



Vehicle Autonomous Navigation with Context Awareness

Federico Faruffini, Alessandro Corrêa Victorino, Marie-Hélène Abel

► To cite this version:

Federico Faruffini, Alessandro Corrêa Victorino, Marie-Hélène Abel. Vehicle Autonomous Navigation with Context Awareness. 2nd IEEE International Conference on Human-Machine Systems (ICHMS 2021), Sep 2021, Magdeburg, Germany. pp.1-4, 10.1109/ICHMS53169.2021.9582660 . hal-03523227

HAL Id: hal-03523227

<https://hal.science/hal-03523227>

Submitted on 12 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vehicle Autonomous Navigation with Context Awareness

Federico Faruffini, Alessandro Correa Victorino, Marie-Hélène Abel

Abstract—Nowadays, many models performing global robotic navigation exist, and they are capable to drive safely and autonomously, and to reach their set destination. However, most of them don't take into account the information coming from the context in which such navigation is occurring, resulting in a severe information loss. Without Context-Aware Navigation, it is not possible to build a model to let the vehicle adapt its behaviour to the situation, in the way a human driver spontaneously does. It is therefore needed a study on how to connect the contextual information with the robot's control loop. For our solution we will use semantic structures known as ontologies, that help the vehicle reason in real-time and change its own behaviour in function of given contextual information. After a definition of the Context of Navigation, in this paper we propose an approach to the problem of encoding the Context Awareness in the Autonomous Navigation's controller. Finally, such approach is put to the test in a simulator, to discuss the results achieved.

Index Terms—Automation Systems, Intelligent Transportation, Autonomous Mobile Robots

I. INTRODUCTION

The problem of autonomous car navigation has been a research topic for years now, and since the DARPA Grand Challenges [1] a faster development in this field has begun. Current state-of-the-art mathematical models are now able to perform correctly the global navigation between two set points in a safe way, performing obstacle detection if needed. However, despite all these technological advancements in the field, most of these models don't take into account most of the contextual information of the navigation. One of the components of the Context of Navigation, for instance, is represented by the passengers: they could have a preference on the driving styles of the car or on the way overtaking is handled, and so on. These are information that a human driver usually takes into account while driving and adapts his driving to. By missing the contextual information it is not possible for the autonomous vehicle to fully adapt to the situation, resulting in a behaviour that is always the same, no matter the situation. A related problem is that of perceived safety: standard mathematical models for autonomous navigation don't include human-like trajectories. This leads to discomfort for the passengers, since the robotic component is too much perceivable. Instead, as in [2], a robot driving in a familiar way to the passengers could help them to feel safer and more comfortable.

Université de technologie de Compiègne, CNRS, Heudiasyc (Heuristics and Diagnosis of Complex Systems), CS 60 319 - 60 203 Compiègne Cedex federico.faruffini@etu.utc.fr, alessandro.victorino@hds.utc.fr, marie-helene.abel@hds.utc.fr

With the use of the Context of Navigation and a method to reason over it, it is possible to solve both problems.

II. PROBLEM STATEMENT

As previously said, there exist just a few studies on the interaction between the context modelling and the vehicle control loop. This results in the impossibility to have a context-aware control of the car, since many information are not taken into account. A new solution to incorporate these missing information is required in order to meet the passengers' needs and preferences, as well as to adapt the driving style to an even safer one, for both the passengers and the vehicle loads.

We will now introduce a simple example to show why the contextual information is so important to the navigation.

Let's consider an autonomous car with no driver and one or more passengers. During its trip, the vehicle may approach a speed bump or may need to overtake an obstacle in front of itself. The passengers may have some preferences on the driving style used by the car to perform these operations. For example, they may want an acceleration that is smoother than the standard one provided by the autonomous system. Another example could be the desire to set the maximum speed of the car to be below the maximum one permitted by the traffic laws. In this paper we will see how it is possible to modify the behaviour of the vehicle on the basis of these information.

III. RELATED WORKS

In the field of autonomous cars, global navigation can be handled in different ways. The first one is GPS-based, as it exploits information from a GPS sensor to estimate the current position of the car and enables it to drive to its destination. This solution still presents some issues, being highly expensive and being subject to GPS sensor fails, as those depicted in [3]. A cheaper and more robust way to handle global navigation is through Visual Servoing (VS). This method is based on the images caught by cameras placed on the car, and lets it proceed forward in the center of its lane, even in the cases of turns or roundabouts. We will see a more complex image-based VS algorithm proposed by Lima and Victorino in [4] in Section V.

