25th ABCM International Congress of Mechanical Engineering
October 20-25, 2019, Uberlândia, MG, Brazil

COB-2019-1683
DEPLOYMENT OF A SMOOTH CLOTHOID-BASED PATH PLANNING FOR AUTONOMOUS NAVIGATION OF A CAR-LIKE ROBOT

Renan Moreira Pinto
Andrés Eduardo Baquero Velasquez
renanmoreira@usp.br, andresbaquero@sc.usp.br

Ingrid Lorena Argote Pedraza
Vitor Akihiro Hisano Higuti
Mateus Valverde Gasparino
Arthur Jose Vieira Porto
Marcelo Becker
ingridargote@usp.br, vitor.higuti@usp.br, mateus.gasparino@usp.br, ajvporto@sc.usp.br, becker@sc.usp.br
University of São Paulo
Av. Trabalhador São-Carlense, 400
São Carlos, SP, Brazil.

Abstract. *Path and trajectory planning systems are a big challenge in the robot autonomous navigation on unstructured environments. Some of current path planning algorithms generate a path, with angular turns that require an aggressive control action. One alternative to obtain a less aggressive control action is the use of the smooth trajectories, which must be generated considering the robot kinematic, the feasible curvature, the robot dimensions, and the robot speed. For that reason, the focus of this work was the development of a clothoids-based path planning to obtain smooth trajectories for a car-like mobile robot. This approach showed a great adaptability to work in non-structured environments, while considering the kinematics constraints. A predictive system determines the vehicles states together with a description of waypoints allowing the reduction of action maneuvers. A fuzzy logic set was used on the robot angular velocities to control its movement. An IMU (Inertial Measurement Unit) and a GNSS (Global Navigation Satellite System) were used to measure the robot moving state. The proposed algorithm was validated by a series of experiments made in an outdoor environment and the obtained results showed the controller's ability to maintain the accuracy while retaining a smooth drive.*

Keywords: *Autonomous systems, Clothoids, Trajectory control, Trajectory generation.*

1. INTRODUCTION

Although the trajectory generation is a classic problem in robot navigation and there are, in the literature, many papers with novelty solutions, currently this problem persists quite complex. Reeds and Shepp (1990) describe different representations of path generators where some authors use paths composed by arcs connected by straight lines which are easy to produce but they show discontinuities. Also, Reeds and Shepp (1990) proposed a method using three connected curves, with a fixed radius, which represents the minimum permissible curvature for a given vehicle. Another example of a path generator is found in Scheuer and Fraichard (1997) where the authors use a continuous curvature but only work for a finite number of paths. The authors in Shin and Singh (1990), and Scheuer and Fraichard (1997) proposed the use of three clothoid curves to soft the path along the trajectory and a segment of fixed curvature K to minimize the discontinuities and the changes in the angular system.

Clothoids are the basis of most studies of terrestrial locomotion and they are mainly used in the generation of road drawings (Eliou and Kaliabetsos, 2013). Also, clothoids can be used on controller since it produces low lateral push and acceleration (Lima *et al.*, 2015). Another two applications of the use of clothoids in path generator were found in Atmosudiro *et al.* (2017) and Frego *et al.* (2016). In the first one, the authors use the concepts of the clothoids in real time for smoothing the path and in the latter, the clothoids are used to minimize the time to avoid obstacles.

For the aforementioned reasons, this study is focused on the development and implementation of a clothoid-based controller able to generate a smooth trajectory considering kinematic, feasible curvature, robot dimensions, and speed to keep the accuracy while the control actions are reduce.

2. METHODOLOGY

This section exposes the materials used on the experiments and the formulation of the clothoid-based path planning in the reference to a clothoid base path trajectory.

2.1 Sensors

The sensors used in this work are the LSM9DS1 IMU and the GNSS module HMC58983. The linear acceleration of the LSM9DS1 can be set to $\pm$ 2, 4, 8 or 16 g. Its gyroscope supports $\pm$ 245, 500 and 2000 $/s$, and its magnetometer has a range of 4, 8, 12 or 16 gauss. On the other hand, the GNSS module has 10 Hz update rate for cold start and it is 2.5 m for the horizontal accuracy of the GPS antenna. To verify the values of the horizontal precision, the GNNS module was positioned in an georeferenced place and its readings were taken during several days in order to obtain its real precision and the average number of satellites used. As the result was determined that the average number of satellites in the area was 10 and the horizontal error had a variation of 5 m. Therefore it was established that this value would be considered in the tests.

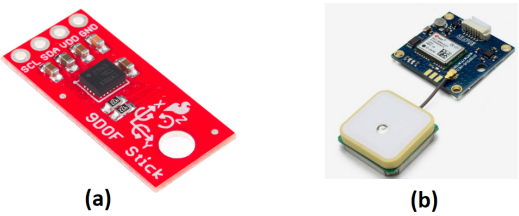


Figure 1. (a) IMU LSM9DS, (b) UBLOX NEO-M8n HMC5983.

