



HAL
open science

Chapter 22. Implementing Context Awareness in Autonomous Vehicles

Federico Faruffini, Alessandro Corrêa Victorino, Marie-Hélène Abel

► **To cite this version:**

Federico Faruffini, Alessandro Corrêa Victorino, Marie-Hélène Abel. Chapter 22. Implementing Context Awareness in Autonomous Vehicles. Handbook of Human-Machine Systems, 1, Wiley, pp.257-265, 2023, 978-1-119-86363-2. 10.1002/9781119863663.ch22 . hal-04390046

HAL Id: hal-04390046

<https://hal.science/hal-04390046v1>

Submitted on 12 Jan 2024

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.

Chapter 22

Implementing context awareness in autonomous vehicles

Federico Faruffini,^{1,2*} Alessandro Correa-Victorino,¹ and Marie-Hélène Abel¹

¹*Heudiasyc Laboratory, University of Technology of Compiègne, 60200, Compiègne, 57 Av. de Landshut, France*

²*DIBRIS, University of Genoa, 16145, Genoa, Via All'Opera Pia, 13, Italy*

*Corresponding Author: Federico Faruffini; federico.faruffini@libero.it

Abstract: One of the problems still present in autonomous driving is the understanding of the surrounding context in which the navigation occurs. Besides recognizing the road components, potential obstacles and vehicles, autonomous cars don't consider the whole information which are intrinsic to the situation, which are instead considered by a human driver.

In this chapter we show how an ontology-based architecture for the Context of Navigation is sufficient to encode such information and to enable reasoning over it. We also show how the control loop of the autonomous vehicle can be modified to adapt to the situation.

Keywords: Context Awareness, Autonomous Vehicles, Ontology

22.1. Introduction

Many autonomous driving models exist nowadays, and some of them are fully capable of performing safe global navigation avoiding obstacles on their path. However, a problem which is still present to our day is the big difference between a person and an autonomous car in the way they drive, as this relies on many factors. One of the most important is the implicit information about the situation that a human driver naturally considers while driving, while autonomous vehicles don't. The totality of this implicit pieces of information specific to the driving scenario is the contextual information. In this chapter we are going to illustrate how the Context of Navigation can be used to reduce the gap between a robotic driver and a human one, by fully adapting its driving style and choices to the situation. In particular, we will show an approach we developed, based on modelling the context and using it in the control loop of the autonomous car to modify its standard behaviour.

Before to proceed, let's make a few examples to give the reader an idea of the scenarios which are possible to encode into the Context of Navigation. Our first example regards an autonomous car transporting a pregnant woman. In such situation, for example, a human driver would probably take smoother turns, accelerate slowly, and when possible take safer roads. In this scenario it is important to know the state of the pregnancy and how many

weeks the pregnancy has lasted so far, as we know there are specific weeks in which it is more dangerous to a pregnant woman to be on a car or drive it. These and many other information can be stored in the Context of Navigation, to let the car understand them.

Another example is the one of objects which are being carried in our car: a human driver would care about their size, positioning in the vehicle, properties (is it at risk of melting, or fragile?) and would decide on how to adapt his driving style to it. For instance, in the case of a heavy package on the upper luggage rack, the driver would avoid steering too fast as it could cause instability problems.

The structure of this chapter is the following: at first, we are going to discuss some background information on ontologies and autonomous driving in Sec. 22.2, then in Sec. 22.2.1 we will give a formal definition of the Context of Navigation. Later, in Sec. 22.3 we will present the state of the art studies on related fields. Then, we will proceed with Sec. 22.4, in which we show our method to build the Context of Navigation, reason over it and use it to suggest to the control loop how to operate. We will also see some tests at the end of the section. Finally, in Sec. 22.5 we will draft our conclusions and in Sec. 22.6 we will discuss the future research challenges for this study field.

22.2. Background

In this section we are going to recall some topics which will be useful to the reader to understand our implementation strategies.

