GRD Journal for Engineering | Volume 1 | Issue 12 | November 2016
ISSN: 2455-5703

Implementation of Low-Level Control in Autonomous Ground Vehicle

Sundar Ganesh C S
Assistant Professor
Department of Robotics and Automation Engineering
PSG College of Technology

Rajkumar A
Junior Research Fellow
Department of Robotics and Automation Engineering
PSG College of Technology

Abstract

The goal of this paper is to compute odometry of vehicle using low-level controls combined with inertial measurement unit. The Low-Level control includes design of Drive-by-Wire mechanisms for steering, brake and accelerator systems with appropriate motors and encoder. Experimentation with encoders and DC motors of steering and brake has been carried out first with various embedded modules to choose best suitable module. The experimentation has led to choosing BeagleBone Black (BBB), A low-cost, open-source community-supported development platform for real-time analysis provided by the TI Sitara AM3358 ARM Cortex-A8 processor with Linux-based operating system. Using BBB dedicated hardware module for high CPR (Counts per Revolution) encoders, the vehicle position is evaluated. Using BBB serial cape, it is interfaced to Roboteq motor controller (used for steering and brake motor) and steering encoder for steering wheel position control. The major task of the paper is the evaluation of odometry from using vehicle rear wheel encoders combined with inertial measurement unit. The paper is carried out on a dune buggy; petrol powered motor vehicle with Ackermann drive platform type and mobility. Initially, Drive-by-wire mechanism for steering, brake and accelerator is designed. Autonomous steering control of vehicle is carried out with feedback from steering motor encoder and steering hand wheel encoder connected to axle of steering system. Using IMU (Inertial Measurement Unit) yaw angle and rear wheel axle encoder position value, the odometry of vehicle are computed. Combined with inertial measurement units, they have proven to be a precise and low-cost sensor for vehicle odometry evaluation.
Keywords- Autonomous vehicle, Odometry, X-by-wire, and Ackermann drive mechanism, Low-level control

I. INTRODUCTION

The Low-Level control includes design of Drive-by-Wire mechanisms for steering, brake and accelerator and control of low-level systems with appropriate embedded module. The goal of this project is to compute odometer of vehicle using low-level controls combined with inertial measurement unit and integration of x-by wire mechanism.

A guidance system for autonomous vehicles navigation in semi-structured outdoor environments. It integrates redundant encoder's data and absolute positioning data provided by landmarks and artificial beacons. Natural features are localized using a laser range sensor, and magnetic sensing rulers were developed to detect magnetic markers buried in the ground. In the first fusion stage, data from four wheel encoders and one steering encoder are fused by means of an EKF, providing robust odometric information, namely in face of undesirable effects of wheels slippage. Next, a second fusion stage is processed for integrating odometric and absolute positioning data. Simulation and real experiments using a four-wheel actuated electrical vehicle are presented [1]. Visual odometer is vital to the future of mobile robotics. In this paper, we demonstrate a method that combines information from optic flow and stereo to estimate and control the current position of a quad rotor along a pre-defined trajectory. The absolute translation in 3D is computed by combining the optic flow measurements between successive frames and stereo-based height over ground. The current 3D position, as estimated from path integration of the incremental translations, is controlled in closed loop to follow the prescribed trajectory. The performance of the system is evaluated by measuring the error between the initial and final positions in closed circuits. This error is approximately 1.7% of the total path length [2].

An autonomous vehicle-target assignment problem where a group of vehicles are expected to optimally assign themselves to a set of targets. We introduce a game theoretical formulation of the problem in which the vehicles are viewed as self-interested decision makers. Thus, we seek the optimization of a global utility function through autonomous vehicles that are capable of making individually rational decisions to optimize their own utility functions. The first important aspect of the problem is to choose the utility functions of the vehicles in such a way that the objectives of the vehicles are localized to each vehicle yet aligned with a global utility function. The second important aspect of the problem is to equip the vehicles with an appropriate negotiation mechanism by which each vehicle pursues the optimization of its own utility function. We present several design procedures and accompanying caveats for vehicle utility design. We present two new negotiation mechanisms, namely, "generalized regret monitoring with fading memory and inertia" and "selective spatial adaptive play," and provide accompanying proofs of their convergence [3]. This report explores the impacts that autonomous (also called self-driving, driverless or robotic) vehicles are likely to have on travel demands and transportation planning. It discusses autonomous vehicle benefits and costs, predicts their likely development and implementation based on experience with previous vehicle technologies, and explores how they will affect planning decisions such as optimal road, parking and public transit supply. The analysis indicates that some benefits, such as independent mobility for affluent non-drivers, may begin in the 2020s or 2030s, but most impacts, including reduced traffic and parking congestion (and therefore road and parking facility supply requirements), independent mobility for low-income people (and therefore reduced need to subsidize transit), increased safety, energy conservation and pollution reductions, will only be significant when autonomous vehicles become common and affordable, probably in the 2040s to 2060s, and some benefits may require prohibiting human-driven vehicles on certain roadways, which could take longer[4].

We have demonstrated that a reservation-based approach can efficiently and safely govern interactions of multiple autonomous vehicles at intersections. Such an approach alleviates many traditional problems associated with intersections, in terms of both safety and efficiency. However, the system relies on all vehicles being equipped with the requisite technology — a restriction that would make implementing such a system in the real world extremely difficult. In this paper, we extend this system to allow for incremental deploy ability. The modified system is able to accommodate traditional human-operated vehicles using existing infrastructure. Furthermore, we show that as the number of autonomous vehicles on the road increases, traffic delays decrease monotonically toward the levels exhibited in our previous work [5]. This paper presents a guidance system for autonomous vehicles navigation in semi-structured outdoor environments. It integrates redundant encoder's data and absolute positioning data provided by landmarks and artificial beacons. Natural features are localized using a laser range sensor, and magnetic sensing rulers were developed to detect magnetic markers buried in the ground. In the first fusion stage, data from four wheel encoders and one steering encoder are fused by means of an EKF, providing robust odometric information, namely in face of undesirable effects of wheels slippage. Next, a second fusion stage is processed for integrating odometric and absolute positioning data. Simulation and real experiments using a four-wheel actuated electrical vehicle are presented [6].

