Online Trajectory Planning With a Modified Potential Field Method on Distributed Architectures for Autonomous Vehicles

Farid Bounini¹, Denis Gingras¹, Herve Pollart², Dominique Gruyer³

¹. LIV– Université de Sherbrooke, Canada, farid.bounini@usherbrooke.ca, denis.gingras@usherbrooke.ca
². OPAL-RT Technologies inc. Canada, herve.pollart@opal-rt.com
³. LiVIC, IFSTTAR, France, domenique.gruyer@ifsttar.fr

Abstract
This paper presents an online trajectory planning strategy with a modified potential field method on distributed architectures for autonomous vehicles. The approach overcomes the well-known artificial potential field method (APFM) issue, which is due to local minima that induce the APFM to stick in. Thus, the standard APFM is no longer useful. The advantage of the new proposed method reverse to those that resort to the global optimization methods is the low computing time which borders up the A-Star (A*) method. The strategy consists of looking for a realistic path in the potential field—according to the potential gradient descent algorithm (PGDA)—and affix a repulsive potential, to the current state, in the case of blocking configuration, a local minimum. When the PGDA reaches the global minimum, a new artificial potential field will be constructed with only one minimum which matches to the final destination of the vehicle, the global minimum. Finally, to determine the achievable trajectory, a second iteration is performed by a PGDA. In addition, other modifications are proposed to adopt the MAPF method on distributed architectures for parallel computing. The preliminary simulation results show that the MAPF method could be a good solution for self-driving vehicles in a non-static environment.

KEYWORDS:
Trajectory Planning; Artificial Potential Field; self-driving vehicles.

Introduction
One of the most important branches of artificial intelligence, which is applied to robotic navigation and automated driving, is trajectory planning. It consists of finding a set of sequences that allows a robot to travel from an initial to a global state [1]. It could also be defined as moving objects in a house without hitting the walls and the other objects in it [1]. In the early work, the robots’ operational environment is considered to be deterministic or inert, as in [2]. Gradually, technology development prompted the new robotic generations to become partially autonomous. Therefore, the vehicle must be suitably equipped with the perception, localization, data fusion, decision-making and control abilities. If the first generation of robots were simple automata—relied on basic intelligence form to repeat series of actions in a static environment—the coming generations will necessitate a quite sophisticated brain [1, 3]. In fact, these needs are a consequence of the increase in the number of degrees of freedom (DOF) of self-governing vehicles. The new generation, so-called third generation, must be able to map their trajectories and react instantaneously to their surroundings [1]. Autonomous vehicles are a part of the mobile robots. The self-driving requirements result in developing a new robotics field on a large scale of vehicle navigation in an uncertain environment. Another aspect of artificial intelligence, which characterizes the autonomous vehicle, is traveling from the starting point “A” to the arrival point “B” without involving human driver [6–7]. In other words, the vehicle moves on its own. The most decisions are fully made autonomously: e.g. itinerary (the planned route), the obstacles’ detection and avoidance, the vehicles’ dynamic control, and communication with its environment (other vehicles, infrastructure, etc.) will be made autonomously by itself. Intelligent and self-driving vehicles are definitely changing the operating mode of the existing vehicles. They are, furthermore, influencing the vision of vehicle manufacturers, the interaction of humans with the vehicles, and the automotive market in general [8]. There are mainly three aspects that characterize an autonomous vehicle: perception, decision, and action:
1) Perception: The navigation in an instructed environment requires a perfect understanding of the vehicle’s surroundings. The vehicle must imperatively identify and determine the positions of each mobile and static obstacle at each stamp time.
2) Decision: The collected data must be handled by a set of algorithms (filtering, position predicting, path planning, and decision-making.). Indeed, an autonomous vehicle should define a sequence of actions leading it from an initial position to a goal one. In a real environment. The vehicle is subjected to many
uncertainties [1, 4]. For instance, sensors’ noise, static and mobile obstacles, etc. Sensors’ measurement will, even though they are assumed to be insignificant in the short term, certainly cause errors at long-term. Then, the resulting inaccuracies produce localization issues, and the robot gradually gets far away from its pre-planned path. The countermeasure to such issues is to fuse the exteroceptive and proprioceptive sensors’ measurements, and include the vehicle dynamic’s constraints and a digital map of the vehicles’ environment [1,3–5].

