International Journal of Artificial Intelligence & Robotics (IJAIR)
Vol.6, No.1, 2024, pp.8-18
E - ISSN : 2686-6269

Design of Predictive Control System for Lane Change in Autonomous Vehicle

Shervind Maharani Mega Permatasari[1], Bambang Lelono Widjiantoro[2], Katherin Indriawati[3]
[1,2,3]Engineering Physics, Industrial Technology and Systems Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[1]mshervind@gmail.com*; [2]blelono@ep.its.ac.id; [3]kindriawati@gmail.com
*corresponding author

ABSTRACT

The automobile business is introducing a lot of autonomous vehicles in the modern day. Lane changes are one of the most complicated urban scenarios in which autonomous vehicles are used. Self-driving automobiles must thus interact with human-driven vehicles in a certain way. In this work, we concentrate on the autonomous vehicle's lane-changing control system for obstacle avoidance. This study employs a predictive control system as its methodology. The vehicle's next movements can be predicted by this control system. The vehicle's position, which is adjusted by the steering angle, is the controllable variable. The vehicle's position, which is adjusted by the steering angle, is the controllable variable. It is clear from the numerical simulation results that the predictive control system executes control actions on lane changes correctly, avoiding collisions with the running vehicle obstacles. RMSE (Root-Mean Square Error) is a performance metric that is derived from the difference between the vehicle's lateral position and the reference trajectory value. The RMSE of the planned predictive control is 0.9681.

Keywords: Autonomous Vehicle; Lane Change; Model Predictive Control.

Article History

Received : Apr, 10th 2024	Accepted : May, 25th 2024	Published : May, 31th 2024

I. INTRODUCTION

The development of the world's automotive industry is currently competing to assemble land vehicles that are comfortable and certainly safe. Technology and artificial intelligence have advanced quite quickly in recent years. The development of autonomous vehicles is one engineering outcome in the realm of transportation technology. An autonomous system is one that can run on its own [1].

Vehicles that have autonomous systems are ones that can run without the need for human intervention. The primary goal of research on autonomous vehicles is to enhance the driving safety system. According to the data obtained, about 94% of traffic accidents are caused by human error [2]. IRTAD (International Road Traffic and Accident Database) announced that in 2019, more than 1.3 million people died, and 10 million people were injured in traffic accidents. Pedestrians were the main victims of about 50% of the total incidents [3]. Driverless four-wheeled vehicle technology, also known as autonomous vehicles, will help reduce traffic accident rates [4].

In previous research on designing lane change systems in autonomous vehicles, many research methods have been used to design an autonomous vehicle trajectory. However, some methods still require a lot of research and testing data, and their implementation is complicated and expensive. Cell decomposition algorithms such as Repidly-explore Random Tree (RRT) are one of the methods used for collision-free path planning, but they incur huge computational and memory costs. Other research uses a trajectory planning approach based on discrete optimization for autonomous vehicles. This approach can select an optimal path from a set of candidate trajectories. This method uses the global route from the digital map obtained before performing local trajectory planning. The centerline or so-called frenet reference path representing the global route is constructed using a cubic spline to generate candidate paths, the frenet coordinate system (s - ρ) is used, and the directional information for the global route is combined with vehicle maneuvers by adjusting the lateral offset to the centerline. A cost function is designed and used to select the optimal path from multiple candidate paths. The results show that the control system with a rapidly-explored random tree (RRT) is less good at generating path extension and smoother control system results [11].

The previous research conducted experiments using perception systems, rate design systems, and control systems that can produce better and more efficient values—starting from the perception system. The perception system in autonomous vehicles is how the vehicle can observe its surroundings. This system uses many sensors to get accurate results. This process includes

DOI: http://dx.doi.org/10.25139/ijair.v6i1.8141

 E - ISSN : 2686-6269

detection. This process includes detection, understanding, and interpretation of the environment around the vehicle, both static and dynamic things [5]. In addition, this perception system must also be able to review ego vehicles against other vehicles to produce simulation results in the form of designed ego vehicle movements. Ego vehicle is a name for the designed vehicle. In this research, the perception system that will be generated only displays the simulated form of the ego vehicle against the lane design. These results produce the form of ego vehicle position, lane design form, and maneuvering simulation.

Path planning is designing a path from a starting position to a destination position in an environment that has static or dynamic obstacles, passing through predefined waypoints [7]. Path planning algorithms that allow vehicles to find the shortest path or optimal path between two points. The optimal path can be a path that minimizes the number of turns, the number of violations, the number of collisions, or whatever is needed in the program. Path planning is classified into two types: global path planning and local path planning [8]. The path planning algorithm in this study uses a discrete algorithm, namely *A**.

Control systems in autonomous vehicles can perform path-switching actions using control systems, one of which is a predictive control system. This research uses a predictive control system as one of the good trajectory planning methods while producing good optimal values in control [12] because of its flexibility and ability to calculate the optimal value solution in the presence of better hard and soft constraints [9]. The advantage of using predictive control methods is that they can predict the future dynamics of a system and optimize the prediction horizon associated with current information [10], from existing research in trajectory planning.