Visual odometer is vital to the future of mobile robotics. In this paper, we demonstrate a method that combines information from optic flow and stereo to estimate and control the current position of a quad rotor along a pre-defined trajectory. The absolute translation in 3D is computed by combining the optic flow measurements between successive frames and stereo-based height over ground. The current 3D position, as estimated from path integration of the incremental translations, is controlled in closed loop to follow the prescribed trajectory. The performance of the system is evaluated by measuring the error between the initial and final positions in closed circuits. This error is approximately 1.7% of the total path length. [7]. we consider an autonomous vehicle-target assignment problem where a group of vehicles are expected to optimally assign themselves to a set of targets. We introduce a game theoretical formulation of the problem in which the vehicles are viewed as self-interested decision makers. Thus, we seek the optimization of a global utility function through autonomous vehicles that are capable of making individually rational decisions to optimize their own utility functions. The first important aspect of the problem is to choose the utility functions of the vehicles in such a way that the objectives of the vehicles are localized to each vehicle yet aligned with a global utility function. The second important aspect of the problem is to equip the vehicles with an appropriate negotiation mechanism by which each vehicle pursues the optimization of its own utility function [8].

This report explores the impacts that autonomous (also called self-driving, driverless or robotic) vehicles are likely to have on travel demands and transportation planning. It discusses autonomous vehicle benefits and costs, predicts their likely development and implementation based on experience with previous vehicle technologies, and explores how they will affect planning decisions such as optimal road, parking and public transit supply. The analysis indicates that some benefits, such as independent mobility for affluent non-drivers, may begin in the 2020s or 2030s, but most impacts, including reduced traffic and parking congestion (and therefore road and parking facility supply requirements), independent mobility for low-income people (and therefore reduced need to subsidize transit), increased safety, energy conservation and pollution reductions, will only be significant when autonomous vehicles become common and affordable, probably in the 2040s to 2060s, and some benefits may require prohibiting human-driven vehicles on certain roadways, which could take longer[9]. In modern urban settings, automobile traffic and collisions lead to endless frustration as well as significant loss of life, property, and productivity. Recent advances in artificial intelligence suggest that autonomous vehicle navigation may soon be a reality. In previous work, we have demonstrated that a reservation-based approach can efficiently and safely govern interactions of multiple autonomous vehicles at intersections. Such an approach alleviates many traditional problems associated with intersections, in terms of both safety and efficiency. However, the system relies on all vehicles being equipped with the requisite technology — a restriction that would make implementing such a system in the real world extremely difficult. In this paper, we extend this system to allow for incremental deploy ability. The modified system is able to accommodate traditional human-operated vehicles using existing infrastructure. Furthermore, we show that as the number of autonomous vehicles on the road increases, traffic delays decrease monotonically toward the levels exhibited in our previous work. Finally, we develop a method for switching between various human-usable configurations while the system is running, in order to facilitate an even smoother transition. The work is fully implemented and tested in our custom simulator, and we present detailed experimental results attesting to its effectiveness[10].

Boss is an autonomous vehicle that uses on-board sensors (global positioning system, lasers, radars, and cameras) to track other vehicles, detect static obstacles, and localize itself relative to a road model. A three-layer planning system combines mission, behavioral, and motion planning to drive in urban environments. The mission planning layer considers which street to take to achieve a mission goal. The behavioral layer determines when to change lanes and precedence at intersections and performs error recovery maneuvers. The motion planning layer selects actions to avoid obstacles while making progress toward local goals. The system was developed from the ground up to address the requirements of the DARPA Urban Challenge using a spiral system development process with a heavy emphasis on regular, regressive system testing. During the National

Qualification Event and the 85-km Urban Challenge Final Event, Boss demonstrated some of its capabilities, qualifying first and winning the challenge. C 2008 Wiley Periodicals, Inc. [11].

II. VEHICLE ARCHITECTURE

This chapter details about the vehicle architecture that is used in the paper implementation and details about the steering and drive mechanisms. The dune buggy vehicle is petrol powered motor vehicle with Ackermann drive platform type and mobility. The vehicle has two DC motors each for steering and brake. It is auto clutch system with forward and reverse system. A servo motor is connected to throttle for speed control. Two encoders are connected to rear wheel axle for position estimation. Battery and UPS are also provided for powering electronic circuit boards and processors. The Fig 2.1 is the illustration of the vehicle with sensors and processors positioned at appropriate places.

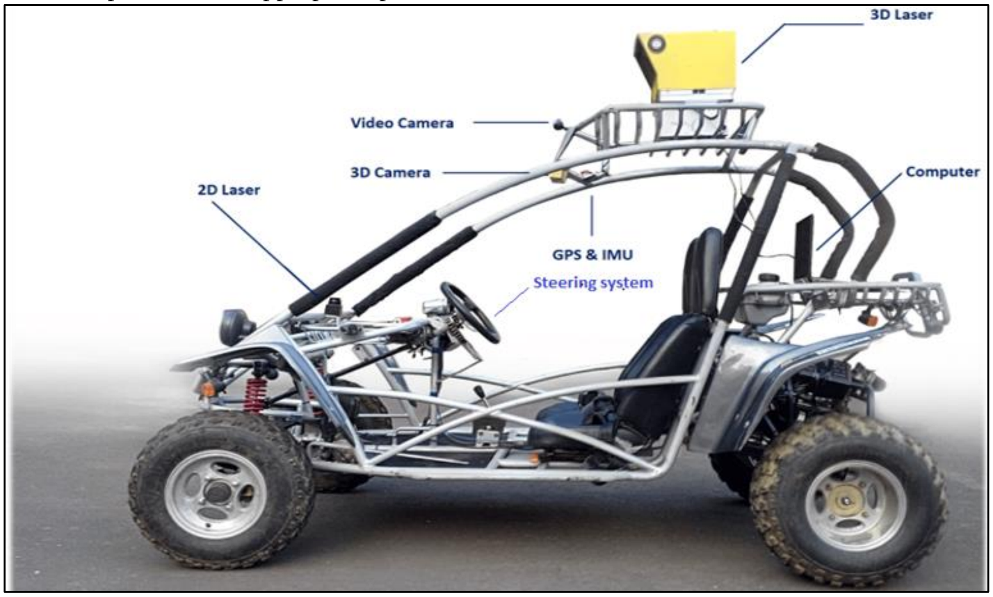

Fig. 2.1: Autonomous Vehicle

A. Ackermann Drive Mechanism

Ackermann steering geometry is an arrangement of linkages in the steering of a designed to solve the problem of wheels on the inside and outside of a turn needing to trace out circles of different radius. The intention of Ackermann geometry is to avoid the need for tires to slip sideways when following the path around a curve. The geometrical solution to this is for all wheels to have their axles arranged as radii of a circle with a common center point. As the rear wheels are fixed, this center point must be on a line extended from the rear axle. Intersecting the axes of the front wheels on this line as well requires that the inside front wheel is turned, when steering, through a greater angle than the outside wheel. The Fig 2.2 shows the Ackermann geometry and approximation of the system.