3) Action: Once the decision is made, and a satisfactory trajectory is defined, the control signals are generated and transmitted to the actuators to act on vehicle dynamics. This allows the vehicle to avoid static and mobile obstacles, and to change its course according to the new environment configurations in real time.

The paper describes a new approach for autonomous vehicle real-time navigation in static and highly dynamic environment on distributed architectures. The strategy stands on a modified artificial potential field method (MAPFM). The proposed MAPFM allows to eliminate the local minima that are due to the vehicles environment configuration, and find a realistic trajectory for the vehicle path planning on distributed architectures. The remaining of the paper incorporates a total of nine parts: The first part presents some related work on trajectory planning algorithms that stand on artificial potential field method, or combined with other path planning methods. The second part describes two classical planning trajectory algorithms, A-Star and artificial potential field methods, according to the literature, that highlights their advantages and their drawbacks. The third part introduces briefly some trajectory planning techniques with APF approach. In the fourth part, the major problem of the potential field method is presented and solved by the suggested strategy, which overcomes the APF local minima and adapted it for online path planning. In the fifth one, some simulation results are presented and discussed by comparing the two methods (APF and A*) results. The sixth and the seventh parts highlights the adapted MAPF for vehicle path on distributed architectures. The eighth part describes the simulated scenario and the results of the MAPF on the distributed architectures. Finally, the last part ends this paper with a conclusion and future works.

Related Work
The perception is often ranked among the three central aspects of an intelligent vehicle. In this work, we assume that the proprioceptive sensors data fusion is already performed according to specific algorithms. For instance, in [8], You Li et al. deal with perception issues frequently encountered in smart vehicles. The problem was merging data that came from two on-board pinhole cameras and a LIDAR (Light Detection And Ranging). The algorithm estimates the translation and the rotation matrices of the two cameras, in 3D, with the proposed methods in P. Nunez et al. [9–10]. The relative rotation and translation matrices of the LIDAR are as well defined to each camera and the stereo system (the left camera is taken as the main stereo system). The different parameters, intrinsic and extrinsic, are determined through experiments on a specific panel: a chessboard shape modeled by a 3D plane. Numerous measures were used to estimate the various parameters of rotation and translation matrices of each sensor.

The artificial potential field (APF) method is widely used throughout the literature. It is based on the uptake of the robot to a particle constrained to move in an artificial PF. The field is a combination of an attractive field, assigned to the global position or to the target, and a set of repulsive fields related to the obstacles around the robot in its environment [11–13]. The robot’s trajectory is determined interactively by a potential gradient descent algorithm. The concept was initially introduced by Khatib [2] to control a manipulator arm motion, avoid the neighbouring obstacles, and to plan the arm’s motion. The artificial potential field method (APFM) is a simple and a very effective strategy to path planning. It is quite fast for online trajectory planning, and easily handles kine-dynamic constraints that satisfy the velocity, the acceleration, and the force/torque bounds at the same time. Thanks to its simple concept and effectiveness for online applications, the APF method is applied in different planning challenges, for instance: Ferry Rippun and al. [14] applied the potential field method on robotic soccer, which is implemented, in their study, in a multi-cooperative autonomous vehicle. The proposed approach called Double Target Potential Field (DTPF). The main objective was to increase the effectiveness of the vehicle movement and achieve two challenging actions: 1-passing the ball to another vehicle, and 2-kicking the ball into the goal. To solve the path planning problem in the 3D environment for the rotary-wing flying robot, Jianhao Tan and al. resort, in their work [15], to artificial potential field method to smooth the offline predetermined A-Star path, which is indeed the shortest path but a non-smooth one, it goes from the robot to the target according to flying vehicle’s environment. The predetermined A* trajectory is used as a
reference one in the case of APF method to smooth the flying vehicle trajectory for online applications. The combination of the two methods—APF and A*—solved the lacks of both methods (non-smoothness and local minima issues, respectively). Nevertheless, the proposed method needs a primary map for the first run. The APF methods have been extended to address problems in which the goal is not reachable due to the obstacle proximity [16], and navigation in narrow passages, in unknown semi-structured environments, is required [17]. Other recent work [18] has focused on the modification of the computation of the potential field according to some fuzzy criteria, which are added to the path planning method APF. The strategy was made up according to: 1-the velocity vector of the vehicle, 2-the modified potential field force function, and 3-their integration of the fuzzy controller, which reposes on adjusting the factors of repulsive potential field in real time.