2.2 Mobile Robots Platforms

To validate the proposed system, the HelvisIII and RAM were used. HelvisIII is an electric small-scale car-like platform developed by the Laboratory of Mobile Robotics (LabRoM) of the University of Sao Paulo. Its main characteristics can be found in Higuti *et al.* (2016); Velasquez *et al.* (2016). RAM is a mobile agriculture robot developed by the University of Sao Paulo and a detailed description of its main characteristics can be found in Sousa (2016). Figure 2 shows both robots.

Figure 2. (a) Car-like experimental platform HelvisIII, (b) RAM - Mobile agriculture Robot.

2.3 General System

Movement planning is critical for environments that may require the computation of new paths, especially in off-road conditions, which can be divided into two layers: a long-term global system and a local planning that respects the kinematic limits of the vehicle and avoids obstacles. As can be seen in Fig. 3, the movement planning system was subdivided in three sections: 1. The trajectory generation; 2. The path following; and 3. The error analysis. The latter system performs as a local planning system based on simple heuristic rules.

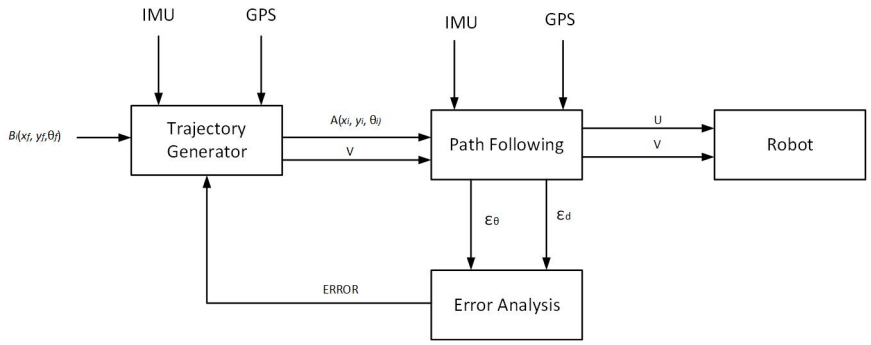


Figure 3. General Path-planning System.

2.4 Trajectory generation

To use the clothoid-based movement planning system, the trajectory generation system was divided by three possibles situations show in Fig. 4. The first step is the creation of a reference path using a clothoid transition curve. In case of errors, the system will try to create a sub-routine from the last state to some waypoint ahead in the reference path. If not possible, it will generate a new path to the final state from the current position.

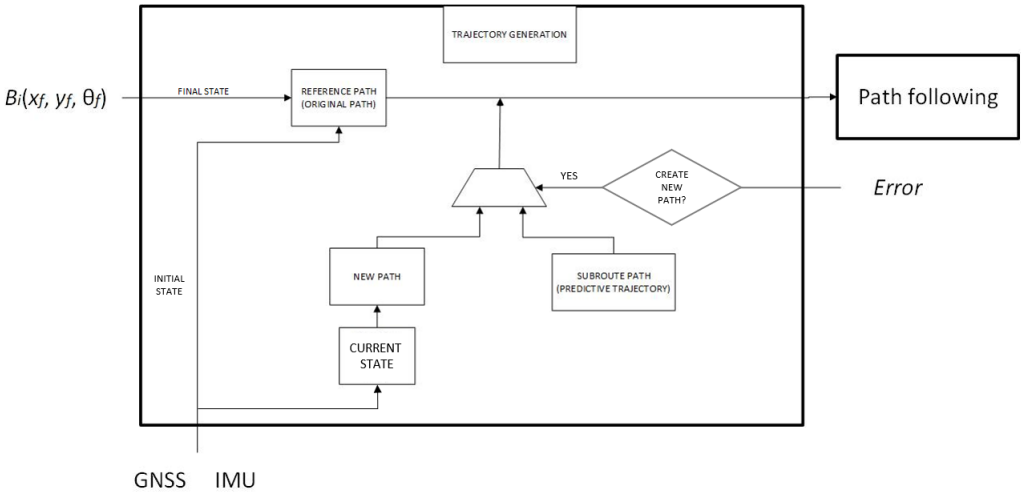


Figure 4. Trajectory Generation System.

To use the clothoid as a transition curve, it is necessary to determine the curve constant C_n for an initial robot state $A(x_i, y_i, \theta_i)$ and a final state $B(x_f, y_f, \theta_f)$. Curve constant is the tangent which forms a constant angle with a fixed direction and it is calculated with Eq. 1.

$$Y_n = m_n X_n + C_n \tag{1}$$

where m is the angular coefficient of the tangent line at point n, thus $m_n = \tan(\theta_n)$, and X_n, Y_n, C_n are the robot position and curve constant at a point n, respectively. Applying the position and orientation of the robot in the A and B states, the Eq. 1 can be redefined as Eq. 2.

$$\begin{aligned} C_i &= \tan(\theta_i)x_i - y_i \\ C_f &= \tan(\theta_f)x_f - y_f \end{aligned} \tag{2}$$

where C_i is the curve constant at point A and C_f is the curve constant at point B. The slowest curve constant C_t will be at the transition point t. The maximal radius allowed between A and B is shown in Eq. 3. To set a radius efficiency for the available mobile robots, this radius, which is the minimal one, is based on the Ackermann steering system. For that reason, the radius was determined using the robot width and length on Eq. 4.

