Development A Real Time Stationary Obstacles Avoidance Scheme for Autonomous Vehicle Navigation System

Abdalla Al-Salah, Saleh Zein-Sabatto
Department of Electrical and Computer Engineering
Tennessee State University
Nashville, TN, USA

Abstract

Motion planning for autonomous vehicles in urban environments is an important and challenging task due to the vehicle dynamic constraints, involvement of stationary and moving obstacles, other vehicles and unpredictable pedestrian movements. The proposed algorithm will present a real time motion planning scheme of autonomous vehicles. Also, will allow autonomous vehicles to avoid collisions with multi stationary obstacles. A linear model predictive control (MPC) algorithm using Laguerre Functions is developed and used where the controller manipulated variables are the acceleration and the front wheels steering angle. Knowledge of the surrounding environment, the state of the autonomous vehicle and other constraints of the MPC are incorporated in the prediction model of the controller to predict safe trajectory for the vehicle and to avoid collision with other obstacles. The proposed algorithm will improve autonomous vehicles navigation behavior in unstructured environments.

Keywords- autonomous vehicles; model predictive control (MPC); Laguerre function; path tracking; obstacle avoidance.

1. Introduction

The new technologies in communication and robotics have made significant leaps in bringing computerization into the transportation industries. Autonomous vehicle technologies become the most heavily researched automotive technologies. Therefore, most automobile manufactories' goal is to create a fully autonomous vehicle that can navigate itself on highways, urban environments, and in any weather circumstances. The majority of the new automobile models include features such as adaptive cruise control and parking assist systems that allow them to steer themselves into parking spaces. [1, 2]. Autonomous vehicles technology aims to reduce crashes, energy consumption, lower harmful emissions, and congestion while at the same time provide mobility to the elderly and disabled, and increase road capacity. Autonomous vehicles use a variety of techniques to detect their surroundings, such as radar, laser, LiDAR light, GPS, odometers and computer vision. Advanced control systems of the autonomous vehicles are capable to interpret sensory data to distinguish between the surrounded obstacles on the road, in order to identify optimal navigation paths to the desired destination. The most technological challenges autonomous vehicles face are sensor perception and decision-making under conditions of uncertainty. Since the autonomous vehicle's technology lies on variety of used sensors and algorithms, to detect, recognize, locate objects, and path planning [1, 2-4].

Many navigation and path planning techniques of mobile robotics have been used, modified, and applied to autonomous vehicles. The navigation problem can be decomposed into three subtasks: mapping and modeling the environment, path planning, and path traversal without colliding.

Path planning is an important subtask of autonomous vehicle's navigation problem. Path planning techniques were categorized into four groups, according to their applications in automated driving: graph search such as (Dijkstra algorithm, A-star algorithm, State Lattice algorithm), sampling such as (Probabilistic Roadmap Method (PRM), Rapidly-exploring Random Tree (RRT)), interpolating such as (Lines and circles, Clothoid Curves, Polynomial Curves, B´ezier Curves) and numerical optimization such as function optimization that has been implemented to improve the potential field method (PFM) [3, 4]. Also, path planning algorithms are described as belonging into classical or intelligent algorithms. The classical algorithms use heuristics, fixed conditional rules, to restrict the application of the base algorithm. And the intelligent algorithms use meta-heuristics, changing conditional rules, to optimize an initial solution [5, 6].

Path tracking control systems are used to control autonomous vehicle to follow the provided reference path by the path planning system, while avoiding a stationary obstacle using throttle and steering maneuver. It's a complicated task due to the

 | DOI: 10.20533/ijicss.9781.9083.20346.2020.0009

requirement of tracking accuracy and vehicle dynamic stability simultaneously [7].

This paper presents a model predictive control approach using Laguerre function for autonomous vehicles navigation and stationary obstacle avoidance, the vehicle model is presented in Section II. The considered predictive control model is described in Section III. Simulation results presented in section IV. Finally, conclusion is included in section VI.

2. Vehicle Dynamics Model

Vehicle dynamics refer to vehicle motion in the lateral and longitudinal directions [8]. Lateral dynamics or transverse dynamics are controlled by the vehicle stability analysis, which is referred to the vehicle handling stability and side slip, which caused by the tire lateral force, yawing and roll motion [8, 9]. Longitudinal dynamics concern with the vehicle behavior while driving such as accelerating, braking and ride comfort such as wheels traction on different road surfaces under different conditions [8, 9]. A simple nonlinear model to describe the dynamics of the vehicle using four states is shown below [10].

