HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Marie-Christin H. Oczko

How to Apply Haptic Signals on Bicycles for Safety

Bachelor Thesis in Computer Science

15. August 2019

Please cite as:
Marie-Christin H. Oczko, "How to Apply Haptic Signals on Bicycles for Safety," Bachelor Thesis (Bachelorarbeit), Heinz Nixdorf Institute, Paderborn University, Germany, August 2019.

CCS

Distributed Embedded Systems (CCS Labs)
Heinz Nixdorf Institute, Paderborn University, Germany

Fürstenallee 11 · 33102 Paderborn · Germany

http://www.ccs-labs.org/

How to Apply Haptic Signals on Bicycles for Safety

Bachelor Thesis in Computer Science

vorgelegt von

Marie-Christin H. Oczko

geb. am 28. Juni 1997
in Bielefeld

angefertigt in der Fachgruppe

Distributed Embedded Systems
(CCS Labs)

Heinz Nixdorf Institut
Universität Paderborn

Betreuer:	**Falko Dressler**
Gutachter:	**Falko Dressler**
	Ingrid Scharlau

Abgabe der Arbeit: **15. August 2019**

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde.
Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Declaration

I declare that the work is entirely my own and was produced with no assistance from third parties.
I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly.

(Marie-Christin H. Oczko)
Paderborn, 15. August 2019

Abstract

Safety for vulnerable road users, like cyclists, is still a major challenge. While the number and quality of assistance systems for cars is increasing day by day, the amount of research on bicycle safety is significantly lower. To improve cyclists' safety, I developed a haptic collision warning system for cyclists based on Inter-Vehicle Communications (IVC). Integrating two vibration motors on the bicycle's handles to convey directional information, the cyclist's attention is directed towards hazards approaching from the corresponding direction to avoid potentially fatal accidents. To gain first insight into the usefulness of the haptic signals, I carry out a psychological study in a simulation environment. I extend the Virtual Cycling Environment (VCE) to design a safe environment for a laboratory experiment without compromising the participant's safety. Instructive simulation scenarios are created and a supportive questionnaire is designed. The results show that the majority of the participants of the experiment react faster to possible dangers with supportive haptic signals. In real life, this might help to prevent traffic accidents involving cyclists.

Kurzfassung

Die Sicherheit von nicht-motorisierten Verkehrsteilnehmern ist ein akutes Problem. Während die Anzahl der verfügbaren Assistenzsysteme für Autos immer weiter zu nimmt und diese ständig verbessert werden, sind gleichartige Systeme für Radfahrer noch kaum erforscht. Um Fahrradfahrern eine sichere Teilnahme am Straßenverkehr zu ermöglichen, habe ich ein haptisches Kollisions-Warnungssystem entworfen, dass auf drahtloser Kommunikation zwischen Fahrzeugen basiert. Durch das Anbringen eines Vibrationsmotors auf jeder Seite des Lenkers, wird dem Radfahrer die Richtung mitgeteilt, aus der sich eine Gefahr nähert. Zusätzlich warnt das System vor allgemein gefährlichen Situationen oder einer Gefahr direkt vor dem Fahrer, indem beide Motoren gleichzeitig vibrieren. Um die Effektivität der Vibrationen und ihre Auswirkungen auf mögliche Nutzer zu evaluieren führe ich ein psychologisches Experiment mit 17 Teilnehmern durch. Durch Erweitern der bereits existierenden Virtual Cycling Environment (VCE) erhalte ich eine sichere Umgebung, die Versuche ohne Gefährdung potenzieller Teilnehmer ermöglicht. Zusätzlich entwickele ich passende Simulationsszenarien und einen begleitenden Fragebogen. Die Ergebnisse der Studie legen den Schluss nahe, dass das entwickelte System Nutzer dabei unterstützt, früher auf mögliche Gefahren zu reagieren. Im echten Leben könnte die Anwendung des haptischen Kollisions-Warnungssystems die Anzahl der Fahrradunfälle reduzieren.

