



Dynamic Context Awareness in Autonomous Navigation

Selim Chefchaoui Moussaoui, Hugo Pousseur, Alessandro Corrêa Victorino,
Marie-Hélène Abel

► To cite this version:

Selim Chefchaoui Moussaoui, Hugo Pousseur, Alessandro Corrêa Victorino, Marie-Hélène Abel. Dynamic Context Awareness in Autonomous Navigation. IEEE International Conference on Systems, Man, and Cybernetics (SMC 2022), Oct 2022, Prague, Czech Republic. pp.2808-2813, 10.1109/SMC53654.2022.9945347 . hal-03934443

HAL Id: hal-03934443

<https://hal.science/hal-03934443>

Submitted on 11 Jan 2023

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.

Dynamic Context Awareness in Autonomous Navigation

Sélim Chefchaoui Moussaoui, Hugo Pousseur, Alessandro Corrêa Victorino and Marie-Hélène Abel

Heudiasyc Laboratory (Heuristics and Diagnosis of Complex System), UMR CNRS 7253

Computer Science Department, Université de Technologie de Compiègne (UTC)

Abstract—Many studies faced the problem of vehicle autonomous navigation in different fields, but nowadays just a few of them uses all the implicit information coming from the context in which such navigation is occurring. This results in a huge potential information loss that prevents us from adapting the vehicle's behavior to each different situation it may be in. In a previous work, we defined a method to model the static context of navigation using ontologies and take it into account in the command law when performing a local navigation task. In this paper, we extend our model of the context of navigation, and define a software architecture able to update the context dynamically, by using sensor information. The method is tested with real-time experiments on driving simulator. They show that the Context of Navigation can be effectively updated during the navigation and leads to a smarter vehicle's behavior on the road.

Index Terms—Intelligent Transportation Systems, System Modeling and Control

I. INTRODUCTION

The problem of autonomous navigation has been studied for years now, and nowadays many functioning solutions for it have been proposed and approved. These solutions generally allow vehicles to perform both local tasks (e.g. going through an intersection or avoiding an obstacle) using sensor information, and global tasks (e.g. following a desired itinerary) using other information sources like maps. By combining local and global tasks, efficient autonomous navigation can be performed nowadays, such that autonomous navigation solutions are now included by car manufacturers.

However, autonomous navigation systems do not take into account the contextual information coming from the vehicle and its environment. It follows that the vehicle has the same behavior regardless of the state of the passengers (e.g. pregnant or injured passenger) or the vehicle components (e.g. battery level). This contextual information which is not directly coming from the sensors constitute what we call the context of navigation. Being able to take it into account in the control law would lead to a smarter navigation that adapts better to the vehicle's environment.

In previous works [1] [2], we have shown that ontologies are an efficient tool to model such contextual information, as

they can be used with rules to provide driving suggestions from the modeled context. After that, we developed a control method based on visual servoing which was able to take these suggestions into account in the control law, when performing a local navigation task.

However, the applications of these works were limited, because the context was defined in a static way and could not be updated during the driving. This static definition was sufficient for some elements that are not regularly updated, like the passenger's state. But it appears as a strong limitation when we want to consider contextual elements that may change in real-time, like the presence of an obstacle. The purpose of this paper is to provide an solution to this problem, by proposing a wider context model, and a framework able to update dynamically the knowledge base obtained from the context of navigation ontology and to take it into account in real-time in the control law. It appears as the natural extension of the capabilities of context aware navigation, towards its consideration and its application in autonomous vehicles.

The paper will proceed as follows. We are first going through the state-of-the-art work on context aware navigation in Sec. II. We then propose a new ontology to model the context of navigation that considers the global information in Sec. III. We then remind the handling of local context in Sec. IV. After this, we focus on the applicative aspects, by explaining the dynamic consideration of the context in Sec. V. Finally, Sec. VI is dedicated to the tests and results obtained in simulation and is followed by a conclusion in Sec. VII.