$$R_{max} = \tan(\theta_t)x_n + C_t \tag{3}$$

$$R_{min} = \frac{width}{length} \tag{4}$$

Due to the fact that the robot usually initiates with a longitudinal movement and only starts steering when the cruise speed is reached, it is established that the robot requires an initial straight line movement correction. This value is determined by the time on the robot acceleration value. Such correction is given by d term in Eq. 5, which describes the center of the curvature at the first clothoid set.

$$(m_i^2 - 1)x_i^2 + (2C_i m_i)x_i + (C_i^2 - d^2) = 0 \tag{5}$$

At the point where the directions of motion for the initial and final curves are known and the radius of the curvature R_{curve} will be a value between the minimal and maximum radius, the next step is to determine the center point of each of the curves through Eq. 6 and Eq. 7.

$$C_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + R_{curve} \begin{bmatrix} \cos\lambda & -\sin\lambda \\ \sin\lambda & \cos\lambda \end{bmatrix} \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} \tag{6}$$

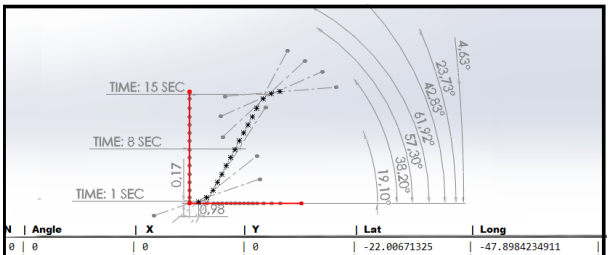
$$C_f = \begin{bmatrix} x_f \\ y_f \end{bmatrix} + R_{curve} \begin{bmatrix} \cos\lambda & -\sin\lambda \\ \sin\lambda & \cos\lambda \end{bmatrix} \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix} \tag{7}$$

where $\lambda \in [-\pi, \pi)$ represents a constant value based on the heading states and it depends on the direction of motion of the analyzed curve. After center of curvature is defined, the next step is to calculate two angles with respect to the global system: the first one between the centers of curvature (Eq. 8) and the second one for the straight path (Eq. 9).

$$\theta_C = \tan^{-1}\left(\frac{C_{y_f} - C_{y_i}}{C_{x_f} - C_{y_i}}\right) \tag{8}$$

$$\theta_{st} = \theta_c + \frac{\pi}{2}\sin\lambda + \tan^{-1}\left(\sqrt{\sqrt{\left(C_{y_f} - C_{y_i}\right)^2 + \left(C_{x_f} - C_{x_i}\right)^2}\left(r_i + r_f\right)^2}\left(r_i - r_f\right)\right)\cos\lambda \tag{9}$$

The values of θ_c and θ_{st} are used to define the arc of curvature and the points of intersection. This can be done by a method similar to that presented in the equations above. With this data is possible to draw a set of points based on the kinematic analysis and to define the position of the vehicle at each instant t = 1 second as shown in Fig. 5. The time-based approach was used based on the platforms limitation of turning speed per second. This also allows the error system to reduce the movement speed. At the end the system will send to the path following system a sequence of points to follow and the cruise speed of the system.



N	Angle	X	Y	Lat	Long
0	0	0	0	-22.00671325	-47.8984234911
1	19.0985931710274	0.981584090388457	0.165129161055787	-22.0067147334378	-47.89841461306272
2	38.1971863420549	1.85510940920921	0.642338217669156	-22.0067190204454	-47.8984067124633
3	57.2957795130823	2.52441295442369	1.37909308239558	-22.0067256390827	-47.8984006589474
4	61.927513064147	2.99500118971781	2.26144602357205	-22.0067335657004	-47.89839640271175
5	61.927513064147	3.46558942501193	3.14379896474852	-22.0067411492318	-47.89839214647646
6	61.927513064147	3.93617766030604	4.02615190592499	-22.0067494189356	-47.89838789024127
7	61.927513064147	4.40676589560016	4.90850484710146	-22.0067573455533	-47.89838363400598
8	61.927513064147	4.87735413089428	5.79085778827793	-22.0067652721709	-47.89837937777069
9	61.927513064147	5.34794236618841	6.6732107294544	-22.0067731987886	-47.89837512153510
10	61.927513064147	5.81853060148251	7.55556367063087	-22.0067811254062	-47.8983708653311
11	61.927513064147	6.28911883677663	8.43791661180735	-22.0067890520239	-47.89836660090647 12
12	61.927513064147	6.75970707207075	9.32026955298382	-22.0067969786415	-47.89836235282941 3
13	61.927513064147	7.23029530736487	10.2026224941603	-22.0068049052592	-47.8983589659411 4
14	42.8289198931196	7.96056530229085	11.2001604745518	-22.0068138666416	-47.8983514916671 15
15	23.7303267220922	8.7927028552183	11.7463491409873	-22.0068187733275	-47.89834396539916

Figure 5. Path Generator and waypoints determination system simulated results for one single path.