Contents

Chapter 1

Introduction

The number of people riding bicycles instead of driving cars is increasing daily[1]. People see the need to be more environmentally friendly or use the opportunity of being faster and more flexible, especially in urban traffic. Additionally, they take advantage of reduced costs in comparison to traveling by car. Sadly, safety for vulnerable road users is still a problem. In 2017, 3,180 people were killed in traffic accidents in Germany. 12% were cyclists and their proportion is increasing significantly every year[2]. To support safer driving behavior on the roads and to reverse the trend of increasing cyclists' death, Advanced Driver Assistance Systems (ADASs) have been integrated into modern cars.

Most approaches to prevent accidents focus on cars while the amount of research on bicycle safety is significantly lower [1]. For cars, many different kinds of signals have been developed and tested with respect to their usefulness and influence on the driver [2], [3]. However, "[...] it is proved that more than half of the accidents that involve bicyclists are caused by the cyclist itself" (cf. [4], p.491). Therefore, ADAS for cyclists are clearly as important as ADAS for cars. Such systems should help cyclists to recognize potentially dangerous situations and obstacles in the surrounding environment. Especially, elderly people and children may have limited attention and are unable to recognize potentially dangerous situations. This group has a higher probability to be involved in crashes [5] (also cf. footnote[3]). Moreover, there are a lot of situations where every cyclist is simply distracted and needs support.

[1] Bundesministerium für Verkehr und digitale Infrastruktur, "Fahrrad-Monitor Deutschland 2017 - Ergebnisse einer repräsentativen Online-Befragung", `https://www.bmvi.de/SharedDocs/DE/Anlage/G/fahrradmonitor-2017-ergebnisse.pdf?__blob=publicationFile`, accessed 2019-03-09

[2] DESTATIS - Statistisches Bundesamt, "Unfallentwicklung auf deutschen Straßen 2017", `https://www.destatis.de/DE/PresseService/Presse/Pressekonferenzen/2018/verkehrsunfaelle_2017/Pressebroschuere_unfallentwicklung.pdf?__blob=publicationFile`, accessed 2019-03-09

[3] European Comission, "Proactive Safety for Pedestrians and Cyclists: Accident Analysis, Naturalistc Observations and Project Implications",2016, `https://cordis.europa.eu/project/rcn/193275/factsheet/en`, accessed 2019-08-11

Three different kinds of signals seem to be promising for bicycles: visual signals (e.g., light or warning signs projected on streets or on glasses), auditory signals (e.g., warning sounds,) and haptic ones (vibration/touch). In this thesis, I focus on how to apply haptic signals on bicycles for safety. The designed haptic signals should support the cyclist's reaction to dangerous situations without too much cognitive effort. I decided to concentrate on haptic signals as they only minimally compromise the cyclist's visual and auditory perception and attention. A road user has to be focused as much as possible on the surrounding traffic and environment. First works on haptic signals on bicycles were already successfully tested and published [1], [6], [7]. For my work, I choose haptic signals based on vibrations to inform the cyclist about approaching dangers. In contrast to cars, possible implementations of signals on bicycles are much more spatially limited. I designed a bicycle-based system by sending vibrations to the handles of the bicycle. Especially the work of Matviienko et al. [7] supports my choice as I discuss in Section 2.5 on page 12.

A bicycle does not have the safety cell around the driver that a car has. Simply taking an automatic response on critical events is not as easily feasible on bicycles as it is on cars because supporting features might be missing. For example, in cars passive safety like seatbelts supports the safe use of electrically controllable emergency brakes, while for bicycles automatic brake usage could cause uncontrollable events and physical harm to the cyclist. Consequently, systems that can alert the cyclist to dangers are the most promising solution (cf. [8]). Based on typical accidents involving cyclists, I identified a suitable ADAS, a collision warning (and avoidance) system, and inserted haptic warnings. My resulting Haptic Collison Warning System (HCWS) is based on Inter-Vehicle Communication (IVC) and therefore has access to information on the surrounding car traffic. By that, all vehicles approaching in a given radius can be checked for possible threats. Opposite to sensors this enables the systems to detect danger earlier, e.g., by being able to detect possible threats around corners. Based on the position of the cars, driving direction and speed, it can predict possible accidents and helps to avoid them.