II. RELATED WORK

Some ontology-based solutions have been proposed in the past as the premises of context aware navigation solutions. Many of them were focused on *Advanced Driving Assistance Systems* (ADAS) solutions, i.e. electronic systems built to help the driver with parking and driving functions. Zhao et al. proposed an ontology-based system to prevent the driver from exceeding the speed limit [3] or to help him in intersections [4]. Their model was based on three different ontologies taking into account information on different levels: the Map Ontology, the Control Ontology and the Car Ontology.

More recently, Armand et al. in [5] modelled the inference rules for an ADAS system to allow the use of reasoners to reason over the ontology representing the contextual information, in the case of ADAS. Thus, the reasoner was able to propose

This research was funded in part by the Labex MS2T ("Laboratory of Excellence" on the Control of Technological Systems-of-Systems) a joint research entity between the Université de Technologie de Compiègne (UTC), the French National Centre for Scientific Research (CNRS) and Sorbonne Universités. Research was carried out in Heudiasyc Laboratory, UMR CNRS 7253, in Compiègne, France.

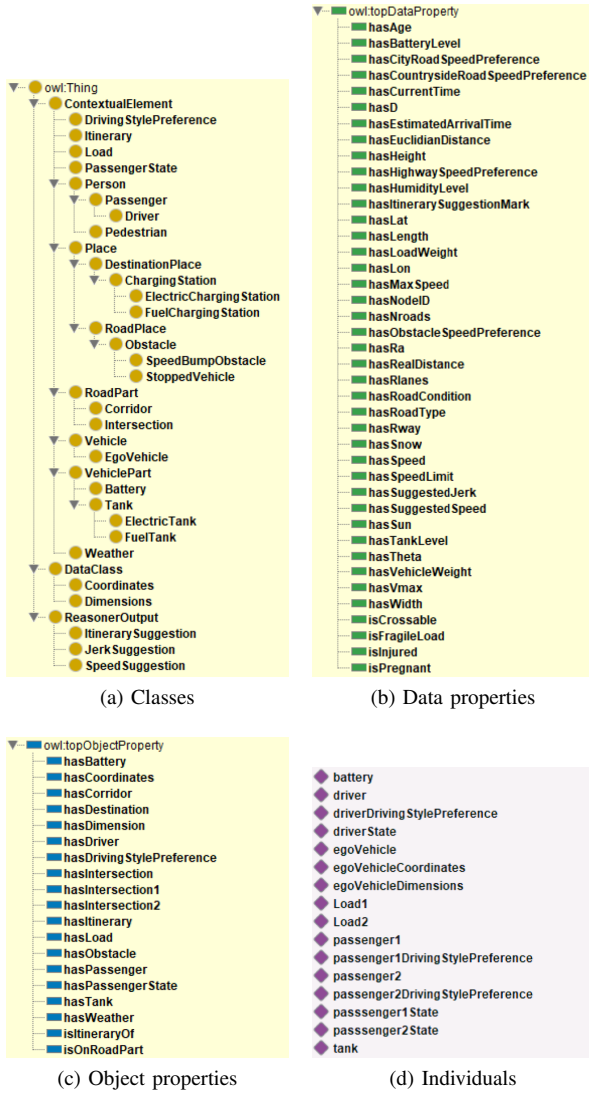


Fig. 1. The OWL ontology of the navigation context shown in Protege. The OWL file created in Protege handles the classes (a), data properties (b) and object properties (c). Individuals are created by the `OntologyHandler` module when initializing or updating the ontology to constitute the knowledge base. Individuals shown in (d) are the one created in the initialization.

suggestions for the driving assistance system on the decisions it should take.

From these previous works, Faruffini et al. proposed a solution to take into account these reasoner suggestions directly in the control law [1]. The approach is a modification of an image-based (or visual servoing) controller that modulates its output to consider more or less the contextual suggestion. This solution has been demonstrated on simulated vehicles, and has proved to be an efficient solution to consider the context in a local navigation law. This approach is reminded in Sec. IV.

