Honda Research Institute EU
**Honda Research Institute Europe GmbH**
https://www.honda-ri.de/

# A Speech-Based On-Demand Intersection Assistant Prototype

**Dennis Orth, Bram Bolder, Nico Andreas Steinhardt, Mark Dunn, Dorothea Kolossa, Martin Ernst Heckmann**

**2018**

**Preprint:**

This is an accepted article published in 2018 IEEE Intelligent Vehicle Symposium. The final authenticated version is available online at: https://doi.org/[DOI not available]

PREPRINT. © 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

# A Speech-Based On-Demand Intersection Assistant Prototype

Dennis Orth[1,2], Bram Bolder[2], Nico Steinhardt[2], Mark Dunn[2], Dorothea Kolossa[1], Martin Heckmann[2]

***Abstract*— We have recently proposed a speech-based on-demand intersection assistant which helps the driver to handle urban intersections by informing him of the traffic situation on the right hand side and recommending suitable gaps in traffic. In a previous user study, conducted in a simulator, we could show that the system is in general well accepted and preferred by drivers compared to driving without assistance or with only visual support. In this paper, we report on an implementation of this system and its evaluation in real urban traffic. We use LIDAR sensors for the perception of the traffic environment. A scene analyzer estimates the gaps between the vehicles in real time. The result of this analysis is provided to a dialog manager, which uses it to inform the driver of approaching vehicles and suitable gaps. While approaching the intersection, the driver can activate the system via a wake-up-word and control it with subsequent speech commands. The design of the data analyzer and dialog manager is based on evaluations at real intersections. The resulting system can provide suitable support to the driver in a wide range of traffic situations.**

## I. INTRODUCTION

Due to the dynamic and possibly confusing crossing vehicle paths, many accidents take place at intersections [1], [2]. In a response to this, existing rear-end collision avoidance systems are being extended to cover more potentially dangerous situations at intersection. One of these is the left-turn with oncoming traffic. For example Volvo [3] and AUDI [4] provide assistants, which help to avoid a collision in such a scenario. Furthermore, the systems are being extended to also cover multi-directional collisions. One example is the BAS PLUS system from Mercedes. It includes a cross-traffic assist, which gives audio-visual warnings if a collision is likely to occur with traffic approaching from the right or left and increases the brake pressure up to a full emergency break once the driver starts breaking [5]. But a limitation of these systems is their relatively small field of view. Due to this and to avoid false alarms, these system can only intervene very late. Such systems are hence only suited for emergency situations. Currently two approaches are mainly pursued to extend the operational range of them. One direction is driver intent estimation and maneuver prediction [6], [7], [8], [9]. This offers the possibility to react more in advance while keeping false alarms low. Another direction is to extend the field of view of these systems beyond the limitations of on-board sensors. Here new cooperation approaches which utilize vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communication [10], [11] are investigated and their application is evaluated with new kinds of assistance systems [12]. E.g., BMW recently proposed a motorcycle prototype, which has implemented V2V technology to assist in crossing and left-turning [13].

With our previously proposed speech-based intersection assistant [14] in the context of our *Assistance On-Demand* (AOD) concept, we follow a different approach. This system is designed to support the driver in the maneuver decisions and is, in contrast to the previously mentioned approaches, activated and controlled on-demand by using speech commands from the driver. The intention is to reduce the number of false alarms by this means, since those could distract and annoy the driver [15] and could even lead to a deactivation of the system. Our current application scenario is the left turn at urban intersections. The system will provide support in situations where the driver is approaching the intersection from a minor road and has no right of way. After the driver has activated the system via speech, the system will inform about the current traffic situation on the right and recommend suitable gaps. The latter will not include direct action recommendations, because we do not want to encourage the driver to perform the left turn without carefully checking the traffic on the right. In [14] we could already show in a simulator study that the system is well accepted by the participants and preferred compared to driving without assistance or with a visual support. To further improve the utility of the system, we have investigated individual drivers' gap acceptance and developed methods to efficiently estimate personalized gap recommendations [16], [17]. We could then show that these personalized recommendations clearly improved the acceptance of the system compared to identical recommendations for all drivers. They also enhanced the monitoring of the traffic situation and further decreased the perceived workload [18].

