run on almost any system with medium compute capabilities. It is robust to various changes in the sensor or robot configurations.

In this paper we describe each of the separate modules categorically by breaking down the entire task into three main components namely Mechanical Design, Electronic Subsystems and Software Stack.

* Denotes equal contribution.
The authors are with the Indian Institute of Technology Kharagpur.
† Department of Computer Science & Engineering.
‡ Department of Aerospace Engineering.
§ Department of Electrical Engineering.
¶ Department of Mechanical Engineering.
** Department of Mathematics.
xii Department of Mining Engineering.

II. System Design

A. Mechanical Design & Analysis

This section deals with the mechanical vertical of the autonomous robot ranging from material selection to structural optimisation. Eklavya 6.0's mechanical design pays particular attention to being lightweight, modular, compatible, easy to assemble and disassemble, and maneuverability of the robot.

1) Chassis: The robot chassis is designed on SolidWorks and analyzed on the simulation software ANSYS. Teak wood is used as the main constituent of the chassis due to its vibration damping properties and high dimensional stability.

The truss structure of the chassis is best suited to handle shear stress encountered by the vehicle. Use of triangular structures has led to significant decrease in the vibration of the chassis, and led to enhancement of the stability of the vehicle; necessary for optimal performance of all the sensors. The horizontal beams stabilize the base and prevent bending of the structure providing rigidity to the structure.The simulation results showed that the strain induced due to random vibrations in the wooden frame was reduced by nearly 35% (as shown in fig.2).

2) Drive System: Locomotion in Eklavya 6.0 is achieved by two pneumatic wheels and a caster wheel. The differential drive configuration allows zero-radius turns [1] about the powered wheel axis for easy maneuverability. The position

978-1-5386-9245-5/19/$31.00 ©2019 IEEE
DOI 10.1109/IRC.2019.00043

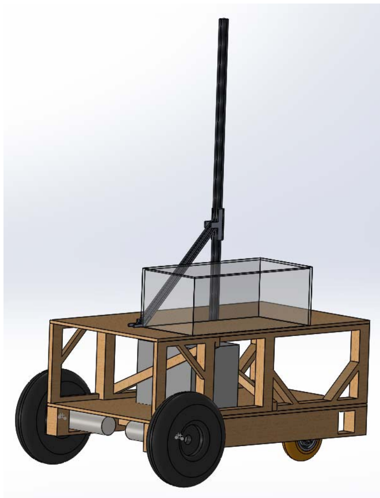

Fig. 1. CAD model of Eklavya 6.0.

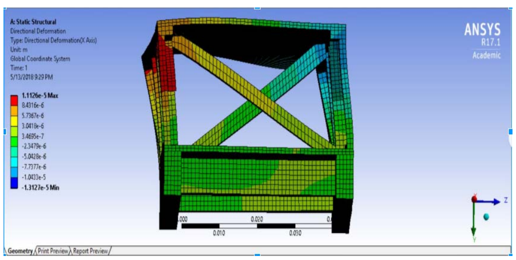

Fig. 2. Structure with Cross-Member

of the centre of mass of the robot has been kept significantly low and is located closer to the axis of the powered wheels.

B. Embedded architecture and design

The embedded architecture and design links the mechanical actuators with the decision making software stack. The architecture includes the required safety measures and is also energy efficient.

1) Sensors and Actuators: Robot gathers raw data from the environment using the sensors and extracts the useful information by processing it.

The proposed system consists of 2 high torque DC motors, Roboteq MDC2230 Motor Controller, sensors like Lidar, Camera, Encoders, GPS and IMU, RF transceiver for Wireless Electronic Stop and a Laptop. Our system uses a front-facing camera to observe the lanes, a 2D Lidar to detect the static obstacles, and GPS, IMU and encoders to localize itself in the environment. The specifications of the sensors and actuators used in our system are given in table I and II.

2) Power Analysis: It is necessary for a system to ensure that all the electronic components receive adequate amount of power to ensure the safety and proper functioning of the components.