Dynamics of the lobal X position of the car center is modeled by:

$$xPos = -S * sin\theta * \theta + cos\theta * S \quad (1)$$

Dynamics of the Global Y position of the car center is modeled by:

$$yPos = -S * cos\theta * \theta + sin\theta * S \quad (2)$$

Heading angle of the car (0 when the car facing east, counterclockwise positive) is given by:

$$\theta = (tan(\delta)/cL) * S + (S * (tan(\delta)^2) + 1)/cL) * \delta \quad (3)$$

cL refers to car length. Finally, the speed (S) of the car positive, it is updated using the following equation:

$$S = 0.5 * T \quad (4)$$

There are two manipulated/ inputs variables to the above vehicle model:

T – Throttle (positive when accelerating, negative when decelerating).

δ - Steering angle (0 when aligned with car, counterclockwise positive).

3. Development of Model Predictive Control

Model predictive control (MPC) is an important control technique that used to control process while respecting a set of constraints for difficult multivariable control problems; therefore, it's widely used, due to its ability to achieve optimal control, incorporate constraints and easy extensions. MPC has three approaches, with a unique structure for each model. Finite impulse response (FIR) models and step response models, dynamic matrix control (DMC), and the quadratic DMC [11, 12].

MPC is a suitable control methodology for autonomous vehicles. Because it considers the future shape of the path while computing the control signal during prediction horizon overall circumstances. Also, it takes into account the dynamic limitation of the vehicle while implementing the vehicle constraints [7, 8]. There are several approaches to designing a Model Predictive Controller.

A. State-Space Approach

State-space approach is the selected model in designing the control system in this paper [12]. The controlled system described by the following state space equation:

$$x_m(j+1) = A_m x_m(j) + B_m u(j) \quad (5)$$

$$y(j) = C_m x_m(j) \quad (6)$$

Where $u(j)$ is the manipulated variable or input variable; $y(j)$ is the output; x_m is the state variable vector; and j is sample time.

The changes in the state variable are defined as:

$$\Delta x_m(j) = x_m(j) - x_m(j-1) \quad (7)$$

Hence, the state-space difference equation is written as:

$$\Delta x_m(j+1) = A_m \Delta x_m(j) + B_m \Delta u(j) \quad (8)$$

By concatenating $\Delta x_m(j)$ with the output $y(j)$ this will lead to a new state variable and output vector as:

$$x(j) = [\Delta x_m(j)^T \; y(j)^T]^T \quad (9)$$

$$y(j+1) = C_m A_m \Delta x_m(j) + y(j) + C_m B_m \Delta u(j) \quad (10)$$

Putting equations 9 and 10 together a new state-space model will be constructed as follow:

$$\overbrace{\begin{bmatrix}\Delta x_m(j+1)\\ y(j+1)\end{bmatrix}}^{x(j+1)} = \overbrace{\begin{bmatrix}A_m & O_m^T\\ C_m A_m & [I]\end{bmatrix}}^{A} \overbrace{\begin{bmatrix}\Delta x_m(j)\\ y(j)\end{bmatrix}}^{x(j)} + \overbrace{\begin{bmatrix}B_m\\ C_m B_m\end{bmatrix}}^{B} \Delta u(k)$$

$$y(j) = \overbrace{[O_m \quad I\,]}^{C} \begin{bmatrix}\Delta x_m(j)\\ y(j)\end{bmatrix}$$

Where $O_m = \overbrace{[00 \;\dots\; 0\,]}^{n1}$, $A_m, B_m and\ C_m$ are the augmented matrixes of the model.