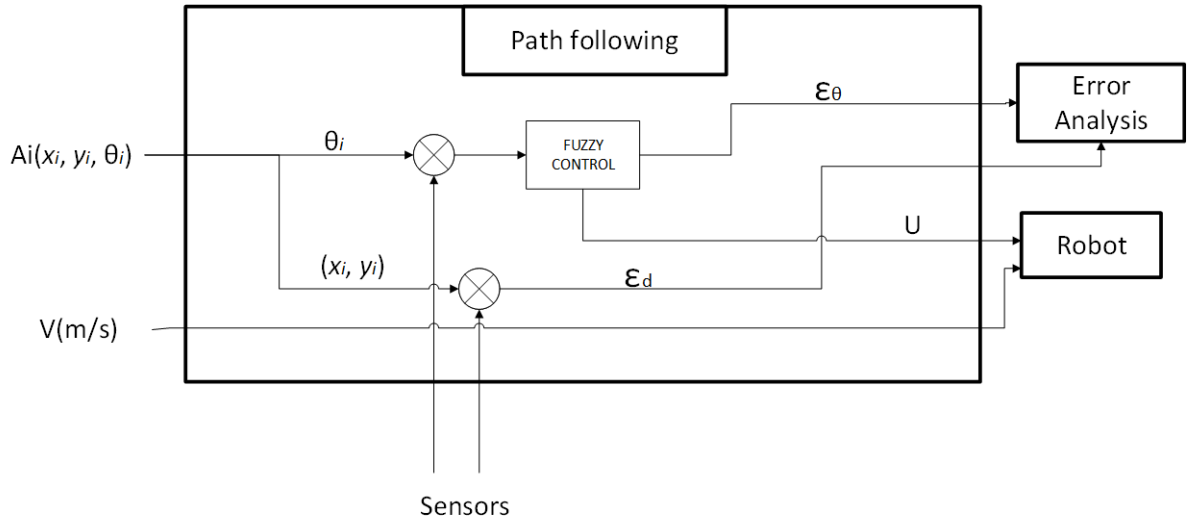


Figure 6. Path Generator and waypoint determination system for one single path.

2.5 Path following

Figure 6 shows a diagram of the path following subsystem, which has two steps: 1. The error calculation; and 2. The fuzzy controller. The latter receives the value of the next step, generates the steering speed of the vehicle for the current step and sends the heading error to Error Analysis.

Vehicle orientation control was deployed using a fuzzy closed loop control. The input θ corresponds to the heading angle of the next step. The error signal ε_θ is the input of the fuzzy controller block that calculates an appropriate control action signal U, which will be applied on the actuators of the mobile robot represented by the robot block. Triangular set definitions were adopted similarly to exposed in Yu and Zhang (2008). Input Fuzzy Sets MN (very negative), N (negative), Z (zero), P (positive) and MP (very positive) are represented in Fig. 7(a). The output fuzzy sets: R (right), C (Center) and L (left) are represented in Fig. 7 (b).

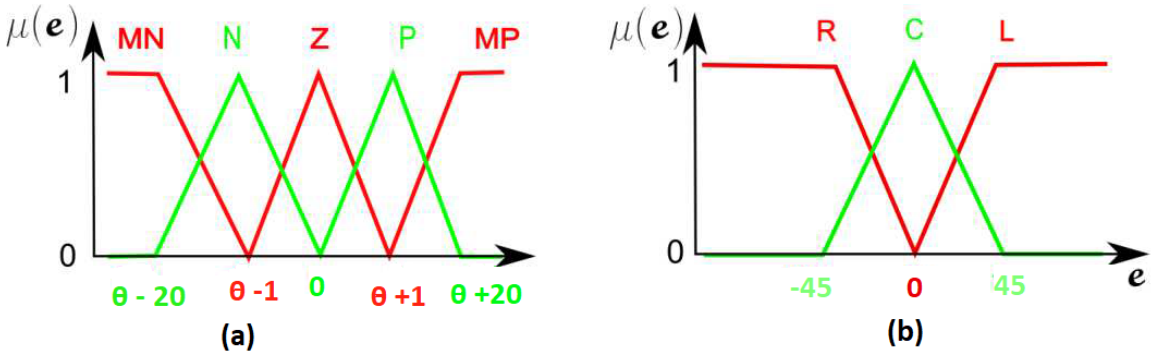


Figure 7. Fuzzy logic sets.

The defuzzified output is employed to decide the necessary commands to be applied. Those commands are shown in Tab. 1. The controller is based on the RAM's steering capabilities of the mobile robot: the input was set such that the minimal steering speed was 0.0174 rad/s and the maximum was 0.087 rad/s. If the system requires a greater steering speed, then the error will be sent to the Error Analysis and the movement speed will decrease to give more time for the system to complete steering.