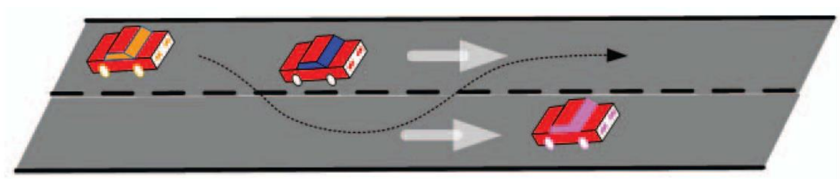


Fig.1. Manoeuvre Phase

Fig.1 is an illustration of an overtaking vehicle. When the red vehicle with the yellow window detects the lane and obstacles in front of it, the vehicle starts performing lane change actions to the right lane, and trajectory tracking continues in the right lane until the autonomous vehicle passes the red vehicle with the blue window. Then, a second lane change occurs to push the autonomous vehicle back into the left lane. During the overtaking maneuver, the autonomous vehicle must pass the red vehicle with blue windows at a sufficient distance to avoid collision with the purple-windowed vehicle [5]. They are creating a lane change system using the predictive control system. As explained in the paragraph above, this research uses a predictive control system as a lane design system while producing good control system results, utilizing a perception system in the form of the occupancy grid method results in the position of the vehicle with the environment around the ego vehicle. These results are processed to determine the optimal value in producing an efficient and good control system using a predictive control system. So, this research aims to produce the optimal value of the predictive control system in the lane change design system in autonomous vehicles.

II. METHOD

A. Vehicle Modeling

A ground vehicle that is often used in conducting autonomous research is a car. Cars have two front wheels for turning and two fixed rear wheels. This model can also be called the Ackermann model [11]. The canter of rotation of the car chassis lies on the line through the rear wheels at the intersection with the line perpendicular to the front wheels in Fig.2.

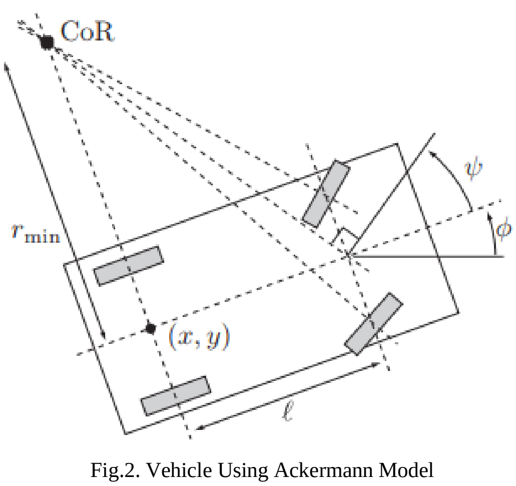


Fig.2. Vehicle Using Ackermann Model

DOI: http://dx.doi.org/10.25139/ijair.v6i1.8141

 E - ISSN : 2686-6269

Fig. 2 is an image of the ego vehicle model designed for this study. The rotation canter of the car chassis lies on the line through the rear wheels at the intersection with the line perpendicular to the front wheels in Fig. 2. The kinematic model of the Ackermann vehicle by ignoring the turning angle of four wheels is as in Equation (1) [11].

$$\dot{q} = \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} (tan\ \psi)/l & 0 \\ \cos\phi & 0 \\ \sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{1}$$

$$\dot{x} = v\cos\dot{\phi} \tag{2}$$

$$\dot{y} = v\sin\dot{\phi} \tag{3}$$

$$\dot{\phi} = \frac{v}{l}\tan\psi \tag{4}$$

$$\dot{\psi} = w \tag{5}$$

Equations 1 to 5 are vehicle modeling used in this study. Equation (1) is from the value of the model used, namely the Ackermann model. The Equations (2) until (5) are the position of the vehicle against the front and rear wheels to the steering angle. Where, the $\dot{q}$ variable is Ackermann vehicle modeling, the $\dot{\phi}$ variable is rolling angle, the $\dot{x}$ variable is midpoint x on the rear wheel, the $\dot{y}$ variable is midpoint y on the rear wheel, the $\dot{\psi}$ variable is steering angle, the l variable is vehicle length from the center point of the front wheels to the center point of the rear wheels, the v variable is the velocity (km/h), and the w variable is turnover (rpm). Table I is a parameter of vehicle modeling, a table that contains the information and values used in this study.

TABLE I
ACKERMANN PARAMETERS VEHICLE

Symbol	PARAMETERS	Value
ϕ	The angular position of the vehicle	0.1
x	Position of the vehicle with respect to the axis x (m)	0.5
y	Position of the vehicle with respect to the axis y (m)	0.3
ψ	Vehicle steering angle position (rad)	0.5
l	Distance between front and rear wheel axis (m)	0.2
v	velocity (m/s)	10
w	turnover (rad/s)	3

B. Environment

At this point, the scenario driving designer is used to develop a road condition scenario scheme by adding parameters for the road's length and width, the path's geometry, and the amount of obstacles—such as vehicle objects—as needed. A lane design parameter table is shown in Table II.

TABLE II
PARAMETERS ENVIRONMENT

PARAMETERS	Value
Shape of the world	Straight turn
Long	200 meters
Width	7 Meters
Turn	60º and 130º angle
Ego vehicle	1 vehicle
obstacles	4 vehicles, constant dynamics

The parameters consist of the environmental model dimensions of the lane design. In this study, the road is designed to turn. The road conditions used in Fig.3 are a design drawing for the design of the lanes used. The image is from the overall lane model, which is in the form of straight turns, blue ego vehicles, and other obstacles.