Ontologies. The first point we'd like to discuss is that of ontologies. According to [Studer et al., 1998], an ontology is “a formal, explicit specification of a shared conceptualization”. It is a tool to represent data in a domain of discourse in such a way that it is both understandable by humans and computers. Ontologies, differently from standard relational databases, have a really interesting property to our scope, as they allow for reasoning on their data to infer new knowledge over the existing one. Another difference from databases is the way data are stored in ontologies: instead of tuples, ontologies use triples. The structure of a triple is *Subject-Predicate-Object*, for instance

Mark isOwnerOf RedCar
Mark hasAge 43

The information stored in an ontology make up the so-called knowledge graph. The standard language for ontology definition is OWL¹, the Web Ontology Language, and it allows for complex triple structures. It defines Classes for the instances in our triple store and properties for them. In the first example, *Mark* is an instance of the *Person* class, while *RedCar* could be an instance of a *Car* class. This latter class could be in turn a child class to a more generic *Vehicle* class. The link between *Mark* and *RedCar* is *isOwnerOf*, which is an object property (class to class). In the second example *hasAge* is a data property, as *43* is just a value and not an object. With restrictions over the classes and properties we

¹<https://www.w3.org/OWL/>

shape our ontology and a first simple reasoning. One of the ways it is possible to obtain a more complex reasoning over an ontology is the Semantic Web Rule Language (SWRL) by [Horrocks et al., 2004]. SWRL lets the programmer define inference rules as follows. One or more boolean variables are considered as antecedent, one as a consequent: if all the ones in the antecedent are true, then the consequent is true in turn. An example of SWRL rule is the following:

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

This rule lets the car infer that it must take the preferred driving style of the one passenger on board. SWRL rule consequents can be used as antecedents in other rules, so that a multi layered structure is created for a more deep reasoning in our ontology.

Autonomous driving. Starting in 2004, the Defense Advanced Research Projects Agency (DARPA) of the USA held three major competitions for autonomous cars [Thrun et al., 2006, Thrun, 2006]. Since then, this field has seen a lot of interest by the scientific community and many autonomous vehicle models currently exist. Much of them are based on sensors (internal - endoceptive, external - exteroceptive) like GPS systems in order to navigate. Some of such models suffer from sensor problems, as temporary disconnection or sensor failures. Another kind of autonomous vehicle model is based on Visual Servoing (VS), meaning the car drives itself by using a frontal camera. This approach was proven to be more stable than standard sensor-based ones. Since we based our solution on an existing one which relies on Visual Servoing, we are now going to illustrate how it operates. Such approach is called the Image-Based Dynamic Window Approach (IDWA) [Lima and Victorino, 2016] and it combines VS to drive the car with the Dynamic Window Approach to perform obstacle avoidance.

In Fig. 22.1 we show how the way the car is driven in the center of the lane by IDWA. The model computes the limits of the lane (the yellow lines) in order to compute the center of the lane (the red line). Then, the position of the car is computed and the control needed to get to the center (if there is some position error) is computed.

When an obstacle is perceived, the control is given to the DWA module, which computes the best couple of tangential and rotational velocities (v, w) to overcome it safely and comfortably. This is achieved by finding the best couple which optimized the following cost function:

$$IDWA(v, w) = \alpha \cdot heading(v, w) + \beta \cdot dist(v, w) + \gamma \cdot velocity(v) \quad (22.1)$$

where α , β and γ are real-valued parameters to be properly tuned. The *heading* function aims at keeping the car in the center of the lane, the *dist* function is used to avoid obstacles and the *velocity* function's objective is to keep the car's speed close to a desired setpoint.

