Engineering and Technology Journal e-ISSN: 2456-3358

Volume 09 Issue 01 January-2024, Page No.- 3381-3391

DOI: 10.47191/etj/v9i01.19, I.F. – 7.136

A Comprehensive Review on Creating Autonomous Car Systems

Namareq Odey[1], Ali Fadhil Marhoon[2]

[1,2]Department of Electrical Engineering, University of Basrah, Basrah, Iraq

ABSTRACT: Autonomous vehicles are around us and are finding trillions of recent developed applications, starting from driverless cars to automatically observation in critical areas. Continual progress in these technologies within the earlier decades makes these inventions possible. However, the planning of these technologies which must be surmounted, to apply efficient, useful supremely important and safe tasks of those independent units are daunting as well as numerous. Over the past decade, many researches are published within the domain of autonomous vehicles. Yet, most of them concentrate only on a selected technological area, such as vehicle control, visual environment perception ... etc. During this paper we present a brief yet comprehensive overview on the most significant key ingredients of autonomous cars, it covers almost all bases, from available models and types, their functions, importance, in addition to the most important related work.

KEYWORDS: Self-Driving Cars, Sensing, Localization, Object Tracking, Path Planning

I. INTRODUCTION

Autonomous cars (ACs) are popularly referred to as self-driving (SDCs), an autonomous vehicle, or robot car. The goal of an ACs is to automatically drive with absence of a driver interfering. ACs aim to progress everything beginning from road safety arriving into universal mobility, taking in consideration reducing the prices of driving. The whole concept autonomous cars on its own could seem complex, but the thought behind it's quite simple and lies within the remit of current technology.

It can be achieved by applying the computing power, and mixing it with reliable sensors, intelligent algorithms are also can be considered as other components [1]. Towards providing clearness levels' field within SDC industry; organizations like US Department of Transportation's National Highway Traffic Safety Administration (NHTSA) as well as German Federal Highway Research Institute (BASt) have declared definitions respecting SDCs supported their automation degree. One among the foremost references that commonly used is J3016 Driving Automation Taxonomy [1-4]; which is published in International Society for Automotive Engineers (SAE), in which six levels concerning autonomous vehicles are described, starting from no automation to full automation.

Beginning with Level 0, the driving force monitors and performs all driving tasks' aspects. Level 0 autonomy might contain providing warnings to the driving force, but the vehicle might not hold any reaction. While Level 1 supplements the functions of driver assistance, which enables the system to acquire either lateral (steering) or longitudinal (acceleration/deceleration) control. Moreover, it supports the vehicle driving force which concerns controlling almost all vehicle features. They appear in the least the encompassing environments, acceleration, braking, and steering.

On the other hand, Level 2, or Partial Automation, concerns the system taking up in both lateral and longitudinal control in well defined circumstances. The driving force remains required within the vehicle itself in order to monitor environmental factors besides critical safety tasks. Furthermore, in Level 3, which is known as Conditional Automation, the vehicle could perform all environmental monitoring by means of sensors. More specifically, the vehicle itself could drive in autonomous mode for certain situations; nevertheless, driving force would be able to take over during the situation in which the vehicle exceeds the control limit. Moreover, Level 4 is High Automation. During this autonomy level, vehicle will control the tasks of brakes, steering, in addition to vehicle acceleration. More particularly, it will monitor the vehicle itself as well as pedestrians, and the whole highway. However, the person is still needed in uncontrollable situations, like congested places in streets and cities.

Finally, in Level 5, a complete automation can acquire, which means no human driver is needed; the vehicle autonomously will control all critical functions like brakes, pedals, steering, and the whole environment, it will identify and react to every unique driving condition like traffic jams. Autonomous driving does not represent a single technology, but rather occupy a high complex system consists from several subsystems. It might be divided it into three main components which are: algorithms, which involve sensing, perception, as well as decision making which is required in for critical cases; in addition to client systems, which comprise the software as well as the hardware platform; and finally the cloud platform, which include high definition (HD) maps, simulation, deep learning model training, and data storage.

However, most review literatures on autonomous vehicles concerns only the algorithms of just one technology branch in SDCs, while the present paper provides a comprehensive review for the most technologies used in the end to end implementation for SDCs. This review provides a leading guide for students, researchers, or any person interested with the field of autonomous technologies.

II. CREATING AUTONOMOUS VEHICLE SYSTEMS

Autonomous vehicle systems are very complex, involving three essential subsystems [1], which are:

- Algorithms for localization, perception, planning as well as control.
- Client subsystems, like robotic software in addition to the hardware.
- Cloud subsystem, where simulation, data storage, deep learning model training and high-definition (HD) mapping are incorporated.

More specifically, the algorithm subsystem takes out information from sensor data from the vehicle's environment and takes decisions about vehicle's behaviors. On the other hand, the client platform combines these algorithms to satisfy reliability and real time requirements, whereas the cloud subsystem affords offline computing and storage requirements for SDCs. In this review we are focusing on Algorithm subsystem, which is briefly explaining the theories and technologies used in each part of it in next sections.

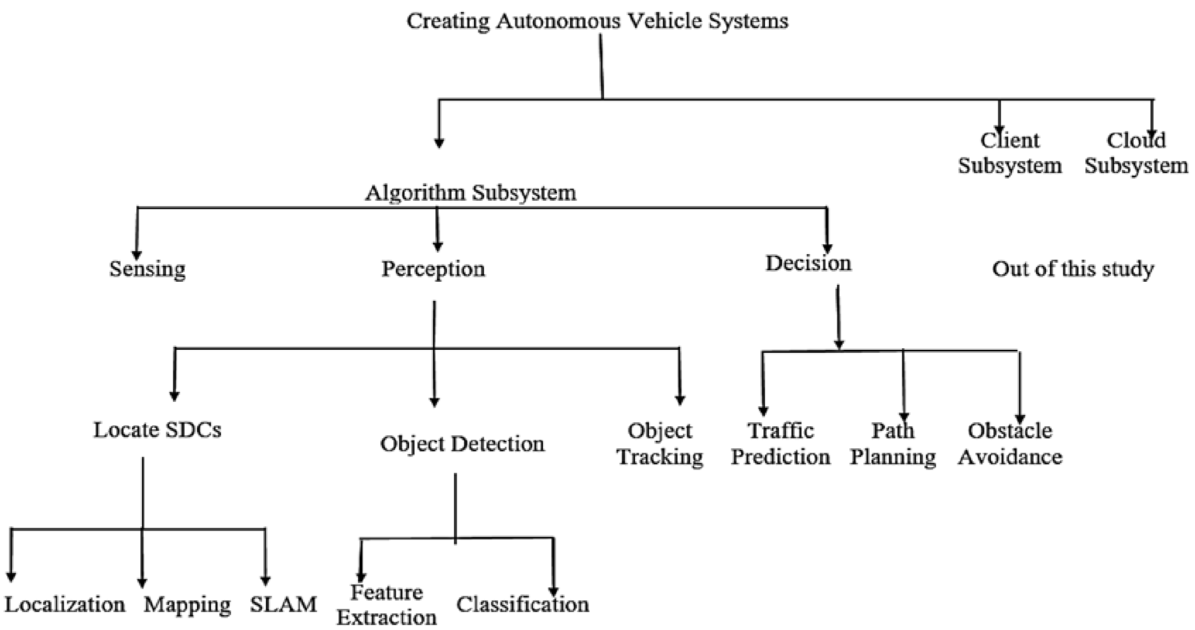
extraction of information from sensor data. Perception; which is the vehicle's awareness of its surroundings. In addition to decision, which is the execution of actions to reach destinations safely [1].