DOI: http://dx.doi.org/10.25139/ijair.v6i1.8141

E - ISSN : 2686-6269

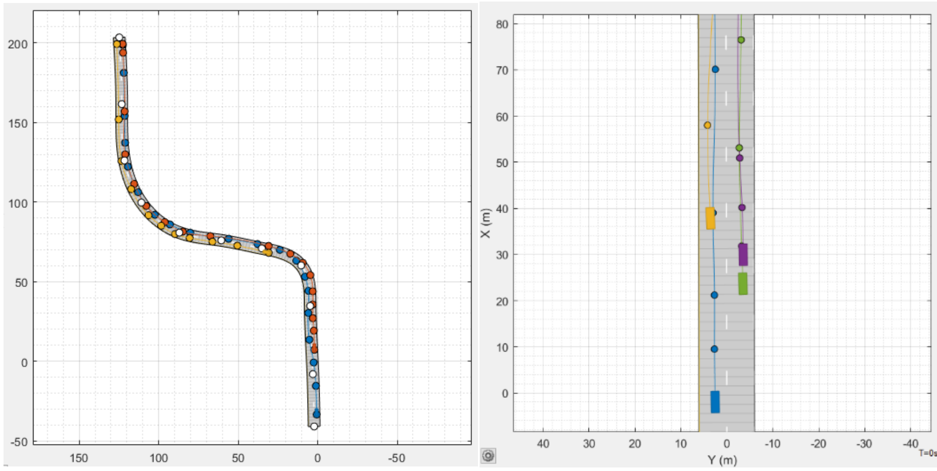


Fig.3. Schematic of the Environment

C. *Path Planner*

Path Planner is one of the most fundamental problems. Path planning allows the vehicle to find the optimal path between two points. Path planning can be categorized into two, namely, global path planning and local path planning [8]. Fig.4 is a block diagram design of path planning consisting of 3 blocks. Namely, the occupancy grid block is used as the perception system. Then, the A* block is a path planner and the reference signal block, which is useful for converting the signal generated from A* into a signal that the predictive control system can input.

Fig.4. Path Planning Block Diagram

The A* algorithm is one of the global planning methods that uses a discrete search method to find the destination node by utilizing a graph model. The graph consists of nodes, which can be formed by cells of the underlying grid map. Equation (6) is used by the A* algorithm to calculate the best path through the network by taking into account the cost that is determined by the connections between the nodes. The A* algorithm uses two heuristic values, h(n), which are Manhattan distance using Equation (8) and Euclidean distance using Equation (7). [5]. Where the $F(n)$ variable is the cost required to move, the $g(n)$ variable is the cost movement from the starting point to the current node, the $h(n)$ variable is the cost of the movement path from the current node to the destination node, the x_n variable is point x at the current node, the x_{goal} variable is point x at the destination node, the y_n variable is point y at the current node and the y_{goal} variable is point y at a destination node.

$$F(n) = g(n) + h(n) \tag{6}$$

$$h(n) = |x_n - x_{goal}| + |y_n - y_{goal}| \tag{7}$$

$$h(n) = \sqrt{(x_n - x_{goal})^2 + (y_n - y_{goal})^2} \tag{8}$$

D. *Object Detection*

Object detection is the process of recognizing an object. This object detection indicates that the object is clearly visible in the frame, and its position is also clearly determined. Object detection can be said to be a way to find and determine the position of objects in a frame [6]. LiDAR consists of a distance scan that continuously rotates clockwise through a motor attached using a coir. It provides distance scanning data up to an 8-meter radius and transmits the obtained data through a communication interface [5]. LiDAR uses modulated infrared signals, after which the returned signals are detected and sampled by a visual acquisition module. Through the digital signal processing module, the sampled data is processed to produce distance and angle information between the object and the LiDAR [5]. The capability to function in space, a full field of view that encompasses 360 degrees, a compact size, a reasonably lightweight form factor, and a very low power consumption. The parameters of the sensor that was utilized, specifically the camera, are listed in Table III. It is composed of values, parameters, and units of measurement. When it comes to the construction of detecting objects that are utilized by ego vehicles, this table contains the parameter values that are utilized. It will assist in the detection of ego vehicles and the environment around them.

DOI: http://dx.doi.org/10.25139/ijair.v6i1.8141

 E - ISSN : 2686-6269

TABLE III
CAMERA SENSOR SPECIFICATIONS IN AUTONOMOUS VEHICLES

PARAMETERS	UNITS	Value
Maximum Range	Meters	150
Detection Probability	-	1
Accuracy Bounding Box	Pixels	1
Focal Length	Pixel	[800, 800]
Maximum Occlusions	*Fraction*	0.5

Fig.5 shows the placement of the camera sensor and the designed Lidar. The camera sensor is located on the front roof of the vehicle. In comparison, the Lidar is placed in the center of the vehicle's roof. This placement is usually adjusted to the general placement of sensors. This placement is placed in that section in order to facilitate the detection of obstacles around the ego vehicle. In Fig.6 are the parameters of the camera sensor used in the vehicle design that can help the maker reset the parameters according to what is needed. Fig.7 shows the parameters of the Lidar sensor used in the vehicle design, which can help the maker to reset the parameters according to what is needed.

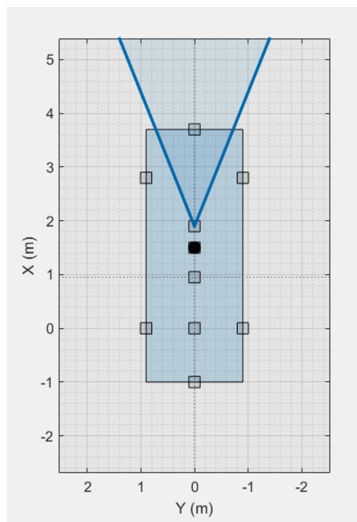


Fig.5. Camera Sensors And Lidar Installed on An Autonomous Vehicle in The Scenario.