The next step in the predictive control system design procedures is to calculate the predicted plant output within optimization window *Np* as the number of samples, with the future control signal as adjustable variables. Based on the state-space augmented model (*A*, *B*, *C*), the future state variables are calculated using the future control parameters set sequentially.

$$x(j_i + 1|\, j_i) = Ax(j_i) + B\Delta u(j_i)$$

$$x(j_i + 2|\, j_i) = Ax(j_i + 1|\, j_i) + B\Delta u(j_i + 1|\, j_i) \\ = A^2 x(j_i) + AB\Delta u(j_i + 1|\, j_i) + B\Delta u(j_i + 1)$$

$\vdots$

$$x(j_i + N_p|j_i) = A^{Np}x(j_i) + A^{Np-1}B\Delta u(j_i) + A^{Np-2}B\Delta u(j_i + 1) + A^{Np-Nc}B\Delta u(j_i + N_c - 1) \quad (11)$$

From the predicted state variables, the predicted output variables are calculated.

$$y(j_i + 1|\, j_i) = CAx(j_i) + CB\Delta u(j_i)$$

$$y(j_i + 2|\, j_i) = CA^2 x(j_i) + CAB\Delta u(j_i + 1|\, j_i) + CB\Delta u(j_i + 1)$$

$\vdots$

$$y(j_i + N_p|j_i) = CA^{Np}x(j_i) + CA^{Np-1}B\Delta u(j_i) + CA^{Np-2}B\Delta u(j_i + 1) + CA^{Np-Nc}B\Delta u(j_i + N_c - 1) \quad (12)$$

Define vectors *Y* and ΔU, where *Y* is the predicted output data vector, and ΔU is the parameter vector for the control sequence as:

$$Y = [\, y(j_i + 1|\, j_i) \quad y(j_i + 2|\, j_i) \;\dots\; y(j_i + N_p|\, j_i)\,]^T \quad (13)$$

$$\Delta U = [\Delta u(j_i) \;\; \Delta u(j_i + 1) \;\dots\; \Delta u(j_i + N_c - 1)]^T \quad (14)$$

$$Y = Fx(j_i) + \phi\, \Delta U \quad (15)$$

$$F = \begin{bmatrix} CA\\ CA^2\\ CA^3\\ \vdots\\ CA^{Np}\end{bmatrix},\ \phi = \begin{bmatrix} CB & 0 & 0 & \dots & 0\\ CAB & CB & 0 & \dots & 0\\ CA^2B & CAB & CB & \dots & 0\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ CA^{Np-1}B & CA^{Np-2}B & CA^{Np-3}B & \dots & CA^{Np-Nc}B\end{bmatrix}$$

Hence, *F* and ϕ are matrices used in the prediction equation.

B. Use of Laguerre Function

The principle idea of using Laguerre functions is to reformulating the predictive control problem, reduce the MPC computational complexity, and simplify the solutions, by introducing a sequence of discrete orthonormal functions with few fitting coefficients into the design procedure [7, 12].
The discrete time sequence of Laguerre networks is described by z- transform expression as follow:

$$\Gamma_N(z) = \frac{\sqrt{1-p^2}}{1-pz^{-1}} * \left(\frac{z^{-1}-p}{1-pz^{-1}}\right)^{nT-1} \quad (16)$$

This set of discrete-time Laguerre functions are expressed in a vector form as

$$Lg_i(j)^T = [\, lg_1^i(j) \quad lg_2^i(j) \;\dots\; lg_{Ni}^i(j)\,] \quad (17)$$

The set of discrete-time Laguerre functions satisfies the following difference equation

$$Lg\,(j+1) = A_{lg}\, Lg(j) \quad (18)$$

Where matrix A_{lg} is ($nT \times nT$), *nT* is the number of

terms. The initial condition of Laguerre functions is given by:

$$Lg(0)^T = \sqrt{\beta}\left[1 \;\; -p \quad p^2 \quad -p^3 \dots \left(-1^{nT-1}\, p^{nT-1}\right)\right] \quad (19)$$

Where "p" is the pole, and it's between $0 \le p < 1$ and $\beta = (1 - p^2)$. Note, "p" *and* "nT" are denoted for the controller stability.
In case $nT = 4$ then:

$$A_{lg} = \begin{bmatrix} I & 0 & 0 & 0\\ \beta & \alpha & 0 & 0\\ -\alpha\beta & \beta & \alpha & 0\\ \alpha^2\beta & -\alpha\beta & \beta & \alpha\end{bmatrix},\ L(0) = \sqrt{\beta}\begin{bmatrix}1\\ -\alpha\\ \alpha^2\\ -\alpha^3\end{bmatrix}$$