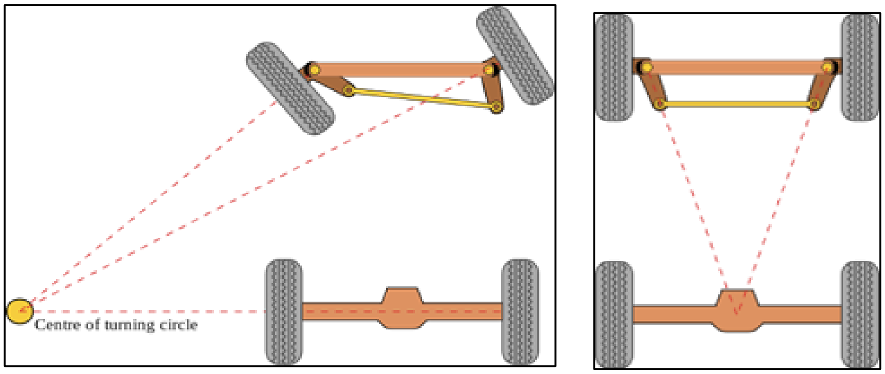

Fig. 2.2: Ackermann geometry

B. Ackermann Angled Steering Arms

The steering arms in the Fig 2.3 is angled inwards to create a means for the wheel angles to change at a different rate. This is the basis of the Ackerman Steering Principle and creates this unequal angular movement of the wheels. This unequal angular movement occurs because of the relative position of the steering arm pivot point (A) around how the steering arm pivot point moves around the king pin pivot point (B). The Fig 2.3 shows the Ackermann angled steering arms.

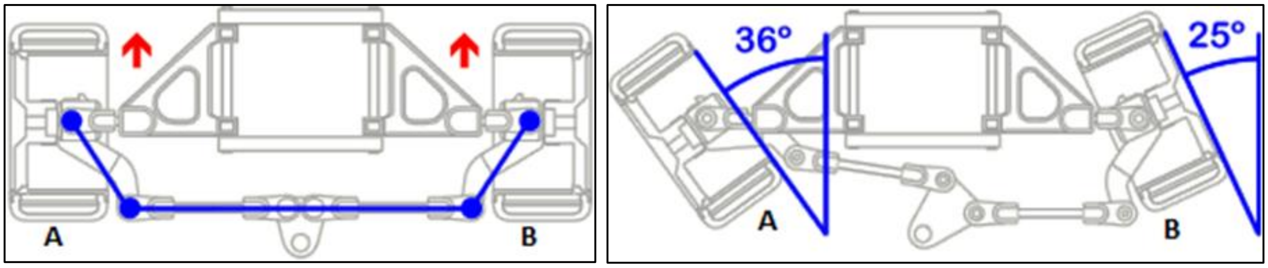

Fig. 2.3: a) Steering arms b) Front wheel steering angle

As the steering arms are angled, the pivot point (A) is not vertically aligned and is, in a straight ahead position, part way round the circle. Because of this, a Right movement of the steering arm will cause the pivot point to move a greater distance in the forward direction than a Left movement of the steering arm. You can reset the animation to its Starting position if required.

C. Odometry

Odometry is the use of data from motion sensors to estimate change in position over time. The change in position data are evaluated from encoders connected to rear wheel axle. The encoder output pulses are converted to distance travelled and are used for position evaluation. The orientation of vehicle is determined through steering wheel angle. The determination of steering wheel angle involves a mapping function between steering hand wheel and steering wheel. As steering hand wheel has 360 deg rotation and steering wheel can rotate at 25 deg alone as according to vehicle Ackermann drive mode.

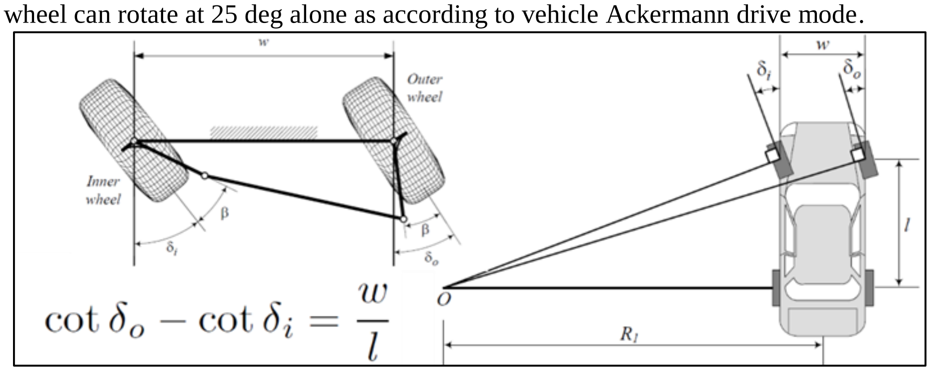

Fig. 2.4: Ackermann geometry

The figure 2.4 shows the Ackermann drive model. Consider a front-wheel-steering 4W S vehicle that is turning to the left, as shown in Figure 2.4. When the vehicle is moving very slowly, there is a kinematic condition between the inner and outer wheels that allows them to turn slip-free. The condition is called the Ackerman condition. Where, δ i is the steer angle of the inner wheel, and δo is the steer angle of the outer wheel. The inner and outer wheels are defined based on the turning center O. The distance between the steer axes of the steerable wheels is called the track and is shown by w. The distance between the front and rear axles is called the wheelbase and is shown by l. Track w and wheelbase l are considered as kinematic width and length of the vehicle.

A device that provides steering according to the Ackerman condition is called Ackerman steering, Ackerman mechanism, or Ackerman geometry. There is no four-bar linkage steering mechanism that can provide the Ackerman condition perfectly. However, we may design a multi-bar linkages to work close to the condition and be exact at a few angles.