From this state of the art, we will propose a generic framework for context-aware autonomous driving, able to infer in either global and local navigation law, and a related ontology for the context model. In this paper, the developed framework will focus on local navigation by taking into account the dynamic evolution of the navigation context.

III. CONTEXT MODELING

In this section, we detail the ontology-based model developed for the context of navigation, which uses elements of the Navigation Context Ontology (NCO) created in [2].

A. Ontologies

Before going through the model, we need to remind some definitions and properties of ontologies and reasoners, that are the tools used to model the context of the vehicle autonomous navigation.

An ontology can be defined as a semantic structure to store real world data with a formal representation. This kind of representation has been originally developed for the semantic web, which aimed to make web pages machine-readable. More recently, the use of ontologies for robotic applications have been considered as stated in the previous section. Given this historical consideration, we model ontologies using the tools originally developed for the semantic web. These tools are the *Ontology Web Language* (OWL) that is the language describing the ontology, the *Semantic Web Rule Language* (SWRL) that describes the rules applied to our ontology, and the Protege software [6] as a front-end ontology editor.

To model a system using ontologies, we need to define classes corresponding to the semantic concepts, and properties that are the links between the classes (object properties) or between classes and numerical data (data properties). These abstract definitions are together forming the *Terminological Box* (TBox). From these definitions, we can define individuals that are equivalent to the class instances. These individuals constitute what we call the *Assertional Box* (ABox). Together, TBox and ABox form a *Knowledge Base* (KB).

The main benefit of using an ontology structure is the possibility to exploit reasoners. Reasoners are software that can operate logical reasoning on the KB to make deductions that provide new information on it. That is the reason why an ontology-based system is adapted to our problem and the ones mentioned in the previous section. More generally, ontology-based systems are useful tools for proposing recommendations or suggestions to their users. That is the way ontology has been used in previously mentioned works and is being used in this work.

B. Context model

The Context of Navigation has two components: the Static Context and the Dynamic Context. The ontology we modeled in [2] was focused on the modeling of the Static Context, i.e. elements that are not updated in real-time. These were mainly information on the passengers' states (to adapt the driving to a pregnant or injured passenger for example) or on the vehicle state (its loads for example). Here, we consider again these static elements, and we add information that vary during the trip with respect to the navigation: this is the Dynamic Context, which models for example the vehicle speed or the presence of an obstacle in front of the vehicle. These elements require to be able to update the context dynamically (when the vehicle enters a new type of road), at varying moments during the