However, sections below discuss in details the algorithms subsystem employed; in terms of sensing, perception including their three functions which are locating SDCs, object detection, and object tracking) with subsections of the most common methods used for each function, in addition a subsection of the latest researches used to improve each technology branch are summarized at the end of each section.

V. SENSING

SDCs comprise various significant sensors, since each sensor have benefits as well as weaknesses. The information from various sensors need to be linked to amplify reliability together with security [1, 2, 3, 5]. Some of sensors used in SDCs are obtained below:

- Global Positioning System/Inertial Measurement Unit (GPS/IMU): This type of sensors assists the SDCs to locate themself via declare some inertial modifications as well as position approximations at a high rate, for instance, 200 Hz. GPS may be precise localization sensor, yet its update rate is belated, with approximately 10 Hz, therefore unable of giving real-time updates. In other hand, IMU errors collected over tir **Figure1** Algorithm Subsystem Structure in Cre position estimatio times, at or above 200 Hz. By a combination of both GPS and IMU, precise as well as real time updates for SDCs localization are often provided.
- Light Detection and Ranging (LiDAR): It is employed for localization, mapping, as well as obstacle



III. ALGORITHM SUBSYSTEM

The algorithms in creating autonomous vehicle systems as depicted in Figure 1, consist of sensing; which is the

avoidance. It operates through reflecting a beam off surfaces and computes the time of reflection in order to find the distance. As a result of its great preciseness, LiDAR could be employed to create HD maps, in order to recognize obstacle, to assign a moving vehicle against HD maps, etc. a LiDAR unit, like Velodyne 64-beam laser, rotates at 10 Hz as well as takes approximately 1.3 million readings per second.

- Cameras: Cameras are generally employed for object tracking as well as object recognition missions like lane detection, pedestrian detection, and traffic signal detection etc. to improve SDCs safety, current realizations most often mount about eight 1080p cameras round the car. The cameras are usually working at 60 Hz, and generating about 1.8 GB of data per second.
- Radar and Sonar: This system is usually utilized in the final line of defense for obstacle avoidance. The info produced via this system indicates the distance and velocity from the closest object ahead to the path of the vehicle. Upon detecting an object not far ahead, then SDCs should utilize the brakes or turn to ward off the obstacle. Thus, the info produced through this system doesn't need lots of processing and typically is delivered to the control processor.

VI. PERCEPTION

The data collected from sensor system is imparted to the perception step in order to afford a comprehending of SDC’s environment. Three major functions are included in this stage. First locate SDCs on their maps, this is done through three different techniques namely: localization, mapping and the commonly used and more efficient is the Simultaneous Localization and mapping SLAM method. The second and third functions are object detection and object tracking.
All of these functions are explained in detailed below:

A. Perception First Function (Locate SDCs):

The purpose of locate SDCs function is to provide complete as well as accurate consideration of the vehicles dynamic environment as possible in to provide a basis for decision making. SDCs locating can be composed in to three major sub functions, which are Localization, Mapping as well as Simultaneous Localization and Mapping. Each one of these techniques has its own advantages and disadvantages as depicted below.

1) Localization

Localization is the procedure of finding out the SDC’s position and orientation found on a map. Localization depends deeply on sensors like GNSS, odometry, LiDAR, IMUs or cameras. Ideal tactics of localization depend on either scan matching, where SDC correspond its view of the environment with the map, or on dead-reckoning, where SDC employ its knowledge of heading, time and speed to trace from a known location on the map to a replacement location [2,3,6,7] some of the most localization procedures are briefly depicted below:

- Localization Using GNSS: Global Navigation Satellite Systems (GNSSs) are a conventional localization method as they provide an uncomplicated as well as economical procedure for SDCs to localize. In this method the concept of trilateration to find the SDC’s absolute position is utilized. But this method demands line-of-sight with no less than three satellites, thus it is not convenient for various operating domains where the satellites are concealed, e.g., tunnel, urban canyon, indoors, etc. An additional week point is the comparatively low accuracy. The accuracy can be enhanced using Real-Time Kinematic (RTK) or Differential-GPS base stations [2,3,6,7].
- Localization Using Wheel Odometry: Wheel odometry localization is a localization method that uses wheel sensors as well as heading sensor. The process of localization is achieved by utilizing dead-reckoning, an uncomplicated procedure employed in obsolete sea navigation. This approximates the SDC’s position depend on the distance traveled corresponding to a recognized initial point as well as projected direction. This method acts in any active environments. This method has issues caused by wheel slip, inconsistent road surfaces, etc. thus, the localization determination is usually used in the short term to recover for momentary inconvenience of other localization methods [2,3,6,7].
- Localization Using INS: Inertial Navigation System (INS) localization is a localization method that doesn't demand any exterior sources. INS localization is depending on utilizing the dead-reckoning method on rotation and motion and measurements afforded by an IMU (inertial measurement unit), which usually contains gyroscopes, accelerometers as well as magnetometers. INS localization usually offers extra precise pose approximation than wheel odometry, but it is up until now affected by cumulate errors, so it requires to be modified occasionally by other localization methods [2,3,6,7].
- Localization with External References: Additional technique to realize SDC’s localization is by connecting extra auxiliary devices or infrastructure with the functional environment this infrastructure can be taken form passive devices, like visual markers and magnets, or active devices, like Bluetooth and Wi-Fi. This technique is generally utilized for indoor environments. Though, enhancing infrastructure is rarely possible, that makes this technique inconvenient to work in wide areas [2,3,6,7].
- Localization Using LiDAR: LiDAR localization technique utilizes ‘natural ‘landmarks, like, walls, buildings, and trees which is found in the working environment, this method is more convenient for large areas, in situations where additional infrastructure is too pricey, or unfeasible This technique can be utilized in global as well as local localization. The operation is typically achieved by doing scan matching, this method tries to find the geometric alignment of two scans. The geometric alignment is related to the SDC’s rotation and translation. By tracking the rotation as well as translation, the pose can be approximated by increasing the consequent pose variation from the initial

point. In case of the global localization, scan matching approaches also can be used to recognize loop closures [2,3,6,7].

