Journal of Engineering Science and Technology Review 12 (3) (2019) 73 - 77

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Trajectory Planning Exploration for Autonomous Terrestrial Vehicles Based on Splines Techniques in Matlab

Robinson Jimenez-Moreno[1,*], Ruben Dario Hernandez[2] and Edilberto Mejia[2]

[1]*Department of Mechatronics Engineering, Militar Nueva Granada University Bogotá, Colombia.*
[2]*Department of Biomedical Engineering, Militar Nueva Granada University Bogotá, Colombia.*

Received 14 January 2019; Accepted 11 June 2019

Abstract

In this article, the development of an algorithm for the implementation of the Splines technique in trajectory planning oriented to autonomous vehicles manned or unmanned in single-lane lane change processes in the double lane for obstacle evasion is exposed. Two different environments are used that emulate short trajectories, in variable environments in time and space, using Splines in a moving vehicle, for the vehicular overpass by lane change. A graphic interface is developed in Matlab® that allows exploring in a visual way the trajectory generated by the vehicle; it is possible to show that it is not possible to define a single path, starting from a particular object.

Keywords: Trajectory Planning, Autonomous Driving Vehicle, Splines, Object detection.

1. Introduction

With the advance of computational algorithms and the arrival of artificial intelligence [1], the implementation of autonomous vehicles [2] represents a high impact factor. However, previous studies in this field, use different techniques such as trajectory planning through routes delimited on highways with magnets [3]. Another of these examples is the trajectory planning of dynamic systems for agile autonomous vehicles [4], where an algorithm is applied to cars whose dynamics are described either by ordinary differential equations or by high-level hybrid representations, applicable to terrestrial robots or autonomous helicopters. On the other hand, these methods are based on the need for control and planning of smooth trajectories, for example, those required in robotic manipulators [5]. In [6], an investigation is presented using the Laplace equations, where constraints are generated for the generation of a potential function on the regions of the configuration space of an end effector.

With the continuous technological advance in the development of hardware and software tools [7], it is increasingly easier to implement techniques such as Splines [8] [9] for the calculation of trajectories in autonomous driving systems. This type of application has a function or mathematical representation of the trajectory of an object, where it is possible to generate an approximation through a new continuous path with "smoothing".

In this way, this article proposes a trajectory planning system oriented to autonomous vehicles unmanned or manned in single-lane lane change processes for double carriageway through the implementation of the Spline technique.

The article is divided into three sections, where the first section, corresponding to methods and materials, exposes the information referring to the design parameters, development environments of the algorithm and the Spline function implemented. The second section presents the results and discussions regarding the implementation of the Spline function on the trajectory vector, as well as its operation through a user interface. Finally, the conclusions obtained respecting to the application of the implemented algorithm are presented.

2. Methods and Materials

The general process for the development of the spline planning system is presented in Fig. 1.

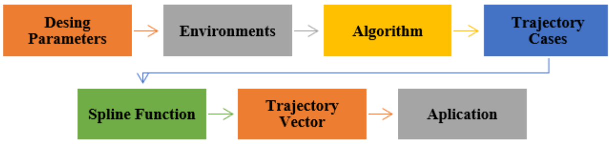


Fig. 1. General block diagram.

Each of the blocks is explained below.

Design Parameters: The Tab. 1 shows the 7 parameters considered in the design. The speed of the vehicle is presumed constant, but when taking into account as a variable value, a linear approach can be implemented as defined by Eq 1., where "*ODD*" is the detection distance of the object in meters, "*Vcar*" is the current vehicle speed expressed in km/h and "*Drmax*" is the maximum reaction time when driving in seconds. These allowed to establish a dynamic use of the system, by varying the detection distance of the object starting from a maximum conduction reaction time of 1s [10] [11].