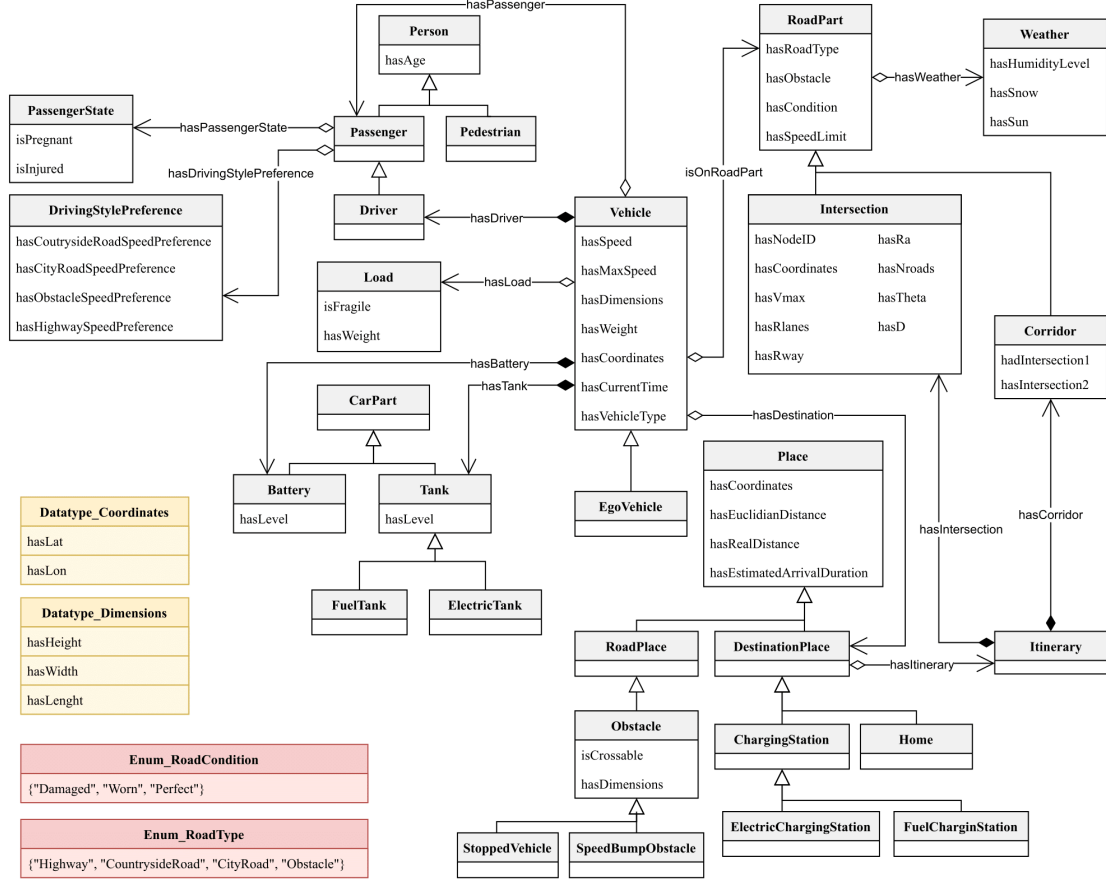


Fig. 2. The context of navigation implemented in the OWL ontology, presented under the form of a UML class diagram. This representation is useful to understand in a single diagram the links between the different classes and their properties. In the bottom-left corner are displayed the enumerations and the classes that represent only a data type.

TABLE I
THE DESCRIPTION OF CRITICAL POINTS (CP) ATTRIBUTES. EACH CP REPRESENTS A ROAD INTERSECTION AND IS PART OF AN ITINERARY.

Attribute	Meaning
CP DESCRIPTION	
NodeID	The ID of the CP, numbered from the beginning to the end of the itinerary
Lat	The latitude of the CP
Lon	The longitude of the CP
V_{max}	The maximum speed limit of the road
R_{lanes}	The number of available lanes in the road
R_{way}	The road classification (one way road or not)
R_a	The CP classification (roundabout or not)
N_{roads}	The number of roads at the CP
CP GEOMETRY	
Θ_1 to $\Theta_{N_{roads}}$	The angles for each road at the CP, ordered counter-clockwise
D_1 to $D_{N_{roads}}$	The directions for each road at the CP, ordered counter-clockwise

travel. These dynamic elements are used on our test scenario (see Sec. VI).

In addition, we extended the definition of our ontology by including classes that model global navigation elements that can be used by a reasoner to provide suggestions for the

global navigation. The global navigation elements modeled in our ontology are places and itineraries. A place is simply modeled by its coordinates, from which we can compute the distance to other places or its presence in an itinerary. For itineraries, we use the same model as [7]. In this model, an itinerary is represented as a suite of intersections (called Critical Points) and corridors. The description of intersection attributes is detailed in Tab. I.

We transcribed the information we want to model in OWL using the Protege software. The overview of the ontology classes, properties and individuals is given in Fig. 1. The same implementation is also described in Fig. 2 in the form of a UML class diagram, to understand better the links between classes and properties. This model presents several local and global elements that will be updated at a certain frequency (e.g. the road state) or when a specific event occurs (e.g. the detection of an obstacle). The *OntologyHandler* module presented in Sec. V is in charge of separating these elements.

In addition to the ontological model, we created a set of SWRL rules which allow the reasoner to suggest a maximum vehicle speed, that is taken into account by our controller. An example rule is given in Sec. IV.