C. Controller Performance Optimization

The purpose of using cost function to optimize the performance of the controller is to minimize the error between the reference signal and the output signal [12]. The optimal controller, achieving this task called discrete-time linear quadratic regulator, which is based on three steps: (1) use algebraic Riccati equation to solve for the Riccati matrix P_∞ using (21), (2) Using P_∞ to construct the state feedback gain matrix *G* using (22) and, (3) gain *G* is used to define the control signal and the state variable using (24). The discrete-time algebraic Riccati equation is:

$$\mathrm{P}(\infty) = A_m\{\mathrm{P}(\infty) - \mathrm{P}(\infty)C_m^T \\ (\Gamma + C_m\mathrm{P}(\infty)C_m^T)^{-1}\, C_m P_m(\infty)\}\, A_m^T + \Theta \quad (20)$$

$$A^T(P_\infty - P_\infty B(R + B^T P_\infty B)^{-1}B^T P_\infty)A + Q - P_\infty = 0 \quad (21)$$

$$G = (R + B^T P_\infty B)^{-1} B^T P_\infty A \quad (22)$$

R and Q are the weight matrices in the predictive control cost function.

The system control law is given by:

$$\Delta u_i(j) = Lg_i(j)^T \eta_i \quad (23)$$

$$\Delta u(j_i) = \begin{bmatrix} Lg_1(0)^T & O_2^T & \cdots & O_m^T \\ O_1^T & Lg_2(0)^T & \cdots & O_m^T \\ \vdots & \vdots & \ddots & \vdots \\ O_1^T & O_2^T & \cdots & Lg_m(0)^T \end{bmatrix} \eta$$

The $Lg_i(j)$ and η are the Laguerre network description of the *ith* control.
The control change $\Delta u(j)$ in the form of linear state feedback control is written as:

$$\Delta u(j) = -G_{mpc}\, x(j) \quad (24)$$

The state feedback controller gain matrix is:

$$G_{mpc} = (Lg0)^T \Omega^{-1} \Psi \quad (25)$$

$$G = \begin{bmatrix} Lg_1(0)^T & O_2^T & \cdots & O_m^T \\ O_1^T & Lg_2(0)^T & \cdots & O_m^T \\ \vdots & \vdots & \ddots & \vdots \\ O_1^T & O_2^T & \cdots & Lg_m(0)^T \end{bmatrix} \Omega^{-1}\Psi$$

The optimal solution of the parameter vector (η) is:

$$\eta = -[\textstyle\sum_{n=1}^{Np} \phi(n) Q \phi(n)^T + R_L]^{-1} (\sum_{n=1}^{Np} \phi(n) Q A^m) x(j_i) \quad (26)$$

Where n is the number of inputs. If we defined Phi, Omega and PSI as follows:

$$\phi(n)_r^T = \textstyle\sum_{i=0}^{n-1} A^{n-i-1} B_r\, Lg_r(i)^T \quad (27)$$

$$\Omega = \textstyle\sum_{n=1}^{Np} \phi(n) Q \phi(n)^T + R_{Lg} \quad (28)$$

$$\Psi = \textstyle\sum_{n=1}^{Np} \phi(n) Q A^n \quad (29)$$

Where Phi (ϕ) is matrix used in the prediction equation, Omega (Ω) and PSI (Ψ) are matrices in the cost of predictive control.

Putting equations (28) and (29) together will lead to the following solution:

$$\eta = -\Omega^{-1}\Psi x(j_i) \quad (30)$$

Equation (30) provides the optimal solution of the parameter vector η, when the system does not apply any control constraints either on the input or output signals.

Finally, the cost function equation used to optimize the controller performance is:

$$J = \frac{1}{2}\eta^T \Omega \eta + \eta^T \Psi\, x(j_i) \quad (31)$$

D. Constrained Control Using Laguerre Functions

As explained earlier, one of the most important features of using MPC is the ability to handle hard constraints in the controller model design. The control constraints require real time optimization using quadratic method. Where the constraints are decomposed into upper limit and lower limit with opposite sign.
Three types of constraints could be applied on the MPC model design.