*E-mail address: Robinson.jimenez@unimilitar.edu.co

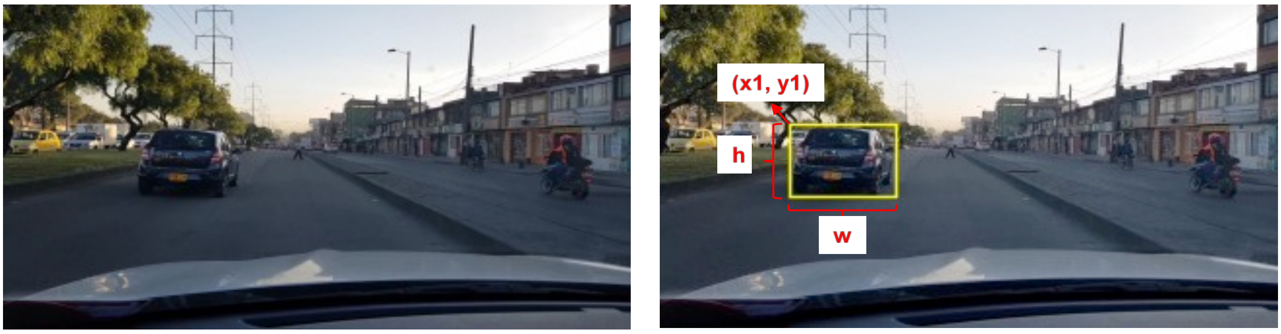
ISSN: 1791-2377 © 2019 Eastern Macedonia and Thrace Institute of Technology. All rights reserved.
doi:10.25103/jestr.123.10

Table 1. Selected design parameters

Parameter	*Value*
Lanes	2
Driving direction	Unique
Constant vehicle speed	60 km/h
Study objects	1
Object detection distance	16.67 m
Rail width	3.2 m
Max driving reaction time	1 s

$$ODD = \frac{(V_{car})\cdot 1000 \cdot Dr_{max}}{3600} \qquad (1)$$

Environments: The planning process for the lane change, two main scenarios extracted from [12] were used, based on images with resolution 320x180 pixels, that correspond to change to the right lane and change to the left lane, using 10 states for each one. In which, on the image of the camera (see Fig. 2-a), the position and dimension of the study object was highlighted with a yellow rectangle (see Fig. 2-b). In addition to indicating the approximate position of the vehicle on the current lane defined in the vector representation of Eq. 2. Where the array is composed by the parameters [x1, y1, w, h, car_position], where x1 and y1 refer to the coordinate of the RoI (Region of Interest) [13] in the upper left corner, while w and h correspond to the width and height of the RoI respectively, finally car_position is the current position of the vehicle between the lanes, based on the spatial distribution defined by Fig. 3.



a.

b.

Fig. 2. a) Original frame. b) Study object detected and marked.

$$vt = [68, 68, 55, 40, 1.6] \qquad (2)$$

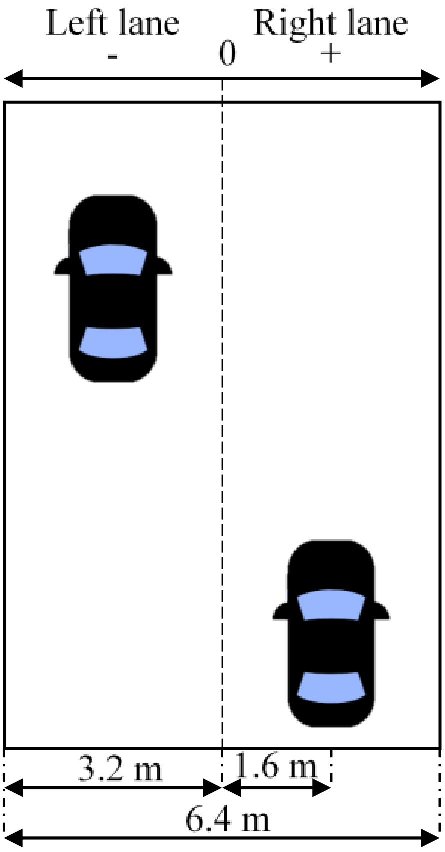


Fig. 3. Spatial distribution for lanes.