Table 1. Fuzzy output

if e = Mn	then y = R	steering left max speed
if e = Mn	then y = R	steering left
if e = Z	then y = C	no steering
if e = p	then y = L	steering right
if e = Mp	then y = L	steering right max speed

2.6 Error Analysis

This section presents the Error Analysis, which considers the robot kinematic model (at low speed) and the path determined by the clothoid. Its motion can be described by the kinematic equations present in Eq. 10.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v(t)\cos(\theta(t)) \\ v(t)\sin(\theta(t)) \\ \frac{v(t)}{L}\tan(\psi(t)) \end{bmatrix} \tag{10}$$

where $v(t)$ is the longitudinal robot speed, L is the distance between wheel axles and $\psi(t)$ is the steering angle of the front wheels at time t.

As Assume $v(t) \neq 0$ and $v(t)$ is a continuous function and consider $v(t)dt = ds$ to translate Eq. 10 to space-domain. Then, knowing that $d\theta/ds$ is the same as $\kappa(s)$, which represents the curvature along the path, and that it is a linearly varying function, then we can say that Eq. 10 describes the clothoid and that $\kappa(s)$ can be defined as $\kappa(s) = Cs + K_0$ where C is the clothoid curvature slope. The system can be defined by N points with a function, then they can be rewritten as Eq. 11-14. More details can be found in Lima *et al.* (2015).

$$x_{i+1} = x_i + \int_{S_i}^{t_{S+1}} \left(\cos(\theta_i) + \kappa_i(s - s_i) + C_{i+1}\frac{(s - s_i)^2}{2} \right) ds \tag{11}$$

$$y_{i+1} = y_i + \int_{S_i}^{S_{i+1}} \left(\sin(\theta_i) + \kappa_i(s - s_i) + C_{i+1}\frac{(s - s_i)^2}{2} \right) ds \tag{12}$$

$$\theta_{i+1} = \theta_i + \kappa_i l_{i+1} + C_i\frac{l_{i+1}^2}{2} \tag{13}$$

$$\kappa_{i+1} = \kappa_i + C_{i+1}l_{i+1} \tag{14}$$

where i = 1,...,N, C_i andl_i are the i-th clothoid curvature slope and arc-length, respectively.

The system assumes that the vehicle is always close to the original reference and the distance between the base path and predictive path through the trajectory can be computed using simple trigonometric relations. An invisible barrier is set over each waypoint radius, in case when the system will overcome these barriers on the next step it corrects the path creating a new sequence of points distributed in the remaining trajectory, as shown in Fig. 8.

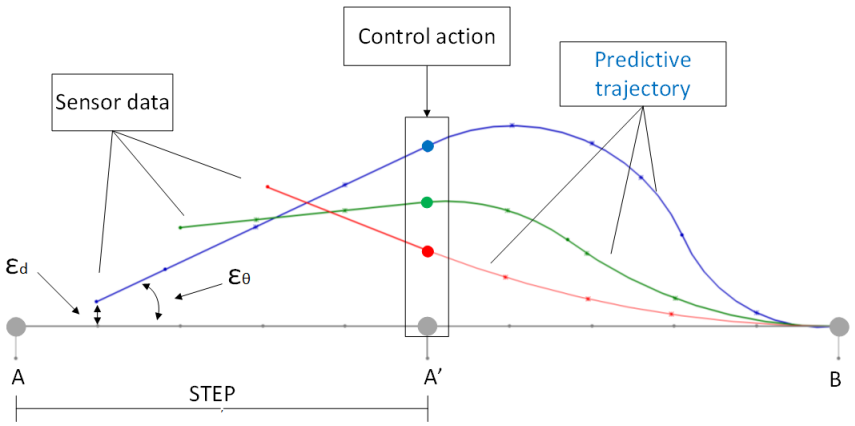


Figure 8. Predictive trajectory path is represented by clothoids describe with several waypoint that will make the vehicle follow the reference path.

The barriers limits is determined by a practical analysis of the vehicle kinematics, sensors and distance from the final position it allows that the vehicle can follow the course smoothly just by monitoring the distance until the action time up. This allows that even with a huge disturbance the system will distribute the error along the path letting the actions at the minimal required to complete the path. Figure 9 shows the schematic of the path planning.

As the values of IMU and GPS are updated every 100 milliseconds and the loop time of the system is 1 second then these readings are accumulated in counters and then the system can verify the values in order to generate instructions to the trajectory generator. These instruction are described in Tab. 2.The actions will be execute if they the counter limit between one step to another and will reset for every step.