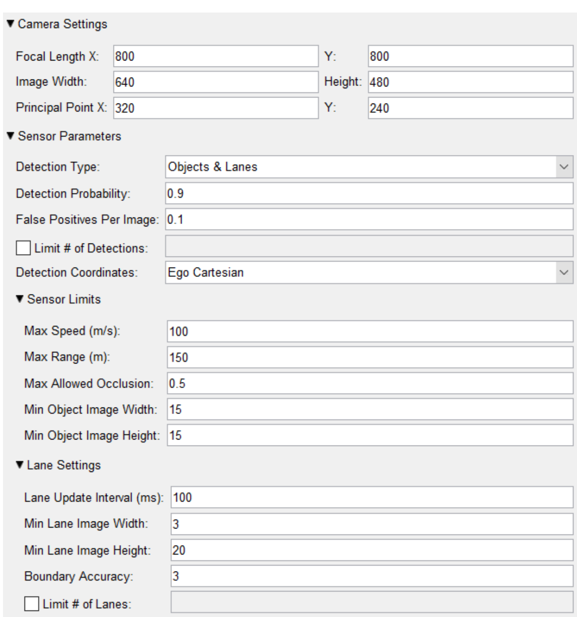


Fig.6. Properties and Configuration of Edged Camera

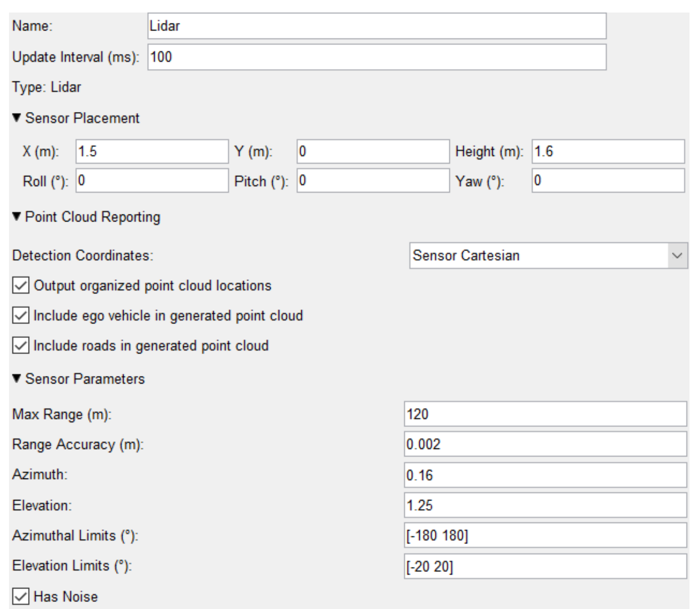


Fig.7. Properties and Configuration of Edged LiDAR

DOI: http://dx.doi.org/10.25139/ijair.v6i1.8141

E. Occupancy Grid Generator

This system's occupancy grid block consists of two blocks: occupancy visualisation and occupancy grid generate. It is necessary to have data on actors (obstacles), ego vehicles, and lanes in order to feed the occupancy process. After that, the occupancy grid was generated. Sensor data and information gathered from the vehicle's surroundings are utilized to create occupancy grids. A two-dimensional depiction of an ambient space separated into tiny cells is called an occupancy grid. The occupancy grid shows the percentage of a given site that is either occupied or vacant. In order to identify objects and determine their presence or distance, this method entails gathering data from sensors like Lidar, radar, or cameras. After processing the sensor data, a grid reflecting the neighborhood's condition is created. Grids based on sensor data can be generated in this process using mapping methods like the grid-based occupancy mapping approach or the A* algorithm [6]

Moreover, data is processed to create a grid, lane canter, and egoPose (the autonomous vehicle position state) when the occupancy grid, which represents data in two dimensions, is formed. Occupancy grid visualization is required in order to show or visualize the data's outcomes. The occupancy grid visualization technique is used to show the occupancy grid visualization once it has been generated. Users may view and comprehend the state of the environment in graphical form due to this representation. The occupancy grid is shown in this visualization as a two-dimensional grid view, with each cell having a distinct color or intensity. Whereas empty cells show spaces devoid of items, filled cells show things or barriers that have been discovered. A graphical display that is directly simulated and generates output that is delivered to an external display environment can be this type of visualization.

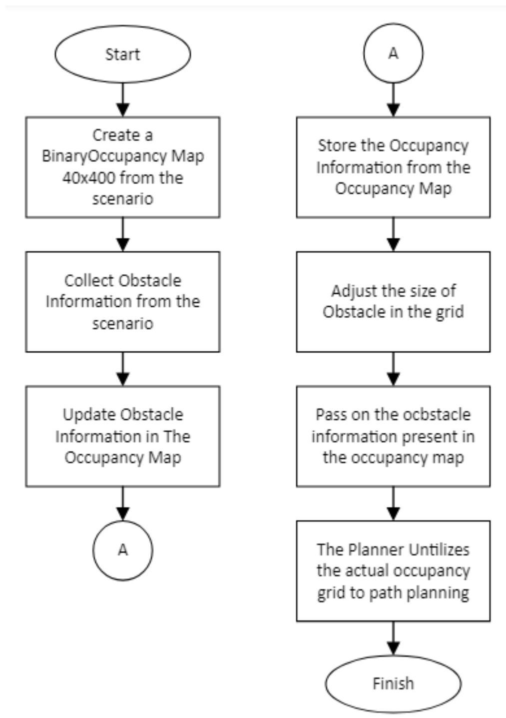


Fig. 8. Dynamic Map Generating Flowchart [6]

Generate a discrete grid that contains information about the environment and cars around the ego vehicle. Before designing a path with the A* algorithmthe output produced by the A* Algorithm, which includes the grid, must be adjusted as a consequence of establishing a scenario path that has been created in the numerical driving scenario designer program. The binary occupancy grid function will be used to build the grid [6]. An illustration of the occupancy grid maker's flow is shown in Fig. 8. Researchers can adjust the occupancy grid as needed with the use of this flowchart. A reference path variable from the A* algorithm path design enters the predictive control signal as a result of the created grid, and this variable is then used to turn the generated reference signal into a control signal.

F. Predictive Control System

Predictive control systems originated in the late 1970s and have come a long way since then, both in research and industry. The term predictive control is used to obtain a control signal by minimizing a cost function with constraints. The underlying ideas in each type of predictive control are [5].

1) Use of process models to predict process output at a future time (horizon).
2) Calculation of the control sequence minimizes the cost function with constraints.