Algorithm: The control algorithm receives as input signals, the current lane of the vehicle according to its location defined between 2 lanes of one-way, if it detects a nearby object or not, next to information about the RoI with greater weight, to either avoid collision or avoid the object identified by a lane change.

For the trajectory planning stage, based on the identification of the relative angular position of the object within the visual range of the vehicle after its detection, considering Fig. 2 and 3, three main cases are defined, which are:

- Object on the left side.
- Object on the right side.
- Object in the center.

When the position object is detected, the angle of rotation of the vehicle is calculated for guarantee the correct evasion of the object and thus avoid a possible collision. It is achieved by implementing the defined cases, if it is necessary for the vehicle to turn to the right or to the left, for which the following spatial relationships are considered:

Turn Right: As shown in Fig. 4, the angle of rotation θ_f is calculated, following Eq. 3 to 6.

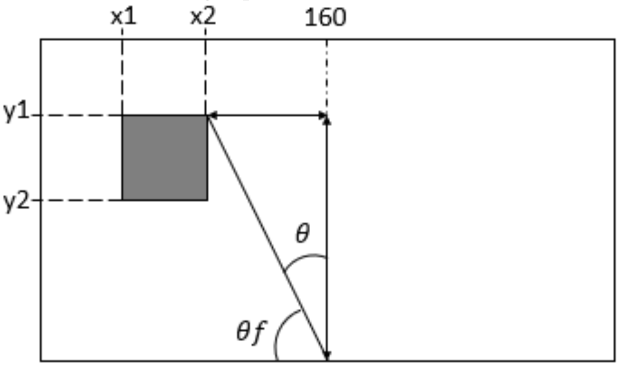


Fig. 4. Graphic definition for turn right.

$$x_\theta = 180 - y_1 \quad (3)$$

$$y_\theta = 160 - x_2 \quad (4)$$

$$\theta = \tan^{-1}\left(\frac{y_\theta}{x_\theta}\right) \quad (5)$$

$$\theta_f = 90 - \theta \quad (6)$$

Where, the values (x1, y1) and (x2, y2), represent the initial and final coordinates (respectively) of the identified RoI, θ is the angle that is formed between the upper right point of RoI and the middle vertical of the scene and θ_f is the resulting angle of the object of study which guarantees the vehicle's turn.

Turn Left: As shown in Fig. 5, the angle of rotation θ_f is calculated, following Eq. 7 to 10.

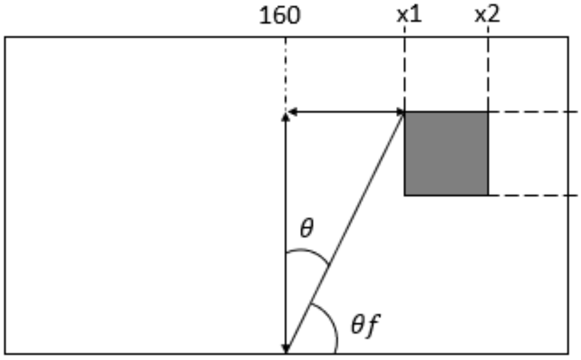


Fig. 5. Graphic definition for turn left.

$$x_\theta = 180 - y_1 \quad (7)$$

$$y_\theta = x_1 - 160 \quad (8)$$

$$\theta = \tan^{-1}\left(\frac{y_\theta}{x_\theta}\right) \quad (9)$$

$$\theta_f = 90 - \theta \quad (10)$$

Where, the values (x1, y1) and (x2, y2), represent the initial and final coordinates (respectively) of the identified RoI, θ is the angle that is formed between the upper left point of RoI and the middle vertical of the scene and θ_f is the resulting angle of the object of study which guarantees the vehicle's turn.

Object in the center: Starting from the representation for objects in the center, as shown in Fig. 6, it is chosen to use a vehicle stop function, this in order to avoid a possible collision, since it is not possible to determine whether the object is in the current vehicle lane or not, due to geometric similarities or changes in perspective of the object.