2) Mapping

The localization procedures work on the hypothesis that a faultless map is existing in advance. The preference of map category relies upon various factors, include: the sensors used, processing power of the computing platform, the localization technique used, the memory, etc. Mapping is the procedure used to build a precise depiction of SDC environments. This may be depending on a local map or on global map. The map must be precise and accurate so that the SDC be capable to work safely. There are three common models of map: Occupancy Grid Maps, Relational Maps as well as Feature Maps [3,6,7].

- Occupancy Grid Maps: Occupancy grid maps are without doubt the most common sort of maps in SDCs as well as robotics. This technique discretizes the surrounding into a grid. Each cell in the grid comprises the probability of occupancy of either occupied or not based on their generic depiction, this technique is also a common map option in case of multi-sensor data fusion.
- Relational Maps: This method determines the relation between the elements of the surrounding. One common example of this method is the pose-constraint map. In which, the elements in the map are the vehicle poses, i.e., location as well as heading, which are represented using a graph elements (nodes) and joined to each other by edges. These edges perform the spatial constraints between the poses, typically depend on odometry calculations.
- Feature Maps: This method can be also called landmark maps, comprise physical elements, and their locations in the surrounding. Maps depend on features have more compact representation because of their higher abstraction level, and they are more robust to small alterations in sensor observations. Furthermore, choosing the correct features might be challenging. And, implementing feature extraction as well as matching online increases computational overhead.

3) Simultaneous Localization and Mapping (SLAM)

The simultaneous localization and mapping (SLAM) is used when a vehicle has neither an accurate map nor an accurate location. These algorithms aim to build a map of the surrounding at the same time as trying to locate the vehicle within that map. If SLAM algorithm recognizes it has returned to a location it already visited, then it performs loop closure. The main forms of SLAM are the filtering approach, utilizing either Kalman or particle filters, and the optimization approach, utilizing either graph-based techniques or bundle adjustment [8,9].

➢ Filtering Approach: The filtering approach points to a technique that approximates the current unknown state depend on past observations. It iteratively updates its internal belief by combining new observations. There are two significant variants of this approach: The Kalman filter as well as the particle filter [3].

- Kalman Filter

Kalman filters are a category of Bayesian filters that assume all noise in the system is Gaussian. The Kalman filter is a recursive procedure with two prime steps: the prediction step as well as the update step. The prediction step utilizes the recent state estimate as well as the error covariance estimate from the preceding iteration (or initial estimate) to calculate the predicted state estimate and predicted error covariance. The update step involves editing the predicted state estimate measured in the preceding step by taking the recent measurement into consideration to produce an updated state estimate as well as updated error covariance estimate. The Bayesian filtering method is a probabilistic technique that utilizes the recursive Bayesian inference structure to predict the unknown probability distribution. Two of the most popular Kalman filter approaches used in SLAM algorithms are the Extended Kalman Filter (EKF- SLAM) as well as the Unscented Kalman Filter (UKF-SLAM). The Extended Kalman Filter (EKF) together with Unscented Kalman Filter (UKF) is utilized to solve systems that include non-linear motion and/or measurement models. This makes them ideal to be used in SDCs, since most sensors utilize polar coordinate system (angle and distance), that introduces non-linear terms when converted to Cartesian coordinates (x, y, z). EKF estimates the non-linear probability distribution by using linearization depend on the Taylor first order expansion. On the other hand, UKF is depending on the Unscented Transformation (UT) that utilizes a group of specially chosen weighted points, i.e., sigma points, to implement the unknown probability distribution. EKF-SLAM together with UKF-SLAM is used to solve the SLAM problem. Because of their structures, both methods are useful for utilizing with landmark-based maps, as well as consume almost the same computational time. Yet, both methods are less useful for large-scale maps since the complexity boosts as the number of landmarks boosts [3,9].

- Particle Filter

The Particle Filter (PF) is based on a technique called Sequential Monte Carlo (SMC), which estimates an unknown probabilistic distribution as the sum of weighted samples or 'particles ', drawn randomly from a proposal distribution utilizing the sampling technique each particle in the particle filter comprises a hypothesis of the map as well as vehicle pose. For each iteration, the estimated map as well as the pose of each particle is edited with respect to the vehicle's motion model as well as sensor calculations. The weight of each particle is also reassigned depend on the observation likelihood. Yet, the PF suffers from the dimensionality, due to the number of particles are grows exponentially with the system dimension A way to solve this issue is to minimize the size of the state-space. One standard approach to do this is by utilizing the Rao- Blackwellization method to the PF, another approach to improve PF performance is by utilizing a better proposal distribution. A proposal distribution is defined as optimal if it minimizes the conditional alterations of the importance weights. One way to achieve this is to incorporate recent observations in the proposal distribution. The optimal proposal distribution needs fewer particles to do the same performance as the standard approach, and makes the algorithm more robust to large motion uncertainty [10-15]

➢ Optimization Approach: The optimization procedure works by using a smoothing concept. Which means that all measurements as well as poses from the beginning till the current observation are utilized to find the most probable overall trajectory. Since all previous observations are taken into consideration, the optimization technique is a solution to the full SLAM problem. Methods depend on the optimization methodology are usually implemented by two processes: front-end as well as back-end, the front- end process, which is sensor-dependent process, is responsible for excerpting the features from sensor data, as well as performing data association. While the back-end process is responsible for determining the optimal configuration that is consistent together with all observations. This process is called solving Maximum a Posteriori (MAP) estimation problem [16].