To find the vehicle's turning radius R, we define an equivalent bicycle model, as shown in Figure 2.5. The radius of rotation R is perpendicular to the vehicle's velocity vector v at the mass center C. Using the geometry shown in the bicycle model, we have

$$R^2 = a^2 + R^2 \ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

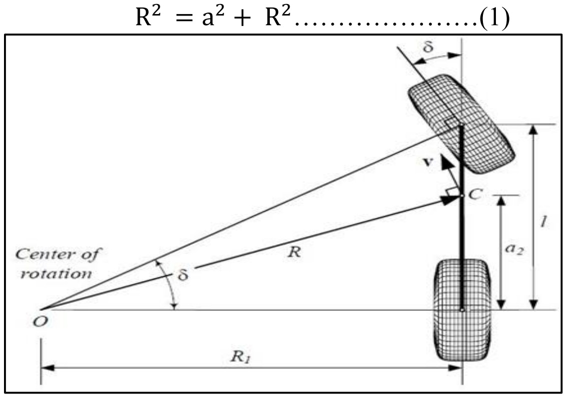

Fig. 2.5: Equivalent bicycle model of front wheel steering vehicle

III. DESIGN OF DRIVE-BY-MECHANISM

A. Drive-by-Wire Mechanism

Drive by wire (DbW), Steer-by-wire (SbW), or x-by-wire technology in the automotive industry is the use of electrical or electro-mechanical systems for performing vehicle functions traditionally achieved by mechanical linkages. This technology replaces the traditional mechanical control systems with electronic control systems using electromechanical actuators and human-machine interfaces such as pedal and steering feel emulators. Components such as the steering column, intermediate shafts, pumps, hoses, belts, coolers and vacuum servos and master cylinders are eliminated from the vehicle.

B. Steer-by-Wire Mechanism

The steer-by-wire mechanism is built with chain sprocket and idler systems. A steering encoder – Dan Foss Steering Angle Sensor Absolute (SASA) is connected to steering column shaft. It provides steering angle information. It is CAN data type output which is parsed and used as 360 deg hand wheel angle rotation. A feedback from steering motor is used for accurate steering wheel position and calibration to steering front wheel according to Ackermann Drive model. The Fig 3.1 shows the steer-by-wire mechanism implemented.

Fig. 3.1: Steer-by-Wire mechanism

C. Brake-by-Wire Mechanism

In the Brake-by-wire mechanism a dc motor is fixed to the brake pedal. The Brake pedal is connected to DC motor shaft by links and couplings. The DC motor is controlled in torque mode of Robotic motor controller, where the amount of current drawn by motor governs the brake pressure and as shown in Fig 3.2.

Fig. 3.2: Brake-by-Wire mechanism

A Brake-by-Wire system implemented with a single processor and an actuator per wheel can significantly increase the quality of the braking, in particular by reducing the stopping distance. Moreover, this technology provides more precise braking by adapting to the pressure the driver puts on the pedal. Like Steer-by-Wire, there is a significant decrease in the weight of the vehicle in removing the hydraulic braking system, and therefore significantly lower costs.

D. Throttle-by-Wire Mechanism

In throttle-by-wire mechanism a servo motor is fixed to carburetor. The servo motor is actuated using pwm signal from an embedded module. The Fig 3.3 shows the throttle by wire mechanism.

Fig. 3.3: Throttle-by-wire mechanism

Throttle–by-wire systems enables the control of the engine torque without touching the gas pedal, steer-by-wire systems allow autonomous steering of the vehicle, brake-by-wire systems delivers distributed brake force without touching the brake pedal, shift-by-wire systems enables the automatic selection of the proper gear.

IV. EMBEDDED SYSTEM ARCHITECTURE

The In-Vehicle PC is the upper-level control which runs navigation and planners for determination of path and sends steering angle and vehicle speed command to Low-Level control module – Microprocessor which takes care of actuators and sensors. Fig 4.1 illustrates the developed embedded system architecture.

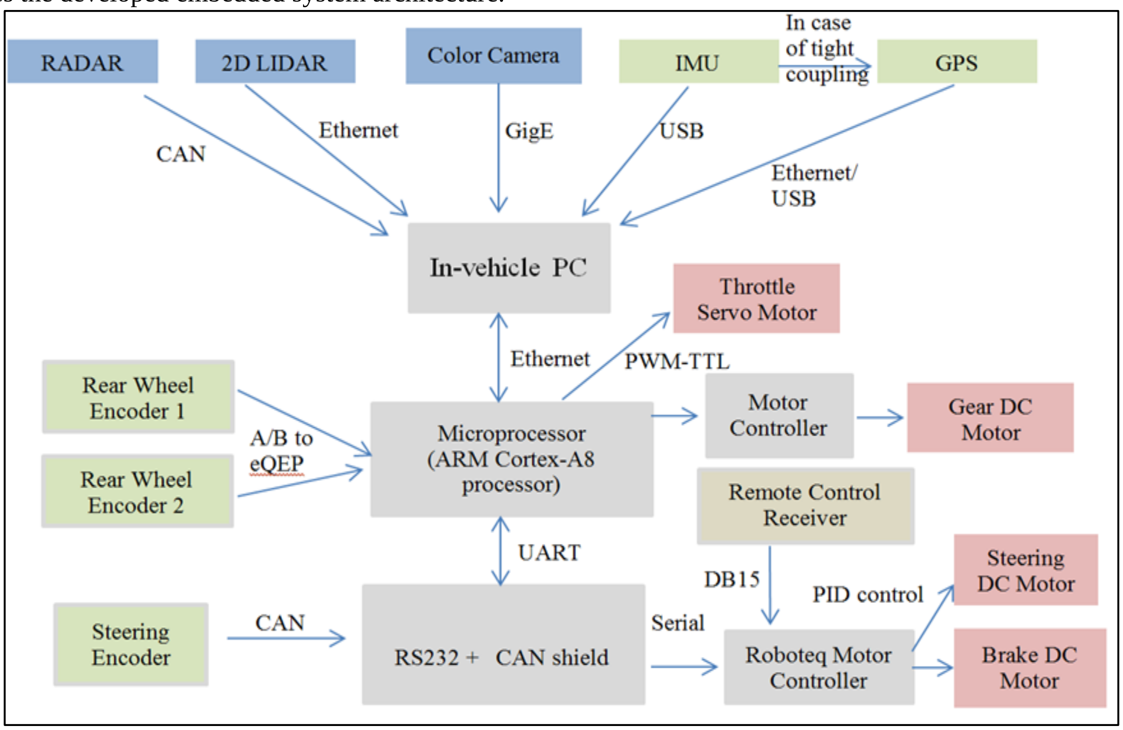

Fig. 4.4: Block diagram

A. *Intel Galileo*

Galileo is a microcontroller board based on the Intel Quark SoC X1000 Application Processor, a 32-bit Intel Pentium-class system on a chip. The Intel Galileo embedded module is interfaced to high CPR encoder of 3200 pulses per revolution. The Intel Galileo is programmed using its dedicated IDE to compute velocity from encoder pulses. Experimentation of interfacing and examining with encoder. Evaluating the velocity showed that at higher acceleration of vehicle and rear wheels rotating at high speed the output velocity data overflows and doesn't give output.