Regarding the context, many studies on the navigation have been published, mainly on the aspects of the context that are dynamic to the navigation itself. Examples can be the other vehicles, the obstacles, the sections of the path to reach the goal and so on. Some of these studies exploit semantic structures known as ontologies, which allow to

store real world data and reason on them, inferring new information. Shlenoff et al. in [5] propose to use ontologies to aid the navigation planning, in the case of the detection of a possible obstacle. In particular, they show how such structure would allow for reasoning on the possible effects of the collision with it, and for a decision on how to handle it. For example, a collision with a newspaper would not be avoided, while one with a big metallic cylinder would be, since it would probably produce an extensive damage to the vehicle, its passengers and its load. Regele in [6] proposed a way to use ontologies to model traffic laws, in particular at difficult intersections. In his work, he models the road as a directed graph, and through the use of labels he defines the relationship among them. For instance, two merging lanes are marked as conflicting so that, when the car is in one of them, it knows it has to be careful for possible vehicles coming from the other lane. Regele's proposition is simple and reduces the information the car needs to process in real time, since it is not required for it to know the full topology of the intersection, but just the parts related to its movement. Finally, Armand et al. in [7] proposed an extremely simple way to implement a human-like driving behaviour in autonomous cars. This study too is based on ontologies and deals with the inference the car can make when travelling in presence of other cars or pedestrians. A human-like behaviour of the car could also enhance the perceived safety by the passengers, as familiar manoeuvres could make them more comfortable with the idea of a robotic driver, as depicted by Elbanhawi et al. in [8].

As previously said, these studies didn't consider - or considered just a small fraction of - the context of navigation, resulting in a huge information loss.

IV. CONTEXT MODELLING

To do our semantic modelling of the Context of Navigation we used a semantic technology tool called ontology. Before talking about the Context of Navigation for our tests, we will briefly introduce ontologies and the reasoning over them.

A. Ontologies

An ontology is a way to represent a domain of discourse, in such a way that it is understandable by both humans and computers. In other words, it provides a structure to store real world data in an organized way. Ontologies have some differences from standard relational databases, and some advantages to our scope. The main advantage is the possibility to reason on them, inferring new information over the asserted one. This operation is done by pieces of software called Reasoners. A popular language for ontologies is OWL, the Web Ontology Language, and its standard editor, Protégé [9], was developed by the Stanford University. Each information in an OWL ontology is stored as a triple, in the form *Subject - Predicate - Object*, for example in our context we could have the following triples:

$$\text{MyCar} \quad \text{hasPassenger} \quad \text{Emily} \quad (1)$$

$$\text{MyCar} \quad \text{hasMaxSpeed} \quad 50 \quad (2)$$

With all the triples in our storage we build a graph, called the Knowledge Graph, and we obtain our Knowledge Base. OWL Ontologies make the Open World Assumption [10], for which the statements that are not explicitly said to be true are not assumed to be false, and this allows for a better reasoning over them. Ontologies have 2 components: a terminological component (TBox) and an assertional one (ABox).

The TBox contains the Classes for our Context. For example, we could have a class called *EgoVehicle*, and another one called *Obstacle*. In the TBox we find also the relationships between Classes, called Object Properties, or between a Class and some data, called Data Properties. We have an example of Object Property in (1) and of Data Property in (2).

The ontology ABox contains the instances of the Classes defined in the TBox and the real world data assigned to them: following our example, MyCar is an instance of the class EgoVehicle.

B. Reasoning over the ontology information

A first level of reasoning and inference is done, at run-time, by the ontology itself. This is the case of Defined Classes. We have an example in Fig. 1a: Infant, OldPerson, Passenger and Toddler are Defined Classes. They all are of type Person, but satisfy different other conditions, for instance the Toddler class is defined as having the age value between 1 and 4. The good thing about Defined Classes is

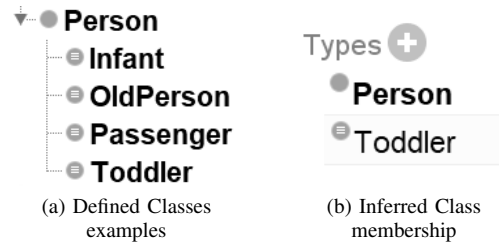


Fig. 1: Inference over Defined Classes

that we don't need to specify them while creating an instance, since they are inferred by the system at run-time. So, if we create an instance *Ginny* of type *Person* and assign it the age value 2, at run-time the system will infer that *Ginny* is also of type *Toddler*, as we can see in Fig. 1b.