Trajectory Planning Methods

One of the most crucial tasks of autonomous vehicles is the path planning. It stands on the proprioceptive and the exteroceptive sensors’ measurements [1]. The vehicle shapes its surrounding environment in two or three dimensions and locates itself in it. Then, it uses trajectory planning algorithms to determine its path to move and avoid any collision with the obstacles. The path planning methods are multiple, and the most-used approaches are highlighted in [1]: visibility graph, cell decomposition, Dijkstra [5], etc. This section describes two of these methods: “A Star” the well-known robot trajectory planning that will be compared to “the modified potential field” method to overcome the local minima issue.

A-Star:

“A-Star” is one of the most effective trajectory planning methods. It is used in the robot navigation issues in congested environment and in multiple digital games such as mazes [19]. The fundamental idea is to find the shortest path, and to optimize Dijkstra’s criterion [20]. A* stands on Euclidean horizon to evaluate the distance between the different states. The basic algorithm does not take into account the obstacles between the current state and the final state. Atypical mechanisms overcome such as issues when the robot gets stuck. The procedures are: 1- To look for other paths when the robot is in a sticking state; 2- To avoid any path leading to blocking nodes. To do so, two path lists are created. The first is the OPEN list, it includes the set of paths to explore the parent nodes and find the optimal path. The second is the CLOSED list leading to dead nodes. Sometimes, the path searching is time-consuming. To overcome such constraints, optimization criteria are associated with the planning issue. Some others opted for heuristic solutions as shown in [21].

Artificial Potential Field Method:

The APF method can be easily implemented and executed in real time for control and navigation purposes. The basic concept of the vehicle motion, in the potential field, may be interpreted as a moving particle in a field of two electric particles with different signs [1]. Analogically, the positively charged particle is the vehicle. The global position is negatively charged, and the obstacles are considered as a set of particles with the same charge as the vehicle. The field potential gradient may be interpreted as forces that constrain the positive particle to reach the position of the negative one. The barriers create repulsive forces and push the robot to move away from obstacles [20–21]. At each configuration Qi, the positive APF forms an attractive force, which is defined as the negative gradient of the field. It denotes the favorable direction of the vehicle motion to reach the global position Qg. The combination of the attractive and the repulsive forces drives safely the vehicle to its destination.

Many forms of the positive potential exist. The most commonly used throughout the literature are parabolic and conical functions. The robot takes advantage of the first function when it is far away from the goal state. And it uses the second one, once it gets in a circle defined by its center that corresponds to the global position and its unit radius.