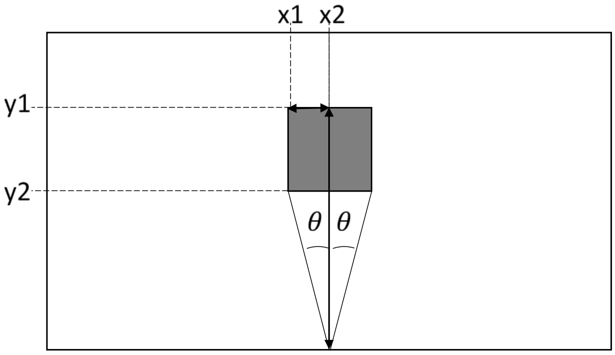


Fig. 6. Graphic for center object position definition.

Spline function: Considered the properties of the splines [8] [9], the Smoothing-spline technique [14] was selected, which when plotting the cubic spline curve. Starting from the mathematical definition represented in equation (11) [14], where *"p"* is the smoothing parameter, *"w"* are the specific weights and *"x"* and *"y"* the Cartesian components of the original trajectory function, a unit value is selected for the parameter *p* (p = 1), this passes through all data sets while preserving the original comportment of the trajectory to the maximum, which makes it the best choice among the object different Spline techniques for the desired application.

$$p \sum_i w_i (y_i - s(x_i))^2 + (1-p) \int \left(\frac{d^2 s}{dx^2}\right)^2 dx \quad (11)$$

Trajectory vector: When the angle of rotation has been determined for lane change, a trajectory vector of 4 points is estimated. This trajectory is going to load to the Spline function for the 2 case studies. It is then, that when considering the comportment of the Smoothing-Spline technique, which generates a smoothing on a function, starting from vertices or study points, it establishes a linear trajectory formed by points of study in vertices (see Figs 7-a and 8-a), which would not represent a change at the moment of generating the "smoothing", it is necessary to generate a trajectory with points of study close to the desired vertices (see Figs 7-b and 8-b) on which the Spline function can be applied (see Figs. 7-c and 8-c).

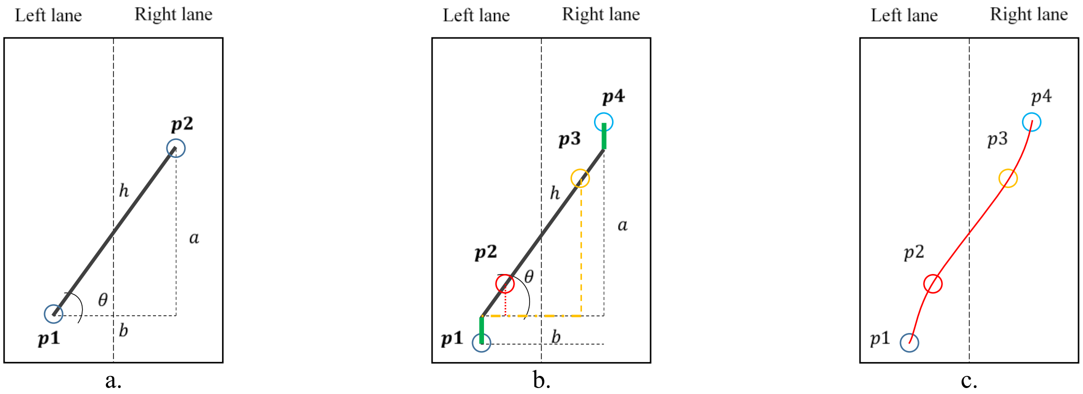


Fig. 7. a) Original linear trajectory. b) Fixed 4-point trajectory. c) Spline trajectory. (Turn-to-right case).