1- Constraints on the rate of change of the control variable $\Delta u(j)$.

$$\Delta u_m^{min} \le \Delta u_m(j) \le \Delta u_m^{max} \quad (32)$$

Using Laguerre functions in the design, the incremental control signal represented as *Δu(ji + m)* where m = 0, 1, 2, 3, ... for a multi-input system. The user is free to use any possible number of constraints imposed on the solution. So, Δu_{min} and Δu_{max} are vectors containing individual limits for each control variable. the matrix form for the constraints at a future sample (m):

$$\Delta u^{min} \le \begin{bmatrix} Lg_1(m)^T & O_2^T & \cdots & O_m^T \\ O_1^T & Lg_2(m)^T & \cdots & O_m^T \\ \vdots & \vdots & \ddots & \vdots \\ O_1^T & O_2^T & \cdots & Lg_m(m)^T \end{bmatrix} \eta \le \Delta u^{max}$$

o_j denotes a zero vector that has the same dimension as $Lg_j(m)$.

2- Constraints on the Control Signal Amplitudes.

$$u_m^{min} \le u_m(j) \le u_m^{max} \quad (33)$$

The control signal increment is $u(j) = \sum_{i=0}^{j-1} \Delta u(i)$. The inequality constraint for the future time j = 1, 2, ...

$$u^{min} \le \begin{bmatrix} \sum_{i=0}^{j-1} Lg_1(i)^T & O_2^T & \cdots & O_m^T \\ O_1^T & \sum_{i=0}^{j-1} Lg_2(i)^T & \cdots & O_m^T \\ \vdots & \vdots & \ddots & \vdots \\ O_1^T & O_2^T & \cdots & \sum_{i=0}^{j-1} Lg_m(i)^T \end{bmatrix} \eta + u(j_i - 1) \le u^{max}$$

3- Constraints on the output.

$$Y^{min} \le Fx(j_i) + \Phi \Delta U \le Y^{max} \quad (34)$$

After formulating the constraints as a portion of the controller design requirements, then these constraints have to be translated to linear inequalities and link them with the original model's predictive control problem. The main purpose of this task is to determine the constrained variables using ΔU parameter vector as that used in the predictive control design. Therefore, based the parameter vector ΔU constraints are expressed in a set of linear equations. The vector ΔU is often called a decision variable.

$$\Delta U^{min} \leq \Delta U \leq \Delta U^{max} \quad (35)$$

In a matrix form this becomes

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta U \leq \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}$$

In the case of a manipulated variable constraint

$$\begin{bmatrix} u(ji) \\ u(ji+1) \\ u(ji+2) \\ \vdots \\ u(ji+N_c-1) \end{bmatrix} = \begin{bmatrix} I \\ I \\ I \\ \vdots \\ I \end{bmatrix} u(ji-1) + \begin{bmatrix} I & 0 & 0 & \dots & 0 \\ I & I & 0 & \dots & 0 \\ I & I & I & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ I & I & \dots & I & I \end{bmatrix} \begin{bmatrix} u(ji) \\ u(ji+1) \\ u(ji+2) \\ \vdots \\ u(ji+N_c-1) \end{bmatrix}$$

The constraints have been expressed as the inequalities with linear in the parameter ΔU, then these constraints have to combine with the original cost function J used in the design of predictive control.

In case of applying inequality constraints of the form $H\Delta U \leq \gamma$

$$\begin{bmatrix} H1 \\ H2 \\ H3 \end{bmatrix} \Delta U \leq \begin{bmatrix} \gamma 1 \\ \gamma 2 \\ \gamma 3 \end{bmatrix}$$

Where H *is* a matrix reflecting the constraints as follow:

$$H_1 = \begin{bmatrix} -C_2 \\ C_2 \end{bmatrix} \quad , \quad \gamma_1 = \begin{bmatrix} -U^{min} + C_1 u(j_i - 1) \\ U^{max} - C_1 u(j_i - 1) \end{bmatrix}$$

$$H_2 = \begin{bmatrix} -I \\ I \end{bmatrix} \quad , \quad \gamma_2 = \begin{bmatrix} -\Delta U^{min} \\ \Delta U^{max} \end{bmatrix}$$

$$H_3 = \begin{bmatrix} -\Phi \\ \Phi \end{bmatrix} \quad , \quad \gamma_3 = \begin{bmatrix} -Y^{min} + Fx(j_i) \\ Y^{max} - Fx(j_i) \end{bmatrix}$$