However, the effectiveness of the HCWS depends on whether there is an appropriate reaction of the cyclist to the given warnings. Consequently, my HCWS and its influence on the driving behavior of the cyclist has to be under examination. Therefore, reactions on incidents caused by haptic signals must be understood and evaluated. As mostly dangerous situations are to be examined, human-in-the-loop simulations are a promising approach. They are suitable laboratory experiments that do not put test subjects into real physical danger and with minor disturbing factors [9]. Additionally, by conducting laboratory experiments, the reproducibility of the experiments becomes more reliable. By Heinovski et al. [10] it is investigated that the Virtual Cycling Environment (VCE) as published by Buse, Sommer, and Dressler [11] is qualified for "conducting ADAS-testing experiments" (extracted from [8], p. 10). Hence, I use the VCE as a core of my Virtual Haptic Cycling Environment (VHCE) in which I implemented the HCWS. Thereby, the modular implementation supports the future reuse of haptic signals built-in on the bicycle. I integrated haptic signals on a stationary bicycle trainer that is connected to my VHCE. Every time the HCWS detects a possible danger, it sends vibrations to the handles of the bicycle. Thereby, it is possible to change some variables, in my case, to switch on or off the haptic signal and to alter the warning distance to the current danger (i.e., other vehicles) when a warning is sent. As a result, participants are able to ride a real bicycle through a virtual environment, while receiving haptic warnings. Moreover, I designed and conducted a psychological experiment with test subjects to give first insight into the usefulness of my VHCE. I analyzed both collected data of their ride and a questionnaire filled in by the participants afterwards.

Summarized, my main achievement is the development of a VHCE based upon the VCE that supports real-life traffic simulations, especially for urban traffic. At intersections the cyclist follows a given, randomly generated route and has the possibility to choose an arbitrary direction (steering left, right or go straight ahead). Therefore, yield and priority signs are taken into account. Like in real life, the other traffic participants may behave contrary to road traffic regulations. However, I assume that the cyclist themself acts according to traffic rules. The VHCE contains my HCWS that is based on IVC and sends signals to give directional cues to the cyclist. It is possible to adjust it by setting different options for patterns and side of vibrations. To interact with the environment, a physical stationary bicycle trainer is equipped with vibration motors. The cyclist can react with any desired action on incoming haptic signals, for example to ignore it, to stop, to speed up or to bypass the dangerous vehicle. Furthermore, the latency of the implemented HCWS is considered. Additionally, I give a first impression of the benefits of the implemented haptic signals by setting up an experiment about the ways cyclists react to dangerous situations at intersections with and without haptic feedback.

Chapter 2

Fundamentals

In this chapter I summarize and explain necessary fundamental knowledge for this thesis. After the discussion of the Virtual Cycling Environment which is the core of my VHCE and short explanations about the usage of protocol buffers, I concentrate on relevant facts that influence the choice of my simulation scenarios and the design of haptic signals. Finally, I outline interesting research that has been done in the context of my thesis.

2.1 Virtual Cycling Environment

My work builds on the Virtual Cycling Environment (VCE) as it is described in "Modeling Cycling Behavior to improve Bicyclists' Safety at Intersections" [10] without the warning collision system introduced there and with minor changes to improve the visualization introduced in [8]. In the following, I name it the *core VCE*. Figure 2.2 gives an overview about my resulting simulation environment, the VHCE, which I built on top of the core VCE for this thesis. To simplify the comparison to the core VCE, the figure is based upon Figure 2 of Heinovski et al. [10]. The green parts are components I modified (blue highlighted with green core) or inserted (green highlighted) to integrate the handling of haptic signals. The grey faded component is not used for my experiments. In the following, I concentrate on the description of the core VCE, my modifications are discussed in Chapter 3 on page 16.

In this thesis, haptic signals are to be integrated into an already existing bicycle that is put on a stationary trainer with exercise rollers as shown in Figure 2.1. It is extended by sensors and applications to measure its current speed and steering angle. The handlebar of the bicycle is equipped with an Android smartphone. Based on free available information of the smartphone's accelerometer and magnetometer, Stratmann implemented his android application "BicycleTelemetry" [10] that

Figure 2.1 – Bicycle on stationary trainer with exercise rollers

computes the steering angle of the bicycle. In order to measure its velocity, an infrared sensor is fixed at the spokes of the bicycle's rear wheel and connected to a commercial off-the-shelf Raspberry Pi (as described in [10]). The physical bicycle together with its sensors are the input device of the User Interface of the core VCE. The cyclist can ride the virtual bicycle in the VCE by cycling the physical bicycle