B. *Arduino Uno*

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. Experimenting with Arduino Uno interfacing the high CPR encoder. Output data overflow and switching of values randomly have been found out. It has been found out that interrupt capability of Arduino Uno is not dedicated module for the high CPR encoders. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

C. Beaglebone Black

The BeagleBone Black (B BONE-BLACK-4G) is a low cost, high-expansion focused BeagleBoard using a low cost Sitara AM3358 ARM® Cortex -A8 processor from Texas Instruments. BeagleBone Black has an Enhanced Quadrature output type encoder decoder module (eQep) which can seamlessly output velocity without overflow and data interruption. It has a support for connecting additional module to able to interface with serial port devices and CAN devices. Hence it is an appropriate module for the various elements in low-level control of Autonomous Ground Vehicle.

D. Roboteq Motor Controller

Roboteq's MDC22xx controller is designed to convert commands received from an RC radio, Analog Joystick, wireless modem, PC (via RS232 or USB) or microcomputer into high voltage and high current output for driving one or two DC motors. The controller features a high-performance 32-bit microcomputer and quadrature encoder inputs to perform advanced motion control algorithms in Open Loop or Close Loop (Speed or Position) modes. The MDC2230 used in this implementation, features several Analog, Pulse and Digital I/Os which can be remapped as command or feedback inputs, limit switches, or many other functions. The controller's two motor channels can either be operated independently or mixed to set the direction and rotation of a vehicle by coordinating the motion of each motor. Numerous safety features are incorporated into the controller to ensure reliable and safe operation.

The controller's operation can be extensively automated and customized using Basic Language scripts. The controller can be reprogrammed in the field with the latest features by downloading new operating software from Roboteq. The Roboteq controllers are fitted with many safety features ensuring a secure power-on start, automatic stop in case of command loss, overcurrent protection, and overheat protection.

E. Steering Control

The RoboteQ is the high current motor controller. It is used to control two DC motors connected to Brake and Steering. The Roboteq is a dedicated motor controller developed specially for the application of motor control. Using RoboteQ GUI – Roborun Utility various control configurations can be set. In the application of Steering control, the configuration Closed-loop position relative is used and in Brake control Closed-loop torque control are used.

In closed loop position relative mode the controller accepts a command ranging from -1000 to +1000, from serial/USB, analog joystick, or pulse. The controller reads a position feedback sensor and converts the signal into a -1000 to +1000 feedback value at the sensor's min and max range respectively. The controller then moves the motor so that the feedback matches the command, using a controlled acceleration, set velocity and controlled deceleration. This mode requires several settings to be configured properly but results in very smoothly controlled motion.

In closed loop torque mode of motor controller, the motor is driven in a manner that it produces a desired amount of torque regardless of speed. This is achieved by using the motor current as the feedback. Torque mode is mostly used in electric vehicles since applying a higher command gives more “push”, similarly to how a gas engine would respond to stepping on a pedal. Likewise, releasing the throttle will cause the controller to adjust the power output so that the zero amps flow through the motor.

F. Throttle Servo Motor

The throttle servo motor is interfaced to embedded module and controlled by pwm (pulse-width-modulation) of embedded module. The servo motor is a 180 deg high torque motor. The pwm pulses are determined by duration of pulses corresponding to angle of servo motor shaft. Rear wheel encoders are used for velocity feedback of vehicle. The rear wheel encoders are 5 V DC operated 3 200 pulses per revolution encoder. The encoders are interfaced to embedded module and upon processing of input pulses provided by encoder the vehicle velocity is computed by the embedded module. The RS232 + CAN cape module is connected to BeagleBone Black embedded module. It provides the embedded system to interface to devices that has serial port communication only available. In the software architecture the RS232 + CAN cape help to interface with RoboteQ motor controller and steering encoder.

V. SYSTEM INTEGRATION

A. Interfacing BBB with Rear Axle Encoders

Interfacing BeagleBone Black with rear axle encoders involved following sub tasks such as enabling eQEP modules in BeagleBone Black. The BeagleBone Black is an ARM based microprocessor. It is an operating system based file system management. As BBB can also run on Linux, the kernel is the communication between user code and System-on-a-chip (SOC). The Kernel of BBB is modified to 3.8 versions from factory default kernel to enable eQEP’s (enhanced Quadrature-Encoder

Pulse). The BBB has device tree overlays which are an indentification to find components of SOC. The kernel module acts as a driver to get the device tree working. To enabe eQEP, it is first patched and compiled along with kernel and enabled through device tree overlays.

B. Interfacing BBB with Steering Encoder

Interfacing BeagleBone Black with steering encoders involved following sub tasks such as interfacing with plug-in boards and retrieving data from BBB. The BeagleBone has support for CAN output type devices. Using plug-in boards called – cape, the BeagleBone is interfaced to steering encoder CAN output type device. The cape has EEPROM which automatically enabled CAN in device tree overlay of BBB whenever the cape is connected to BBB. The command ip link set is used to enable can0 type in BBB. The frequency of bits per second transmitted by the encoder device is 290kbps.The steering encoder CAN output data format is encoded in program and steering angle is provided as output.

C. Serial Cape for Beaglebone Black

This BeagleBone Serial Cape is the most flexible way to connect to other devices with a serial connection. This cape allows for simultaneous use of up to two RS232 ports and one isolated CAN bus, RS232, or one RS485.

In the application, the serial cape as shown in Fig 5.1(a) is used to interface with Roboteq motor controller through serial communication. Also in interfacing with CAN output type steering encoder. The Fig 5.1(b) shows the embedded module in the autonomous vehicle. Quickly and cleanly add an RS-232 serial level converter to your BeagleBone Black with this "micro" cape.