• Graph-Based SLAM: This method treats the SLAM issue through creating a pose- constraint graph also determining the configuration which is consistent with the graph. Every node within the graph performs a vehicle pose, and is joined to another node by an edge that performs the spatial constraints between the poses. Sensor-specific and front-end component are used to create the graph and perform data association. The back-end component resolves the estimation through utilizing some non-linear square techniques, like Gauss-Newton or Levenberg-Marquardt methods [16-17].

• Bundle Adjustment (BA): (BA) is a visual reconstruction method that seeks to optimize the 3D framework as well as the parameters named (pose and/or calibration) [18]. BA is modeled as non- linear least-squares case in which the goal is to determine the optimal configuration, which reduces a cost function. Re-projection error, is one of the popular cost functions used. Whose can be determined as the difference between the predicted 2D projection and the recognized feature position for every 3D corresponding point on the image plan.

4) Locate SDCs Related Works:

Yanlei Gu et al [19] displays a precise vehicle self-localization system as well as assesses the suggested system in various urban environment classes. The improved system uses a global navigation satellite system (GNSS) positioning technique as the key method. On the other hand, Xiaoli Meng et al [20] enhance the localization accuracy by using the UKF-based GNSS/IMU/DMI fusion method. point cloud-based lateral correction is also proposed. The proposed method represses falsely-detected curb points using the RANSAC procedure. While Xingxing Guang et al [21] integrate the Inertial Measurement Unit (IMU) data together with the line feature parameters from a camera to enhance the navigation reliability. The experimental results explore that the suggested method can improve the stability, accuracy, as well as reliability of the navigation system.

Furthermore, Khaoula Lassoued et al. [22] propose two localization methods sharing GNSS errors. The proposed approach depend on set inversion technique with constraint propagation offers a significant improvement in terms of accuracy as well as confidence domains associated with usual standalone techniques. While Miguel Ángel et al. [23] offer an algorithm concerning localization of autonomous vehicles by using Monte Carlo Localization (MCL) scheme, and improving the particles' weights through the addition of Kalman filtered Global Navigation Satellite System (GNSS) information. The proposed scheme is evaluated by means of KITTI dataset evidencing that it enhancing localization with respect to classic GNSS and Inertial Navigation System (INS) fusion as well as Adaptive Monte Carlo Localization (AMCL), further, it is tested also in the platform of autonomous vehicle in the Intelligent Systems Lab of University Carlos III de of Madrid, giving qualitative outcomes.

Moreover, Giovanni A. Santos et al. [24] proposes an enhanced localization framework for autonomous vehicles by using tensor together with antenna array depend on GNSS receiver. This method provides a positioning five times more precise than state-of-the- art single antenna depend on GNSS receivers, minimizing the positioning error from 149 cm to 30 cm. Wherease Uche Onyekpe et al. [25] suggest a Neural Network embedded algorithm to enhance the localization of autonomous vehicles as well as robots alike in challenging GPS deprived surroundings. By approximating the displacement as well as the orientation rate of the vehicle within a GPS outage period, this method overtakes the INS in all examined scenarios, via giving up to 89.55% enhancement on the displacement estimation besides 93.35% on the orientation rate estimation. On the other hand, Xieyuanli Chen et al. [26] utilize 3D LiDAR scans in order to solve the dilemma of autonomous cars localization in a large-scale map at outdoor environment. The proposed approach is tested on various datasets in order to verify the accuracy and reliability of the given scheme in locating an autonomous car in various environments and work online at the LiDAR frame rate to follow the vehicle pose.

B. Perception Second Function (Object Detection)

The target of perception is to sense the dynamic surroundings of SDCs as well as build a trusty and comprehensive framework of it, depend on sensory data. for the SDCs to be intelligent as well as safe, perception units should be capable to recolonize dynamic objects such as pedestrians, other vehicles and cyclists, and to realize static objects like lane boundaries, traffic signs, road surface as well as lights, to trace dynamic objects in 3D, etc. as all the following driving planning, decision, as well as control relies on accurate perception output, its significance cannot be overstated. One of the primary skills that an SDC demands is object detection. This is high-priority for any SDC to be capable to safely movement because it permits it to detect dynamic and static objects. Object detection comprises of processing the image, extracting features within it as well as classifying those features in order to build a semantic framework (or map) [3,4,27]. The common used feature extraction methods are: Scale-Invariant Feature Transform (SIFT), Histogram of oriented gradients (HOG), as well as Maximally Stable Extremal Regions (MSER). Where Classification process can be achieved using Random Forest (RF), Support Vector Machine (SVM), Artificial Neural Network (ANN) or other machine-learning techniques.

1) Feature Extraction

The major task of object detection lies in designing feature descriptors that produce each class distinguishable from others [3]. Some of the majority techniques used in feature extraction depicted below:

- Scale-Invariant Feature Transform (SIFT) [28]: performs an image as a collection of invariant key points, i.e., features within an image that are never changing of rotation, scale, translation, illumination as well as other viewing conditions. For every key points, a vector of 128 numbers is measured as its fingerprint by using a histogram of orientations as well as gradient magnitudes then, the fingerprint can be compared with a set of previously defined fingerprints to find if they are identical.
- Histogram of oriented gradients The (HOG) descriptor [29]: utilizes the histogram of edge directions or intensity gradients, as illustration of local object shape and appearance. this technique divides the image for cells, and a histogram of for every pixel within the cell is measured. Usually, the normalization of the contrast of local histograms is performed by using the average intensity value over a block of connected cells, then all local histograms are connected to produce the final descriptor.
- Maximally Stable Extremal Regions (MSER) [30]: is a detection technique that operates by detecting the property variants of a block of pixels relative to its environments. MSER characterizes an image as regions that are maximally unchanged, despite the intensity do change. each region is generally presented by an ellipsoid fitted to the actual shape. Compared to the previous techniques MSER is faster and unchanged to affine transformations, like skewing.

2) Classification

The eventual step of the object detection process is to classify the features extracted within the previous steps into a group of predefined classes, as 'car ', 'pedestrian ', 'truck ', etc. Usually, the classification process is executed by a machine-learning procedure [3]. Some of the extensively used classifiers contain random forest (RF), support vector machine (SVM) as well as artificial neural networks (ANNs).