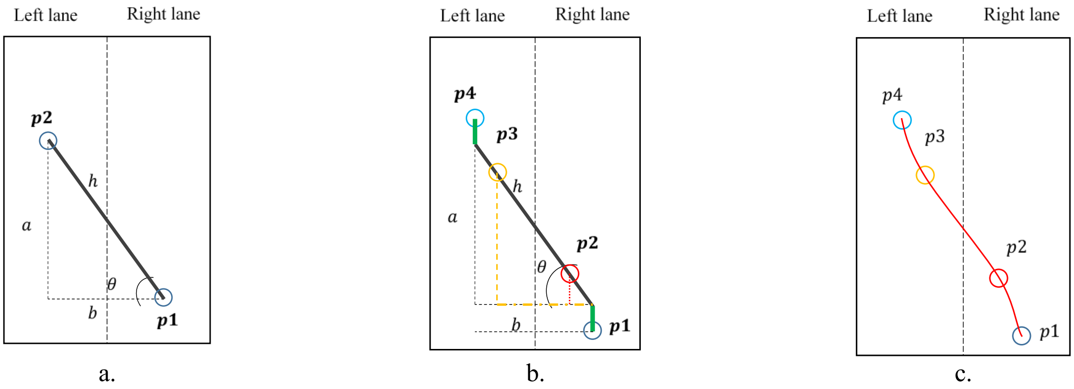
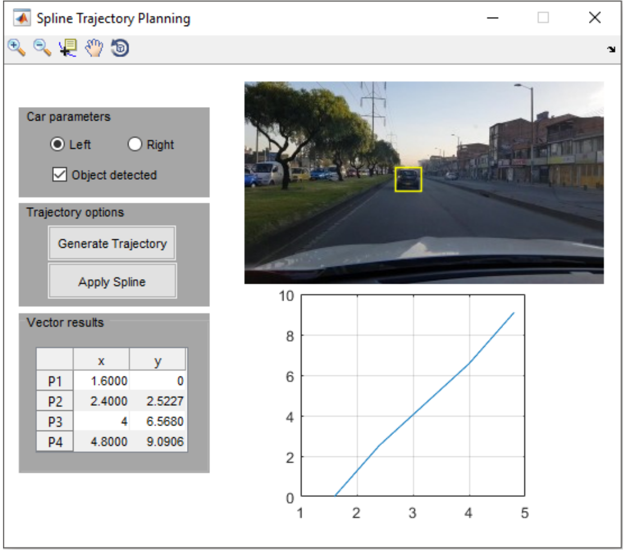


Fig. 8. a) Original linear trajectory. b) Fixed 4-point trajectory. c) Spline trajectory. (Turn-to-left case).

3. Results and Discussions

Each of the two scenarios was validated for the ten states, verifying the application of the Spline and obtaining the results shown in Tab. 2, referring to the trajectory vectors. Finding that, for both cases, when the vehicle reaches the final position in both scenarios, the same output trajectory vector is obtained in its component in Y. Once the scenario and the reference point (vehicle to evade) must be identified, the trajectory is generated as illustrated in Fig. 9, with this vector it is possible to generate the "smoothing" on said route (see Fig. 10).


Spline Trajectory Planning

Car parameters

● Left ○ Right

☑ Object detected

Trajectory options

Generate Trajectory

Apply Spline

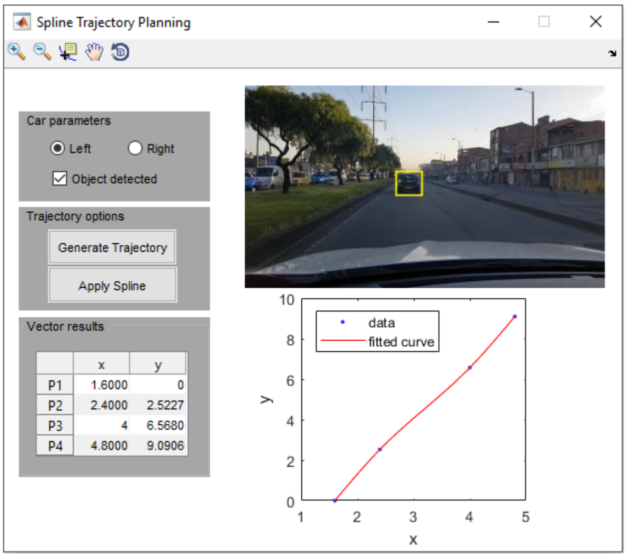
Vector results

	x	y
P1	1.6000	0
P2	2.4000	2.5227
P3	4	6.5680
P4	4.8000	9.0906

Fig. 9. Graphic user interface for trajectory planning with Splines in Matlab.