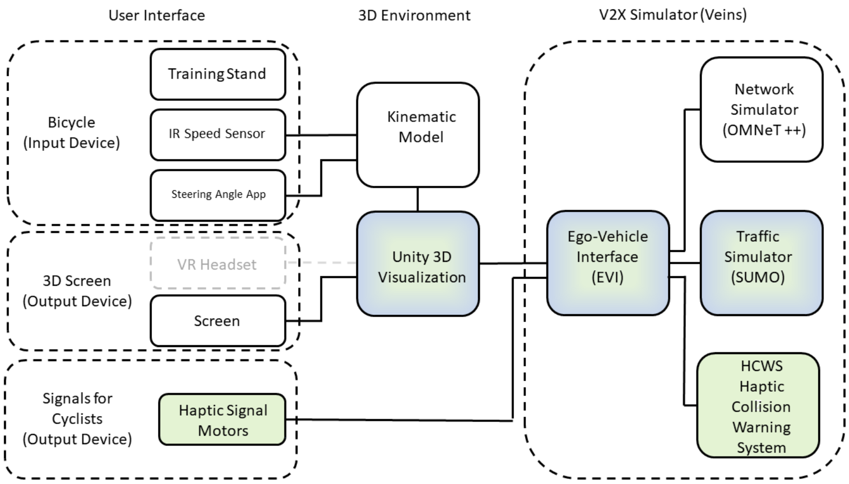


Figure 2.2 – My VHCE based on Figure 2 of [10]

on the training stand as it is discussed by Heinovski et al. [10]. To interact with the virtual world and to visualize the bicycle's surroundings, e.g., other vehicles, the road, and buildings, the cyclist can look either through 3D glasses or on three monitors arranged in front of him. These are the output devices of the User Interface of the core VCE.

As described by Buse, Sommer, and Dressler [11], the system architecture of the VCE consists of two different simulation platforms that are connected by an interface that is called the Ego Vehicle Interface (EVI) [12]. The first system (called 3D Environment) consists of a component called Kinematic Model and the Unity Visualization Component that shows the 3D simulation to the cyclist. As this system interacts with the physical bicycle, it runs in real-time. All data collected in real-time during the cyclist's physical ride is transmitted over an IP network via UDP to the Kinematic Model. This module prepares the bicycle's position and orientation and sends the computed coordinates to Unity[4]. The Unity component creates a 3D visualization of the bicycle, of its position in the virtual world and of its surroundings like road networks and buildings. Specific information about the scene around the bicycle, e.g., its position relative to other vehicles, is obtained via EVI from the Simulator of Urban MObility (SUMO) of the second simulation system. Obviously, it is necessary that both simulators, the 3D Visualization and SUMO, have shared information about the structure of the simulation scenarios. For this, our Unity implementation accepts the same input files as SUMO.

The second simulation framework is the open-source VEhIcular NetworkS (Veins) simulator [13] that is used for evaluating Vehicular Ad-hoc NETworks (VANET) [11] and to "enable cars to become part of a complex interaction system, a smart city" (extracted from [12], p. 33). As described in Sommer, German, and Dressler [13], its purpose is to improve the evaluation of IVC protocols. Veins handles the simulation of the bicycle and the cyclist's environment, for example by setting it up, running, and monitoring it. To simulate vehicle networks, Veins couples bidirectionally two important components via TCP: the simulation package SUMO to simulate road traffic [14] and the Network Simulator OMNeT++ to model communication patterns of VANET nodes [15]. Both simulators are working in parallel. The necessary communication of the components is assured by defining a common Traffic Control Interface (TraCI) that enables realistic simulations and comparable research results. Furthermore, Veins supports the integration of additional components, for example a possible ADAS. In general, the modularity of the core VCE enables an easy exchange and extension of its components. I used this feature in my work for the integration of the HCWS instead of the old ADAS being described in [10].