- Random Forest: RF [31] is a combination of several decision trees that are automatically produced by random chosen of data as well as feature subsets. The classification result is produced by taking the majority results of all the decision trees. RF is more robust to overfitting as it comprises the model with random noise. It also has lower variance because of the trees averaging effect.
- Support Vector Machine (SVM): (SVM) [32] can be considered as one among the foremost conventional and effective techniques for classification, which intends to find out a separating hyperplane which distinguishes collections of various class labels. In most cases, it might be not possible to disjoin the classes utilizing a linear function. yet, in that case a highly dimensional plane could be used. so, by using of some kernel functions or non-linear mapping, the input data is converted into a high-dimensional space, then classified based on the separating hyperplane.
- Artificial Neural Network (ANN): (ANN) is a system consists of a multi-layered of interconnected neurons. Nonlinear classification usually utilizes a specific sort of ANN, which is the multilayer perceptron (MLP). An MLP includes not less than three layers namely: (input layer, hidden layer as well as output layer), each feature can be represented by a neuron at the input layer. the multilayer perceptron is trained by using the backpropagation method [3], which frequently updates the weight of each neuron in the forward and backward directions till the right classification achieved. In forward stage, the input data is multiplied together with the weight of the neuron then some activation functions applied after that, the output of the current layer is propagated to the following layer till the result from the final layer is calculated. The error between the current and the desired result is calculated, and then the weight of each neuron is updated from the output layer to the input layer to decrease the error.

C. Perception Third Function (Object Tracking)

It is aim is to approximate the state of objects like object speed, object acceleration together with object location ...etc. SDCs have to track numerous traffic objects to maintain safe distance as well as predict object trajectories, which are difficult to find in case of dynamic objects. particle filter is a common filter which is utilized for object tracking [33], tracking by detection is another common technique used for object tracking. In this technique the object detector is utilized on consecutive frames, and the detected objects are joined throughout the frames. Those techniques have some uncertainties namely: false position from detector as well as missed direction. The produced uncertainties could be solved by means of Markovian Decision Process (MDP) so object tracking can be expressed as an MDP operation.

D. Object Detection and Tracking Related Works:

Harshithal R et al. [34] suggest a design of a system for real time pedestrian detection for self-driving cars then the system performance is evaluated by utilizing images and video inputs datasets. The proposed model is implemented by utilizing of Histograms of Oriented Gradients Feature extraction as well as Support Vector Machine (SVM) Classifier. The suggested system is able to detect pedestrians with accuracy of 98.31% besides when its closer to camera, will be able to achieve 100% recognition accuracy. While Chaowei Hu et al [35], suggest an embedding Convolutional Neural Network (CNN) based fast obstacle detection procedure, the proposed model goal comprises three key parts: single image obstacles detection, adjacent images obstacles tracking, and finally video obstacles detection and tracking. The suggested model is evaluated on KITTI datasets together with collected datasets. The results show that the proposed model achieves high accuracy.

Junekyo Jhung et al. [36] present an end-to-end steering controller with CNN-depend on closed-loop feedback for autonomous vehicles that enhances performance associated with traditional CNN-based approaches. The performance of the suggested system has been tested under simulations as well as on-road tests. On the other hand, Xinping Gu et al. [37] propose a lane changing prediction model of autonomous vehicles depend on the data of US-101 and I-80 segments in the NGSIM dataset. The simulations results display that the lane changing model depends on random forest have higher precision.

Furthermore, Rodolfo Valiente et al. [38] propose a new approach by sharing images between cooperative self-driving vehicles to enhance the control reliability of steering angle. This approach utilizes CNN. The proposed model shows lowest RMSE value respect to the other existing models. Wherease, Laura García Cuenca et al. [39] uses Supervised learning algorithms (support vector machine, linear regression, as well as deep learning) to build predictive models to approximate the vehicle speed as well as steering angle. The proposed models were tested with a dataset. The results displayed that the steering angle as well as vehicle speed offer significant information for driving behavior prediction.

On the other hand, Yonggang Liu et al. [40] propose an autonomous lane change decision-making model by analyzing the parameters of the autonomous vehicle lane change. Then, a support vector machine (SVM) algorithm is adopted. The results demonstrate that the SVM model fulfill better results than the rule-based lane change. While Amal Hbaieb et al. [41] propose a real time pedestrian and vehicle detectors, the proposed model utilizes Histogram of Oriented Gradient (HOG) descriptor, Support Vector Machine (SVM) classifier together with Haar feature based cascade classifier, and the first two methods are used for pedestrian detection where the last method is used for vehicle detection. The suggested model shows good detection accuracy around 90% for pedestrian detection and 80% for vehicle detection. Moreover, Pranav KB et al. [42] propose implementation of a real-time pedestrian detection system based on CNN algorithm for autonomous vehicles. The system is trained with PETA–CUHK dataset, INRIA dataset, as well as real-time video input. The results show recognition accuracies ranging between 96.73 – 100%. The proposed system can also be used as a driver assistance system in non- autonomous cars. . Moreover, Namareq O. et al (2023) [43] paper provides an approach to deep learning; which combines the benefits of both convolutional neural network CNN together with Dense technique. This approach learns based on features extracted from the feature extraction technique which is linear discriminant analysis (LDA) combined with feature expansion techniques namely: standard deviation, min, max, mod, variance and mean. The presented approach has proven its success in both testing and training data and achieving 100% accuracy in both terms.

VII.

VIII. DECISION

The generation of effective and safe action plan in real time of SDC is achieved in decision platform. Three main operations are involved in this stage. Including Traffic Prediction, Path Panning and Obstacle Avoidance [1,3].

A. Traffic Prediction

The traffic prediction target is to estimate the actions of the detected objects in the nearby future, gives the detailed information corresponding to the prediction outcomes together with the points of the spatial temporal trajectory, then push them through other sub modules. Various factors have to be taken in this stage of prediction such as the historical behavior, the map features as well as the surrounding scenarios. These kinds of predictions are categorical and can be resolved by machine learning classification techniques [8,9,10]. This information is not enough since information like speed, headings, timing information together with trajectory points have to be known for the predicted trajectories which can be achieved by producing the spatial temporal trajectories to keep tracking of lane sequences. One of the possible techniques to resolve this issue in SDCs is the Kalman Filter [44,45].

B. Path Planning

Path planning of SDC in a moving surrounding is a very complicated issue especially when the car is demanded to be utilized in its full susceptibility. The path planning actions can be depicted by employing a top-down method comprising a hierarchy of three layers: route, behavioral, and motion planning [1,3]. All described below.