It is necessary to take into account that in cases when the vehicle maintains its position in constant x, Spline function will not be generated, since it would be applied on a straight line, which is not a valid case. As can be seen in Tab. 2, after completing the trajectory by indicating to the system the desired final points (x4), it achieves completed a final position with regard a vector whose component *"x"* has the same value in each of the points. However, the output vector in its component *"y"* varies in increasing direction as the vehicle continues to advance, until reaching a stable state with difference 0, between the current position value and the desired reference defined by its component in *"x"* for the lane change.

Due to the complexity of the problem in dynamic terms because of the large number of variables involved, it is based on a static exploration which allows initial validations, taking into account the position as the main variable of study of the desired trajectory, that is why the design parameters of Table 2 are selected as initial points.


Spline Trajectory Planning

Car parameters

● Left ○ Right

☑ Object detected

Trajectory options

Generate Trajectory

Apply Spline

Vector results

	x	y
P1	1.6000	0
P2	2.4000	2.5227
P3	4	6.5680
P4	4.8000	9.0906

Fig. 10. Graphic user interface for trajectory planning with Splines in Matlab.

Table 2. Trajectory vectors for 2 cases.

			To right			*To left*		
	p1	***p2***	***p3***	***p4***	***p1***	***p2***	***p3***	***p4***
x1	1.6	2.4	4	4.8	4.8	4	2.4	1.6
y1	0	2.5227	6.568	9.0906	0	3.8592	10.5775	14.4367
x2	2.1	2.775	4.125	4.8	4.1	3.475	2.225	1.6
y2	0	2.4519	6.3557	8.8077	0	3.436	9.308	12.744
x3	2.8	3.3	4.3	4.8	3.5	3.025	2.075	1.6
y3	0	2.2919	5.8756	8.1675	0	3.372	9.1161	12.4881
x4	3.3	3.675	4.425	4.8	3.2	2.8	2	1.6
y4	0	1.987	4.961	6.948	0	3.4397	9.319	12.7587
x5	3.9	4.125	4.575	4.8	3.1	2.725	1.975	1.6
y5	0	1.512	3.5359	5.0479	0	4.0088	11.0264	15.0353
x6	4.2	4.35	4.65	4.8	2.8	2.5	1.9	1.6
y6	0	1.3998	3.1995	4.5993	0	4.3484	12.0452	16.3936
x7	4.4	4.5	4.7	4.8	2.4	2.2	1.8	1.6
y7	0	1.1046	2.3139	3.4185	0	4.5442	12.6325	17.1767
x8	4.6	4.65	4.75	4.8	2.1	1.975	1.725	1.6
y8	0	0.7945	1.3835	2.1779	0	3.0276	8.0828	11.1104
x9	4.7	4.725	4.775	4.8	1.7	1.675	1.625	1.6
y9	0	0.6543	0.9628	1.6171	0	1.6628	3.9883	5.6511
x10	**4.8**	**4.8**	**4.8**	**4.8**	**1.6**	**1.6**	**1.6**	**1.6**
y10	**0**	**0.5**	**0.5**	**1**	**0**	**0.5**	**0.5**	**1**

4. Conclusions

It was possible to validate the Smoothing-spline function, with respect to the inclusion of its Smoothing parameter (p =

1) when controlling the level of "smoothing" on an input function, in this case, the trajectory vector makes this technique a viable option for adjusting the displacement for lane change in autonomous or semi-assisted driving systems.