Fits completely inside the BeagleBone Black or Orange enclosure (sold separately), giving a sleek solution for BeagleBone serial port integration. Micro cape leaves all P8 and most of P9 I/O pins available for other applications. CBB-TTL-232 micro-cape is configured as a DTE (Device Terminal Equipment) similar to a standard PC. The DB9-male pin-2 is an input to the BeagleBone (RxD). The DB9-male pin-3 is an output (TxD).Changing configuration from UART0 (default) to UART4 requires soldering (see Manual for more details).

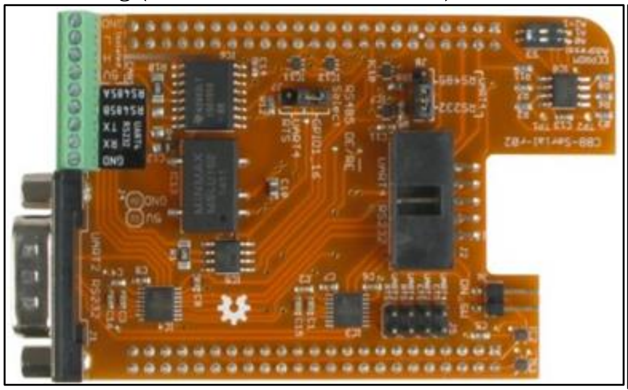

Fig. 5.1: (a) Serial cape

Fig. 5.2: (b) Serial cape in interface with tor controller

D. Interfacing BBB with Servomotor

The BeagleBone Black is interfaced to servo motor using vero board and connected to BBB through stackable pins. The function of vero board is to provide supply voltage for servo motor for it to operate. The throttle servo motor is interfaced to embedded module and controlled by pwm(pulse-width-modulation) of embedded module. The servo motor is a 180 deg high torque motor. The pwm pulses are determined by duration of pulses corresponding to angle of servo motor shaft. The position of the servo motor is set by the length of a pulse. The servo expects to receive a pulse roughly every 20 milliseconds. If that pulse is high for 1 millisecond or less, then the servo angle will be zero, if it is 1.5 milliseconds, then it will be at its centre position and if it is 2 milliseconds or more it will be at 180 degrees.

This example uses the PWM feature of the GPIO library to generate the pulses for the servo. The PWM frequency is set to 60 Hz so that the servo will receive a pulse roughly every 17 milliseconds. The length of the pulse is changed by adjusting the duty cycle over the fairly narrow range of 3 to 14.5 percent.

E. Power Circuit Board

In the design of circuit board for encoders and motor, the electrical specifications of BeagleBone Black are to be considered first. The BeagleBone Black is operated at 3.3 V stage. The encoders are operated at 5 V and a voltage conversion should take place to convert the encoder output to 3.3 V. The steering encoder operates between 9 V and 30 V is also powered. In the design of power circuit for encoder. The encoder operates at a voltage range of 5 V. In the embedded board using voltage regulators the 12 V source supply is converted to 5 V range to supply encoders. The BBB operates at 3.3 V and hence input pins should be provided at the specified voltage range. The encoders output which are also 5 V is converted to 3.3 V by voltage divider circuit made of appropriate resistor values. The 3.3 V output are then connected to BBB through stackable pins. In the design of power circuit for servo motor. The voltage regulator used for powering the encoder is connected parallel to second voltage regulator to able to provide approximately 2 Amps current. As two encoders and a servo motor can consume to approximately 1.5 Amps. The servo motor signal pin are connected to BBB pwm pin through stackable pins.

Servo motor has three wires. They are power, ground, and signal. The power and ground are connected to vero board. The signal is connected to BeagleBone Black pwm pin. The pwm device tree overlay is enabled using pwm device tree overlay file.

In the Fig 5.3 the power circuit is shown. In the power circuit board. There are stackable pins, voltage regulator IC's, capacitors and voltage divider circuit. The stackable pins are used to connect to BBB. The voltage divider circuit powers two encoders and a servo motor. The supply voltage is 12V and output voltage is 5V at a current of 2Amps.

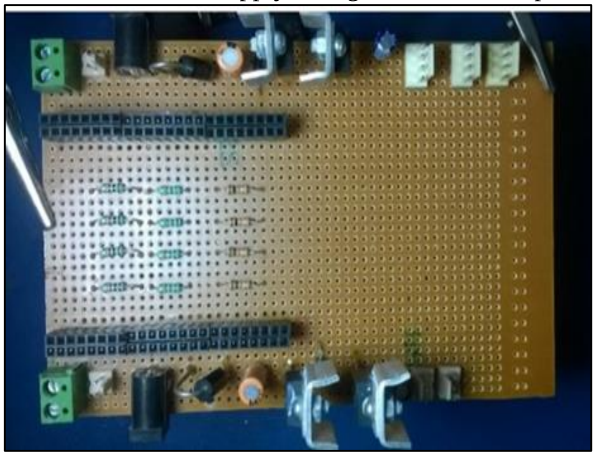

Fig. 5.3: Power circuit board

Fig. 5.4: Embedded system for Autonomous Ground Vehicle

In the Fig 5.4 the embedded system for low-level control is shown. The embedded system is placed on the front tray of vehicle. A 12V supply of battery is converted to AC power supply using Inverter. The power circuit board, BeagleBone Black and motor controller get power through the inverter.

VI. RESULTS AND DISCUSSION

A. Odometry Validation

Odometry is the use of data from motion sensors to estimate change in position over time. The change in position data are evaluated from encoders connected to rear wheel axle. The encoder output pulses are converted to distance travelled and are used for position evaluation. The orientation of vehicle is determined through steering wheel angle. The determination of steering wheel angle involves a mapping function between steering hand wheel and steering wheel. As steering hand wheel has 360 deg rotation and steering wheel can rotate at 25 deg alone as according to vehicle Ackermann drive model.

In the implementation the position x and y are determined from encoder connected to rear wheel axle. The orientation is determined through steering wheel angle. The steering wheel angle is provided by mapping function between steering hand wheel and steering wheel. As the vehicle is accelerated the odometry of vehicle i.e. the position and orientation of the vehicle are been published through the program.