IV. LOCAL CONTEXT AWARENESS

In this section, we remind briefly how we embedded context awareness in a local driving problem.

A. Problem formulation

We want to control our vehicle to make it drive autonomously in the middle of its lane, while avoiding possible obstacles. The control of the vehicle is given as $u = [v \ \omega]^T$ where v represents the longitudinal speed of the car and ω the angular velocity. We use the kinematic model of a front-wheel car with Ackerman's approximation for the steering angle ϕ , which is written as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \cdot \cos(\phi) \\ \sin(\theta) \cdot \cos(\phi) \\ \sin(\theta)/l \\ 0 \end{bmatrix} v + \begin{bmatrix} x_0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (1)$$

where x and y represent the vehicle's coordinates in a 2D plane, while θ and ϕ represent its orientation and yaw rate.

In a previous work, D. Fox et al. developed an approach called *Dynamic Window Approach* (DWA) [8]. It consists in finding the couple (v, ω) that maximizes the following objective function:

$$\text{DWA}(v, \omega) = \alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega) \quad (2)$$

where α , β and γ are parameters to be tuned. The heading function is based on the final orientation of the robot to lead it to the final goal position. The dist function prioritizes movement in areas free of obstacles, while the velocity function prioritizes longitudinal speeds close to the desired constant set point.

This method was then adapted in D. Lima [9] to be able to compute the heading function based on visual information. That is the *Image-based Dynamic Window Approach* (IDWA). This autonomous navigation controller is adapted in this paper, in order to take into account the context of the navigation. This is done by adding in this controller framework the information provided by a reasoner, as described in the next section.

B. Reasoner output

As said in Sec. III, one of the main advantages of the ontologies is the possibility to use rules that allow reasoning on the KB. Thus, rules can be used to provide suggestions

given a KB. In our case, the reasoner results in a suggested speed, that is used by the controller as explained after.

We define a set of rules to provide speed suggestion using the SWRL language. For example, the following rule aims to provide suggested speed w.r.t. the passenger preferences:

```
EgoVehicle(?v) ∧
SpeedSuggestion(?sg) ∧
hasPassenger(?v, ?p) ∧
hasDrivingStylePreference(?p, ?dsp) ∧
hasRoadType(?rp, "CityRoad") ∧
hasCityRoadSpeedPreference(?sp) ∧
isOnRoadPart(?v, ?rp)
→ hasSuggestedSpeed(?sg, ?sp)
```

The suggested speed is then used by the ICDWA robotic control loop, as described in the next section.

C. Image and Context-based Dynamic Window Approach

To take the suggested speed into account in the control law, F. Faruffini et al. defined a new approach called *Image and Context-based Dynamic Window Approach* (ICDWA) [1].

To allow the ICDWA to take the suggested speed v_{CN} into account, we have modified the IDWA by adding a fourth function to the maximization problem. It results:

$$\text{ICDWA}(v, \omega) = \alpha \cdot \text{heading}(v, \omega) + \beta \cdot \text{dist}(v, \omega) + \gamma \cdot \text{velocity}(v, \omega) + \delta \cdot \text{context}(v, \omega) \quad (3)$$

The parameter γ represents at which point the suggested speed is considered by the controller: for $\gamma = 0$ the suggestion is not considered at all, for $\gamma = 1$ it is the only speed to be considered.

The context(v, ω) function is decoupled as:

$$\text{context}(v, \omega) = \varepsilon \cdot (\text{context}_v(v) + \text{context}_\omega(\omega)) \quad (4)$$

where $\varepsilon \in [0, 1]$ and $\text{context}_v(v)$ is expressed as follow (the principle is the same for $\text{context}_\omega(\omega)$):

$$\text{context}_v(v) = \begin{cases} \sigma(x) & \text{if } x \neq 0 \\ \frac{1}{2} + \frac{1}{4}x & \text{otherwise} \end{cases} \quad (5)$$

where $\sigma(x)$ represents the sigmoid function and x the difference between the current speed and the speed suggested by the reasoner: $x = v - v_{CN}$.