For trajectory planning processes in autonomous vehicles, it is necessary to implement sensorics or additional methods, which provide more information to the system regarding both the state of the "world" and the relationship between the vehicle and the "world". This can be observed with the fact of requiring to know the proximity of the object of study for the decision making of the vehicle, since only with the detection or identification of the object is not enough, since a very large object that is at a great distance can present a RoI with the same dimensions and position as a small object at a short distance.

Finally, for future work, it is suggested to have or provide parameter information such as relative positions of the vehicle with respect to its environment to generate a better approximation of the trajectory function or know additional variables such as the speed or relative acceleration of both the vehicle and the object of study to achieve a function of dynamic trajectory closer to reality.

Acknowledgement

The authors are grateful to the Nueva Granada Military University, which, through its Vice chancellor for research, finances the present project with code IMP-ING-2290 (2017-2018) and titled "Prototype of robot assistance for surgery", from which the present work is derived.

References

[1] D. Marr, “Artificial Intelligence -- A Personal View”, 1977, vol. 9, no 1, p. 37-48.

[2] T. Lozano-Pérez, “Autonomous robot vehicles”. Springer Science & Business Media, 2012.

[3] P Ü. Özgüner, C. Stiller, and K. Redmill, “Systems for safety and autonomous behavior in cars: The DARPA Grand Challenge experience,” Proc. IEEE, 2007, vol. 95, no. 2, pp. 397–412.

[4] E. Frazzoli, M.A. Dahleh and E. Feron, “Real-time motion planning for agile autonomous vehicles”, Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148), Arlington, VA, USA, 2001, vol. 1, pp. 43-49. doi: 10.1109/ACC.2001.945511.

[5] F. Pfeiffer and R. Johanni, "A concept for manipulator trajectory planning," Proceedings. 1986 IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, 1986, pp. 1399-1405. doi: 10.1109/ROBOT.1986.1087500.

[6] C. I. Connolly, J. B. Burns and R. Weiss, "Path planning using Laplace's equation," Proceedings., IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 1990, vol. 3, pp. 2102-2106. doi: 10.1109/ROBOT.1990.126315.

[7] H. Crompton, "A Historical Overview of M-Learning: Toward Learner-Centered Education" In Handbook of mobile learning, pp. 41-52. Routledge, 2013.

[8] L. Schumaker, “Spline functions: basic theory”. Cambridge University Press, 2007.

[9] G. Wahba, “Spline models for observational data”. vol. 59. Siam, 1990.

[10] J.D. Lee, D.V. McGehee, T.L. Brown and M.L. Reyes, “Collision Warning Timing, Driver Distraction, and Driver Response to Imminent Rear-End Collisions in a High-Fidelity Driving Simulator”, In Human factors, 2002, vol. 44, Issue 2, pp. 314 – 334. doi: https://doi.org/10.1518/0018720024497844.

[11] N.L. Broen and D.P. Chiang, “Braking Response Times for 100 Drivers in the Avoidance of an Unexpected Obstacle as Measured in a Driving Simulator”, In Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 1996, vol. 40, Issue 18, pp. 900 – 904. doi: https://doi.org/10.1177/154193129604001807.

[12] fwtt and YouTube, Bogotá, Autonorte. Fabrica Bavaria, Castilla a Centro Chia en 33min. [online]. Colombia, 2015 Disponible en: https://www.youtube.com/watch?v=ZvZMaF8r2AM&t=783s.

[13] P. Constante, A. Gordon, O. Chang, E. Pruna, F. Acuna and I. Escobar, “Artificial Vision Techniques to Optimize Strawberry's Industrial Classification”, IEEE Latin America Transactions, 2016, vol 14, no 6, p. 2576-2581. doi: 10.1109/TLA.2016.7555221.

[14] Matlab, Smoothing Spline, [online], E.E.U.U., 2018, Disponible en: https://www.mathworks.com/help/curvefit/smoothing-splines.html