Parabolic functions:

The mathematical expression (1) formulates the shape of the parabolic APF.

\[
U_{pa}^{k^a}(Q) = \frac{1}{2} k^a \left\| d(Q) \right\|^2
\]  

(1)

Where \( k^a > 0 \) is a positive constant, \( d(Q) = Q_g - Q \) is the Euclidean distance between the global state and the current state. \( Q_g = [x_g, y_g]^T \) and \( Q = [x, y]^T \) in 2D, or \( Q_g = [x_g, y_g, z_g]^T \) and \( Q = [x, y, z]^T \) in 3D. The function (1) is perpetually positive, and its global minimum equals to zero. The approach assigns the
highest potential field to the starting state. It can be considered as a particle of a mass “m” and located at a relative height “h” to the global state. The particle holds a potential energy $E_p = mh$, which is transformed into kinetic energy as the vehicle approaches its destination. The APF shape provides the vehicle the best direction and the optimal path. The potential gradient is defined as a proportional vector of the difference between the global $Q_g$ and the current states. The further the robot is from its endpoint, the greater is the gradient, and the bigger the attractive forces. The resulting force of this field is the negative gradient (2).

$$f^\alpha (Q) = -\nabla U^\alpha (Q) = -k^\alpha d (Q)$$

Conical function:
The conical gradient features are very interesting, because its gradient is constantly dissimilar to the parabolic potential field gradient. The mathematical formulation (3) defines the shape of the conical potential and its gradient.

$$U^\alpha (Q) = \frac{1}{2} k^\alpha \| d (Q) \|$$
$$f^\alpha (Q) = -\nabla U^\alpha (Q) = -k^\alpha d (Q)$$

Repulsive Potential Field:
Several forms are quoted in literature. The hyperbolic function is introduced for the first time by Khatib [2, 6]. It associates to every obstacle a repulsive charge and affects each spatial cell according to (4).

$$U^\alpha (Q) = \frac{1}{2} k^\alpha \left( \frac{1}{n_i(Q)} - \frac{1}{n_o^\alpha} \right)^2 \text{if } n_i(Q) \leq n_o^\alpha$$
$$0 \text{ otherwise}$$

Where $k^\alpha > 0$ is a positive constant. $n_o^\alpha$ is the distance of influence of the $i$th obstacle. $n_i(Q)$ is the smallest distance between the current robot state and the $i$th obstacle. The potential is defined as being zero outside the range of the objects’ influence, positive inside, and infinite above the objects.

$$n_i(Q) = \min_{Q_o \in O} |Q - Q_o|$$

The resulting forces are expressed as follows.

$$f^\alpha_i = -\nabla U^\alpha_i (Q) = \left\{ \begin{array}{ll} k^\alpha (Q - Q_o) \sqrt{\frac{1}{n_i(Q)} - \frac{1}{n_o^\alpha}} \text{if } n_i(Q) \leq n_o^\alpha \\ 0 \text{ otherwise} \end{array} \right.$$  (6)

Sometimes, the vehicle’s final position is too close to an obstacle. The vehicle environment configuration creates local minima in the neighborhood of the global state. The solution, to such issues, is to multiply the repulsive potential, according to each obstacle, by the distance between the current state and the goal.

$$U^\alpha_i (Q) = \frac{k^\alpha}{2} \left( \frac{1}{n_i(Q)} - \frac{1}{n_o^\alpha} \right)^2 d (Q) \text{if } n_i(Q) \leq n_o^\alpha$$
$$0 \text{ otherwise}$$

The total potential field and forces are the sums of two attractive and repulsive fields and forces respectively.

$$U^\alpha(Q) = U^\alpha_1(Q) + U^\alpha_2(Q) = U^\alpha(Q) + \sum_{i=1}^{n} U^\alpha_i(Q)$$
$$f^\alpha(Q) = f^\alpha_1(Q) + f^\alpha_2(Q) = f^\alpha(Q) + \sum_{i=1}^{n} f^\alpha_i(Q)$$

Trajectory Planning Techniques With APF Approach
Three strategies of path planning are briefly outlined. Constraints, such as vehicle dynamic, could be
assigned to these three techniques in order to formulate an optimization problem.