The V2X Simulator applies the commonly used simulator SUMO for simulating both the virtual bicycle itself (being called the ego vehicle) and the virtual world it is moving through, e.g., road networks, moving vehicles, and buildings [14]. As described in [14], SUMO uses a microscopic approach. This implies that each vehicle is defined by a unique identifier, its departure time and route. Additionally, it is

[4] `https://unity.com`

possible to add further information like the type of the vehicle. Road networks are mapped to graphs using nodes to represent intersections, edges to represent roads, and attributes can be added to define their types. It is possible to create and simulate very large-scale grids. Although I used only the grid network, SUMO scenarios can also be generated from real-world OpenStreetMap data. That highly supports the realism of the simulation. SUMO can be applied for research on IVC. Trace files of the vehicles' movement can be generated by SUMO that reflect the flow between the communication nodes in the OMNeT++ Simulation. In other words, the traces can be used to feed a communication simulator with realistic vehicle behavior as described in [10]. Vice versa, the IVC must have an impact on the behavior of the vehicles in realistic simulation experiments. That means, the communication nodes being simulated in OMNeT++ also have to interact with the movement of vehicles in SUMO. Both traffic simulation and communication simulation are an important requirement if the interaction of cyclists with realistically timed haptic signals is examined. Therefore, Veins is my tool of choice for this thesis. Veins enables the road traffic and the network traffic to influence each other in both directions.
As OMNeT++ is an event-based simulator and SUMO a time-discrete one, their bi-directional communication is quite simple. Both modules are running in a "non-real time fashion with flexible time granularity – that is, time progress in the simulation (simulation time) is fully decoupled from real (or wall-clock) time" (extracted from [12], p. 33). As described in [14], SUMO is implemented to be a time-discrete and space-continuous traffic flow simulator. Therefore, the simulated vehicles change their state in predefined, discrete intervals of simulation time. Both simulators buffer any commands, arriving in between time steps, to guarantee synchronous execution at defined intervals.

In this thesis the influence of haptic signals on the safety of cyclists is to be examined by simulating the bicycle, named ego vehicle, and the cyclist's reaction on given haptic signals. Therefore, a specific integration interface, the EVI, is needed to build a bridge between the discrete-event VANET simulator Veins and the real-time 3D Driving Simulator as described by Buse in [12]. EVI enables real time hardware-in-the-loop applications by coordinating the two simulation systems and all necessary communication. Through this, it solves the problem of the different simulation time models (time-discrete versus wall clock time). As a result, a person can ride their own virtual vehicle, the ego vehicle, through the 3D simulation environment while being presented a possible VANET application to test on the three monitors. The cyclist's behavior sends messages to the EVI to trigger actions in the VANET simulation at a matching time. Results needed in the next steps are precomputed there and forwarded to the EVI to trigger the corresponding visualization.

2.2 Protocol Buffers

I use Google's protocol buffers[5] to send messages from the Veins simulation to the Raspberry Pi that is connected to the motors mounted on the physical bicycle. One of my reasons for using protocol buffers, being freely available since 2008, is that they have already been used in the implemented communication of the components of the core VCE and the EVI. An important advantage of protocol buffers, as described on Google's website[5], is their independence from programming languages and platforms. Therefore, it is possible to work with protocol buffers for all the languages that are applied in the core VCE, especially for Python, C#, and C++. Furthermore, the application of protocol buffers guarantees consistency for my communications.

As stated in the definition, protocol buffers are used to serialize structured data such as communication protocols and data storage. At first the requested data structure is defined. As described on Google's website, a message of the protocol buffer consists of name-value pairs, each being a numbered field. These fields can either be optional, required, or repeated. Their possible values vary from standard values like strings, booleans, or numbers to more complex types. A field defines the (possible) content of a message of the described kind. In a next step, the protocol buffer compiler translates the files. The generated source code can be used for comfortably reading or writing the structured data to and from a huge number of different data streams[5]. Moreover, extensions in messages can be included without too much additional work and with still ensuring backwards-compatibility. Older versions simply ignore the fields introduced later. This feature supports an easy adaption of my implemented communication with the Raspberry Pi to possibly improved granularities of haptic signals in the future.

2.3 Dangerous Traffic Situations

To identify useful scenarios for testing the HCWS, it is important to determine situations where signals would be most helpful. Therefore, as a first step scenarios are considered that typically lead to collisions between cyclists and other traffic participants.