TABLE I
ACTUATOR SPECIFICATION

Geared Motor	Operating Voltage:- 24V, Current:-Max-30A No load:- 1.12A, Rated torque:- 142 Kg cm, Gearbox Ration:- 1:6
Roboteq MDC2230	60A Dual DC motor driver Dual Encoder inputs with 32-bit counters.

TABLE II
SENSOR SPECIFICATION

Vector Nav	3-axis accelerometer, 3-axis gyroscope, 3-axis magnetometer, barometric pressure sensor. GPS-aided Inertial Navigation System (INS).
HOKUYO 2D Lidar	Range of 30 m in 270 degree Plane of device Millimeter resolution in a 270 arc. Accuracy 50 mm within a range of 0.1-30 m.
BFLY- 23S6 Camera	On-camera image processing like color interpolation, gamma, and LUT 16 MByte frame buffer; LED status indicator.
Planetary Encoder	2 Channel Quadrature Encoder 2000 CPR

TABLE III
POWER ANALYSIS

Component	**Quantity**	**Rated Power**
Planetary Encoder Geared Motor	2	100W X 2
Roboteq MDC2230 Motor Driver	1	10 W
BFLY- 23S6 Camera	1	2.5W
VectorNav VN-200	1	0.5W
HOKUYO UTM-30LX Lidar	1	8.4 W
GPS	1	0.5W
Flashlight	1	4.8 W
Total		226.7W

C. Localization and Mapping

Eklavya 6.0 uses high precision encoders, IMU and GPS whose data is passed through two levels of Extended Kalman Filters(EKF) [2]. The EKF algorithm does sensor fusion to improve location accuracy of the bot by at least ten-fold from traditional GPS readings.

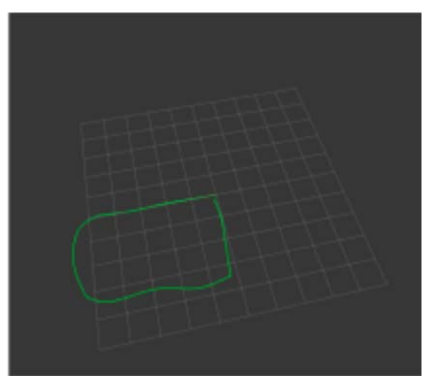

Fig. 3. Loop Closure in Eklavya 6.0

D. Trajectory Planning & Controls

The trajectory Generation and controls module is responsible for optimal trajectory planning and execution of control commands for the said path in an efficient fashion. The proposed planning and controls stack uses a heuristic search based global path-planner in conjunction with an elastic-band based local-planner for trajectory optimization and the output finally, is converted into control commands and feeded into a closed loop control system.

1) Planner: The entire path planning stack can be broadly be deconstructed into two parts:

- Global Planner : The global planner provides the interpolated path between two consecutive way-points which are targets provided by the GPS navigation module or the lane detection module, considering the global obstacle map. The global-planner used is a fast implementation of A^* heuristic search algorithm [3].
- Local Planner : As it is infeasible to use the global planner repeatedly in short intervals, the use of a quick local planner which generates negotiable trajectories is essential. The local-planner used is an Elastic Band [4] based trajectory optimization algorithm. An Elastic band is a dynamic deformable trajectory which lies on a collision-free path taking into consideration the holonomic constraints of the robot prototype.
  We have used TEB-Timed Elastic Band [5] as our local planner which samples multiple elastic band trajectories.
  $$\mathcal{H}(\bar{\tau}) = \int_{\bar{\tau}} \mathcal{F}(z)dz \quad (1)$$
  The eqn. (1) shows the H-signature's relation over various trajectories between start and goal points. This is used to distinguish between various homotopy classes thus reducing the redundancy of the sampling.