1) Route Planning

In route planning, the SDCs performs the calculations to determine the best route to travel from the present location to the destination depend on the information provided by a map about the road network. The route planning calculations need to take into consideration external factors, like real-time traffic information, the expected energy consumption, the user 's preference whether prefer to use toll roads or not, etc. path Route planning typically utilizes particular algorithm named the shortest path algorithm. The shortest path algorithm can be defined as determining the shortest path between two nodes in a map. One of the popular shortest path techniques is Dijkstra's algorithm. The Dijkstra's algorithm begins by initializing the distance value of all nodes to infinity. then for all directly nodes from the starting node a new value or cost is measured, and the value is updated if the new distance is shorter. This process repeats over the entire map till all nodes been traversed. The shortest path to any destination node can be calculated by summing the cost of the node as well as the set of registered edges to reach that node. Faster procedures, like Contraction Hierarchies, implement some precomputation to speed up the procedure [3,46].

2) Behavioral Planning

Behavioral planning determines how to arrive to the next waypoint under the actual driving context, i.e., considering the current road geometry, other traffic participants, perceived obstacles, actual traffic rules (no passing zone, speed limit) limitation of vehicle control, etc. The result of behavioral planning is high-level decision, like lane following, changing lane, overtaking, merging, etc. One of the most issues in this planning step includes expecting the behavior of moving objects in the surrounding. This is crucial for mixed traffic surroundings where SDCs paticipate the normal vehicles with road [3,47].

3) Motion Planning

The problem of this planning step is divided into two sub-problems: the path planning and the trajectory planning. In Path planning step the task of determining the shortest collision-free path from the source point to the destination point While in Trajectory planning step the task is determining the motion sequence, as function of time, to fulfill a smooth drive amongst the desired path. From here a path can be considered as a collection of trajectories with a specification of the vehicle's velocity, acceleration and occasionally jerk (change of acceleration) at each point. to determine the best path, the SDC's environment created by the map information have to be merged with the information collected by the sensors as well as other sources in a discrete representation.

Suitable representations comprise driving corridors or occupancy grids. In Driving corridors represent the free space by taking consideration of all physical boundaries and detected obstacles, like the allowed lane and road boundaries. While in the case of occupancy grids, the SDC's environment is spilt into 2D grid cells. Each cell includes the probability of being occupied by an object. Each of these representations has its own advantages as well as its own disadvantages [3,48].

C. Obstacle Avoidance

In the field of autonomous driving, safety is an essential concern. So that at least two levels of avoidance procedures are utilized to make sure that the SDCs would not collide with obstacles nor other cars. The first procedure is proactive which is depend on traffic predictions at runtime, the traffic prediction technique implements measures like time to collision or predicted minimum distance. The avoidance technique, depending on this information, is triggered to implement local path planning. In the case that the first level fails, the second procedure namely the reactive technique which uses radar data will be gaining control. Once the radar reveals an obstacle;The reactive technique will exceed the current control to keep the vehicle far away from the obstacles [1].

D. Decision Related Works

Seho Shin et al [49] present the DO-RRT scheme to provide efficient path planning in a narrow space for autonomous vehicle driving, which involves many changes in onward and backward directions. They suggest a model of determining the preferred vehicle direction based on magnetic-field, using the nonholonomic vehicle constraints in addition to the geometric obstacles constraints. Simulation experiments in narrow parking spaces relieve its benefit in iteration number as well as the length of the planned path over the conventional nonholonomic RRT algorithm. Domokos Kiss et al [50] present a scheme of car-like robots for global planning that produce continuous curvature paths. The (introductory) RTR path planner is able to design paths comprising straight movements in addition to turning by applying T∗TS local planner to the RTR path in a second approximation phase to get the final path taking into account the continuous and bounded curvature conditions. Simulation experiments demonstrated that their algorithm is useful in few or high difficult situations and the acquired paths are seemed to be natural in a large degree.

Nor Badariyah et al [51] propose an algorithm depending on visibility graph (VG), named Equilateral Space Oriented Visibility Graph (ESOVG), in which the obstacles number for path planning is minimized by ignoring the obstacles which sited outer the space. Simulation results show that the given scheme has an enhancement of 90% as compared to conventional VG. Furthermore, it is appropriate to be used in real-time to accelerate the autonomous cars development. While Chaymaa Laminia et al [52] propose a GAs fitness function that optimizes the mobile robot energy consumed by minimizing the turn's number in its path in order to attain its goal. Moreover, they suggest a GA Improved Same Adjacency Crossover for path planning dilemma. Simulation experiments with several environments and different sizes prove that the presented scheme gives the optimal path with more optimal average turns values and average iterations numbers as compared to other GA methods.

Melih O¨ zcan et al. [53] present a vehicle pattern that able to capture the dynamics of both onward and backward driving in high and low speed. They address the motion planning issue in this model, and provide a model that integrates Sequential Composition of Controllers (SCC) in addition to Rapidly Exploring Random Trees (RRT). Furthermore, simulations show the effectiveness and robustness of the given method. On the other hand, Yuying Chen et al [54] propose a hierarchical method which disconnects path planning from temporal planning. A path which achieves the kinematic constraints is produced via a modified bidirectional rapidly exploring random tree (bi-RRT) scheme. Moreover, each node timestamp in the path is optimized utilizing sequential quadratic programming (SQP) together with feasible searching bounds that specified utilizing safe intervals (SIs). Simulation as well as real test experiments under various scenarios verify the effectiveness of the proposed method. Furthermore, Jean-Baptiste Receveur et al [55] propose a genetic algorithm-potential field combined scheme, that able to argument a strategy of a specific distance. Moreover, a global optimal trajectory is acquired using multi-criteria optimization. While Jacob Miller et al [56] describe a software stack of an autonomous vehicle that integrates SLAM and LiDAR obstacle detection by Euclidean clustering motion planning and obstacle using RRTs and MPC. The presented algorithms are tested under Gazebo world simulation with a vehicle model of 2018 Ford Fusion Fybrid.

IX. CONCLUSION & FUTURE WORKS

Self – driving vehicles depend on complicated technology, and it is difficult trying to get all the information of how they work exactly. This review sets out to give the core concepts in self-driving car research field. This study tackles the field of SDCs from a technical viewpoint. It illustrates the big picture of SDCs are and how they are improved, covering all the bases, starting from the importance of automated cars to the major functions of perception, passing through the significance and types of sensors.

It can be concluded that the deep learning technologies are the best choice in object detection and classification, on the other hand, the choice of path planning and tracking algorithm depends on the application circumstances, finally the best determination of the SDCs is depend on the corporation of the location sensors and SLAM algorithms.

However, the present paper provides a comprehensive review regarding technologies that used in algorithm subsystem of creating autonomous vehicles; future works may concern client and cloud subsystems.