In the minimization process with inequality constraints, the number of decision variables U could be less than the number of constraints. The inequality constraints $H\Delta U \leq \gamma$ may contain active and inactive constraints. An inequality $H_i x \leq \gamma_i$ constraints to be active if $H_i x = \gamma_i$ and inactive if $H_i x < \gamma_i$, where H_i with γ_i form the ith inequality constraint and are the *ith* row of H matrix and the *ith* element of γ vector, respectively. In case to define the active and inactive constraints in terms of the Lagrange multipliers (λ), Kuhn-Tucker conditions are introduced.

$$Ex + F + H^T\lambda = 0, \quad Hx - \gamma \leq 0$$

$$\lambda^T(Hx - \gamma) = 0, \quad \lambda \geq 0$$

In terms of the set of active constraints, *Zact* refers to the index set of active constraints. So, the necessary conditions become

$$Ex + F + \textstyle\sum_{i \in Zact} \lambda_i H_i^T = 0 \quad (36)$$

H_i is the ith row of H matrix, for the *ith* row equation (37) means it's an equality constraint, hence an active constraint. For equation (38) the constraint is satisfied, hence the constraint inactive. In case λ (39) is non-negative the constraint is active, but if λ (40) is zero the constraint is inactive.

$$H_i x + \gamma_i = 0 \quad i \in Z_{act} \quad (37)$$

$$H_i x - \gamma_i < 0 \quad i \notin Z_{act} \quad (38)$$

$$\lambda_i \geq 0 \quad i \in Z_{act} \quad (39)$$

$$\lambda_i = 0 \quad i \notin Z_{act} \quad (40)$$

where E, F, M and γ are matrices and vectors in the quadratic programming problem. E is assumed to be positive and symmetric definite.
Therefore, if inequality constraints are applied to MPC model design the cost function J is written as:

$$J = \frac{1}{2}\eta^T \Omega \eta + \eta^T \Psi\, x(j_i) \quad (41)$$

And the optimal parameter η *is:*

$$\eta = -\Omega \backslash (\Psi * Xf) \ -\Omega \ \backslash H^T \lambda \quad (42)$$

Xf is the state feedback, and the feedback control gain is written as:

$$G = \gamma + H\Omega^{-1}(\Psi * Xf) \quad (43)$$

Where

$$\lambda = -(HE^{-1}H^T)^{-1}(\gamma + HE^{-1}F) \quad (44)$$

And

$$\mathrm{F} = (\Psi * Xf) \quad (45)$$

4. Simulation Results

The simulation is carried out using Matlab in implementing and analyzing the developed MPC controller. In the model predictive controller design the fitting coefficients are chosen as: pole" p*=0.2",* number of terms "*nT=15*", also, the prediction horizon *"Np=22"* and the control horizon *"Nc=10".*

The reference values of the vehicle global lateral longitudinal position considered to be "X=0" and

"Y=130", and the speed reference considered to be constant 20 mph.

Unstructured environment has been constructed with four stationary obstacles with five-meters length and two meters wide, and they were placed in random positions on front the vehicle. Finally, a virtual safe zone has been constructed around the obstacle double the length and width of the vehicle, and is centered on the obstacle, in order to prevent the automated vehicle to get too close while it is avoiding and passing.

The first scenario presents the vehicle following desire path while avoiding a stationary obstacle in order to access the desired destination and maintain speed of 20 mph, without applying any constraints on the vehicle dynamics or controller as is showing in Figure 1.

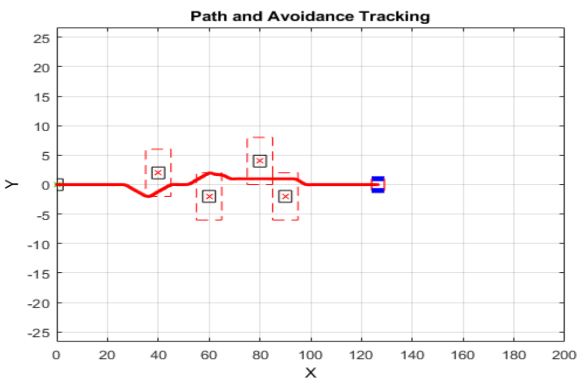

Figure 1. Avoiding Obstacle Scenario

The vehicle starts detecting the obstacles from 20 meters distance and starts the maneuver in order to avoid the obstacles with attempt not to exceed the surrounded safe zone between the dashed lines. The controller has two outputs (control signals) throttle and steering angle, Figure 2 shows the change on the throttle and the total throttle values as is showing the max values (1.06 and 5.2 m/s^2) and min values (-0.26 and -0.54 m/s^2) respectively. And Figure 2 shows the change on the steering angle and the steering angle values as is showing the max values (0.33 and 1.17 radian) and min values (-0.28 and -1.52 radian) respectively.