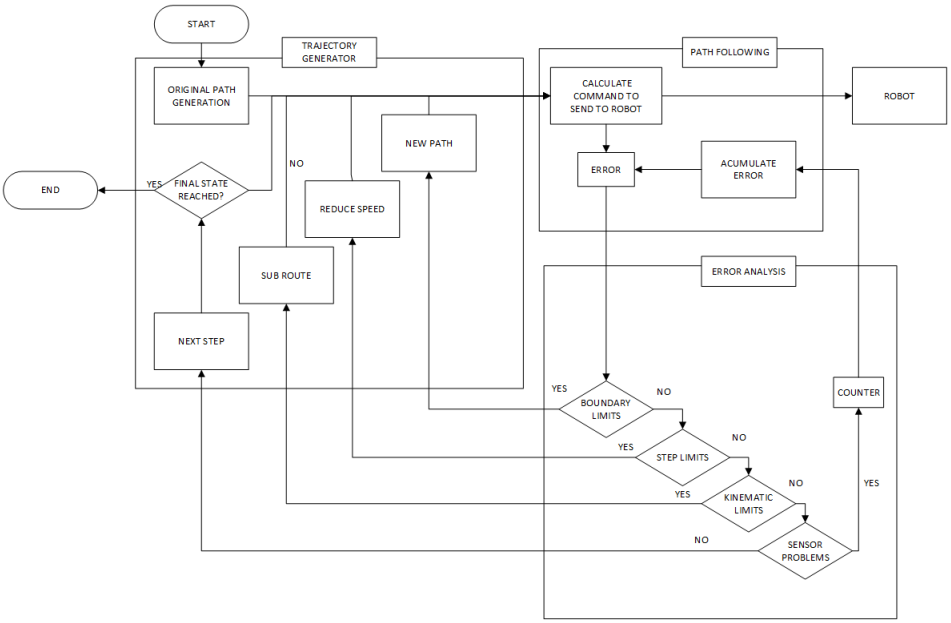


Figure 9. Schematic of the predictive analysis of the system.

Table 2. Control System instructions for the heuristics rules - those actions are taking while the action loop is running to compute before the actions are take place

Data	Error	Possible cause	Correction	Counter
Boundaries limits	passed	turn problems	new path	+5
Distance	0 or Negative	GPS data is wrong	acumulate error	+1
Speed	too Slow or too Faster	Terrain change	speed adjust	+1
Heading - curve	Faster turning	Skidded	Slow turn speed	+1
Heading - curve	very fast turning	Data is old	Sub-route	+2
Heading - destiny	Changed	Data is old	Sub-route	+1

2.7 Experimental results

In essence, the experimental goal consisted to keep the mobile robot in the path on the smoothest way avoiding rough movements while keep the track. Figure 10 shows the path used in the tests we used three test courses and for each one ten tests are made five for each vehicle. The first path is a straight line, the second one is a single curve path and the last one is a clothoid base curve combine of 3 curves set.

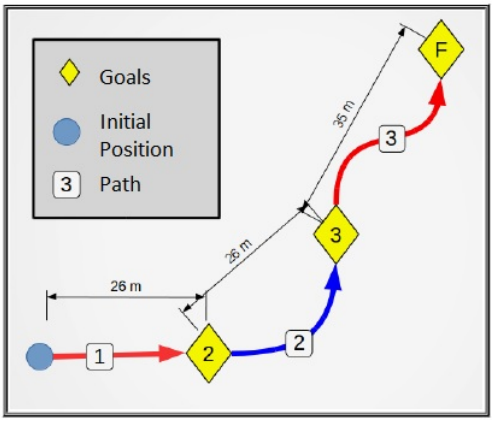


Figure 10. Path development for test with the clothoid based.

Figure 11 and Figure 12 show the path number 1, 2 and 3 generated by the trajectory generator and the GPS data obtained along the path.

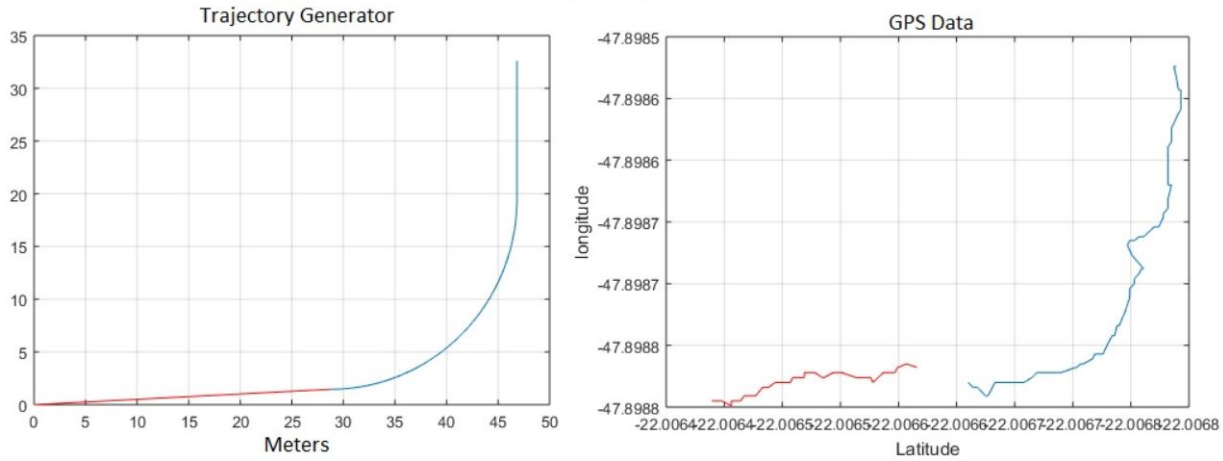


Figure 11. Experimental results using the reference path 1 and 2. The figure on left shows the reference and the vehicle curvature and the figure on the right shows the obtained GNSS data.