When a more complex level of inference rule is required, we can use SWRL, the Semantic Web Rule Language [11]. It allows to define rules that may contain computations of the composition of other higher-level boolean operations. A SWRL rule has two parts: an antecedent and a consequent. We can read a SWRL rule in this way: if the antecedent is true, then the consequent is also true. We can define really useful rules to our scope, for example in the following rule

$$\begin{aligned} & \text{EgoVehicle}(\text{?v}) \wedge \text{hasPassenger}(\text{?v}, \text{?p}) \wedge \text{Person}(\text{?p}) \wedge \\ & \text{hasLocalRoadDrivingPreference}(\text{?p}, \text{?pref}) \rightarrow \\ & \text{hasLocalRoadDrivingStyle}(\text{?v}, \text{?pref}) \end{aligned}$$

we define a simple rule to let the ontology suggestions adapt to the driving style on local roads to the one preferred by the

passenger. Atoms with just one parameter, as *EgoVehicle(?v)* are true if the object used as parameter has the specified Class: in this case, the atom will be true only for all *?v* belonging to the *EgoVehicle* OWL Class, false otherwise. Atoms with two parameters, as *hasPassenger(?v, ?p)* are true for each case in which there is a Property (in this case, an Object Property) that holds. Since all the components of the antecedents are put in a logic AND relationship, just one false value between them will stop the rule from triggering. This rule was taken as example, but we have to notice that it is not able to give a correct result in the case multiple passengers are present, since it will be triggered one time for each of them. This will result in inferring possibly different driving styles for the same car.

Another SWRL rule in our ontology is the following:

$$\begin{aligned} &EgoVehicle(?v) \wedge hasPassenger(?v, ?p) \wedge Person(?p) \wedge \\ &hasPassengerState(?p, ?passengerState) \wedge \\ &InjuredState(?passengerState) \rightarrow \\ &hasDrivingStyle(?v, EmergencyLocalRoadDrivingStyle) \end{aligned}$$

It is used to encode that, when our ego vehicle has at least an injured passenger, its driving style changes automatically an emergency one. In the rule is presented only the version with the local road, but similar rules are possible to be applied to highways and other roads.

A SWRL rule can also use another one as part of its antecedent, allowing for more complex layers of reasoning. For example, the next rule needs the last one to be true in order to be triggered:

$$\begin{aligned} &EgoVehicle(?v) \wedge \\ &hasDrivingStyle(?v, EmergencyLocalRoadDrivingStyle) \rightarrow \\ &canUseEmergencyLane(?v, true) \end{aligned}$$

C. The Context of Navigation

In order to apply the contextual information to the autonomous navigation, we had to define the Context in which the vehicle is operating, by selecting the useful information to our scope. Following the definition of Context by Dey [12], we gave our definition of Context Of Navigation [13]:

The navigation context is any information that can be used to characterize the situation of navigation over a given period of time. Here, navigation is a movement considered relevant to the interaction between a driver and an application, including the driver and the applications themselves.

Furthermore, we define the 2 components of the Context of Navigation. The first component is the Dynamic Context, that comprehends all the information that change with respect to the car's navigation. The second component is the Static Context, and it contains all the information that don't vary with respect to the car's navigation.

If we want to model the driving preferences of the passengers, we are interested in the Static Context. Since a complete Context would be too much complex to model from scratch, we will stick to its parts that are needed for the sake of our example. In Fig. 2 we can see the OWL Classes in

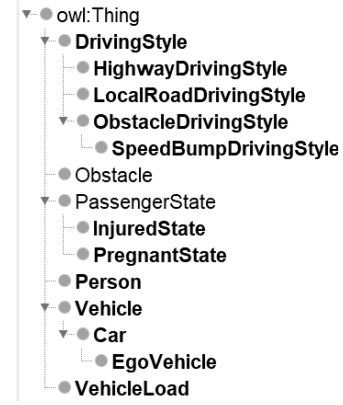


Fig. 2: The Ontology Classes in the Context of Navigation

our ontology, in the Protégè editor. We can see examples on contextual information, as the possibilities to have different driving styles for each kind of road or to approach different obstacles. This way, we have the possibility to encode the driving preferences of specific passengers, and not just to create a single behaviour for the same kind of situation. For instance, we could have the following triples in our ontology:

MyCar hasPassenger Emily
Emily hasDrivingStylePreference Emily_SBDS
Emily_SBDS hasMaxSpeed 5

(please notice that, for the sake of readability, the name of the instance *Emily_SpeedBumpDrivingStyle* was shortened to *Emily_SBDS*). In this example we have the information that Emily prefers overcoming speed bumps at a speed not higher than 5km/h. This example helps to realize how the different contexts could be modelled, as for another passenger we could have different preferences. The Static Context of

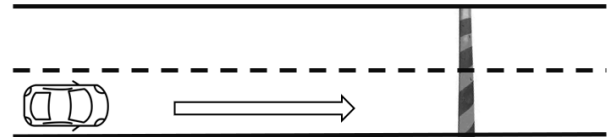


Fig. 3: Scenario 1: the car must drive over a speed bump

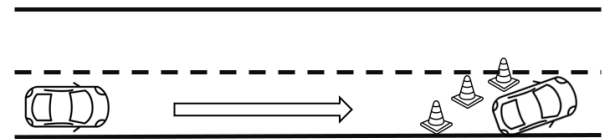


Fig. 4: Scenario 2: the car must avoid a static obstacle

Navigation contains also the information related to the load of the vehicle, as in the class *VehicleLoad* in Fig. 2. This is useful because the objects being transported could affect the behaviour of a human driver, and so should affect the autonomous one too. For instance, with a fragile load, the vehicle must avoid sharp accelerations or decelerations - when possible - to avoid damaging it.

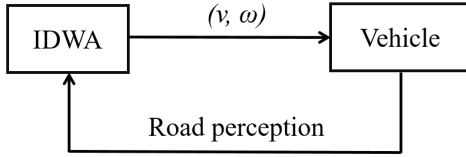


Fig. 5: Block diagram of the IDWA controlled vehicle [4]

In Fig. 3 we can see one of the possible scenarios we can encode in our Context of Navigation: the vehicle proceeds towards a speed bump and has to overcome it. As said before, depending on the on-board passengers and the load we may want to have different behaviours. By the use of triples in sec. IV-C we can force the car to have a specific behaviour. In Fig. 4 we have a second scenario, in which the car has to overcome a static obstacle (a cluster of obstacles in this case) on its path, by changing lane. This can be done, again, by setting a specific preference for the passenger or the carried load in this specific case. We will see this scenario in Sec. VII. Of course, many more scenarios can be thought, as intersections or other more complex situations. These will be left for future studies.

V. AUTONOMOUS NAVIGATION SYSTEM

Lima and Victorino in [4] proposed a vision-based hybrid control approach. It is composed by 2 parts: a VS one and a dynamic window approach (DWA) one, and it takes the name of Image-based Dynamic Window Approach (IDWA). The VS controller, more robust than the second, is used as deliberative controller to drive the car correctly in its lane. The DWA component is used as a reactive controller when an obstacle is perceived. In this case, all the acceptable couples (v, ω) of longitudinal and angular speed are computed, and through an optimization process the best couple is selected to perform the obstacle avoidance. The optimization objective is the maximisation of:

$$\alpha \cdot \text{heading}(v, w) + \beta \cdot \text{dist}(v, w) + \gamma \cdot \text{velocity}(v) \quad (3)$$

where α , β and δ are real-valued parameters to be properly tuned. After the successful obstacle overtaking, the control is brought back to the VS component to continue the navigation. Finally, in the case there is no acceptable couple (v, ω) to safely avoid the obstacle, the vehicle is stopped. The DWA component is obtained by minimizing the sum of three components with related parameters. By modifying these parameters' values, the authors were able to set the response time and the hardness of the steering of the vehicle while driving. The block diagram of the system using the IDWA controller is in Fig. 5: the feedback given by the Vehicle to the IDWA is the road perception, composed of the front image and the laser scans of the forward area. This way it is possible for the VS component to compute the control over the image data, and for the DWA component to start avoiding one obstacle, if present.

This control model was eventually tested on a real autonomous car and was proven to have better performances

than those of a standard dynamic window approach.