The vehicle input corresponds to the total force. \( F = f'(Q) \quad (9) \)

The velocity of the vehicle equals the total force. \( \dot{Q} = f'(Q) \quad (11) \)

The vehicle corresponds to an unitary mass point and moves under the effect of the total force. \( \ddot{Q} = f'(Q) \quad (10) \)

**Artificial Potential Field Method and Local Minima**

The artificial potential field method is still sensitive to narrow paths, which induce chattering issues, and to local minima, which are a source of blocking configurations \([10–13]\). The major drawback in APF methods is the possibility of getting stuck in these local minima \([22]\). They are commonly related to the vehicle’s workspace configuration and especially to weight coefficients, associated to each obstacle during the APF design \([2]\), when attractive forces cancel the repulsive ones. Numerous approaches are put forward to overcome the local minima issue, thus, it remains a difficult problem. Barraquand and al. proposed in \([23]\) a Randomized Path Planner. The approach suggests moving the robot randomly at each time it gets trapped in a local minimum. Therefore, the determined path usually loses some of its smoothness and some configurations take a long time to get the robot far away from the local minimum. Miguel A. Padilla Castaneda and al. \([6]\) suggested driving the vehicle along the nearest barriers in the same clockwise direction each time the vehicle stuck in a local minimum. Some others suggest reshaping the potential field, making it harmonics, and call global optimization algorithms such as genetic algorithms \([6, 24]\).

However, the first suggestion is only available for narrow spaces and minor local minima. The second is insufficient for the vehicle planning trajectory considering the vehicle dynamic. And the third one is considerably time-consuming for real-time applications. The objective of this new approach is to overcome and eliminate the local minima issues. The idea is inspired from pouring a liquid matter, water for instance, with high pressure—attractive potential field—from the initial state until it reaches the global state. The proposed strategy consists of designing the total APF as the classic method according to the two sets of equations (1) and (7). The strategy uses the potential gradient descent algorithm (PGDA) to discover the vehicle path. The PGDA is executed as long as the ongoing state is different from the global minimum. If it cannot go forward, trapped in a local minimum, then the algorithm adds some extra repulsive potential to the current state until it gets free, according to the set equation (7). Once the vehicle destination is reached, the PGDA searches, once again, the final and practicable vehicle path in the new potential-field once again.

The artificial proposed potential field method skeleton for online trajectory planning is given as follows:

1: Design the attractive PF “Ua” according to global state.
2: Design the repulsive APF “Ur” according to each obstacle and its weight coefficients “ki” and No \((N_{0\text{Long}}, N_{0\text{Lat}})\).
3: Assign the initial state Q i to the path vector.
4: while \(d(Q) \neq 0\) do
5:   Call the potential gradient descent algorithm (PGDA) to determine the next state.
6:   if (PGDA is stuck AND \(d(Q) \neq 0\)) //local minima
7:     Add a new repulsive APF “Ur” to “Ua”.
8:   else
9:     Add the current state to the path vector.
10: end if
11: end while
12: Call the potential gradient descent algorithm (PGDA) to determine the new path.

**Simulation Strategy and Preliminary Results**

To test the proposed method, we created a vehicle workspace of ten meters wide and thirty-five meters long. Multiple obstacles produce two different local minima between the initial vehicle position and its destination as illustrated in figure 1 and 2. The area is divided into squares of 0.25 × 0.25 m. With a standard APF method, the potential gradient descent algorithm is attracted and blocked by the nearest minimum even after adding random noise. So to avoid global optimization and following the obstacles, the modified APF approach eliminates the local minima as shown in figure 1 during the first PGDA execution. It is possible to see the position of each local minimum in figure 2. The second operation of PGDA
Adapted MAPF to Vehicle Trajectory Planning