V. DYNAMIC CONTEXT UPDATE

This section focuses on the dynamic update of the context of navigation and our software implementation of this update.

To benefit from the advantages of reasoning, the context used by the reasoner should be dynamically updated. The principle of context update in the case of the ICDWA is shown in Fig. 3. We are now able to update the individuals in the ontology used by the reasoner.

To be able to perform such an update, we developed a Python module named *OntologyHandler*. It uses the

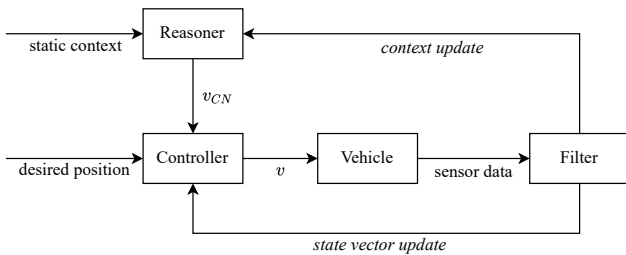


Fig. 3. ICDWA controller architecture that is able to update dynamically the context of navigation, using sensor data from the vehicle.

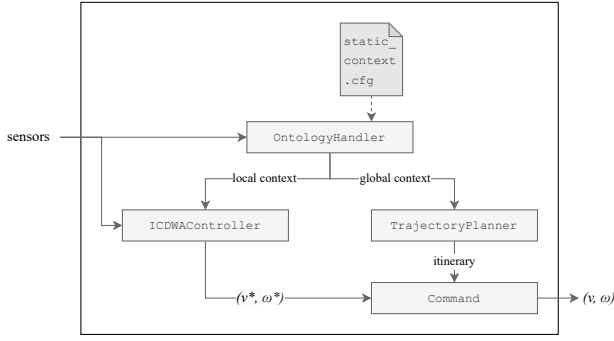


Fig. 4. Details of the software architecture developed for the context-aware navigation framework. The framework is capable of running separated instances of the OWL ontology, and is able to update in real-time the ontology individuals, including those who represent the reasoner output. The different submodules displayed in this figure run as separated threads.

owlready2 API by J-B. Lamy [10]. It has the capabilities of initializing a KB from a configuration file, updating the KB in real-time, and get the KB at any time, including the reasoner output. The advantage of owlready2 compared to the methods tested in [2] is that it accesses directly the whole OWL file. For the reasoning part, we use a modified version of the Pellet reasoner [11] used included in owlready2, as it is the only known reasoner able to infer new data from SWRL rules.

The `OntologyHandler` module is included into a wider framework for context aware navigation. The framework is based on ROS [12] and implements local and global context-aware navigation capabilities. The details of the framework architecture are presented Fig. 4.

If the framework is able to handle both local and global navigation, the requirements for these navigation approaches are not the same when it comes to the consideration of contextual information. For global navigation the context must be updated regularly, but updates can be easily separated by several seconds. However, the handling of local navigation tasks requires to update the context with a high frequency, as it has a direct impact on the vehicle's speed. Then, real-time considerations must be tackled.

In the framework architecture presented in Fig. 4, the different modules — including the reasoning module `OntologyHandler` — are executed in separated threads. This feature allows us to choose the execution frequency for the reasoner, in function of the real-time requirements.

VI. TESTS AND RESULTS

This section aims to show the tests done on our developments about local dynamic context awareness. These tests have been realized under the professional driving simulator SCANeR Studio, ver. 2022.1, which provides a ROS bridge that can be used with our architecture, like we tested on a real vehicle.

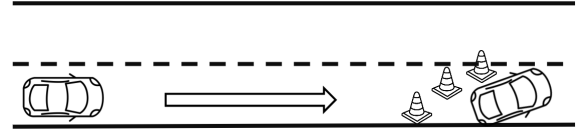


Fig. 5. The principle of our test scenario: the vehicle should avoid the obstacle on its lane.