A German study from 2015 [16] collected data about accidents between cars and bicycles that caused a personal injury and a total claim value of 15,000 € or more. Kuehn, Hummel, and Lang [16] showed that 84% of the incidents occurred at the front of a car (including the left-hand and right-hand front wings). Therefore, the

[5] `https://developers.google.com/protocol-buffers/`

study focused on these cases. To get results, the authors firstly distinguished between four scenarios: bicycle approaching a car from the right (Case A, probability of 42%), from left (Case B, probability of 34%), driving in the same direction as the car (Case C), or the opposite direction (Case D). Even though the probability of serious injuries was the highest in Case D, the amount of accidents in Case D was still much lower than in Case A and B, as Case A and B were much more likely for accident scenarios. As the same argument applied for Case C, the authors of [16] concentrated on a deeper look at cases A and B by dividing each of them into three subcases:

- The bicycle is coming from the right (Case A) and:
  - the car is turning left (9% of A, scenario A1)
  - the car is driving straight ahead (46 % of A, scenario A2, cf. Figure 2.3)

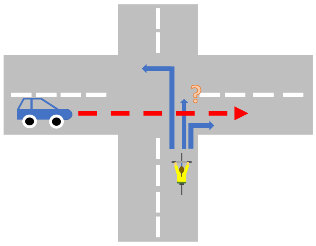


Figure 2.3 – Scenario A2, car is driving straight ahead, bicycle is coming from the right

  - the car is turning right (45% of A, scenario A3, cf. Figure 2.4)
- The bicycle is coming from the left (Case B) and
  - the car is turning left (25% of B, scenario B1, cf. Figure 4.1 on page 38)
  - the car is driving straight ahead (43% of B, scenario B2, cf. Figure 2.5 on page 10)
  - the car is turning right (23% of B, scenario B3)

Finally, the authors [16] identified three most common scenarios for crashes:

- the car is driving straight ahead and the bicycle is coming from the right (cf. Figure 2.3)
- the car is turning right and the bicycle is coming from the right (cf. Figure 2.4)
- the car is driving straight ahead and the bicycle is coming from the left (cf. Figure 2.5)

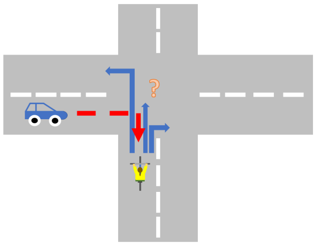


Figure 2.4 – Scenario A3, car is turning right, bicycle is coming from the right

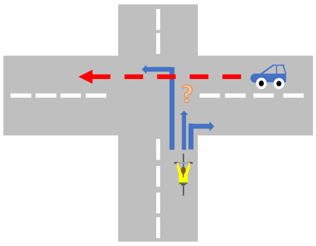


Figure 2.5 – Scenario B2, car is driving stright ahead, bicycle is coming from the left

As less than 8% of cars involved in crashes were parked, this thesis focuses on moving cars. I give more details about my design of simulation scenarios in Section 4 on pages 36ff. .

2.4 Haptic Stimuli

Historically, the word haptic is derivated from the Greek word *haptikos* that is related to "a sense of touch"[6]. Although the context in which the word haptic is used varies (e.g., haptic technology, haptic communication, haptic design, haptic perception), all of them define haptic in relation with "perceptions felt through the skin and [our] ability to sense the positions and movements of [our] limbs" as it is described on page 330 of [17].

In his reference book [18] and in a later published work [17] Goldstein divided the haptic system of the human body into two different subsystems. Figure 2.6 outlines the structure of the haptic system that is also referred to as Haptic Somatosensory System. The first subsystem contains the cutaneous senses being "responsible for perceptions such as touch and pain that are usually caused by stimulation of the skin" (cf. page 330 of [17]). In more detail, the first group is structured in tactile perception (mechanical stimuli), temperature, and pain. The second one is subdivided into