VI. IMAGE-AND-CONTEXT-BASED DYNAMIC WINDOW APPROACH

We will now present a more complex proposal, called ICDWA (Image-and-Context-based Dynamic Window Approach). It is based on IDWA, and a new component is added to the objective function, which becomes:

$$\begin{aligned} \text{ICDWA}(v, w) = & \alpha \cdot \text{heading}(v, w) + \beta \cdot \text{dist}(v, w) + \\ & \gamma \cdot \text{velocity}(v) + \delta \cdot \text{context}(v, w) \end{aligned} \quad (4)$$

with $\delta \in \mathbb{R}$. Now, the problem is how to define the *context* component. We want our reasoner to provide a suggestion from the Context of Navigation of a max longitudinal and rotational speed (v_{CN}, w_{CN}) . Then, we want the *context* component in (4) to give a cost which depends on these values and the real ones (v, w) . The values of v and w can be considered separately in two different cost functions which can be added to form the *context* component, let's call them context_v and context_w :

$$\text{context}(v, w) = \varepsilon \cdot (\text{context}_v(v) + \text{context}_w(w)) \quad (5)$$

The parameter $\varepsilon \in [0, 1]$ in the equation is used to express and control how much we want to trust and rely on the contextual suggestion. For timing reasons, for the moment we just implemented a working solution for the first, keeping the second at a value of 0 in each case. We studied two implementations of ICDWA. In the first one we have, as in the IDWA solution, a brute force computation of (4) to find the better couple. In the second solution we implemented gradient descent to speed up the optimization process.

A. Unoptimized ICDWA

As for the IDWA algorithm, our first solution aims to find the values (v, w) which maximise (4). The implementation of context_v relies on the sigmoid function, so we will now recall it and its derivative:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (6)$$

$$\dot{\sigma}(x) = \sigma(x) \cdot (1 - \sigma(x)) \quad (7)$$

Let's call x the speed difference between the one being evaluated and the maximum proposed by the reasoner:

$$x = v - v_{CN} \quad (8)$$

then, we can implement context_v as:

$$\text{context}_v = \begin{cases} -\sigma(x), & \text{if } x \leq 0 \\ -(\sigma(0) + \dot{\sigma}(0) \cdot x), & \text{otherwise} \end{cases} \quad (9)$$

Since the values of $\sigma(0)$ and $\dot{\sigma}(0)$ can be computed respectively from (6) and (7), we can give a final definition of our context_v component:

$$\text{context}_v = \begin{cases} -\sigma(x), & \text{if } x \leq 0 \\ -\frac{1}{2} - \frac{1}{4}x, & \text{otherwise} \end{cases} \quad (10)$$

This behaviour is shown in Fig. 6, where we can see how the cost is higher for values $v < v_{CN}$, and how it decreases linearly when $v > v_{CN}$.

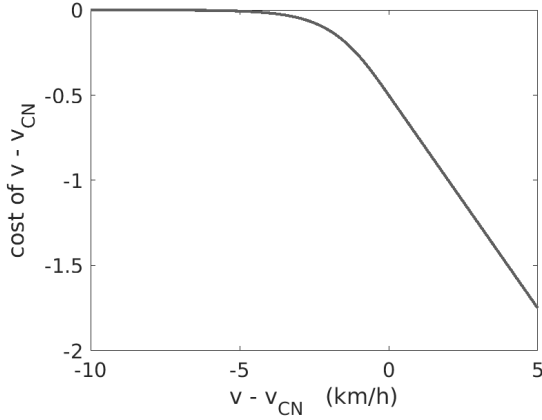


Fig. 6: The cost as in (10)

B. Optimized ICDWA

We also studied an optimized version of our solution, in which the search for the best (v, w) couple is found with gradient descent. Also, the optimization objective in this case was the minimization of (4). For this to be possible, we just inverted the sign for our $context_v$ component, obtaining its new equation:

$$context_v(v) = \begin{cases} \sigma(x), & \text{if } x \leq 0 \\ \frac{1}{2} + \frac{1}{4}x, & \text{otherwise} \end{cases} \quad (11)$$

Its behaviour is the opposite of the one shown in Fig. 6, as it increases as v gets bigger than v_{CN} . The equation in (11) is continuous, derivable and convex, thus it can be used in the optimization by gradient descent.

VII. TESTS

To model our ontology we used the Protégé editor [9] with the Pellet reasoner [14]. To simulate the scenarios we used a professional simulator called SCANeR Studio¹, but since it doesn't have an interface for Java (the language for ontology servers) we had to run the ontology using a Python library called owlready2 [15]. For our simulations we assumed to have precise input from our sensors, as frontal cameras and lidars, since the problem of correct obstacle detection is not being faced in this paper.