To fit the vehicle dynamic, some additional attractive potential field is added to the static and dynamic obstacles according to the relative speed between the obstacles, which could be other vehicles, barriers, etc., and the equipped vehicle. The effect of the relative speed takes places on the obstacle coefficient, in the longitudinal and lateral direction, as follows:

\[
\begin{align*}
    d_{\text{long}}^{s} &= d_{0}^l + v_{r} T_{d} \\
    N_{\text{long}}^{l} &= d_{\text{long}}^{s} / h
\end{align*}
\]  \hspace{1cm} (12)

\[
\begin{align*}
    d_{\text{lat}}^{s} &= d_{0}^l \\
    N_{\text{lat}}^{l} &= d_{\text{lat}}^{s} / h
\end{align*}
\]  \hspace{1cm} (13)

Where: \(d_{0}^{l}\) and \(d_{0}^{l}\) are the minimum, longitudinal and lateral, inter-vehicular distance; \(v_{r}\) is the relative speed of the \(i^{\text{th}}\) target; \(T_{d}\) is the driver’s reaction time—\(T_{d}=0.5\)s for a human driver, and for autonomous system \(T_{d}\) is set to the maximum time stamp of its data processing that can be expressed in
milliseconds—; and \( h \) is the unitary length/size of the sampled space, thus, different values could be assigned to the sampling space in the longitudinal and lateral directions, respectively.

To direct the vehicle from its initial position to its final one, the shape of the repulsive APF of the static and dynamic obstacles, which are on the road, is designed with the same curvature as the road’s ones in order to smooth the APF path. To reduce the time computing of the attractive APF, a modified version of the parabolic APF according to its formulation (1) is given as follows:

\[
U^a(Q) = \frac{1}{2} k^a \left[ x - x_Q \right]^2
\]

(14)

This new formulation allows, in somehow, to design a valid attractive APF for all configuration of the initial and global position of the vehicle. The initial position is set, herein, to the current vehicle position and its final position is set to the maximum laser scanner’s range in the x-direction and to the center of the i-th road lane in the y-direction. Consequently, the effect of the attractive APF (AAPF) takes place just in the x-direction and forces the vehicle to always move forward. The shape of the repulsive APF (RAPF) permits the vehicle to always stay on the road. The curvature of RAPF, which is associated to the obstacle, smooths the proposed MAPF method path, especially in sharp turns. Moreover, other criteria could be associated to the MAPF method so that the PGDA method favors a specific center of road lanes in accordance with the vehicle’s current position that will be presented by extra attractive APF (\( U_{ax} \)) replace all along the desired road lane center. This criterion acts as a guide to the PGDA algorithm from the current position of the vehicle to its global position.

**Path Planning on Distributed Architectures**

To adopt the MAPF method to our project, we suggest parallelizing the attractive and repulsive potential field design on different cores. For instance, a core will be assigned to compute the attractive part of the APF, additional cores could be assigned to construct repulsive APF according to the road lines. The coordinates of the road lines will be delivered, for instance, by an accurate on-board map according to vehicle local or global coordinates which could be delivered by global and local proprioceptive sensors. Another strategy to determine the road lanes coordinates, regarding the vehicle’s ones, is to fuse the vehicle proprio and exteroceptive sensors (Cameras, Lidars, Inertial Navigation Systems INS—Inertial Measurement Unit IMU—, and Global Position Systems GPS) and apply appropriate classification methods to the various objects around the vehicle and assort them according to their size and speed. However, such as methods are extremely time consuming. Thus, in this study, we resort to the first method that uses an accurate on-board numerical map of the racetrack. To this end, the vehicle is equipped with a GPS/INS unity to locate itself in the global and local plans. The GPS/INS unity data are fused according to an extended Kalman filter (EKF). The obstacles are detected by the embedded laser-scanner and processed according to the vehicle position estimate in some auxiliary cores. Finally, another core (in series) is used to fuse the two APFs and determine the vehicle itinerary with the potential gradient descent algorithm.