A. Test scenario

For the tests, we define a scenario that requires dynamic updates, in order to test the principle of dynamic context awareness. In this scenario, we have an intelligent car, driving in a city (thus the speed limitation is 50km/h), on a straight road, with an obstacle at 150m inf front of it in its lane that should be avoided. The principle of the test scenario is shown in Fig. 5.

The vehicle is simulated in SCANeR Studio and communicates via ROS to our contextual navigation framework. To be able to avoid effectively the obstacle, the vehicle should detect the lanes. For this, we interface the deep-learning-based lane detector by A. Shour et al. [13] to our simulation environment.

Concerning the context, the KB is initialized from a configuration file, which states there is a single passenger, with the following speed preferences:

- 110 km/h on highways
- 70 km/h on countryside roads
- 40 km/h on cities
- 5 km/h when crossing an obstacle

As the scenario alternates between city road and the crossing of an obstacle, it is sufficient to test the principle of our context update.

B. Dynamic context update

We want to test if the software architecture detailed in the previous section allow to update the context efficiently. The software has been implemented in Python for ROS and shows to be functional. As stated, we modulated the execution frequency of the `OntologyHandler` module. For the tests, we used a frequency of 20Hz, which appears to be sufficient for our scenario.

C. Local context awareness

The ICDWA method was already tested in [1]. Here, we test it with our new scenario and software architecture.

When driving in the city and avoiding the obstacle, the vehicle's speed is adapted from the passenger's preferences. It results to the velocity profiles of Fig. 6. We used varying values for the coefficient γ of the ICDWA function. As the other parameters α , β and δ , the parameter γ is defined statically in the controller.

The profiles of Fig. 6 show that the velocity is reduced by taking into account the passenger preference. Besides, we can see the impact of the coefficient γ on the context awareness. When $\gamma = 0.5$, the suggestion is taken partially, especially when it comes to avoiding the obstacle: the speed is reduced to

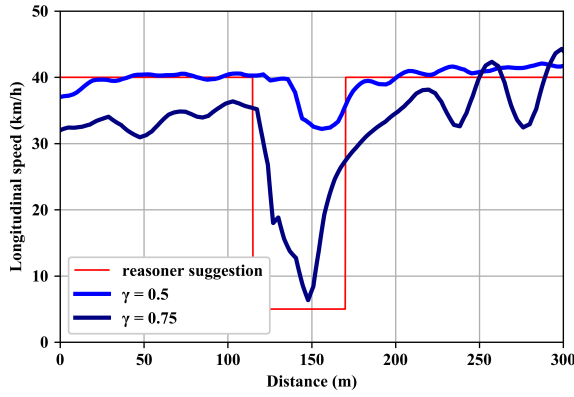


Fig. 6. Velocity profile obtained for the vehicle with our test scenario, with varying coefficient γ . The velocity is reduced by taking into account the passenger preference.

30km/h only, and the situation becomes dangerous. When we increase the coefficient to have $\gamma = 0.75$, the velocity profile matches much more the speed suggested by the reasoner. However, the absolute value of the acceleration is higher for this profile, which is something we want to avoid too. Thus, a compromise has to be found for the choice of the γ coefficient. This study is being performed in our going works.

D. Discussion

From the results presented in Fig. 6, we show that the speed is reduced and does not exceed the driver's speed preferences. The γ parameter seems to be well-tuned between 0.3 and 0.6. The optimal tuning of this parameter could lead to further works on the context consideration.

Due to hardware limitations and a lack of optimization in the bridge between the simulator and the framework, the simulation was performed at a very slow rate (between 4Hz and 10Hz) that caused a few instabilities for the ICDWA controller and high instabilities for the lane detector. To face this limitations, the speeds (both passenger's preferred speed and speed limitation of the road) have been reduced by 5 for the simulation. This issue should be targeted for future implementations on real vehicles.