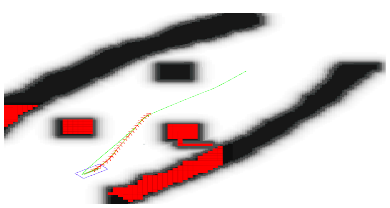

Fig. 4. Path Planning in Lanes with Obstacles

2) Controls: The control architecture of Eklavya 6.0 tries to reject the environmental disturbances and tracks the generated linear and angular speed from the Trajectory generation module which are then converted to the differential velocities of left and right wheel. PID control scheme is chosen because of its ease of implementation and the degree of freedom of tuning three parameters to achieve better performance [6]. The speed feedback is obtained using the two front wheel encoders. The configuration of the mobile robot can be described by five generalized coordinates such as:

$$X = \begin{bmatrix} x & y & \omega_l & \omega_r \end{bmatrix} \quad (2)$$

where x and y are the egocentric coordinates of the vehicle, θ is the orientation angle of the robot (in the world-fixed frame), ω_l is the rotational speed of the left wheel, ω_r is the rotational speed of the right wheel. Using the kinematic model [6], the relation between the parameters can be given as follows

$$\begin{bmatrix} V \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ \frac{R}{2b} & \frac{R}{-2b} \end{bmatrix} \begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix} \quad (3)$$

where R is the radius of the wheel. The position and the orientation of the robot can be determined by the following

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V \\ \dot{\theta} \end{bmatrix} \quad (4)$$

E. Lane Detection & Way-point Generation

A real time lane detection algorithm addressing the complications like glare and shadow effects is presented here. We acquire a RGB image from the camera and pass it through the lane detection pipeline as described further.

In this paper, a lane estimation approach with multiple lane model identification followed by an appropriate lane curve fitting is proposed and implemented [7].

1) Top View Generation: Inverse perspective mapping is used to obtain the top view of the image.

2) Background Substraction and Thresholding: The next computation step is background shedding which is chiefly green coloured regions emerging due to grass. This step condenses the information content and decreases the computational time of the algorithm. In our experiment, the difference of twice the blue channel and green channel was eventually used. The functional white pixels of the lane are segregated from the green background using correlation algorithm which compares the intensity values of the grayscale, B, G and R channels against a selected threshold value.

3) Superpixel Clustering for Outlier Reduction: The thresholded image still contains some outliers, which might have an adverse effect on curve fitting. We have devised a Simple Linear Iterative Clustering (SLIC) based technique for outlier reduction which does not affect the lanes.

4) Lane Model Identification: The suggested lane model identification methodology uses adaptive lane modelling to handle cases like intersections and horizontal appearing lane markers. In the event of horizontal lanes, lanes are modelled as straight lines otherwise, a parabolic lane model is employed. The algorithm optimally switches between a line and quadratic polynomial for the lane model. All models of lines present in the clustered image are determined using the Hough algorithm. Against all the estimated models of line, one with the maximum number of inliers is used for gradient calculation. The computed gradient is correlated with a threshold value to determine the lane model. If the computed gradient of the detected line is inside a specific range, which in our case was observed to be lying between 20 degrees and -20 degrees, the lane model is estimated using a line. Contrarily, a parabola based model estimation is selected.

5) Lane Model Estimation: Two parabolic curves are simultaneously estimated using the RANSAC [8] algorithm. The model of parabolas used are:

$$y^2 = \lambda_1.(x-a), y^2 = \lambda_2.(x-a-w) \tag{5}$$

The choice of parameters λ_1 and λ_2 is made in such a way that the shift between the vertex of the two parabolas is related to the actual width of the lane w. λ_1 and λ_2 represent the curvature of the two polynomials. Curves with w less than a certain threshold and λ_1 and λ_2 having small values with opposite signs are eliminated. The RANSAC confidence simultaneously takes into account the model of both the lanes. In case of single lane, the other lane is predicted by contemplating whether the detected lane is the left or the right lane.

6) Local Waypoint Generation: After determining the equations of lanes, we select and pass the local waypoint between the lanes, to the planner. Waypoint is calculated at a distance of 3 meters ahead and the heading direction is calculated as the average of gradients over all the points on the lane. This point, along with the orientation, is then passed into the planner.

7) Improvements Over Other Approaches: Opposed to other lane fitting approaches which estimate a single lane twice, our model optimizes over both the lanes simultaneously. This ensures that the lanes are apart by an appropriate distance, even when only a single lane is visible. Hence, our model works even in such non-favourable conditions.