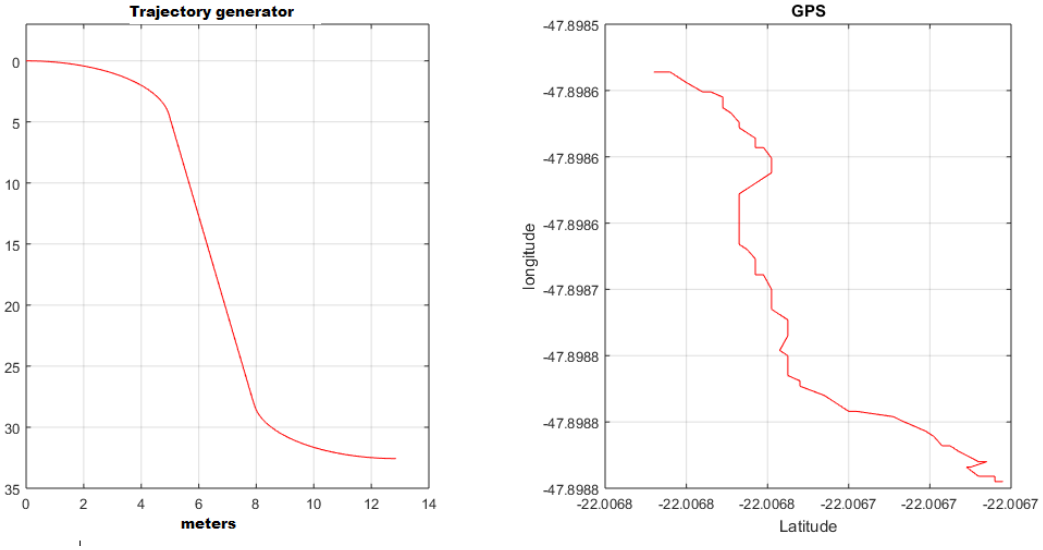


Figure 12. Experimental results using the reference path 3.The figure on left shows the reference and the vehicle curvature and the figure on the right shows the obtained GNSS data.

Table 3 summarizes the obtained results. Finally, the system reads the data every second to determine the next action and to estimate the total time. The crusade speed was set 0.3 m/s on RAM. The speed could be increase until 1 m/s without any loss on the accuracy during the trajectory but at the curvature speed its was limited by the steering speed its would need a action time lower than 1 second. The worst results was take on the single curvature path, based on the idea that the algorithm will try always to maintain the R_{max} on those cases it show that GNSS accuracy was a great impact since of the curvature requires more corrective actions from the IMU and the reference path.

Table 3. Experimental results resume.

Path	**Distance (m)**	**GNSS (m)**	**Final Distance (m)**	**heading Error (°)**	**Action time (s) Predictive / Real**
Lineal$^{(1)}$	26	28 ± 5	0 ± 3	0.3 ± 0.3	96 / 95 $\pm$1
Single curve$^{(1)}$	26	26 ± 2.94	3.6 ± 3.54	1.5 ± 1	139 / 134 $\pm$10
Clothoid$^{(1)}$	35	34 ± 4.86	2.35 ± 1.32	0 ± 3.0	127 / 127$\pm$5

$^{(1)}$ Speed was 0.3 m/s

2.8 Offline sensors test

Because the goal of those tests is to evaluate the vehicle capability to determine its initial position and autonomously follow the created path to the final state, then different initial states were used in each of the experiments. The final state is added to the robot's initial database to generate the trajectory from its actual state, then the algorithm keeps track if an errors on the sensors occurs will set a counter and allow the robot continues the path following the last path calculated. If the counter reaches the limit set then the trajectory reset and will be reconfigured, otherwise, the system will reset the counter when the sensors get back online. Figure 13 shows the experimental offline path test, at the state A, the error on the GNSS reach it limit and the algorithm ignore the GNSS data and keep the track using the offline path, at state B GNSS sensor is back on and its keep the original path until reach the final state. The idea is to show that the algorithm is able to recognize when the GNSS data is having disturbances and will try to maintain the path using the reference path and IMU set. The algorithm will stop if stays more than 10 seconds without any response for GNSS sensor or if the θ value is over $\pi/2$.