In this paper we will go one step further, out of the simulator and into real traffic. We have implemented the intersection assistant in a prototype vehicle and evaluated it in urban traffic. The complexity of real traffic is difficult to recreate in a simulator. Our previous investigations were based on rather simple traffic scenarios containing vehicles driving at constant speed. Additionally, realistic sensor limitations could not be considered fully in the simulator. Extending the system to cope with the dynamic urban traffic in particular required the development of sophisticated dialog management strategies.

In the next section, we will start by describing the general structure of the system and give more conceptual details on the components in the subsequent sections. Following this,

[1]Institute of Communication Acoustics, Ruhr-Universität Bochum, Bochum, Germany
{dennis.orth, dorothea.kolossa}@rub.de
[2]Honda Research Institute Europe GmbH, Offenbach am Main, Germany
{bram.bolder, nico.steinhardt, mark.dunn, martin.heckmann}@honda-ri.de

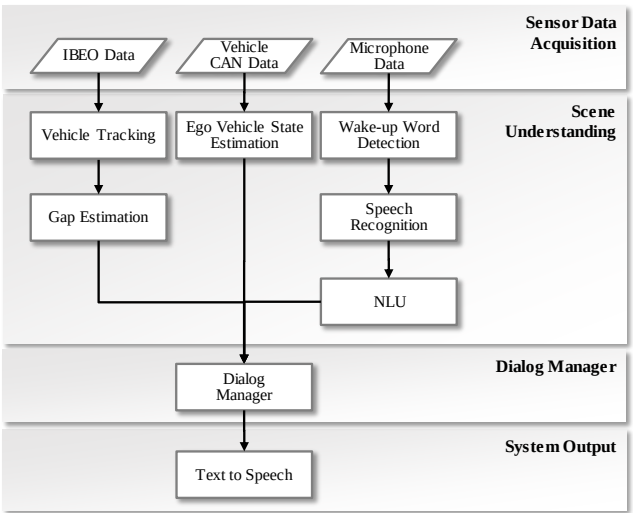



Fig. 1. Overview of the main components of the AOD system.

we will present the implementation details, conclude what has been achieved and give an outlook on what has to be done to obtain an intersection assistant, which is able to cover most of the possible traffic situations with high accuracy.

## II. System Overview

In this section we give an overview of the main building blocks of the AOD system: sensor data acquisition, scene understanding, dialog manager and system output (compare Fig. 1). We will present the core elements, scene understanding and dialog manager, and their sub-components in more detail in Sections III and IV.

### A. Sensor data acquisition

We use LIDAR sensors to estimate position and velocity of all traffic participants in the vicinity of the ego vehicle. The vehicle CAN bus provides access to the ego vehicle's current velocity, acceleration and the state of the accelerator as well as the brake pedal. We acquire the speech commands from the driver via a Bluetooth headset.

### B. Scene understanding

The LIDAR sensor already provides tracks of the different traffic participants. However, the vehicles passing from the left side frequently occlude the vehicles approaching from the right side. To mitigate this problem, we implemented an additional vehicle tracking which is able to cope with such occlusions. Based on these corrected vehicle tracks (see Sec. III-B), we estimate the gaps between vehicles from the right side.

The AOD system should only give feedback on the current traffic situation to the driver while the ego vehicle is standing at an intersection. Therefore, we implemented methods to detect the arrival and departure of the ego vehicle from the intersection (see Sec. III-A).

To facilitate the interaction with the AOD system, the automatic speech recognition (ASR) continuously listens for the wake-up word. We have chosen 'Cora' as wake-up word as an acronym for Cooperative Assistant. Once the wake-up word is detected, the recognition of the driver's utterance is triggered. A natural language understanding (NLU) component analyzes the utterance and forwards the result to the dialog manager.

### C. Dialog Manager

The dialog manager is activated when the NLU detected the intent of the driver to make the system watch right. Then it continuously receives the data from the gap estimation and ego vehicle state estimation module, analyzes it and informs the driver accordingly.

### D. System Output

The system interacts with the environment only via speech. For this it uses a text-to-speech (TTS) module.

## III. Scene Understanding

The target of the scene understanding part of the AOD system is to analyze the traffic scene surrounding the ego vehicle and interpret the drivers' utterances. Additionally, it also evaluates the state of the ego-vehicle.