22.2.1. Basic definitions

A formal definition of the Context of Navigation was given in [Faruffini et al., 2021]:

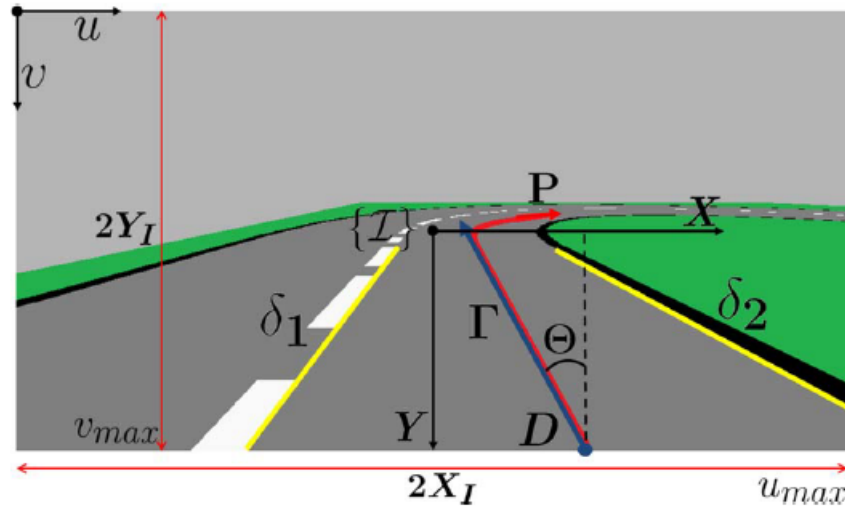


Figure 22.1: IDWA visual servoing in [Lima and Victorino, 2016]

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.

The Context of Navigation has 2 main components: the Static Context and the Dynamic Context. The Static Context comprehends all those information which don't change with respect to the current navigation. Instances can be the preferences of the passengers, their health situation, the type of carried load and its property etc. The Dynamic Context contains the information which may vary during the navigation, as the position of obstacles, other vehicles, state of traffic lights, temperature of the engine, fuel in the tank. Given its modular nature, the Context of Navigation can be easily implemented for different applications through the use of an ontology.

22.3. Related works

As we decided to use ontologies to our scopes, we will now see some of the most interesting studies on ontologies applied to the vehicle navigation.

Regele in [Regele, 2008] proposed an ontology-based structure to model the topology of the road, in order to let the autonomous car understand it. Each road segment is labelled to allow the car to understand which of the lanes are available for it to navigate and which cannot be used. Also, with the same labels the car understands the concept of conflicting/merging lanes, and is enabled to use traffic rules in them in a safe way. One of the key points of this approach is the low computational power required in real time by the car, as it can only be given information on the lanes around itself and not the full struc-

Table 22.1: Related works comparison table.

| Reference | Description |
|-------------------------|--|
| [Regele, 2008] | Ontology-based structure to model the topology of the road and lanes. |
| [Xiong et al., 2016] | Modelling the space surrounding the ego vehicle and label the portions to overtaking manoeuvres. |
| [ZhaoLihua, 2015] | Ontology-based model for a safer ride in uncontrolled intersections by preventing excessive speed. |
| [Janowicz et al., 2018] | Modelling and detecting failures in perception. |
| [Armand et al., 2014] | Ontology for reasoning as a human driver. |

ture, to reason over in real time. Xiong et al. in [Xiong et al., 2016] proposed to divide the space surrounding the ego vehicle and label the portions, for instance *Follower*, *Follower's Follower*, *Leader*, *NearSide Follower*. This way it is easier to model reasoning, for instance when the car has to deal with overtaking a slower vehicle.

Many ontology-based solutions can be found in ADAS (Advanced Driving Assistance Systems), and some of them consider some external information to aid the driver. In the case of Zhao et al. [ZhaoLihua, 2015, Zhao et al., 2015] the authors studied a model for a safer ride by helping the driver in uncontrolled intersections and by preventing him to exceed the speed limit. Since the car's perception is sensor-based, it is important to model and detect failures in them. The SOSA (Sensor, Observation, Sample and Actuator) ontology proposed by Krzysztof et al. [Janowicz et al., 2018] can be used to deal with this problem.

A very interesting study is the one by Armand et al. [Armand et al., 2014]. The authors proposed an ontology to let the car reason in a way which is extremely similar to the one of a human driver. This is important as this way the rules are also easier to be implemented, tested and debugged. These works are resumed in a comparative way in Table 22.1.