E - ISSN : 2686-6269

3) In the preceding strategy, at each sampling time (at time *k*), the horizon is moved to the next sampling time (at time k+1) by involving the use of the first control signal (u(k)) to control the process and the above steps are repeated using the latest information.

Predictive Control System is a control strategy based on solving an open-loop horizon optimal control problem. The controller is based on a discrete-time model of the system that is used to predict the system response to control inputs along a number of discrete time steps, the prediction horizon. The predicted system response will be compared with the reference and tracking error. Control inputs are sorted using a performance cost function. The future behavior of the controlled system is optimized by selecting the best acceptable control input and minimizing the cost function with constraints. Predictive control uses the Receding Horizon Principal control scheme [6].

Model Predictive Control (MPC) is a discrete-time control technique whose main goal is to map out a trajectory of future control inputs or manipulated variables in order to maximize plant output performance in the future within a constrained time frame. At the beginning of the time window, the plant information must be supplied. The state-space model of the plant serves as the foundation for the design of Model Predictive Control systems [5]. The plant outputs at future instants (prediction horizon) of time are explicitly predicted using the state-space model. After resolving optimization issues with quadratic programming (QP) and minimizing an objective function, MPC determines a series of control inputs. Additionally, MPC employs a receding horizon technique in which the plant receives the first control signal of the sequence determined at each time step, with the horizon being shifted toward the future at each instant.. Preventing violations of input and output constraints, maximizing some output variables while maintaining others within predetermined ranges, and regulating a significant number of system variables in the case that a sensor or actuator is unavailable are some of the key goals of an MPC controller. The Model Predictive Controller process consists of three essential steps: feedback correction, optimization, and prediction model.

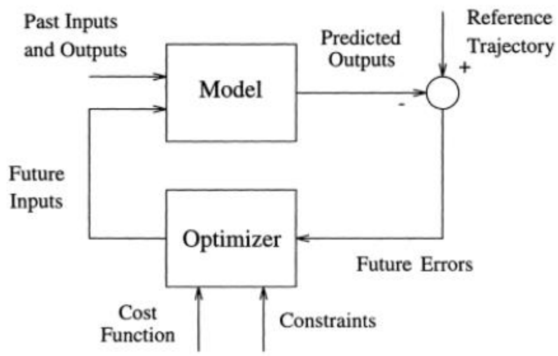


Fig.9. Structure Of How Predictive Control System Works

A basic structure is used to implement predictive control, as shown in Fig.9 [13]. A model is used to predict future outputs based on past and current values and proposed future optimal actions. The optimizer calculates this action by considering a cost function with constraints. The predictive control system uses a mathematical modeling system using Equations (1) to (5) as an Ego Vehicle model. The predictive control system, especially the optimizer process, can be done by minimizing the cost function, as shown in Equations (9) to (11) with constraints [5]. Where, the $J(k)$ variable is the cost function, the N variable is horizon prediction, the Q variable is the state weight matrix, the R variable is the control weight matrix, the u variable is input control, the x variable is vehicle position, and the $a(m|n)$ variable states the value of a at the moment m that was predicted at the time n.

$$\min_{u} J(k) \tag{9}$$

$$J(k) = \sum_{j=1}^{N} \tilde{x}^T(k+j|k)Q\tilde{x}(k+j|k) + \tilde{u}^T(k+j-1|k)R\tilde{u}(k+j-1|k) \tag{10}$$

$$J(k) = \bar{x}^T(k+1)Q\bar{x}(k+1) + \bar{u}^T(k)R\bar{u}(k) \tag{11}$$

constraints,

$$\tilde{x}(k) = x(k) - x_r(k)$$
$$\tilde{u}(k) = u(k) - u_r(k)$$

$$\bar{x}(k+1) = \begin{bmatrix} \tilde{x}(k+1|k) \\ \tilde{x}(k+2|k) \\ \vdots \\ \tilde{x}(k+N|k) \end{bmatrix} ; \bar{u}(k) = \begin{bmatrix} \tilde{u}(k|k) \\ \tilde{x}(k+1|k) \\ \vdots \\ \tilde{x}(k+N-1|k) \end{bmatrix} ; Q = \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{bmatrix} ; R = \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R \end{bmatrix}$$

$$\bar{x}(k+1) = A(k)\tilde{x}(k|k) + B(k)\tilde{u}(k)$$

G. Root Mean Square Error Method

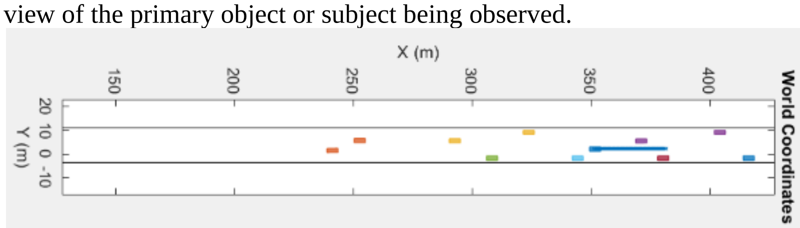
The Root Mean Square Error (RMSE) method is an important error measure for use in statistics and machine learning. RMSE is calculated by taking the square root of the mean square of the difference between the predicted value and the true value. A low RMSE value indicates that a model can accurately predict the true value, as shown in Equation (12) [5]. Where the n variable is the amount of data, the y_i variable is the actual value, and the $\hat{y}_i$ variable is the reference value.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (12)$$