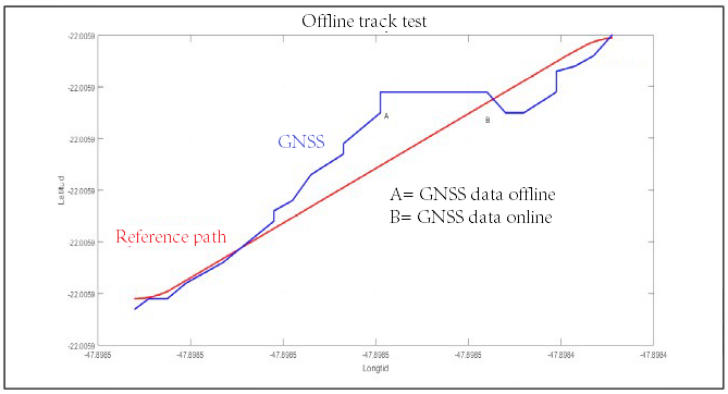


Figure 13. Path development for test with disturbance at point A to B GNSS data is offline and the system was able to autonomous recognize and switch to the reference path based and take only the IMU data while waiting new info, at point B the GNSS data is back and the system was able to continues to the end state.

2.9 Discussions

The maximum number of clothoids used to describe the path is dependent on the maximum allowable deviation between the reference path and the predictive path. This path deviation can be programmed to be inversely proportional to the objective, as control actions throughout the process of trajectory are compensate with a predictive analysis allowing the error to be large at the initial state with less aggressive control actions.

There are limitations to clothoid curves and the curve smoothing system, in this case the physical limitations would be the steering speed of the vehicle and the minimal curve that it can perform based on its wide. The other limitations is the GNSS accuracy on small curves (less than 3 meters), the values on the GNSS are inefficient and the IMU data will lead the movement. The system is capable to maintain the route without the GNSS information for some time this allow the system reboot the sensors data in case of failure and keeps the movement forward the track.

3. CONCLUSION

In this paper, a clothoid based path was applied for path following on mobile robots. This controller formulation is based on three steps: a trajectory generation system, a fuzzy controller and an error analysis system. The clothoid path has been taken as an ideal solution due to its smooth curvature allowing create large prediction distance and distribute the minimal error along the entire trajectory. The heuristics rules are used to create boundaries for the sub route system to reduce the control actions and the curve constant is established on the clothoids. In the end, we provide validation on real reference paths with two different mobile robots. As future work, we can consider time efficiency of the proposed system including the longitudinal dynamics of the robots to adjust the cruise speed and the inclusion of visual sensors to study cases with eventual path blocking.

4. ACKNOWLEDGEMENTS

The authors would like to thank to FAPESP (grant numbers 2017/10401-3 and 2018/10894-2), CAPES and CNPq for the financial support and the scholarships for the students. Also, Renan Moreira Pinto was supported by Coordination of

Superior Level Staff Improvement, Brazil (CAPES), Finance Code 001.

5. REFERENCES

Atmosudiro, A., Verl, A., Lechler, A. and Schwarz, G., 2017. "A clothoid based real-time interpolation concept for path smoothing". pp. 41–46. doi:10.1109/ICCAR.2017.7942658.

Eliou, N. and Kaliabetsos, G., 2013. "A new, simple and accurate transition curve type, for use in road and railway alignment design". *European Transport Research Review*, Vol. 6, pp. 171–179. doi:10.1007/s12544-013-0119-8.

Frego, M., Bertolazzi, E., Biral, F., Fontanelli, D. and Palopoli, L., 2016. "Semi-analytical minimum time solutions for a vehicle following clothoid-based trajectory subject to velocity constraints". In *2016 European Control Conference (ECC)*. pp. 2221–2227. doi:10.1109/ECC.2016.7810621.

Higuti, V.A.H., Velasquez, A.E.B., Borrero Guerrero, H., Magalhães, D.V. and Becker, M., 2016. "Description of helvis 3a: small scale car-like robot for precision agriculture". In *Simpósio do Programa de Pós-Graduação em Engenharia Mecânica - SIPGEM*. EESC-USP.

Lima, P.F., Trincavelli, M., Mårtensson, J. and Wahlberg, B., 2015. "Clothoid-based model predictive control for autonomous driving". In *2015 European Control Conference (ECC)*. pp. 2983–2990. doi:10.1109/ECC.2015.7330991.

Reeds, J.A. and Shepp, L.A., 1990. "Optimal paths for a car that goes both forwards and backwards".

Scheuer, A. and Fraichard, T., 1997. "Continuous-curvature path planning for car-like vehicles". In *IROS*.

Shin, D.H. and Singh, S., 1990. "Path generation for robot vehicles using composite clothoid segments". Technical Report CMU-RI-TR-90-31, Carnegie Mellon University, Pittsburgh, PA.

Sousa, R.V., 2016. "Robô agrícola móvel (ram) : uma arquitetura baseada em comportamentos hierárquicos e difusos para sistemas autônomos de guiagem e navegação". In *São Carlos : Escola de Engenharia de São Carlos, Universidade de São Paulo, 2007*. EESC-USP.

Velasquez, A.E.B., Higuti, V.A.H., Guerrero, H.B., Becker, M., Magalhães, D.V. and Milori, D.M.B.P., 2016. "Helvis - a small-scale agricultural mobile robot prototype for precision agriculture". In *13th International Conference on Precision Agriculture - MO, USA*.

Yu, F. and Zhang, M., 2008. "Generalized triangular norms based product and similarity of fuzzy sets". pp. 286–290. doi:10.1109/FSKD.2008.644.

6. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.