### A. Ego Vehicle State Estimation

The system will only start analyzing the traffic scene after the ego vehicle has arrived at an intersection. Similarly, the system should stop as soon as the driver starts leaving the intersection. We do not use map information to detect the arrival at or departure from the intersection but rather assume that after the driver requested the system the ego vehicle will be standing at the intersection once it stopped moving and it will leave the intersection once it starts moving again.

The wheel-speed sensors of the test vehicle cannot measure velocities below $2\,\mathrm{km\,h^{-1}}$; any velocity below this threshold leads to an output value of $0\,\mathrm{km\,h^{-1}}$. However, while approaching the stop line of an intersection, the vehicle may be within this velocity range for several seconds. This makes a standstill prediction, to avoid premature activation of the assistance function, necessary.

If the measured velocity drops to zero, the standstill prediction is triggered, where $n_0$ denotes the index of the last non-zero velocity measurement. For the prediction we assume that the deceleration is constant from $2\,\mathrm{km\,h^{-1}}$ to $0\,\mathrm{km\,h^{-1}}$. We filter the raw wheel-speed sensor measurements $v(n)$ with a mean filter of length $K$ to obtain a smoothed signal $\bar{v}(n)$. In a buffer of length $M$ we store the most recent values of this signal. This buffer is only updated with nonzero velocity measurements. If the zero velocity trigger occurs, the buffer is held constant and the acceleration is computed from its contents according to:

$$a_{\mathsf{avg}} = \frac{\bar{v}(n_0) - \bar{v}(n_0 - (M-1))}{M\frac{1}{f_{\mathsf{CAN}}}}\,, \tag{1}$$

where $f_{\mathsf{CAN}}$ is the sampling rate of the vehicle CAN bus. Afterwards thresholds for the minimal and maximal averaged acceleration, $a_{\min}$ and $a_{\max}$, are applied to ensure a stable

operation even in case of large measurement outliers inside the measurement buffer:

$$a'_{\mathsf{avg}} = \begin{cases} a_{\mathsf{max}} & \text{if } a_{\mathsf{avg}} > a_{\mathsf{max}} \\ a_{\mathsf{min}} & \text{if } a_{\mathsf{avg}} < a_{\mathsf{min}} \\ a_{\mathsf{avg}} & \text{otherwise} \end{cases} \quad (2)$$

This acceleration is then used to ramp the predicted vehicle velocity down to zero with:

$$v_{\mathsf{extrapolate}}(n) = \bar{v}(n_0) + a'_{\mathsf{avg}} \cdot (n - n_0 + \frac{K-1}{2}) \frac{1}{f_{\mathsf{CAN}}} \quad (3)$$

where the term $\frac{K-1}{2}$ compensates the mean filter delay. As soon as the predicted velocity has reached zero, the standstill signal is set.

If one or more of the following criteria is fulfilled, the standstill signal to the assistance function is released again.

- $v(n) > 0\,\mathrm{km\,h^{-1}}$
- $v(n) = 0\,\mathrm{km\,h^{-1}}$ && accelerator pedal is being pushed
- $v(n) = 0\,\mathrm{km\,h^{-1}}$ && brake pedal is released for more than $t_{\mathsf{brake}}$ seconds. This is especially important for automatic transmission vehicles, which start moving already when only the brake is released

### B. Vehicle Tracking during Occlusions

A vehicle passing from the left side can occlude a vehicle arriving from the right side. As a consequence, the tracking algorithm of the LIDAR sensor can lose track of the vehicle from the right during the passage of the vehicle from the left side. It will then spawn a new track for a new vehicle to which it assigns a new vehicle ID. To mitigate this, we extrapolate the tracks of vehicles whose track was lost with constant speed and orientation by at most $t_{\mathsf{extrapolate}}$. During this time we check if the LIDAR sensor tracking algorithm detects a new vehicle which is compatible with our extrapolation. Compatible means that the difference in heading, speed and position of the two vehicles lies inside an error margin. If this is the case, the tracks of the new vehicle and the extrapolated vehicle are merged and both receive the ID of the extrapolated vehicle.