Based on the first scenario performance and results, the minimum and maximum constraints values were selected as follow:

1- Change over throttle -0.1 and 0.5 m/s^2.
2- Throttle -0.05 and 0.5 m/s^2.
3- Change over steering -0.1 and 0.2 radian.
4- Steering -1 and 1 radian.

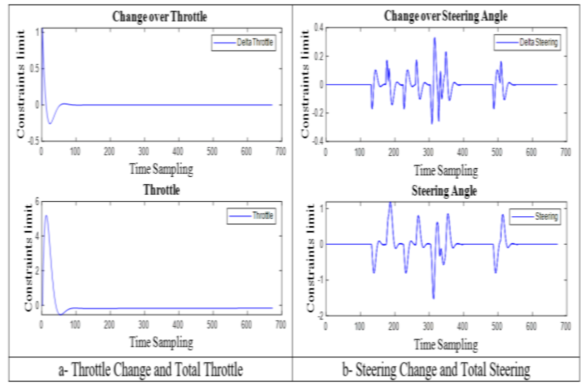

Figure 2. Presents the Control Signals and the Rate of Change Over the Control signals without Applying constraints

The second scenario represents the vehicle following a path while avoiding stationary obstacles in order to access the desired destination, with applying constraints on the vehicle control signals as follow:

Case-1: Apply constraints on the change on the control signal throttle or/and steering (*delta_u*). Figure 3 shows the effect on the change over the throttle after applying min and max constraints (-0.1 and 0.5 m/s^2), the applied changed effects the throttle values with min and max (-2 and 4.6 m/s^2). Figure 3 shows the effect change over the steering after applying min and max constraints (-0.1 and 0.2 radian), and the effects on the steering values with min and max (-0.76 and 0.84 radian).

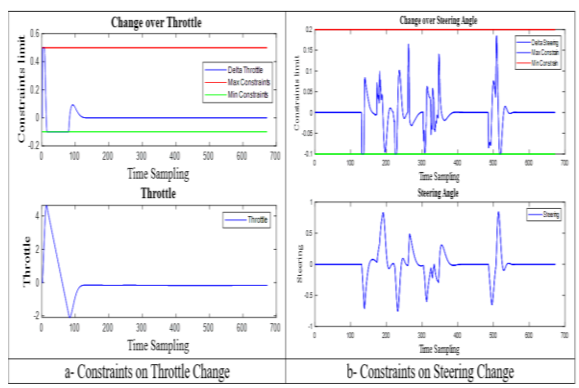

Figure 3. Constraints Applied on the Rate of Change of the Throttle and The Steering Angle

Case-2: Apply constraints on the control signal throttle or/and steering (*u*). Figure 4 shows the effect on the throttle after applying min and max constraints (-0.05 and 0.5 m/s^2) and the effects on the change over the throttle min and max values (-0.023 and 0.38 m/s^2). Figure 4 shows the effect on the steering after applying min and max constraints (-1, 1 radian), which effects the steering min and max values (-0.96 and

0.82 radian) and effects the change over the steering min and max values (-0.22 and 0.25 radian).

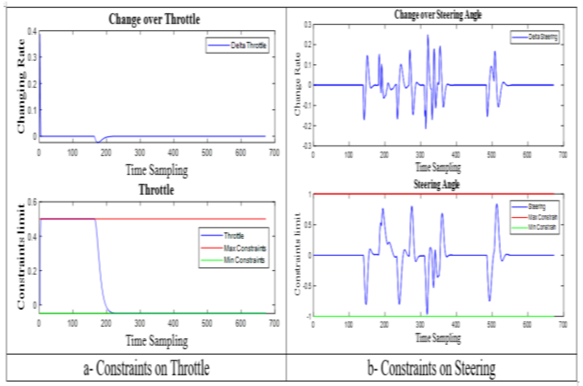

Figure 4. Constraints Applied on the Control Signals Throttle and Steering Angle

Case-3 Apply constraints on both the change on the control signal (*delta_u*) and the control signal (*u*). Figure 5 shows the combination of both constraints on the change of the control signals and the control signals themselves.