At this stage, the main challenge of the proposed method is the CPUs (cores) synchronization and the real-time execution. To this end, we previously developed a real-time simulator of collaborative and autonomous vehicles based on Pro-SiVIC and RTLAB. ESI-Civitec provides us a real-time simulator of infrastructure environment, vehicles’ dynamic, and embedded sensors (Inertial Navigation System, odometers, Light Detection And Ranging, camera, etc.), named Pro-SiVIC. This simulator allows the user to conceive various scenarios for the real-time simulation of the advanced driver assistance systems (ADAS), intelligent and autonomous and evermore for collaborative vehicles. Pro-SiVIC provides the possibility to adjust and to set the sample time for each sensor in real time. The user can activate several operation modes instantly, which can be modified during the simulation: to switch a sensor on or off, to record data in a file, and use Data Distributed Store “DDS” mode to send sensor data for external applications such as Orchestra, which is a “C” application. It extends the RTLAB connectivity capabilities to co-simulation, written in distinctive programming languages or generated by various simulation tools. In this study, Orchestra is mainly used to synchronize the different tools executed during the simulation scenarios (Pro-SiVIC, RTLAB, C/C++, Matlab/Simulink), to broadcast the vehicles’ state estimates, to rout signals with specific subsystems (cores) and Orchestra framework, and for inter-vehicular communication purposes, etc. The RTLAB (Real-Time LABoratory) allows the user to simulate the embedded real-time data processing, such as data fusion of embedded proprio and exteroceptive sensors with the on-board map for accurate ego and objects localization. Opal-RT technologies’ products are used for parallel...
computing and to ensure the real-time execution. The two platforms—RTLAB and Pro-SiVIC—are complimentary [25]. Their association creates a powerful simulator for real-time simulation of ADAS, intelligent/autonomous and collaborative vehicles. For more information, the reader is referred to [25]. The vehicle path planning skeleton, on distributed architectures, is derived from the previous algorithm for online path planning.

1: On the first core: design the attractive APF “$U_a$” according to: the global state; the selected road lane center; the laser scanner maximum range and the selected method.

2: On the second core: design the repulsive APF “$U_r$”, in parallel with the first core, on the distributed architecture, according to: each obstacle and its relative position and speed $N_o$ ($N_o^{long}$, $N_o^{lat}$), and weight coefficients “$k_r$”.

3: On the third core:

4: Retrieve the first two cores results and assign the initial state $Q_i$ to the path vector.

5: while ( $d(Q) \neq 0$ ) do: // while the vehicle current position is different from the $Q_g$.

6: Call the potential gradient descent algorithm (PGDA) to determine the next state.

7: if (PGDA is stuck AND $d(Q) \neq 0$) //local minima

8: Add a new repulsive APF “$U_r$” to “$U_t$”.

9: if (PGDA is stuck N time in the same state)

10: print: PGDA cannot overtake the obstacle.

11: break

12: end if

13: else

14: Add the current state to the path vector.

15: end if

16: end while

17: Call the potential gradient descent algorithm (PGDA) to determine the new path.

Simulation Scenario

The main scenario consists in generating a virtual environment—set here to the standard Pro-SiVIC racing circuit named “HorseRing”—including bitumen road, traffic barriers, landscape, etc. Three cars are involved in this scenario. They are all equipped with proprioceptive sensors: a global position system (GPS) and inertial measurement unity (IMU), attached to the car chassis, which measure vehicles’ position, velocity, acceleration and orientation, and with an exteroceptive sensor, a laser scanner, to scan its environment, detect the static and moving obstacles and measure the relative distance between the vehicle and the obstacles. The acquisition time stamp of the sensors ($T_s^{GPS}$, $T_s^{IMU}$, $T_s^{laser}$), and the parameters’ value of the proposed path planning method are given in the table 2. The time stamp of the cores that are used for vehicle localization and path planning, the time stamp of each block (subsystem) is set to 10 ms. The sensors’ data fusion is processed on the first core, then the results are transferred to the next two cores to compute the repulsive and attractive artificial potential fields, $U_r$ and $U_a$, respectively. Finally, the two APFs are sent to the fourth core to fuse the two APF, find and plot the final vehicle path, as presented in the figure 3. The computation time, for vehicle trajectory planning process, equal to the sum of the computational time of all the blocks in series. For the cores that handles the two APFs design in parallel, the computation time is chosen as the maximum of both. It gives $T_t = T_{NI} + max ([T_{N2}, T_{N3}]) + T_{N4}$. As RTLAB insures the real time execution according to the specified time stamp, as stated above, the time stamps are set herein to 10 ms, that makes $T_t = 3 * T_{NI} = 30$ ms.