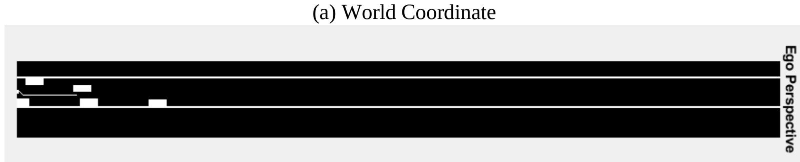
III. RESULT AND DISCUSSION

A. Path Planner

The binary occupancy grid map is helpful for collision avoidance and freeway path optimization in path planning algorithms. The occupancy map uses a binary value of zero or one to indicate the occupancy state. One (1) or "true" denotes an occupied site, whereas zero (0) or "false" denotes a free location. The world coordinate and ego perspective—the outcomes of the occupancy grid—are displayed in Fig. 10. When expressing an object's or point's location in three-dimensional space (x, y, and z) using a fixed or absolute reference, the term "coordinate" refers to a coordinate system. The car is shown in Fig. 10 (a) at coordinates (258, 5.5). It was driven to avoid the car in front of it and came to a stop at coordinates (351, 2). The self-driving car travels for ten seconds. In contrast, the ego perspective in Figure 10 (b) refers to the viewpoint or point of view from the autonomous vehicle's or the subject's point of view. The ego viewpoint, which is typically the reference point when sensing or focussing, characterizes the perspective or point of view of the primary object or subject being observed.



(a) World Coordinate



(b) Ego Perspective

Fig. 10. The Results Of The Occupancy Grid

Table IV above is a table of vehicle position results generated based on world coordinates.. This table, which presents the overall simulation results of the trajectory created by the autonomous vehicle using the A* algorithm, yields the results of the F-value (cost necessary to move). The car is thought to be traveling at a steady 15 m/s. Every one second, the table displays the autonomous vehicle's lateral position data. This indicates that the average distance traveled with the A* trajectory outcome for 10 seconds is 139 meters, and the average estimated cost for the entire run is 2475.

TABLE IV
VEHICLE POSITION RESULTS

TIME(S)	X0 (INITIAL X VEHICLE POINT)	XF (VEHICLE POINT X END)	Y0 (VEHICLE POINT Y INITIAL)	YF (VEHICLE POINT Y END)	F(N)
0	204	206	2	2	236
1	206	216	4	4	228
2	216	236	5	5	218
3	236	251	5.5	5.5	223
4	251	269	5.5	5.5	220
5	269	284	5.5	5.5	223
6	284	305	5.5	5.5	218
7	305	313	5.5	5.5	231
8	313	327	4	4	225
9	327	343	3	3	223
10	343	352	2	2	230

DOI: http://dx.doi.org/10.25139/ijair.v6i1.8141

B. Predictive Control System Results

The horizon open-loop optimum control issue is the foundation of the predictive control system, a control approach. The predictive control is implemented using the same general framework as in Fig. 11.

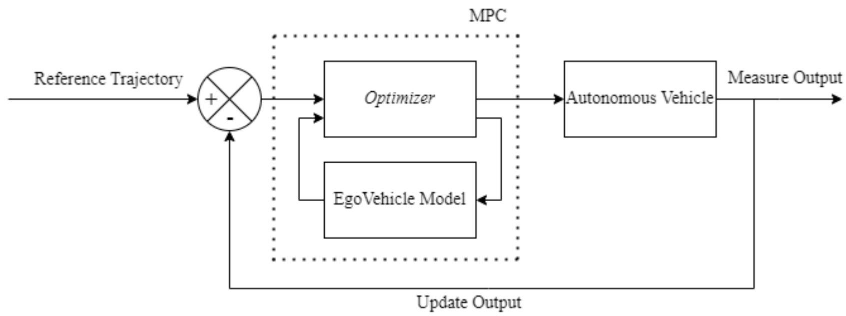


Fig.11. Basic of Structure Predictive Control System

Based on historical, present, and suggested future ideal activities, a model value is used to forecast future outputs. The cost function with constraints is taken into account by the optimizer while calculating this action. Figure 11 [5] is the basic structure of how the predictive control system works. In the predictive control system, an optimization event occurs to produce a good value. The results of this optimization produce a more optimal value. The predictive control system uses a dynamic modeling system specifically using the state space as in Equation (13) and (14). Where, the X_{k+1} variable is state variable -$k+1$, the A_k variable is the state matrix, the X_k variable is state variable -k, the B_k variable is input matrix, the U_k variable is input, the Y_k variable is output, and the C_k variable is the output matrix.

$$X_{k+1} = A_k X_k + B_k U_k \tag{13}$$

$$Y_k = C_k X_k \tag{14}$$

The predictive control system, especially the process *optimizer*, can be done by minimizing the *cost function* as in Equation (15) and (16) with *constraints* [13]. Where, the $U \triangleq \{\Delta u_k, \ldots, \Delta u_{k+N_{u-1}}\}$ variable is predictive *horizon* of inputs to-k until to-N_u, the J variable is cost function, the U variable is input control, the $x(k)$ variable is state variable now -k, the N_y variable is predictive horizon input, the N_u variable is predictive horizon output, the $y_{k+i|k}$ variable is output at time -k, the $r_{k+i|k}$ variable is tracking trajectory setpoint, the Q variable is output weight, the $\Delta u'_{k+i|k}$variable is input weight, the R variable is input weight, and the $\Delta u_{k+i|k}$ is predictive input.

$$\min_{U \triangleq \{\Delta u_k, \ldots, \Delta u_{k+N_{u-1}}\}} \{J(U, x(k))\} \tag{15}$$

$$= \sum_{i=0}^{N_y - 1} \left[(y_{k+i|k} - r_{k+i|k})' Q (y_{k+i|k} - r_{k+i|k}) + \Delta u'_{k+i|k} R \Delta u_{k+i|k} \right] \tag{16}$$

constraints,

$N_u = N_y$
$u_k \in U \; dan \; u_{k+i} \in [umax_{min}],\ \Delta u_{k+i} \in [umax_{min}],\ untuk \; i = 0,1, \ldots, N_u - 1$
$y_k \in Y \; dan \; y_{k+i} \in [ymax_{min}],\ untuk \; i = 0,1, \ldots, N_y - 1$
$\Delta u_k = u_k - u_{k-1} \in \Delta U \; dan \Delta u_{k+i} = 0,\ untuk \; i \geq N_u$
$x_{k|k} = x(k), x_{k+i+1|k} = A(k) x_{k+i|k} + B(k) u_{k+i|k} = u_{k+i-1|k} + \Delta u_{k+i|k}, y_{k+i|k} = C(k) x_{k+i|k}$