The scenario we simulated is the one of an autonomous car that has to avoid three static obstacles in its lane, by temporarily switching to the other lane, as in Fig. 4. Also, there is a passenger on board which prefers a slower speed than the maximum one, which was set to 10km/h for this first tests. A SCANeR Studio screenshot taken from a simulation is shown in Fig. 7.

A. Unoptimized algorithm

We firstly put to the test our unoptimized version of the ICDWA algorithm. The first 3 parameters in (4) were set to $\alpha = 1$, $\beta = 1.5$, $\gamma = 0.1$. We can see in Fig. 8 the



Fig. 7: Simulation in SCANeR Studio

results on a test with $\epsilon = 0$ and one with $\epsilon = 1$, $v_{CN} = 5\text{km/h}$, $\delta = 0.75$. The first test is the one in which we don't take into consideration the contextual component of (4), and the autonomous car proceeds just driven by IDWA. We can see clearly its behaviour pattern: as an obstacle is perceived, the car quickly breaks, then it overcomes it at a safe speed and finally it accelerates again. The second test lets us see the behaviour of ICDWA with a speed which is half of the max one of the car. We can see how the car's speed is generally much lower than in the other test, and that only in some instants the max suggested speed is not respected, as the objective function probably gave it less cost than to the other components. The car proceeded avoiding the obstacles correctly, as well as keeping in the center of the lane when in absence of obstacles. It is interesting to notice that for some values of δ we had much worse performance, as the contextual suggestions were taken too much in consideration with respect to the other components. This led to reducing too much the max distance kept from the obstacles, or interfering with the lane following.

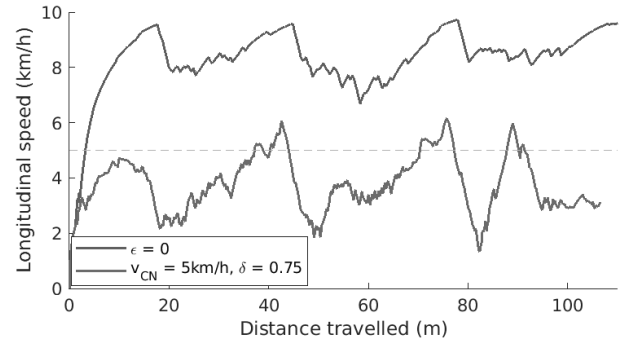


Fig. 8: Test result of unoptimized ICDWA

B. Optimized algorithm

For this tests, we used the following values for the first 3 parameters: $\alpha = 1$, $\beta = 0.3$, $\gamma = 0.1$. As we said before, the optimized version of the ICDWA controller uses gradient descent to quickly find the optimal values of (v, w) , instead of brute-forcing all the combinations. We can see in Fig. 9 3 tests we made with this method. The first one represents the case in which we don't have interest in the contextual information. Even if the positions of the obstacles are the same as before, we can see how the car behaves differently,

¹AVSimulation, SCANeR Studio, <https://www.avsimulation.com/catalog/>

with smoother acceleration/deceleration rates. The other two tests were done with two different v_{CN} values, and we can see how this time the maximum speed suggested was not exceeded by the car. Again, in these tests we have smoother variations with respect to those of Fig. 8. With this approach too, the car was able to avoid obstacles and proceed in the center of the lane when possible, so the tests were again successful. As in the first tests, we found out that a proper

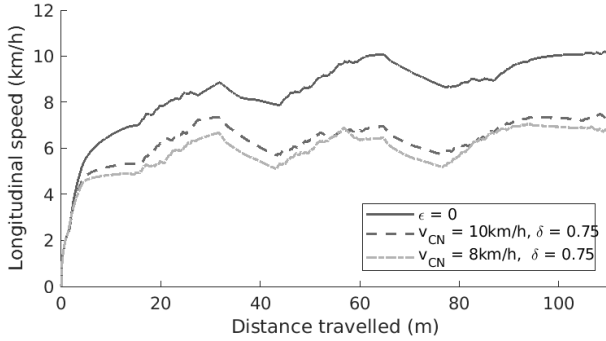


Fig. 9: Test result of optimized ICDWA with different parameters

tuning of the parameters is needed, as for some values of ϵ or δ the performance of the system could decrease, even if in a minor way if compared to that of the unoptimized solution.