22.4. Implementation and Tests

In this section we are going to illustrate more in detail how our structure works. At first, we are going to provide an example of Context of Navigation modelling using an ontology and SWRL. Later in this section we are going to see our solution to implement a context-aware control loop rule called Image-and-Context-based Dynamic Window Approach (ICDWA).

22.4.1. Implementing the Context of Navigation

In Sec. 22.2 we gave an overview of the semantic tools we can use to model the Context of Navigation, which are ontologies and the Semantic Web Rule Language. We will now see how simple it is to use them to create a modular structure to store information over the

Context of Navigation and reason over it. From the definition of the Context of Navigation ([Faruffini et al., 2021]) we can understand how many information can be considered part of it: for the sake of brevity, we are just going to show a small example of it. Also, the complexity of debugging ontologies is known in literature ([Lambrix, 2020]) and a modular approach is often seen as a good solution to avoid needing big debugging operations.

The main purpose of our ontology structure is to reason and give a real time suggestion of maximum speed v and yaw rate w depending on the situation. The control loop rule will take these suggestions in its optimization function to find the best (v, w) for the car.

Let's take the following scenario as our example. We have an autonomous car with a sick passenger called Mark, who is going to the hospital. Fortunately, the situation is not an emergence, so the car must stick to the traffic rules as usual. As the passenger is sick and nauseated, we want the car to adapt its driving behaviour to this situation, to avoid worsening the man's condition. To complicate this scenario, let's assume we have different paths to the destination, with different features as speed bumps, heavy traffic and so on.

How can we model the Context in this case? Of course, we are going to need the triples to store data about our passenger and his condition:

Mark isOwnerOf RedCar
RedCar a EgoVehicle
Mark hasAge 43
Mark hasPassengerState Sick
Mark hasPassengerState Nauseated
RedCar hasPassenger Mark
RedCar hasCurrentFuel 10.4

Also, we can have some rules to support the reasoning. We can show now just a couple of them to understand how they are written:

$$EgoVehicle(?v) \wedge hasPassenger(?v, ?p) \wedge Person(?p) \wedge hasPassengerState(?p, Sick) \rightarrow hasSickPassenger(?v, True)$$

$$EgoVehicle(?v) \wedge hasSickPassenger(?v, ?p) \rightarrow hasDrivingStyle(?v, Sick_Passenger_DrivingStyle)$$

$$EgoVehicle(?v) \wedge hasDrivingStyle(?v, Sick_Passenger_DrivingStyle) \rightarrow hasSuggestedMaxSpeed(?v, 40)$$

So, with these 3 rules we created a reasoning stack that brings the suggested speed of the car to a lower value, which is $40km/h$ not to worsen the passenger's nausea. As we can see, with just a simple structure like this one we get the required suggestion to be sent to the control loop. A similar but slightly more complex structure can be created for the w suggestion.

22.4.2. Control Loop Rule

In this subsection we will present our solution for a control loop rule able to adapt to the situation. It is based on the IDWA ([Lima and Victorino, 2016]) algorithm we presented

in Sec. 22.2, and implements context awareness. Its full name is ICDWA - the Image and Context based Dynamic Window Approach. So, how does the interaction between Context and Control Loop work? As we said in the previous section, our ontology in real time gets us a suggestion of maximum values of speed v and yaw rate w . We can add a component to the cost function in (22.1) a new component called *context* to take the ontology suggestions into account, and we obtain:

$$IDWA(v, w) = \alpha \cdot heading(v, w) + \beta \cdot dist(v, w) + \gamma \cdot velocity(v) + \delta \cdot context(v, w) \quad (22.2)$$

which is the objective function for ICDWA. Let's now discuss the *context* component. First of all, $\delta \in \mathbb{R}$ is a parameter to be tuned with the others in order to assign weight to this component. Secondly, we can say that since v and w are not fully related we can split the component into 2 subcomponents and then add those to get its value:

$$context(v, w) = \epsilon \cdot (context_v(v) + context_w(w)) \quad (22.3)$$

Here, $\epsilon \in [0, 1]$ is a parameter which we will discuss later on. Let's now consider $context_v$: what is the behaviour we want from it? Given a suggestion of maximum speed v_{MAX} , we want it to generate a high cost if $v \geq v_{MAX}$, a lower one otherwise. Since we will find the best values (v, w) from (22.2) through gradient descent, we need it to be continuous, derivable and convex. If we define

$$x = v - v_{CN} \quad (22.4)$$

we can then use the following implementation for $context_v$:

$$context_v(v) = \begin{cases} \sigma(x), & \text{if } x \leq 0 \\ \frac{1}{2} + \frac{1}{4}x, & \text{otherwise} \end{cases} \quad (22.5)$$

where the σ function is the sigmoid function. Our $context_v$ component as defined in (22.5) has a smoothed ramp-like behaviour, which increases as long as we don't follow the ontology's suggestions.

We previously hinted at the parameter ϵ in (22.3). Why to use another gain for the component, and not just one containing both δ and ϵ in it? We decided to do so for an easier model: let's see why we need ϵ and why we need these parameters to be separated. Let us now assume to have a very old passenger in our car, who prefers it to be driven at a very low speed to feel safer on board. Now, is it always correct to follow this desire? Probably, if we have a low-traffic situation there won't be any issues in following it. However, if we have a high-traffic scenario, if we decide to slow down the car we could worsen the traffic situation, and that is not acceptable. By real time reasoning we can compute the optimal value of ϵ so that we can follow the suggestions only when it is possible ($\epsilon = 1$) and ignore them otherwise ($\epsilon = 0$). Now, if we consider also the full range $[0, 1]$, we can have a mixed behaviour, in which we try to adapt to the desires of the passenger while keeping in account the traffic situation. The value of ϵ is computed in real time by a specific part

of the ontology.

To finish, why do we need to separate δ and ϵ ? The reason lies in their nature: δ is tuned in laboratory, and then kept constant throughout the navigation, while ϵ can change at any instant.

22.4.3. Simulations

The Image-and-Context-based Dynamic Window Approach was eventually tested in a professional simulator for the automotive industry and academics, called SCANeR Studio². Java is the native language of most of the software implementations of ontologies, however SCANeR doesn't have a Java interface, so we decided to test our solution with Python code. The Python3 library we used is called owlready2, developed in [Lamy, 2017].

We will now present the results we obtained with a particular test. The scenario we chose is the one we can see in Fig. 22.2, in which the car starts from a speed of 0km/h and needs to proceed in a road with 3 obstacles to be overcome, staying in the center of the lane when possible. The maximum speed for the car was set to 10km/h for this test.