REFERENCES

1. S. Liu, J. Tang, S. Wu, & J. Gaudiot. Creating autonomous vehicle systems. Synthesis Lectures on Computer Science. 2020; 8 (2): i-216.
2. Moody, Joanna, Nathaniel Bailey, and Jinhua Zhao. Public perceptions of autonomous vehicle safety: An international comparison. Safety science. 2020; 121:634. http://dx.doi.org/10.1016/j.ssci.2019.07.022
3. S. Hanky. Introduction to Self-Driving Vehicle Technology. Chapman and Hall/CRC; 2019.
4. Sumit Ranjan, and S. Senthamilarasu. Applied Deep Learning and Computer Vision for Self-Driving Cars. Packt Publishing Ltd; 2020.
5. L. Reddy Cenkeramaddi, J. Bhatia, A. Jha, S. Kumar Vishkarma and J. Soumya. A Survey on Sensors for Autonomous Systems. 15th Conference on Industrial Electronics and Applications (ICIEA) 2020,pp.1182-118,IEEE. http://dx.doi.org/10.1109/ICIEA48937.2020.9248282
6. Kuutti, Sampo, et al. A survey of the state-of-the-art localization techniques and their potentials for autonomous vehicle applications. IEEE Internet of Things Journal. 2018; 5(2):829-846. http://dx.doi.org/10.1109/JIOT.2018.2812300
7. 上條俊介, 古艶磊, and 許立達. Autonomous vehicle technologies: Localization and mapping. 電子情報通信学会 基礎・境界ソサイエティ. Fundamentals Review. 2015; 9 (2):131-141.
8. Fernández-Madrigal, Juan-Antonio, and José Luis Blanco Claraco. Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods. IGI Global; 2013. http://dx.doi.org/10.4018/978-1-4666-2104-6
9. T. Bailey, & H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. IEEE robotics & automation magazine. 2006; 13(3):108-117. http://dx.doi.org/10.1109/MRA.2006.1678144
10. Bengtsson, Thomas, Peter Bickel, and Bo Li. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. Probability and statistics: Essays in honor of David A.Freedman,2008;2:316-334. http://dx.doi.org/10.1214/193940307000000518
11. Doucet, Arnaud, et al. Rao-Blackwellised particle filtering for dynamic Bayesian networks. arXiv preprint arXiv. 2013; 1301.3853.
12. Cadena, Cesar, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. IEEE Transactions on robotics.2016;32(6):1309-1332. http://dx.doi.org/10.1109/TRO.2016.2624754
13. F. Gustafsson. Particle filter theory and practice with positioning applications. IEEE Aerospace and Electronic Systems Magazine. 2010, 25(7):53–82. http://dx.doi.org/10.1109/MAES.2010.5546308

14. John H Halton. Sequential monte carlo techniques for solving non-linear systems. Monte Carlo Methods and Applications MCMA. 2006; 12(2):113–141. http://dx.doi.org/10.1515/156939606777488879
15. E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3D reconstruction in Computer Vision and Pattern Recognition. Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) 2006, 363-370, IEEE.
16. C. Stachniss, John J. Leonard, and S. Thrun. Simultaneous Localization and Mapping. Springer International Publishing. Cham; 2016 pp. 1153–1176. 2016. http://dx.doi.org/10.1007/978-3-319-32552-1_46
17. Grissett, Giorgio, et al. A tutorial on graph-based SLAM. IEEE Intelligent Transportation Systems Magazine.2010;2(4):31-43. http://dx.doi.org/10.1109/MITS.2010.939925
18. B. Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - A modern synthesis. In Proceedings of the International Workshop on Vision Algorithms: Theory and Practice, ICCV '99, London, UK, 2000,298–372,Springer-Verlag. http://dx.doi.org/10.1007/3-540-44480-7_21
19. G. Yanlei, Li-Ta Hsu, and Shunsuke Kamijo. Vehicle localization based on global navigation satellite system aided by three-dimensional map. Transportation Research Record. 2017; 2621(1):55-61.
20. Meng, Xiaoli, Heng Wang, and Bingbing Liu. A robust vehicle localization approach based on gnss/imu/dmi/lidar sensor fusion for autonomous vehicles. Sensors. 2017; 17(9):2140-2159. http://dx.doi.org/10.3390/s17092140
21. Guang, Xingxing, et al. An autonomous vehicle navigation system based on inertial and visual Sensors. Sensors. 2018; 18(9):2952. http://dx.doi.org/10.3390/s18092952
22. L. Khaoula and P. Bonnifait. Cooperative localization for autonomous vehicles sharing GNSS measurements. Cooperative Localization and Navigation. CRC Press; 2019 pp. 521-546.
23. Santos, Giovanni A., et al. Improved localization framework for autonomous vehicles via tensor and antenna array based GNSS receivers. Workshop on Communication Networks and Power Systems 2020,pp.1-6,IEEE. http://dx.doi.org/10.1109/WCNPS50723.2020.9263757
24. Onyekpe, Uche, Vasile Palade, and Stratis Kanarachos. Learning to localise automated vehicles in challenging environments using Inertial Navigation Systems (INS). Applied Sciences. 2021;11(3):1270. http://dx.doi.org/10.3390/app11031270
25. Chen, Xieyuanli, et al. Range Image-based LiDAR Localization for Autonomous Vehicles. International Conference on Robotics and Automation (ICRA) 2021, 5802-5808, IEEE. http://dx.doi.org/10.1109/ICRA48506.2021.9561335
26. Luca Venturi, Krishtof Korda. Hands-On Vision and Behavior for Self-Driving Cars. Packt Publishing Ltd; 2021.
27. David G Lowe. Distinctive image features from scale-invariant key- points. International Journal of Computer Vision. 2004; 60 (2):91–110.
28. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) 2005 (pp. 886–893). IEEE. http://dx.doi.org/10.1109/CVPR.2005.177
29. Matas, Jiri, et al. Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing.2004;22(10):761-767. http://dx.doi.org/10.1016/j.imavis.2004.02.006
30. B. Leo. Random forests. Machine learning. 2001; 45(1): 5-32.
31. J. Giebel, D. Gavrila, and C. Schnörr. A Bayesian framework for multi-cue 3d object tracking. Computer Vision-ECCV.2004;pp.241–252. http://dx.doi.org/10.1007/978-3-540-24673-2_20
32. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L. Online multiperson tracking-by-detection from a single, uncalibrated camera. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011; 33(9):1820–1833. http://dx.doi.org/10.1109/TPAMI.2010.232
33. Harshitha, R., and J. Manikandan. Design of a real-time pedestrian detection system for autonomous vehicles. In Region 10 Symposium (TENSYMP) 2017(pp.1-4).IEEE. http://dx.doi.org/10.1109/TENCONSpring.2017.8069981
34. Hu, Chaowei, et al. Embedding CNN-based fast obstacles detection for autonomous vehicles. SAE Technical Paper, 2018; No. 2018-01-1622. http://dx.doi.org/10.4271/2018-01-1622
35. Jhung, Junekyo, et al. End-to-end steering controller with cnn-based closed-loop feedback for autonomous vehicles. In IEEE intelligent vehicles symposium (IV) 2018, 617-622, IEEE. http://dx.doi.org/10.1109/IVS.2018.8500440
36. Gu, Xinping, et al. Vehicle lane change decision model based on random forest. International Conference on Power, Intelligent Computing and Systems (ICPICS) 2019, 115-120, IEEE. http://dx.doi.org/10.1109/ICPICS47731.2019.8942520
37. Valiente, Rodolfo, et al. Controlling steering angle for cooperative self-driving vehicles utilizing cnn and lstm-based deep networks. In 2019 IEEE intelligent vehicles