Table VI shows the parameters used for predictive control values. These parameters adjust to the needs to produce good values. Then, from these parameters] the results of the predictive control system in the form of predictive control entry signals in the form of reference trajectory signals are displayed in the graph in Fig.12, an image of the simulation results in the form of a trajectory reference graph. This graph is the result of the trajectory based on the design made.

TABLE VI
PARAMETERS PREDICTIVE CONTROL

PARAMETERS	VALUE	DESCRIPTION
N	30	Horizon
Q	5	Output Weight
R	1	Input Weight

E - ISSN : 2686-6269

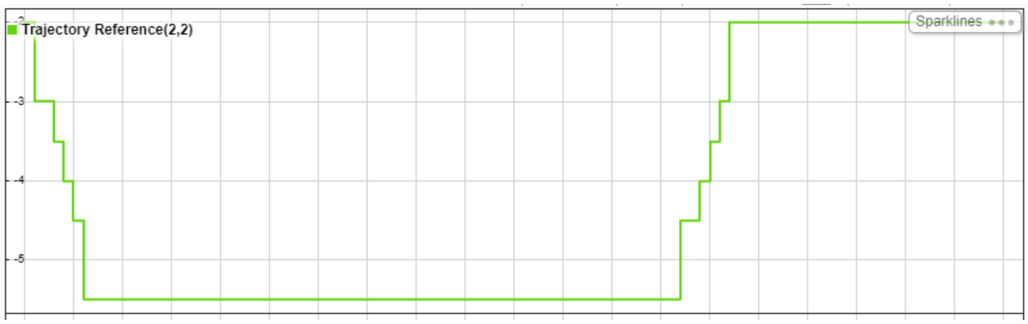


Fig. 12. Trajectory Reference Graph

A vehicle motion graph is produced by the predictive control system processing the data derived from the vehicle's lateral position. A lateral graph of the vehicle position represents the simulation findings in Fig. 13. The vehicle position produced by this system is displayed on this graph. This graph facilitates the process of graphically representing the vehicle's position. When the lead vehicle's relative distance from the ego vehicle exceeds the safe distance, MPC tracks the directed ego vehicle velocity. The MPC regulates the headway when the relative distance is too near to the safe distance. The ego car does an obstacle avoidance maneuver and MPC tracks the intended lateral position when it detects an obstruction and wishes to pass the lead vehicle.

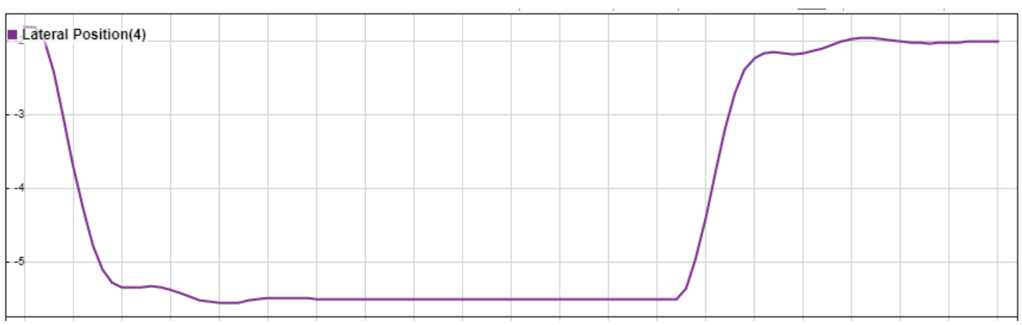


Fig.13. Lateral Position of The Vehicle

The steering angle depicted in Figure 14 represents the predictive control that is produced as a result. within the graph of steering angle. Autonomous vehicles executing lane change actions (maneuvers) demonstrate an increase in amplitude at the 0th and 7th seconds due to the steering angle acting as an actuator in response to the control signal produced by the predictive control system.

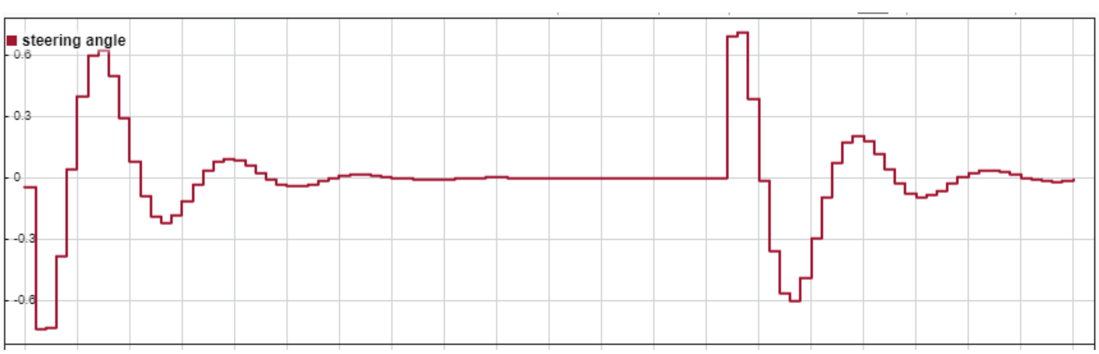


Fig.14. Steering Angle Graph

A graph showing the outcomes of comparing the vehicle's lateral position value with the trajectory reference is shown in Figure 15. The lateral value of the vehicle location and the actual value can be compared by looking at the graph. The graph above's results, as seen in the image above, demonstrate how the error value between the vehicle's lateral position and the reference trajectory value is calculated. The lateral vehicle position error, as determined by the Root-Mean Square Error (RMSE) computation, is 0.9681 relative to the reference trajectory value.