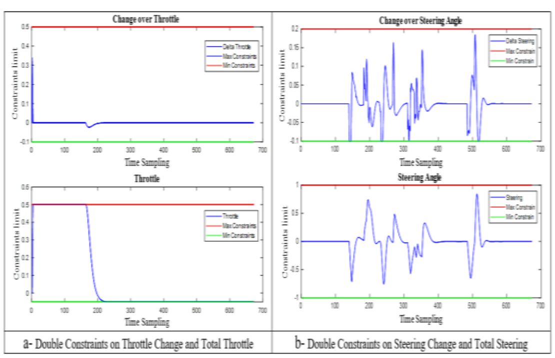

Figure 5. Double Constraints the Control Signals and the Control Signals Change

Therefore, the min and max values of the change on throttle are (-0.023 and 0.33 m/s^2), and the throttle values are (-0.05 and 0.5 m/s^2). The min and max values of the change on steering are (-0.1 and 0.18 radian) and the steering are (-0.76 and 0.82 radian).

5. Conclusion

In this paper, a linear model predictive control strategy was developed for stationary obstacles avoidance of an autonomous driving vehicle in unstructured environment. Simulation results showed the effectiveness of the developed control model in maintaining the vehicle speed and avoiding obstacles encountered by the vehicle. Both non-constraints and constraint scenarios were simulated, and acceptable behaviors of the vehicle were observed in both cases. This approach resulted in an effective prediction strategy by the controller to control the vehicle dynamics using the throttle and steering angle control variables of the vehicle.

6. References

[1] S. A. Bagloee, M. Tavana, M. Asadi and, T. Oliver, "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies", Journal of Modern Transportation, vol. 24, pp. 284-303, August 2016.
[2] D. J. Fagnant and, K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations for capitalizing on self-driven vehicles", Journal of Transportation Research Part A: Policy and Practice, vol. 77, pp. 167-181, July 2015.
[3] D. González, J. Pérez , V. Milanés and, F. Nashashibi, "A Review of Motion Planning Techniques for Automated Vehicles", IEEE Transactions on Intelligent Transportation Systems , vol. 17, pp. 1135 - 1145 , April 2016.
[4] C. C. Lin, W. J. Chuang and, Y. D. Liao, "Path Planning Based on Bezier Curve for Robot Swarms", 2012 Sixth International Conference on Genetic and Evolutionary Computing, August 2012.
[5] J. Pérez, J. Godoy, J. Villagrá and, E. Onieva, "Trajectory generator for autonomous vehicles in urban environments", 2013 IEEE International Conference on Robotics and Automation, October 2013.
[6] S. G. Anavatti, S. L. X. Francis and, M. Garratt, "Path-Planning Modules for Autonomous Vehicles: Current Status and Challenges", 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), October 2015.
[7] B. Zhang, C. Zong, G. Chen and, B. Zhang, "Electrical Vehicle Path Tracking Based Model Predictive Control With a Laguerre Function and Exponential Weight", IEEE Access, vol. 7, pp. 17082 – 17097, January 2019.
[8] R. C. Rafaila, C.F. Caruntu and, G. Livint, "Centralized Model Predictive Control of Autonomous Driving Vehicles with Lyapunov Stability", 20th International Conference on System Theory, Control and Computing (ICSTCC), Sinaia, Romania ,October 2016.
[9] S. Yang, S. Li and, Y. Lu, "An overview on vehicle dynamics", International Journal of Dynamics and Control, December 2013.
[10] Mathworks www.mathworks.com, (Access Date: 12 Dec, 2019).
[11] D. E. Seborg, T. F. Edgar, D. A. Mellichamp and, F. J. Doyle III, "Process dynamics and control," 4th edition, University of California, Santa Barbara, 2016.
[12] L. Wang, "Model Predictive Control System Design and Implementation Using MATLAB", Springer International Publishing, 2009.