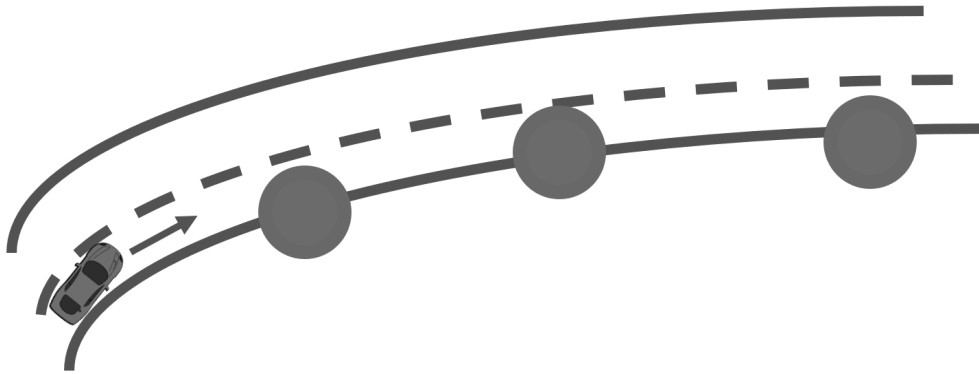


Figure 22.2: The simulation of ICDWA in SCANeR Studio

In Fig. 22.3 we can see the behaviour of the longitudinal speed for an autonomous car with and without context awareness, and see the differences in its behaviour. The solid line, with $\epsilon = 0$, is the one in which we don't consider the contextual suggestions at all, resulting in a pure IDWA control. We can see clearly the pattern that the speed of the car follows for the 3 obstacles. The dark dashed line represents the situation in which we assign the weights $\delta = 0.75$ and $\epsilon = 1$ to the context component. In this case, the speed limit is set to the same as before, but as we can see the car keeps a little below it, as we instructed it through the cost function in (22.5). Finally, the lighter line represents the same situation with a lower value of suggested speed v_{CN} , which was set to 8km/h . We can see how the car keeps a speed which is even lower than the previous test, and respects

²<https://www.avsimulation.com/scaner-studio/>

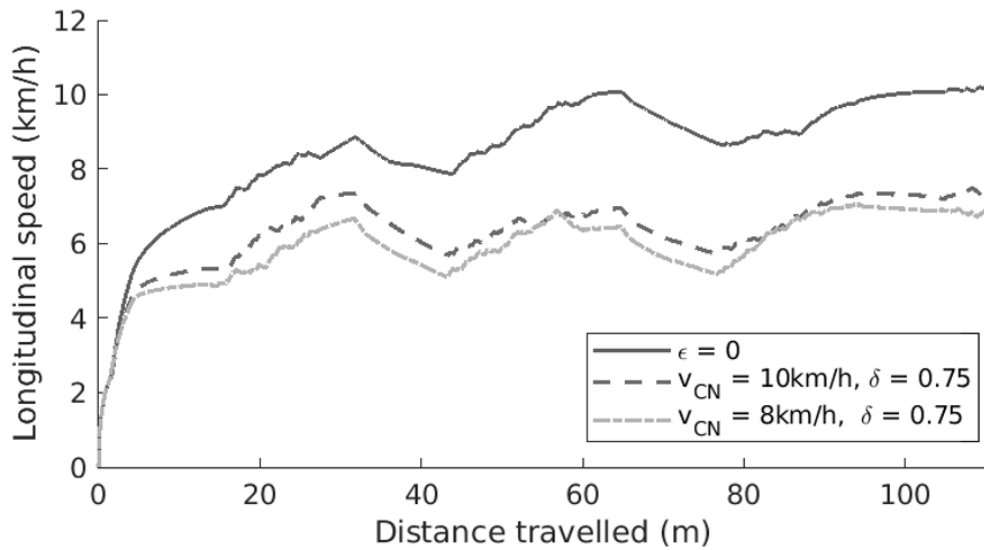


Figure 22.3: The speed profile of the autonomous car with ICDWA

again the suggestions. This was a simple simulation, but it showed promising results.

We performed other tests, varying the values of the parameters and the max speed. All of them were successful (the vehicle did not crash into an obstacle or go off the road), and led to different car behaviours. It was clear the problem now is to find a way to set the 4 parameters in the cost function of ICDWA to optimize the performance. Also, the ϵ parameter was for the moment chosen before runtime, but it must be computed in realtime to be effective.

22.5. Conclusions

In this chapter we presented a new context-aware control loop called ICDWA - the Image-and-Context-based Dynamic Window Approach. We proposed a way to structure the Context of Navigation by giving it a formal definition and an ontology-based structure. We then proceeded to study the way to build reasoning rules over it, and we used the Semantic Web Rule Language for it. We shaped our ontology to be able to output a suggested max speed value to be taken into account by the controller. We then proposed our control law - the ICDWA - which is able to adapt the car's behaviour to the suggestions of the ontology. We showed how we can decide when to ignore such suggestions and how to do that. We finally performed tests on a professional simulator for automotive which led to really promising results.

Compared to other autonomous navigation algorithms as IDWA, our model is able to get a better adaptation to the situation, as it can rely on the context modelling and suggestions coming from the ontology in real time.

22.6. Future research challenges

Even if the proposed structure was proven to have a good performance, many more challenges are still open. In future works the automated tuning of $\alpha, \beta, \gamma, \delta$ will be studied, as well as the computation for the ϵ parameter. Also, the control loop rule is to be expanded with the real time suggestion of the maximum yaw rate ω , and with the jerk computation for v too.