[6] `https://www.lexico.com/en/definition/haptic`, accessed 2019-08-10

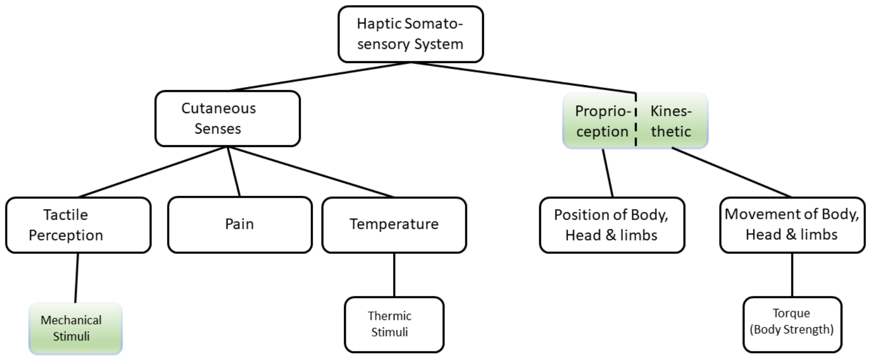


Figure 2.6 – Haptic Somatosensory System inspired by DIN ISO 9241 and [17]

the proprioceptive sense and the kinesthetic sense. Proprioception allows human beings to control their body, head, and limbs intuitively. That means they can feel their position without directly taking a look at them [7]. Kinesthesis enables them to sense their movement. In the context of Computer Science, the frequently-used DIN ISO 9241 defines *haptic* in the scope of interaction between humans and computers[8] and is more or less in line with the above definition.

To improve bicycle safety, I am applying haptic stimuli on the bicycles by adding vibrating handles on handlebars as a signaling device.They are based on the tactile perception of mechanical stimuli sent to the cyclist's hands. In the skin, specific receptors, called mechanoreceptors, respond to its mechanical stimulation. Especially in the palms, the receptors react very sensitively to stimuli as described in [18]. Furthermore, as the cyclist has to recognize which palm is stimulated by vibrations (left, right, or both), kinesthetic and proprioceptive perception are involved, too. In Figure 2.6 I highlighted the parts of the haptic somatosensory system that are involved in processing vibrations.

The work of [7] disproved my first assumption that the usage of vibrotactile warnings at more than two places increases the cyclist's attention and safety. As a result, I restrict the stimulation to the cyclist's palms.

[7] `https://www.brainblogger.com/2009/06/09/what-is-proprioception/`, accessed 2019-08-06

[8] 9241 D. E.I.: Ergonomie der Mensch System Interaktion – Teil 910; Norm; Deutsches Institut für Normung, Berlin; 2011

To use an adequate strength of vibration, I had to take care of its Absolute Threshold. On page 13 of [17] the Absolute Threshold is defined to be the "smallest amount of stimulus energy necessary to detect a stimulus" by the receptor in the skin. In the beginning of this thesis, I had some problems to adjust the strength of vibrations as I discuss in Section 3.2 on pages 17ff.. Another consideration is the frequency of the chosen vibrations. Vibrations address the so called Pacinian corpuscle that can be stimulated efficiently by high rates of vibrations while it has a limited perception of constant pushing [17]. The spreadsheet 11.1 of [18] lists a frequency between 10 and 500 Hz to be optimal. Obviously, such results give only hints how to choose a suitable frequency of vibrations as they measure the quantitative potential of skin only. Nevertheless, our installation is in line with these results (cf. Section 3.2 on pages 17ff.).

2.5 Related Work

The number of works on signals on bicycles is still limited. Some publications focus on using haptic signals to convey information to cyclists. Most approaches try to use signals for navigation, platooning, or collision warning [1], [6], [19], [20]. In the following, I take a look at some of these works to evaluate their effectiveness and to draw conclusions for the development of my haptic signals.