### C. Gap Estimation

The first step of the gap estimation is to identify the traffic participants relevant to the intended left-turn assistance. Relevant are only those vehicles which are approaching from the right side, since the driver is still monitoring the left side. To identify these vehicles reliably we apply two criteria. First, in addition to being on the right side of the ego vehicle[1] they have to have a heading angle in the range of $[180°, 0°]$. This angular range has to be wide enough to allow for slightly tilted positioning of the ego vehicle at the intersection, non-orthogonal T-intersections and moderate errors in the tracking of the surrounding vehicles. Second, they have to arrive towards the ego vehicle in the angular ranges $[0°, -130°]$ or $[0°, 130°]$. The latter filters vehicles which approach from behind (see Fig. 2). Furthermore, vehicles which are driving slower than $1\,\mathrm{m\,s^{-1}}$ are also rejected, in order to filter out unstable detections on parked vehicles and pedestrians. For the remaining vehicles their expected time of arrival at the reference point in front of the ego vehicle (Fig. 2) is calculated. This calculation is based on the current velocity and the distance of these vehicles. We observed that, while standing at the intersection, the LIDAR sensor only yielded reliable estimates for the y-coordinate. Therefore, we use a projection of the vehicle distance onto the y-axis to estimate the time of arrival. This analysis is performed for every time frame independently, i.e., no temporal integration is applied.

[1]More precisely, they have to be on the right side of the trigger point in Fig. 2. The reason will be explained in Sec. IV-A

## IV. Dialog Manager

One of the key features of the AOD system is the situated dialog, i.e., a dialog which is targeted at and embedded in the physical environment. Our scenario is characterized by a highly dynamic environment due to the relatively rapidly moving vehicles. This requires a predictive dialog planning where the future state of the environment has to be considered in relation to the time for speech synthesis and the average time a driver needs for listening to and understanding the message. This prediction is limited mainly by the system's perception range and the accumulating uncertainties in the future environment state with an increasing prediction horizon. We assume that the system can look approximately $D_{\mathsf{predict}}$ meters to the right due to sensor range limitations and that all vehicles are moving with constant speed. Assuming a maximum speed of the traffic participants $v_{\max}$ defined by the current speed limit, the minimal prediction time $T_{\mathsf{predict}}$ (Fig. 2) can be calculated as $T_{\mathsf{predict}} = D_{\mathsf{predict}}/v_{\max}$. In the following we will detail the dialog management approach we devised to tackle the aforementioned problems.

### A. General Dialog Logic

References of the system to traffic participants have to be unambiguous as they might otherwise confuse the driver. Due to sensory limitations only the traffic participants dynamics are known. With current sensors, additional features such as type (car, truck, pedestrian ...), size or color are not

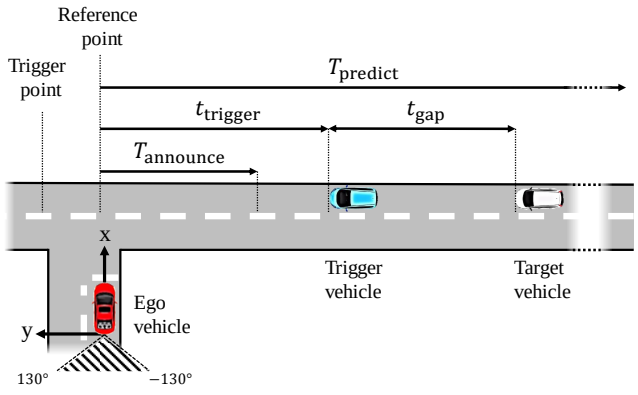



Fig. 2. Top view of the T-intersection, showing the ego vehicle (red) waiting for the vehicles from the right to pass the intersection.

available, or can only be obtained at too late a point in time. Hence the system refers to all traffic participants as 'vehicles'. To achieve the required unambiguous reference, the system always makes reference to the traffic participant approaching from the right with the shortest time of arrival. We will refer to this vehicle as the *trigger vehicle* (Fig. 2). In its approach several conditions are tested for the trigger vehicle, which can trigger an announcement of the system. We refer to the vehicle following the trigger vehicle as the *target vehicle* as the announcement is in most cases targeted at this vehicle. The reference point for the calculation of the time of arrival is in the center of the ego vehicle . We call this time $t_{\mathsf{trigger}}$ for the trigger vehicle. Consequently, we denote the time gap between the trigger and the target vehicle as $t_{\mathsf{gap}}$ and measure it from the front of the trigger vehicle to the front of the target vehicle. Due to aforementioned sensor limitations, we currently don't take the length of the trigger vehicle into account to determine the actual gap between the vehicles.