The evaluation of odometry technique is the most reliable way to compute and measure vehicle position, coupled with sensors which inform the navigation system of the positions of obstacles or landmarks. Such systems often involve preprogramming the vehicle with information about the geometric layout of the environment, and the positions and characteristics of key landmarks.Fig.6.2 shows the validation of odometry data.

In the validation process of odometry data, the encoders of rear wheel which are computed to vehicle position data is published. The mapping function between steering hand wheel and steering wheel publishes the steering wheel angle. The combination of both the data provides the position values x and y, rotation value z.

```
T,1423991610.684283,dSt,-51.405064,cSt,-155.281693,stEnc,98.261688,rbq_val,105.394882,rbq_pid,-8.975969,D,-0.020581,enc1,-63,enc2,-236,distL,-0.008344,distR,-0.032819,dTrans,-0.000834,dR
ot,-0.010382,x,-0.018078,y,0.000036,theta,-0.224941
dLWheel,-0.001112,dRWheel,-0.000556,dTrans,-0.000834,dRot,-0.000181,pose_x,-0.018912,pose_y,0.000039,pose_theta,-0.235323
T,1423991610.821283,dSt,-51.405064,cSt,-155.281693,stEnc,98.261688,rbq_val,96.418915,rbq_pid,-8.975969,D,-0.021138,enc1,-75,enc2,-235,distL,-0.010012,distR,-0.032263,dTrans,-0.000834,dRo
t,-0.010382,x,-0.018912,y,0.000039,theta,-0.235323
dLWheel,-0.001669,dRWheel,0.000556,dTrans,-0.000556,dRot,-0.000121,pose_x,-0.019469,pose_y,0.000041,pose_theta,-0.242245
T,1423991610.960283,dSt,-51.405064,cSt,-155.281693,stEnc,98.261688,rbq_val,87.442947,rbq_pid,-8.975969,D,-0.022250,enc1,-91,enc2,-235,distL,-0.012238,distR,-0.032263,dTrans,-0.000556,dRo
t,-0.006921,x,-0.019469,y,0.000041,theta,-0.242245
dLWheel,-0.002225,dRWheel,0.000000,dTrans,-0.001113,dRot,-0.000242,pose_x,-0.020581,pose_y,0.000046,pose_theta,-0.256087
T,1423991611.098283,dSt,-51.405064,cSt,-155.281693,stEnc,98.261688,rbq_val,78.466980,rbq_pid,-8.975969,D,-0.024475,enc1,-108,enc2,-247,distL,-0.015019,distR,-0.033931,dTrans,-0.001113,dR
ot,-0.013843,x,-0.020581,y,0.000046,theta,-0.256087
dLWheel,-0.002781,dRWheel,-0.001669,dTrans,-0.002225,dRot,-0.000483,pose_x,-0.022806,pose_y,0.000056,pose_theta,-0.283772
T,1423991611.238283,dSt,-51.405064,cSt,-155.281693,stEnc,98.261688,rbq_val,69.491013,rbq_pid,-8.975969,D,-0.026144,enc1,-123,enc2,-259,distL,-0.016687,distR,-0.035600,dTrans,-0.002225,dR
ot,-0.027685,x,-0.022806,y,0.000056,theta,-0.283772
dLWheel,-0.001669,dRWheel,-0.001669,dTrans,-0.001669,dRot,-0.000363,pose_x,-0.024475,pose_y,0.000065,pose_theta,-0.304536
T,1423991611.376283,dSt,-51.405064,cSt,-155.281693,stEnc,98.261688,rbq_val,60.515045,rbq_pid,-8.975969,D,-0.027812,enc1,-135,enc2,-271,distL,-0.018356,distR,-0.037269,dTrans,-0.001669,dR
ot,-0.020764,x,-0.024475,y,0.000065,theta,-0.304536
dLWheel,-0.001669,dRWheel,-0.001669,dTrans,-0.001669,dRot,-0.000363,pose_x,-0.026144,pose_y,0.000074,pose_theta,-0.325300
T,1423991611.514283,dSt,-51.405064,cSt,-155.281693,stEnc,98.261688,rbq_val,51.539078,rbq_pid,-8.975969,D,-0.028925,enc1,-141,enc2,-276,distL,-0.019469,distR,-0.038381,dTrans,-0.001669,dR
ot,-0.020764,x,-0.026144,y,0.000074,theta,-0.325300
dLWheel,-0.001113,dRWheel,-0.001112,dTrans,-0.001112,dRot,-0.000242,pose_x,-0.027256,pose_y,0.000081,pose_theta,-0.339142
```

Fig. 6.5: Odometry validation

In Odometry test results, the hand wheel angle is set at 155 degree right and vehicle start position have been noted. In a completion of circle with 155 degree the odometry data's x, y positions are back to zero position and theta 360 degree has +/- 2degree.

In the results it has been found out that as the vehicle makes a circle the x, y and z values change and after completion of circle the position and orientation retain back to as zero as in initial start position state.

B. Implementation of imu for odometry evaluation

In the implementation of IMU, for yaw data the registers of IMU are accessed and are used for the evaluation of odometry. The following sub-topics explain about the implementation tasks. The Inertial Measurement Unit (IMU) is configured to get Roll, Pitch and Yaw values. The IMU serial cable is connected to BeagleBone Black cape serial port and the pitch, roll and yaw values are published. Using ROS the yaw topic is subscribed and used in odometry function as orientation of vehicle.

The IMU is initially calibrated with 2D calibration utility using IDE of IMU. Several visual odometry approaches use a non-uniform prior on the motion estimate to guide the optimization towards the true solution. This formulation allows us to choose a suitable sensor and motion model depending on the application. In particular, we propose to use a robust sensor model based on the t-distribution, and a motion prior based on a constant velocity model.

The relative rotation priors from the IMU are obtained by a fast integration of the high frequency gyroscopic measurements. Typically, common low-cost IMUs already accomplish this integration internally using a complementary filter along with the gravity direction obtained from the acceleration signals. Experience has shown that the resulting orientation of the IMU is only affected by a slowly changing drift term and that short-term relative orientation of the system can hence be recovered safely, directly from consecutive orientation information delivered by the IMU. The calibrated IMU provide accurate and precise roll, pitch and yaw which are useful in providing precise orientation of vehicle.

In the program for evaluation of odometry the orientation of vehicle is subscribed from the IMU yaw angle. For validation of orientation the vehicle is set to make a circle from and start position and after completion return back to same position. It has been found that orientation of vehicle was accurate and precise.