symposium (IV), 2019, 2423-2428,IEEE. http://dx.doi.org/10.1109/IVS.2019.8814260

38. Garcia Cuenca, Laura, et al. Machine learning techniques for undertaking roundabouts in autonomous driving. Sensors. 2019; 19(10):2386. http://dx.doi.org/10.3390/s19102386
39. Liu, Yonggang, et al. A novel lane change decision-making model of autonomous vehicle based on support vector machine. IEEE Access. 2019,7:26543-26550. http://dx.doi.org/10.1109/ACCESS.2019.2900416
40. Hbaieb, Amal, Jihene Rezgui, and Lamia Chaari. Pedestrian detection for autonomous driving within cooperative communication system. Wireless Communications and Networking Conference (WCNC) 2019, 1-6, IEEE. http://dx.doi.org/10.1109/WCNC.2019.8886037
41. Pranav, K. B., and J. Manikandan. Design and evaluation of a real-time pedestrian detection system for autonomous vehicles. Zooming Innovation in Consumer Technologies Conference (ZINC) 2020, 155-159, IEEE. http://dx.doi.org/10.1109/ZINC50678.2020.9161768
42. Gadepally, Vijay, Ashok Krishnamurthy, and Ümit Özgüner. A framework for estimating long term driver behavior. Journal of advanced transportation. 2017; Vol. 2017. Article ID 3080859. http://dx.doi.org/10.1155/2017/3080859
43. N. Odey, and A. Marhoon. Feature Deep Learning Extraction Approach for Object Detection in Self-Driving Cars. Iraqi Journal for Electrical And Electronic Engineering. 2023; 19 (2): 62-69. https://doi.org/10.37917/ijeee.19.2.8
44. S. Bonnin, T. Weisswange, F. Kummert, and J. Schmuedderich. General behavior prediction by a combination of scenario-specific models. IEEE Transactions on Intelligent Transportation Systems. 2014;15(4):1478–1488. http://dx.doi.org/10.1109/TITS.2014.2299340
45. Kumar, P., Perrollaz, M., Lefevre, S., and Laugier, C. Learning-based approach for online lane change intention prediction. In Proceedings of the IEEE Intelligent Vehicles Symposium 2013, 797–802. IEEE. http://dx.doi.org/10.1109/IVS.2013.6629564
46. Geisberger, Robert, et al. Exact routing in large road networks using contraction hierarchies. Transportation Science.2012;46(3):388-404. http://dx.doi.org/10.1287/trsc.1110.0401
47. Junqing Wei, Jarrod M Snider, Tianyu Gu, John M Dolan, and Bakhtiar Litkouhi. A behavioral planning framework for autonomous driving. Intelligent Vehicles Symposium Proceedings, 2014, 458–464,IEEE. http://dx.doi.org/10.1109/IVS.2014.6856582
48. Christos Katrakazas, Mohammed Quddus, Wen-Hua Chen, and Lipika Deka. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. Transportation Research Part C: Emerging Technologies. 2015; 60: 416–442.
49. Shin, Seho, Joonwoo Ahn, and Jaeheung Park. Desired orientation rrt (do-rrt) for autonomous vehicle in narrow cluttered spaces. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2016, 4736-4741, IEEE. http://dx.doi.org/10.1109/IROS.2016.7759696
50. Kiss, Domokos, and Dávid Papp. Effective navigation in narrow areas: A planning method for autonomous cars. 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI) 2017, 423-430, IEEE. http://dx.doi.org/10.1109/SAMI.2017.7880346
51. Latip, Nor Badariyah Abdul, Rosli Omar, and Sanjoy Kumar Debnath. Optimal path planning using equilateral spaces oriented visibility graph method. International Journal of Electrical and Computer Engineering. 2017; 7(6): pp. 3046. http://dx.doi.org/10.11591/ijece.v7i6.pp3046-3051
52. Lamini, Chaymaa, Said Benhlima, and Ali Elbekri. Genetic algorithm based approach for autonomous mobile robot path planning. Procedia Computer Science.2018;127:180-189. http://dx.doi.org/10.1016/j.procs.2018.01.113
53. Özcan, Melih, and Mustafa Mert Ankarali. Feedback motion planning for a dynamic car model via random sequential composition. International conference on systems, man and cybernetics (SMC). 2019, 4239-4244, IEEE. http://dx.doi.org/10.1109/SMC.2019.8913917
54. Chen, Yuying, Haoyang Ye, and Ming Liu. Hierarchical Trajectory Planning for Autonomous Driving in Low-speed Driving Scenarios Based on RRT and Optimization. arXiv preprint arXiv. 2019; 1904:2606. https://doi.org/10.48550/arXiv.1904.02606
55. Y. Xiang, A. Alahi, and S. Savarese, Learning to track: Online multi-object tracking by decision making. In Proceedings of International Conference on Computer Vision 2015, 4705–4713, IEEE. http://doi.org/10.1109/ICCV.2015.534
56. L. S. Liu, J. Lin, J. Yao, D. He, J. Zheng, J. Huang, & P. Shi. Path planning for smart car based on Dijkstra algorithm and dynamic window approach. Wireless Communications and Mobile Computing. 2021; Article ID 8881684. http://dx.doi.org/10.1155/2021/8881684