We will now illustrate the dialog logic in more detail based on a few possible situations. Assuming the trigger vehicle is still sufficiently far away to make an announcement ($t_{\mathsf{trigger}} > T_{\mathsf{announce}}$) and the gap behind this vehicle $t_{\mathsf{gap}}$ is large enough for the driver to make the turn, the system should announce 'gap after next vehicle'. In this utterance 'next vehicle' refers to the trigger vehicle which triggered the announcement and 'gap' to the target vehicle which is the vehicle which determines if the driver can make the turn. The driver is able to make the turn if $t_{\mathsf{gap}} \geq T_{\mathsf{gap\,crit}}$, where $T_{\mathsf{gap\,crit}}$ is the so called critical gap, a gap just large enough for the driver. As already mentioned above, the system needs time to output the utterance and the driver needs time to understand it, verify the traffic situation and take a decision. We accumulate this time in $T_{\mathsf{announce}}$. Hence, the system should start with outputting 'gap after next vehicle' at the latest when $t_{\mathsf{trigger}} = T_{\mathsf{announce}}$. Due to the limited and situation dependent perception horizon $T_{\mathsf{predict}}$, it is advisable to make the announcement of the fitting gap as late as possible, i.e., when $t_{\mathsf{trigger}} = T_{\mathsf{announce}}$. This maximizes the likelihood that all vehicles relevant for the planned announcement are already detected. In case $t_{\mathsf{gap}} < T_{\mathsf{gap\,crit}}$ the system will announce 'vehicle from the right'. To avoid confusions, this announcement is only made after the trigger vehicle has passed the intersection. This condition is fulfilled when the front of the vehicle passes the trigger point, a point which is $l_{\mathsf{vehicle\,default}}$ to the left of the reference point. The value $l_{\mathsf{vehicle\,default}}$ represents a default vehicle length, which we use as the real vehicle length is difficult to obtain reliably from the LIDAR. The next announcement will only be possible after the trigger vehicle has passed the trigger point. After this it is removed from the environment representation and the vehicle with the then shortest time of arrival is chosen as the new trigger vehicle.

The situations which we encountered in real traffic varied widely and required a differentiation into substantially more possible cases. To make these cases manageable and allow for further extensions we model them as a state machine.

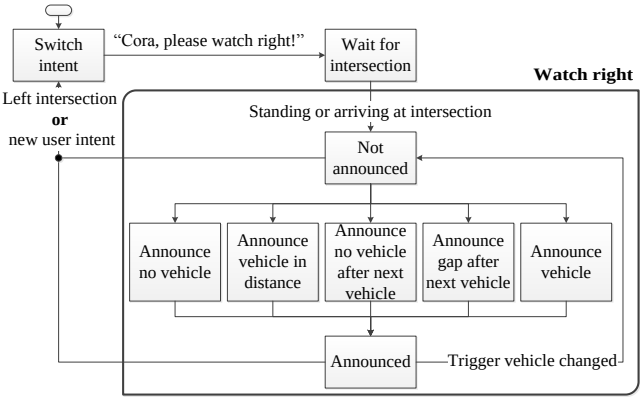



Fig. 3. Overview of the dialog manager state machine.

### B. Top-level State Machine

Fig. 3 shows the hierarchical state machine of the proposed AOD dialog manager. On the top level the state machine iterates through the following states:

- *Switch intent*: The system is waiting for an input from the driver. If the NLU recognizes that the driver's intent is to make the system watch the right-hand side, it will transition to *Wait for Intersection*. Currently there are no other intents implemented.
- *Wait for Intersection*: The system analyzes the state of the ego vehicle. If the arrival at the intersection has been detected, the system outputs 'Okay. I am watching' followed by a transition to *Watch Right*. If the vehicle is still in motion, the system outputs 'Okay. I will watch', waits until the arrival at the intersection is detected and then also transits to *Watch Right*.
- *Watch Right*: In this state the traffic environment is analyzed and feedback is given to the driver. This is implemented in a subordinate state machine, which is explained in more detail in the next section.

### C. Watch Right State Machine

The Watch Right State Machine consists of three different types of states:

- *Not Announced*: The system reaches this state if no announcement has yet been made for the current trigger vehicle. In this state the conditions for transition into one of the *Announcement* states are checked.
- *Announcement*: In this set of states, the system gives feedback to the driver on the current traffic situation. Once the system started to give the feedback for a specific trigger vehicle, it transits immediately to the *Announced* state.
- *Announced*: The system waits in this state until the current trigger vehicle crosses the trigger point. Then a switch to the new trigger vehicle takes place and the system transits back to *Not Announced*.