Our test shows how contextual information can be updated in the ontology of the context of navigation from the vehicle's sensors, integrating dynamic information to the KB, and then inferred back on the vehicle control law through the ICDWA controller. The example displayed in this paper are simple but functional, and it is possible to extend them towards a much more complex and complete context aware navigation. The software framework and the ontology have been designed to be generic enough to handle new kind of information, including information on global navigation, that could be included and tested in the future.

VII. CONCLUSION AND PERSPECTIVES

In this paper, we presented a method to model contextual information and add it dynamically to the navigation law of an autonomous vehicle, in order to provide a smarter

vehicle behavior on the road. We also defined a software framework capable of handling the context by interacting with the ontology of the context of navigation in real-time during the navigation of the vehicle. This framework was tested on simulation with a new scenario, and proved that the context can be effectively updated to infer dynamically to the vehicle behavior.

The promising results obtained proved that this approach could improve the autonomous vehicle future developments. In further works, a much wider ontology of the context of navigation could be developed, including information on the global navigation, in addition to new methods to handle this kind of information. Besides, the ICDWA approach used in this work could be improved too, by choosing wisely at which point the contextual suggestions should be considered, with respect to the safety norms and considerations and to the navigation integrity. Finally, the choice of an adequate moment for updating each element of the context could be defined to avoid useless executions of the reasoner. These points are essential for future research and need to be targeted in the future developments, towards the consideration of the semantic modeling of the context of navigation.

REFERENCES

- [1] F. Faruffini, A. Corrêa Victorino, and M.-H. Abel, "Vehicle Autonomous Navigation with Context Awareness," in *2nd IEEE International Conference on Human-Machine Systems (ICHMS 2021)*. Magdeburg, Germany: IEEE, Sep. 2021, pp. 1–4.
- [2] F. Faruffini, H. Pousseur, A. Corrêa Victorino, and M.-H. Abel, "Context Modelling applied to the Intelligent Vehicle Navigation," in *47th Annual Conference of the IEEE Industrial Electronics Society (IECON 2021)*. Toronto, Canada: IEEE, Oct. 2021, pp. 1–6.
- [3] L. Zhao, R. Ichise, S. Mita, and Y. Sasaki, "An ontology-based intelligent speed adaptation system for autonomous cars," in *The 4th Joint International Semantic Technology Conference*, 11 2014.
- [4] L. Zhao, R. Ichise, T. Yoshikawa, T. Naito, T. Kakinami, and Y. Sasaki, "Ontology-based decision making on uncontrolled intersections and narrow roads," in *2015 IEEE Intelligent Vehicles Symposium*, 06 2015.
- [5] A. Armand, D. Filliat, and J. Ibanez-Guzman, "Ontology-based context awareness for driving assistance systems," in *IEEE Intelligent Vehicles Symposium, Proceedings*, 06 2014, pp. 227–233.
- [6] M. A. Musen, "The protégé project: a look back and a look forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015.
- [7] I.-F. Kenmogne, D. Alves De Lima, and A. Corrêa Victorino, "A Global Navigation Management Architecture Applied to Autonomous Robots in Urban Environments," in *18th IEEE International Conference on Intelligent Transportation Systems (ITSC 2015)*, Las Palmas de Gran Canaria, Spain, Sep. 2015, pp. 2109–2114.
- [8] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.
- [9] D. Lima and A. Victorino, "A hybrid controller for vision-based navigation of autonomous vehicles in urban environments," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 03 2016.
- [10] J.-B. Lamy, "Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies," in *Artificial Intelligence In Medicine*, vol. 80, 2017, pp. 11–18.
- [11] 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.
- [12] Stanford Artificial Intelligence Laboratory et al., "Robotic operating system."
- [13] A. Shour, H. Pousseur, A. Corrêa Victorino, and V. Cherfaoui, "Shared Decision-Making Forward an Autonomous Navigation for Intelligent Vehicles," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2021)*. Melbourne, Australia: IEEE, Oct. 2021, pp. 1634–1640.