VIII. CONCLUSION AND FUTURE WORK

In this paper we showed how a simple ontology-based Context of Navigation can be used to make autonomous car trips more comfortable for the passengers, by trying to adapt to their preferences. We have seen a starting example of OWL ontology that is applicable to this situation, and proposed how to model the interaction between its output and the control loop's. In particular, we proposed the Image-and-Context-based Dynamic Window Approach, giving its first component for the longitudinal speed v . We developed two versions of it, which differ on the presence of an optimization algorithm. Finally, we simulated the results in a professional simulator showing promising results, and confirming the superiority of the optimized version over the standard unoptimized one, which followed the DWA method.

Future works will study the way to handle the jerk and angular speed component of ICDWA, in order to complete the algorithm. The Context of Navigation must be expanded with a modular paradigm in order to let it contain all the possible information of interest, and the related reasoning structure. Also, there is the need for a more complete ontology in order to take into account the characteristics of the car. This way it will be possible to obtain more a realistic reasoning, based on the physical capabilities of the car: in the final end, a small car and a bigger one, with the same exact context and scenario, could react differently to the same input. Another key problem to be faced in future studies is the choice of the values for the parameters. As it was shown,

a proper tuning of α , β , γ and δ is needed to avoid problems in lane keeping and obstacle avoidance. Also, the real-time estimation of the parameter ϵ will be addressed to obtain a better adaptability to different scenarios, not only based on the traffic conditions.

Future works will consider the interaction of the autonomous car with other entities, in the field of study of Vehicle-to-Vehicle communication (V2V) and Vehicle-to-Infrastructure communication (V2I). These aspects will be crucial in order to improve the safety of the trip, as well as the overall performance of the system of systems comprehending all the road users.

Finally, as much information related to the vehicle, the destination, the load and the passengers will have to be stored and processed, data ethics and security will have to be considered at early stages of the project.

ACKNOWLEDGEMENT

This research study was funded by IDEX Sorbonne Université - Labex MS2T.

REFERENCES

- [1] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.
- [2] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *2010 IEEE International Conference on Robotics and Automation*, pp. 987–993, 2010.
- [3] M. Buehler and K. Lagnemma, and S. Singh, "Special issue on the 2007 DARPA urban challenge, Part I," *J. Field Robot.*, vol. 25, pp. 423–566, 08 2008.
- [4] D. Alves de Lima and A. Corrêa Victorino, "A hybrid controller for vision-based navigation of autonomous vehicles in urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 2310–2323, Aug 2016.
- [5] C. Schlenoff, S. Balakirsky, M. Uschold, R. Provine, and S. Smith, "Using ontologies to aid navigation planning in autonomous vehicles," *The Knowledge Engineering Review*, vol. 18, no. 3, p. 243–255, 2003.
- [6] R. Regele, "Using ontology-based traffic models for more efficient decision making of autonomous vehicles," in *Fourth International Conference on Autonomic and Autonomous Systems (ICAS'08)*, pp. 94–99, 2008.
- [7] A. Armand, D. Filliat, and J. Ibañez-Guzman, "Ontology-based context awareness for driving assistance systems," in *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pp. 227–233, June 2014.
- [8] M. Elbanhawi, M. Simic, and R. Jazar, "In the passenger seat: Investigating ride comfort measures in autonomous cars," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 3, pp. 4–17, 2015.
- [9] M. A. Musen, "The protégé project: a look back and a look forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015.
- [10] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. USA: Prentice Hall Press, 3rd ed., 2009.
- [11] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosz, and M. Dean, "Swrl: A semantic web rule language combining owl and ruleml," w3c member submission, World Wide Web Consortium, 2004.
- [12] A. Dey, "Understanding and using context," *Personal and Ubiquitous Computing*, vol. 5, pp. 4–7, 02 2001.
- [13] F. Faruffini, A. Correa-Victorino, and M.-H. Abel, "Towards a semantic model of the context of navigation," in *The Fifth International Conference on Information and Knowledge Systems*, 06 2021.
- [14] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Journal of Web Semantics*, vol. 5, no. 2, pp. 51–53, 2007. Software Engineering and the Semantic Web.
- [15] J.-B. Lamy, "Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies," *Artificial Intelligence in Medicine*, vol. 80, pp. 11–28, 2017.