From the *Not Announced* state, a transition to one of the following announcement states takes place if the corresponding condition is met:

- *Announce No Vehicle*: If there is no vehicle approaching the intersection from the right the system outputs 'No vehicle from the right'. An internal timer assures that this announcement is performed at most every $T_{\mathsf{novehicle}}$ seconds.
- *Announce Vehicle in the Distance*: If there was previously no trigger vehicle and $t_{\mathsf{trigger}} \geq T_{\mathsf{announce}} + T_{\mathsf{gap\,crit}}$ the system outputs 'Vehicle in the distance'. The latter ensures that the distance to the approaching vehicle is still large enough for the driver after the announcement has finished.
- *Announce No Vehicle after Next*: If there is no target vehicle and $t_{\mathsf{trigger}} \leq T_{\mathsf{announce}}$ the system outputs 'No vehicle after the next vehicle'.
- *Announce Gap*: If $t_{\mathsf{gap}} \geq T_{\mathsf{gap\,crit}}$ and $t_{\mathsf{trigger}} \leq T_{\mathsf{announce}}$ the system outputs 'Gap after next vehicle'.
- *Announce Vehicle*: If $t_{\mathsf{gap}} < T_{\mathsf{gap\,crit}}$ the system will wait until the trigger vehicle reaches the trigger point and then outputs 'Vehicle from the right'. This state also has an internal timer ($T_{\mathsf{vehicle}}$) which prevents that in dense traffic situations this message is repeated too frequently.

The system also leaves the state *Announced* if there is no trigger vehicle present at all or if we need to make an additional announcement for the same trigger vehicle. The latter occurs if, e.g., the system has announced 'Vehicle in the distance' and now needs to make again announcements when the vehicle approaches.

In addition to the logic outlined so far we implemented a few shortcuts and additional states. These mainly are needed to handle the arrival at the intersection and dense traffic. However, due to space constraints they are not further detailed here.

The *Not Announced* and the *Announced* states regularly check if a new driver utterance has been received or the ego vehicle has left the intersection. In these cases the system interprets this as a signal to abort the current dialog and will transit back to the *Switch Intent* state.

## V. System Implementation

In the following, we describe the prototype vehicle, the system software architecture and the parameter settings of the different components.

### A. System Hardware and Software

A modified 2012 model-year Honda CR-V was used as the prototype vehicle. In addition to the standard equipment, it features 6 Ibeo Automotive Systems LUX LIDAR sensors, which have an overall coverage of $360°$. The LIDAR raw data is processed in a dedicated Ibeo sensor fusion and object detection & tracking unit. Furthermore, the trunk contains computing hardware to store and process the sensor data. For speech acquisition, we use a Plantronics Voyager 5200 UC Bluetooth headset and for speech feedback the standard audio equipment of the vehicle.

The software infrastructure integrates four different subsystems. As middleware for acquiring the LIDAR and CAN data stream, we use RTMAPS by Intempora. From there the data is sent via nanomsg to the vehicle tracking module and ego vehicle state estimation modules. These are implemented in RTBOS (Real-Time Brain Operating System), a middleware for distributed and multi-threaded systems developed at the Honda Research Institute Europe [19]. The results of these operations are again transmitted via nanosmg to a stand-alone Python program which comprises the gap estimation and dialog manager modules. For the implementation of the dialog manager state machine, we use the package SMACH [20] from the Robot Operating System (ROS). Speech recognition and synthesis are accomplished via the VoCon™ and Vocalizer™ software respectively, both from Nuance. Additional interfaces were written by Linguwerk to enable the communication with the dialog manager, also via nanomsg.

The vehicle CAN uses a sampling rate of $f_{\mathsf{CAN}} = 50\,\mathrm{Hz}$. At this sampling rate the ego vehicle state estimation is performed. The LIDAR has a sampling rate of $f_{\mathsf{LIDAR}} = 25\,\mathrm{Hz}$ which drives the vehicle tracking and gap estimation. Different threads pull the corresponding components for these data and then provide them to the dialog manager. The dialog manager itself runs at a sampling rate of $f_{\mathsf{dialog}} = 10\,\mathrm{Hz}$.