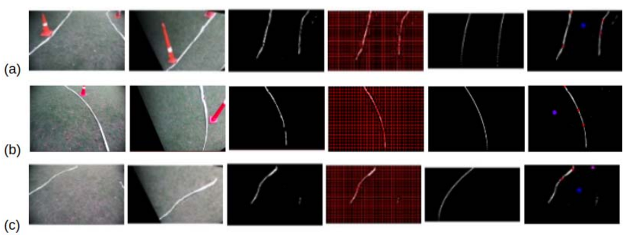

Fig. 5. From left to right (1)Acquired image from the camera (2) Top view image (3) Preprocessed image (4) Clustered image (5) Estimated lane model (6) Estimated heading point. The results clearly demonstrate that objects and white patches appearing close to the lanes do not have any adverse effect on the performance of the algorithm.

III. System Integration

Robotic systems are highly data intensive and involve a lot of concurrent processes. For this very purpose we have chosen ROS:Robot Operating System [9] as the backbone of our autonomous driving system. ROS provides highly optimized networking protocols for the exchange of information through nodes and topics in a concurrent manner. The handy abstraction provided by ROS eliminates the hassle of handcrafting data management solutions. The entire vehicle's software stack was deployed on a medium powered machine running Linux Ubuntu Xenial 16.04 with ROS Kinetic Kame packing a Intel i7-770HQ processor, 16GB of DDR4 RAM and a CUDA capable Graphics Processing Unit (Nvidia GTX1050).

IV. Conclusion

In this paper, we have presented the design and implementation of IIT Kharagpur's entry to IGVC, Eklavya 6.0. We have discussed the advantages of a wooden chassis with adequate design optimisation and have successfully managed to keep the deformations under the considered limits. Robust EKF based localisation has been discussed. It is shown that usage of TEB has allowed the robot to navigate the path, following a smooth trajectory thus leading to an improvement in the time taken to cover the distance, reducing the collision frequency as compared to previously used RRT based planners.

The integrated system is able to efficiently navigate through GPS waypoints, follow lanes and avoid obstacles with significant improvements to the hardware as well as software components. The performance of the systems was found to be satisfactory, although there are several possibilities for future work so that the robustness may be increased. It must be noted that our entry using the proposed architecture received the 2nd prize jointly with Boise State University in the Autonomous Navigation challenge at IGVC 2018.

Youtube link of Eklavya 6.0's final run at the 26^{th} IGVC.

Acknowledgment

We would like to thank team Autonomous Ground Vehicle, IIT Kharagpur and Sponsored Research & Industrial Consultancy, IIT Kharagpur for providing us with resources and funding for this project.

References

[1] Patil R.V. Mohite A.V. Prof. Patil .D.D. Shirsath S.V., Jadhav K.R. Zero turn vehicle. *International Research Journal of Engineering and Technology (IRJET)*, 2016.

[2] T. Moore and D. Stouch. A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.

[3] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, July 1968.

[4] S. Quinlan and O. Khatib. Elastic bands: connecting path planning and control. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 802–807 vol.2, May 1993.

[5] Christoph Rsmann, Frank Hoffmann, and Torsten Bertram. Integrated online trajectory planning and optimization in distinctive topologies. *Robot. Auton. Syst.*, 88(C):142–153, February 2017.

[6] CS Shijin and K Udayakumar. Speed control of wheeled mobile robots using pid with dynamic and kinematic modelling. In *Innovations in Information, Embedded and Communication Systems (ICIIECS), 2017 International Conference on*, pages 1–7. IEEE, 2017.

[7] S. Agrawal, I. K. Deo, S. Haldar, G. R. Kranti Kiran, V. Lodhi, and D. Chakravarty. Off-road lane detection using superpixel clustering and ransac curve fitting. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1942–1946, Nov 2018.

[8] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[9] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. Ros: an open-source robot operating system. In *ICRA Workshop on Open Source Software*, 2009.