![Figure 3: Distributed MAPF on Pro-SiVIC and RTLAB platform](image-url)
The simulation results, figure 4, show that the selected path, the green trajectory, by the potential gradient descent algorithm (PGDA), incites the vehicle to always move forward, as expected, until it meets the center of the right road lane and gets repelled by the repulsive artificial field of the right road border. Once the PGDA stuck in the same state, behind the obstacle represented, in the same figure, by the red vehicle, the proposed method affixes some additional repulsive APF to the current state and looks for the next state by furthering the vehicle motion in the x and y directions. The same operations are performed repeatedly until the PGDA reaches the global or the final state. The effect of the inter-vehicular speed, as expressed previously by the two sets equations (12) and (13), are visible at each time the vehicle gets blocked and needs to move forward and change lane. As can be seen in the right part of the above figure, the parameter $h$, which is the unitary length (size) of the sampled space, determines how smooth will be the vehicle trajectory. However, its value induces to a new trade-off between the smoothness of the trajectory and computing time of the MAPF method will take. Thus, the smaller is the value of $h$, the smoother is the trajectory and the greater is the computing time, and vice versa. A solution to smooth the vehicle trajectory could be an EKF using a basic model of a 2D moving object, or a bicycle dynamic model.

**Conclusion**

This paper presented online trajectory planning with a modified potential field method on distributed architectures for autonomous vehicles. The approach adds dynamically repulsive artificial potential field, to the standard APF, at each the vehicle gets trapped in a local minimum. The operation eliminates any blocking configuration, in the vehicle environment, that is the major challenge of the standard APF method. This operation is continuously repeated as long as the final destination of the vehicle is not reached. Finally, the strategy uses the potential gradient descent algorithm (PGDA) to determine the new feasible path. The simulation results, the comparison between the A$^*$ MAPF performances, show the effectiveness of the suggested strategy and its alignment with real-time path planning. However, it is worthy to emphasis on the fact that the suggested MAPF method computation time depends on the
number and the size of the local minima, that PGDA meets at the first run. A solution for such as problems is to associate an appropriate distance of influence \( N_{oi} \) of each local minimum, in x and y directions, in accordance with its size and its relative speed according to the vehicle. In addition, other criteria are proposed to reshape the repulsive artificial potential field according to the road curvature, and to specify some favorite direction. To reduce, in meanwhile, the computing time of the MAPF method, the different step are computed separately on different cores on distributed architectures. The future work will focus on the integration of the bicycle dynamic model in the path planning, and to design an advanced controller having as an input vector the MAPF trajectories as a reference path.

Acknowledgment
This work is part of COOPERCOM, a 3-year international research project (Canada-France). The authors would like to thank the firms ESI-Civitec in France and Opal-RT in Canada, the MITACs Accelerate Canadian funding program, the National Science and Engineering Research Council (NSERC) of Canada and the "Agence nationale de la recherche (ANR)" in France for supporting the project.

References
8. You Li, Yassine Ruichek and Cindy Cappelle (2011). 3D triangulation based extrinsic calibration between a stereo vision system and a LIDAR. 14th International IEEE Conference on Intelligent Transportation Systems Washington DC, USA.