On the other hand, the Context of Navigation we proposed in our examples is small if compared to what it could be: future works will focus on expanding it with many more information. Some of the latter are the integration of a trajectory planner, traffic rules, obstacle trajectory predictor and many more. Also, a way to input data into the ontology is needed, and it must not take too long for the users, so this will be another interesting part of future studies. Since many data about the potential passengers will be collected and stored, it is important to consider security aspects as well as data ethics in early stages of the project.

22.7. References

- Armand, A., Filliat, D. and Ibañez-Guzman, J. [2014], Ontology-based context awareness for driving assistance systems, in '2014 IEEE Intelligent Vehicles Symposium Proceedings', pp. 227–233.
- Faruffini, F., Correa-Victorino, A. and Abel, M.-H. [2021], Towards a semantic model of the context of navigation, in I. Saad, C. Rosenthal-Sabroux, F. Gargouri and P.-E. Arduin, eds, 'Information and Knowledge Systems. Digital Technologies, Artificial Intelligence and Decision Making', Springer International Publishing, Cham, pp. 168–183.
- Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B. and Dean, M. [2004], 'SWRL: A semantic web rule language combining OWL and RuleML', W3C Member Submission. Available at <http://www.w3.org/Submission/SWRL/>.
URL: <http://www.w3.org/Submission/SWRL/>
- Janowicz, K., Haller, A., Cox, S. J. D., Phuoc, D. L. and Lefrançois, M. [2018], 'SOSA: A lightweight ontology for sensors, observations, samples, and actuators', *CoRR* **abs/1805.09979**.
URL: <http://arxiv.org/abs/1805.09979>
- Lambrix, P. [2020], 'Completing and debugging ontologies: state of the art and challenges'.
- Lamy, J.-B. [2017], 'Owlready: Ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies', *Artificial Intelligence in Medicine* **80**.
- Lima, D. and Victorino, A. [2016], 'A hybrid controller for vision-based navigation of autonomous vehicles in urban environments', *IEEE Transactions on Intelligent Transportation Systems* pp. 1–14.
- Regele, R. [2008], Using ontology-based traffic models for more efficient decision making of autonomous vehicles, in 'Fourth International Conference on Autonomic and Autonomous Systems, ICAS 2008, 16-21 March 2008, Gosier, Guadeloupe', IEEE Computer Society, pp. 94–99.
URL: <https://doi.org/10.1109/ICAS.2008.10>
- Studer, R., Benjamins, V. and Fensel, D. [1998], 'Knowledge engineering: Principles and methods', *Data & Knowledge Engineering* **25**(1), 161–197.
URL: <https://www.sciencedirect.com/science/article/pii/S0169023X97000566>
- Thrun, S. [2006], Winning the darpa grand challenge, in J. Fürnkranz, T. Scheffer and M. Spiliopoulou, eds, 'Machine Learning: ECML 2006', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 4–4.
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P.,

- Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C. and Mahoney, P. [2006], ‘Stanley: The robot that won the darpa grand challenge.’, *J. Field Robotics* **23**, 661–692.
- Xiong, Z., Dixit, V. V. and Waller, S. T. [2016], The development of an ontology for driving context modelling and reasoning, in ‘2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)’, pp. 13–18.
- Zhao, L., Ichise, R., Mita, S. and Sasaki, Y. [2015], Core ontologies for safe autonomous driving., in S. Villata, J. Z. Pan and M. Dragoni, eds, ‘International Semantic Web Conference (Posters & Demos)’, Vol. 1486 of *CEUR Workshop Proceedings*, CEUR-WS.org.
- ZhaoLihua, L. [2015], Ontology-based decision making on uncontrolled intersections and narrow roads, in ‘2015 IEEE Intelligent Vehicles Symposium, IV 2015, Seoul, South Korea, June 28 - July 1, 2015’, IEEE, pp. 83–88.