### B. Ego vehicle state estimation parameters

For the algorithm described in Sec. III-A a buffer size of $M = 50$ samples and a mean filter length $K = 11$ samples was used, which corresponds to $1.0\,\mathrm{s}$ and $0.22\,\mathrm{s}$ respectively. The acceleration was limited by $a_{\min} = 0.3\,\mathrm{m/s^2}$ and $a_{\max} = 8.0\,\mathrm{m/s^2}$. Furthermore, we set the brake release time $t_{\mathsf{brake}}$ to $0.75\,\mathrm{s}$.

### C. Dialog Manager Parameters

For the dialog manager, $T_{\mathsf{announce}} = 2.0\,\mathrm{s}$ is used as an estimate of the cumulative time needed for the system to output one utterance and for the driver to comprehend it. Given the approximate range of the LIDAR of $150\,\mathrm{m}$, the maximum time for which we can predict the traffic situation on the right side ($T_{\mathsf{predict}}$) is $11.0\,\mathrm{s}$ when vehicles are assumed to move at a maximum velocity of $50\,\mathrm{km\,h^{-1}}$, the speed limit in the area we tested the system. As mentioned previously, the actual detection distance unfortunately varies from vehicle to vehicle.

## VI. Current Limitations and Future Work

Relying only on LIDAR sensors entails several limitations in our perception capabilities. In some cases, a leading vehicle might occlude a following vehicle. This becomes more severe, when the road to the right is at an angle of more than $90°$ to our road or if it curves. In these cases, the following vehicle might only be detected very late, which could lead to wrong announcements of the dialog manager. This is particularly problematic as we do not yet check if an announcement made is still valid. As one example, once we announced a fitting gap we currently do not monitor if the situation changes. Therefore, in future

work, we will first implement a process which constantly verifies that the current traffic situation is still in line with the last announcement. Upon a deviation it should start a new announcement to avoid confusion or inadequate behavior of the driver.

So far, in our predictions we assume that the velocity of the traffic vehicles is constant. In general, we did not experience problems with this assumption. However, stopping and very slow vehicles cause problems as their predicted time of arrival is very long, up to infinity. This means that, e.g., situations in which vehicles want to turn into the road we are currently on or in which they stopped at a pedestrian crossing, waiting for pedestrians to cross, can currently not be handled automatically. Parked vehicles and pedestrians are also problematic as they are currently only identified based on their instantaneous velocity. Due to detection errors this velocity might be wrongly estimated and as a consequence they might disturb the dialog manager. To tackle these problems we plan to build a dynamic model of the traffic participants, which also includes their previous behavior and their acceleration.

As the size of the vehicles cannot be recognized while they are approaching, the vehicle length is not taken into account for the calculation of the gaps. Instead, a default length of $4\,\mathrm{m}$ is used. This is acceptable in most cases, but it also entails the risk that long trailer trucks can trigger erroneous announcements of fitting gaps. However, this problem can possibly only be solved by an improved sensor system or with V2I/V2V communication.

At the moment, the system does not communicate the presence of obstacles limiting its field of view to the driver, which also leads to confusing announcements. We will extend the situated dialog so that the driver can optimize the ego vehicle's position. Similarly, the system could also communicate its confidence on the estimated gaps, which may be a valuable information for the driver.

## VII. CONCLUSIONS

In this work we have implemented a prototype of our previously proposed speech-based on-demand intersection assistant in a vehicle to show the feasibility in real traffic and to use the acquired data for further optimization. The system can be activated by speech commands and is able to recognize when the driver stands still at the intersection. It then uses LIDAR sensors to track the vehicles approaching from the right, estimates the gaps between the vehicles and gives the driver recommendations on suitable gaps until it detects the driver starting the left turn.

Through ongoing evaluations of the system in real traffic, we have developed a situated dialog manager, which can support the driver in a wide range of traffic situations. The evaluations with the prototype have helped us to identify many relevant aspects of the possible traffic conditions and options for handling them.

Future work will be aimed at addressing the remaining issues outlined above, in order to arrive at a system that is fully adaptive to the environment, including all possible dynamic changes of the traffic conditions. In addition, we already started an in depth analysis of our system with regard to the prediction of the gaps and the positions of the traffic vehicles [21].