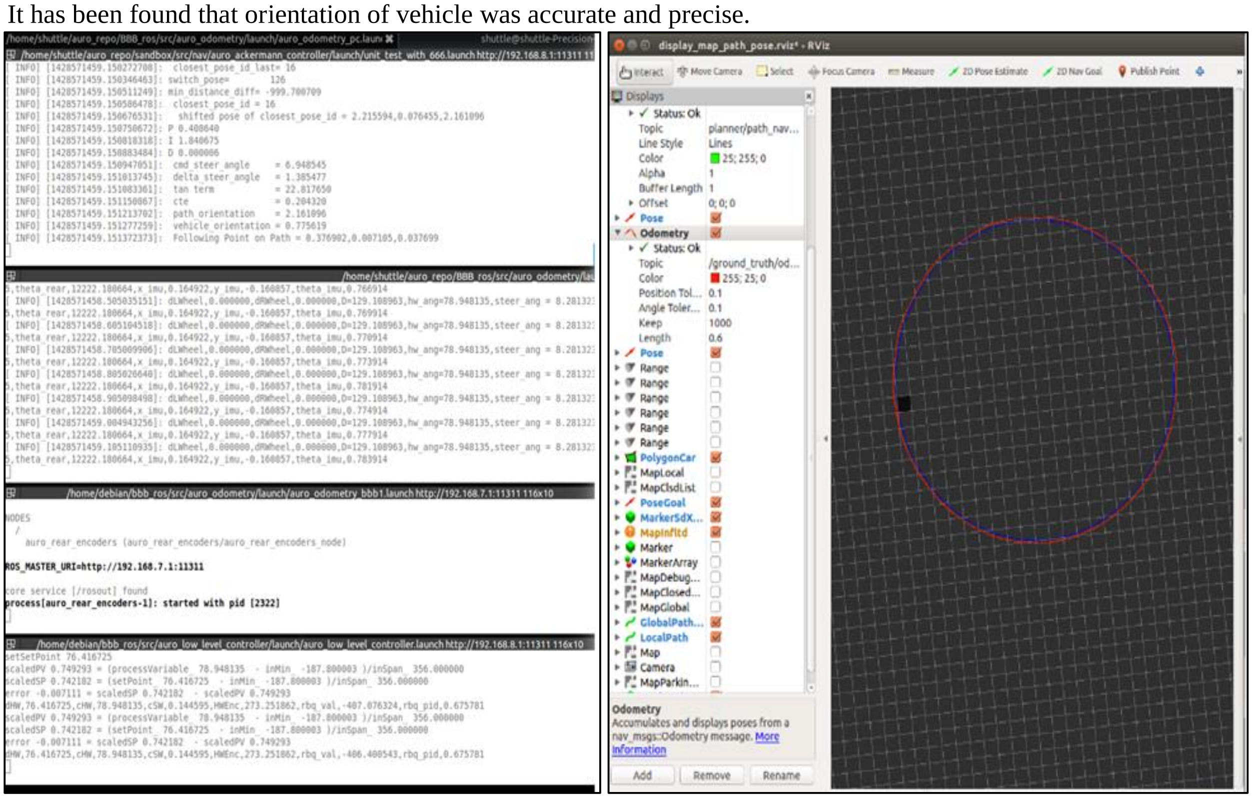

Fig. 6.6: IMU based odometry implementation

VII. Conclusion

This paper helped to gain a thorough knowledge on the interpretation of autonomous vehicles software and hardware architecture. Process study was a part of the odometry evaluation phase, which helped to learn the various operations involved in the determination of autonomous vehicle odometry, and also identify the various techniques involved.

Design of Drive-By-Wire mechanism (DBW) was also an essential step in the paper. Several conceptual designs were proposed before finalizing the ultimate design of the DBW systems. This endeavor helped understand the various parameters to be considered in designing a mechanism for steering, brake and accelerator. Circuit design and building is indispensable for implementation of low-level control. Through this study, a better knowledge of embedded modules was obtained. This exercise helped to aggravate the interests in research and analysis of emerging trends in autonomous ground vehicles. With robust and sophisticated embedded module, the paper on the implementation of low-level control in autonomous ground vehicle will prove to serve in real-time applications in automotive industry. Future developments can be made with sophisticated and rugged embedded module, using Electronic Control Unit (ECU) for autonomous ground vehicle.

REFERENCES

[1] Reza N.Jazar., "Vehicle Dynamics – Theory and Applications", Vol. 2, Issue 2 (2008) ISSN: 2249-3905, pp. 377-454
[2] Campbell, Stefan F. (Stefan Forrest), "Steering control of an autonomous ground vehicle with application to the DARPA Urban Challenge", Massachusetts Institute of Technology, February 2007, pp. 72-85
[3] K J Astorm, "Vehicle/Bicycle Dynamics," Lecture notes on vehicle dynamics, vol. 58, 2011, pp. 347-352
[4] Saeed B. Niku, "Introduction to Robotics: Analysis, Systems, Applications", Pearson Education, Inc., 2003
[5] Prof. Dr. Hans P. Geering, Prof. Charles Brom, "Parkmate enhanced odometry", Swiss Federal Institute of Technology Zurich, Winter 2004/05, pp. 72-85
[6] Reuben Strydom, Saul Thurrowgood, Mandyam V. Srinivasan, "Visual Odometry : Autonomous UAV Navigation using Optic Flow and Stereo",Summer 2005/06.
[7] Gurdal Arslan, Jason R. Marden, Jeff S. Shamma, "Autonomous Vehicle-Target Assignment: A Game-Theoretical Formulation", Winter 2006/06,pp 01-05
[8] Todd Litman Victoria Transport Policy Institute, "Autonomous Vehicle Implementation Predictions Implications for Transport Planning", pp 06-08
[9] Kurt Dresner Peter Stone – "Sharing the Road: Autonomous Vehicles Meet Human Drivers",Pear publications, 2005-06, pp 08-09
[10] Carnegie Mellon University, General Motors Research and Development, Continental AG, Intel Research, "Autonomous Driving in Urban Environments: Boss and the Urban Challenge", 2012, pp 05-09
[11] L. Conde Bento, Urbano Nunes; Fernando Moita and Antonio Surreci, "Sensor Fusion for Precise Autonomous Vehicle Navigation in Outdoor Semi-structured Environments",Winter 2008/06, pp 06-08