DOI: http://dx.doi.org/10.25139/ijair.v6i1.8141

 E - ISSN : 2686-6269

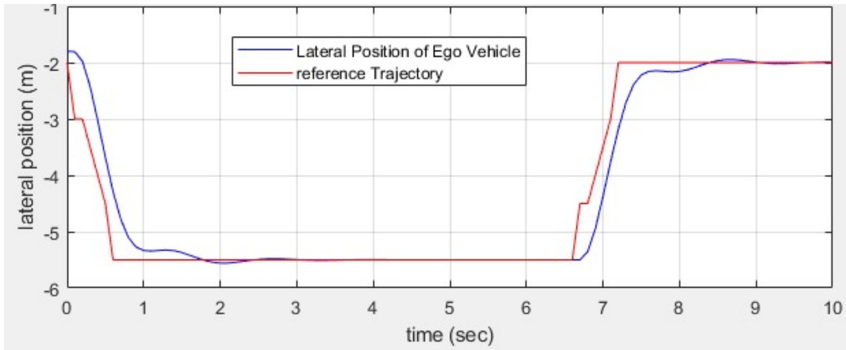


Fig.15. Comparison graph of lateral value of vehicle position with trajectory reference

IV. CONCLUSION

Based on the description that has been explained in the previous chapter, the conclusion that is then obtained in this determination is that the predictive control system is proven to be successful in carrying out its control action on lane changes. It is proven successful because it can produce optimal values in the lane change system. The resulting graph is in the form of a steering angle graph and a lateral graph of the vehicle position. Comparing the value of the trajectory reference results with the lateral position results is done by calculating the error value between the lateral position of the vehicle and the reference trajectory value. This method is numerically simulated using RMSE (Root-Mean Square Error) calculation so that an error of 0.9681 is obtained for the lateral position of the vehicle against the reference trajectory value. This error value proves that the control system is optimal.

ACKNOWLEDGMENTS

The authors extend their gratitude to a number of people and institutions, notably the Engineering Physics autonomous vehicle research team and the Embedded and Cyber-Physical Systems (ECS) laboratory. We especially thank the Republic of Indonesia's Ministry of Research, Technology, and Higher Education for providing essential research support (Contact Number: 1184/PKS/ITS/2024). Their efforts, assistance with funding, and unwavering support made this study project much better.

REFERENCES

[1] S. Thrun et al., "Stanley : The Robot that Won the DARPA Grand Challenge," vol. 23, no. June, pp. 661–692, 2006.W.-K. Chen, Linear Networks and Systems (Book style). Belmont, CA: Wadsworth (1993) 123–135.

[2] C. Urmson et al., "Autonomous Driving in Urban Environments : Boss and the Urban Challenge," vol. 25, no. February, pp. 425–466, 2008..

[3] M. Almaged and A. Mahmood, "Autonomous Model Vehicles : Signal Conditioning and Digital Control Design," no. September 2018, 2019.

[4] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, "An MPC algorithm with combined speed and steering control for obstacle avoidance in autonomous ground vehicles," ASME 2015 Dyn. Syst. Control Conf. DSCC 2015, vol. 3, 2015.

[5] Arfianti, Rida Ayu (2024) Perancangan Sistem Kontrol untuk Obstacle Avoidance pada Autonomous Vehicle dengan Kontrol Prediktif Berbasis Robot Operating System (ROS). Masters thesis, Institut Teknologi Sepuluh Nopember.

[6] Permatasari, Shervind Maharani (2023) Perancangan Sistem Trajectory Generation Pada Autonomous Vehicle Menggunakan A* Algorithm. Other thesis, Institut Teknologi Sepuluh Nopember.

[7] Hu, X., Chen, L., Tang, B., Cao, D., & He, H. (2018). Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. Mechanical Systems and Signal Processing, 100, 482–500. https://doi.org/10.1016/j.ymssp.2017.07.019

[8] Li, X., Sun, Z., Cao, D., He, Z., & Zhu, Q. (2016). Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications. IEEE/ASME Transactions on Mechatronics, 21(2), 740–753. https://doi.org/10.1109/TMECH.2015.2493980

[9] Liu, J., Jayakumar, P., Stein, J. L., & Ersal, T. (2015). An MPC algorithm with combined speed and steering control for obstacle avoidance in autonomous ground vehicles. ASME 2015 Dynamic Systems and Control Conference, DSCC 2015, 3. https://doi.org/10.1115/DSCC2015-9747

[10] Liu, J., Jayakumar, P., Stein, J. L., & Ersal, T. (2019). Improving the robustness of an MPC-based obstacle avoidance algorithm to parametric uncertainty using worst-case scenarios. Vehicle System Dynamics, 57(6), 874–913. https://doi.org/10.1080/00423114.2018.1492141

[11] S. Brown, M. Lee, dan C. Wilson, "Trajectory Planning for Autonomous Vehicles Using the Kinematic Bicycle Model," IEEE Transactions on Intelligent Transportation Systems,

[12] C. Huang, F. Naghdy and H. Du, "Model predictive control-based lane change control system for an autonomous vehicle," 2016 IEEE Region 10 Conference (TENCON), Singapore, 2016, pp. 3349-3354, doi: 10.1109/TENCON.2016.7848673. keywords: {Wheels;Actuators;Radio frequency;Feedforward neural networks;Lane change control;model predictive control;vehicle safety;autonomous vehicle},