## ACKNOWLEDGMENT

We would like to thank Heiko Wersing for fruitful discussions and Martina Hasenjäger for her support during our recordings and various additional contributions.

## REFERENCES

[1] Directorate General for Transport, "Traffic safety basic facts 2016: Junctions," European Commission, Tech. Rep., Jun. 2016.

[2] W. G. Najm, B. Sen, J. D. Smith, and B. N. Campbell, "Analysis of Light Vehicle Crashes and Pre-Crash Scenarios Based on the 2000 General Estimates System," Tech. Rep., 2003.

[3] Volvo Cars. [Online]. Available: www.volvocars.com/us/about/our-innovations/intellisafe

[4] Audi AG. [Online]. Available: https://audi-dialoge.de/en/Kreuzungsassistent

[5] M. Mages, F. Klanner, and A. Stoff, *Intersection Assistance*. Cham: Springer International Publishing, 2016, pp. 1259–1286.

[6] M. Liebner, M. Baumann, F. Klanner, and C. Stiller, "Driver intent inference at urban intersections using the intelligent driver model," in *IEEE Intell. Vehicles Symposium*, 2012.

[7] D. J. Phillips, T. A. Wheeler, and M. J. Kochenderfer, "Generalizable intention prediction of human drivers at intersections," in *2017 IEEE Intell. Vehicles Symposium (IV)*, June 2017, pp. 1665–1670.

[8] A. Zyner, S. Worrall, J. Ward, and E. Nebot, "Long short term memory for driver intent prediction," in *2017 IEEE Intell. Vehicles Symposium (IV)*, June 2017, pp. 1484–1489.

[9] V. Losing, B. Hammer, and H. Wersing, "Personalized maneuver prediction at intersections," in *IEEE 20th Int. Conf. on Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, oct 2017.

[10] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Communications Surveys Tutorials*, no. 4, pp. 584–616, 2011.

[11] S. Kühlmorgen, A. Gonzlez, A. Festag, and G. Fettweis, "Improving communication-based intersection safety by cooperative relaying with joint decoding," in *IEEE Intell. Vehicles Symposium (IV)*, 2017.

[12] F. Naujoks and A. Neukum, "Timing of in-vehicle advisory warnings based on cooperative perception," in *Proc. of the human factors and ergonomics society Europe chapter annual meeting*, 2014, pp. 1–13.

[13] BMW Motorrad. [Online]. Available: www.press.bmwgroup.com/global/article/detail/T0274952EN/bmw-motorrad-presents-the-r-1200-rs-connectedride

[14] N. Schoemig, M. Heckmann, H. Wersing, C. Maag, and A. Neukum, "please watch right evaluation of a speech-based on-demand assistance system for urban intersections," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 54, pp. 196 – 210, 2018.

[15] L. Biester, "Cooperative automation in automobiles," Ph.D. dissertation, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät II, 2009.

[16] D. Orth, D. Kolossa, M. Sarria Paja, K. Schaller, A. Pech, and M. Heckmann, "A maximum likelihood method for driver-specific critical-gap estimation," in *2017 IEEE Intell. Vehicles Symposium (IV)*.

[17] D. Orth, D. Kolossa, and M. Heckmann, "Predicting driver Left-Turn behavior from few training samples using a Maximum-A-Posteriori method," in *IEEE 20th Int. Conf. on Intell. Transp. Syst. (ITSC)*, 2017.

[18] D. Orth, N. Schömig, C. Mark, M. Jagiellowicz-Kaufmann, D. Kolossa, and M. Heckmann, "Benefits of personalization in the context of a speech-based left-turn assistant," in *AutomotiveUI '17*.

[19] A. Ceravola, F. Joublin, M. Dunn, J. Eggert, M. Stein, and C. Goerick, "Integrated research and development environment for real-time distributed embodied intelligent systems," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Oct 2006, pp. 1631–1637.

[20] [Online]. Available: http://wiki.ros.org/smach/

[21] D. Orth, N. Steinhardt, B. Bolder, M. Dunn, D. Kolossa, and M. Heckmann, "Analysis of a speech-based intersection assistant in real urban traffic," in *submitted to: 2018 IEEE 21th Int. Conf. on Intell. Transp. Syst. (ITSC)*, Maui, Hawaii, USA, Nov.