As already mentioned, my design of haptic signals is partially motivated by the promising study of Matviienko et al. [7]. It examines which different warning signals are easily understood by children, and how to combine them to increase their efficiency in dangerous situations. The aim of their study was to investigate the effectiveness of different, single (unimodal) or combined (multimodal) stimuli, i.e., vibration, light, and audio, to assure a better safety for children riding a bike. Using these different factors, they were able to restrict their experiments to a non-steerable interactive bicycle simulator and to concentrate on simplified scenarios. In a first exploratory study without any traffic simulation, the authors tested different stimuli, more detailed vibrations on the handles (left, right, both) or the saddle of the bicycle, three light positions on the handlebar (left, right, entire), (a speaker on the handlebar), and semantically reasonable combinations of lights and vibrations. The children under test had to ride a bike on an empty road and to stop if they noticed an arriving signal. Afterwards, they were asked how the received signals were interpreted. Matviienko et al. [7] found out that "Unimodal signals were the easiest to recognize [100% recognition] and suitable for encoding directional cues" that is discussed in the beginning of [7]. As participants "spent significantly more time perceiving visual than auditory or vibrotactile cues" (extracted from [7], p. (15:)10) and disliked the speaker on the handlebar, the use of vibrations turn out to be the best solution in this context. "However, when vibration was presented in more than two locations, interpretation became ambiguous. For example, the lowest recognition rate occurred for signals with vibration at three locations" (extracted from [7], p. (15:)5). This supports my decision to send vibrations to the handles only. In a second, two-level experiment the authors extended the simulator to explore the efficacy of the signals above (and their combinations) to prevent car-to-cyclist collisions. In a first level, unimodal visual signals, like haptic signals or audio ones (this time in a helmet) were used to give directional hints to the cyclist riding straightforward down a long street with several intersections and cars approaching from the left or right. For situations with higher urgency the children had to stop immediately. Remarkably, the study stated that the safety of children was increased if using all warning signals together. Thus, in the beginning of their study Matviienko et al. noted that when "priming stop actions, reaction time was shorter when all three modalities were used simultaneously". But as above, the haptic (and audio) warning signal turned out to be the favorite one for directional cues.

Many works focus on using haptic solutions as navigation tools, for example, the on-body solutions of [20], [21], or the bicycle based system being described in [19]. In [20] scientists designed a belt for the cyclist generating on-body vibrations for tactile clues on how to navigate to a certain destination. The designers of the belt researched the advantages and disadvantages of a haptic navigation in comparison to a visual one. On one side, they found out that haptic navigation seemed to be less distracting for test subjects. On the other side, people got confused about different information conveyed by signals being too similar. Additionally, test subjects were slower at intersections in comparison to visual navigation, as they had to pay attention more carefully to signals informing about necessary changes of direction. To prevent such confusions, I kept the conveyed information rather limited. Moreover, my solution of integrating vibration motors on a bicycle is easier and cheaper than designing a comfortable belt.
Instead of applying two small vibration motors on the handlebar, another group of researchers developed a haptic navigation tool that consisted of one of these motors worn on each wrist [21]. As described by Camila Escobar Alarcon and Ferrise [21], the left side vibrating instructed the cyclist to turn left, the right side vibrating to turn right. If both sides vibrated, the cyclist reached the requested destination. Moreover, they used the number of repetitions of vibrations to convey information of the distance. One vibration stood for an event in about 60 m, two vibrations meant that the event was in short distance. In first experiments, their result was that users understood the signals easily. For my study, I also consider the distance

of an obstacle to warn early of upcoming dangers and to give the cyclist enough time for a fitting response. Furthermore, it seems promising that first participants connected a vibration on the left as information to turn left [20]. In my design I use a left vibration as an information of danger approaching from the left. Based on the results of the paper, I expect that people understand this information intuitively. Nevertheless, motors worn on wrists are again additional tools that I wanted to avoid. Another study combined visual navigation with haptic signals to support navigation on bicycles for tourists [19]. The authors again used two vibrotacticle motors mounted on the handles of the bicycle. In a first small test, participants were successfully led to an unknown destination and informed about interesting sights near by. Although the haptic signals reduced the amount of visual attention spent on the display, the cyclist's concentration on the surroundings was still weakened. Moreover, even if the cyclist did not look at the display, the attention was disturbed by changes of the visual presentation. Especially people at increased risk like elderly people or children need all their attention to prevent dangerous situations. Therefore, I focused on haptic signals without additional visual support.

Another publication focused on using haptic signals to support driving in a platoon of cyclists [6] and combined cycling with cooperative driving. The authors [6] informed the cyclist via a so-called Cooperative Adaptive Cruise Control to speed up or slow down to gain a requested speed. The system worked "[...]by spinning a rough textured eccentric cylinder forward or backwards that scratched the participant's hand palm" [6]. If it is turning forward, one has to gain speed; and if it is turning backwards, one has to slow down. The system was tested and declared successfully for large platoons up to ten participants. In my work, I decided to use vibrating motors instead of spinning cylinders as the interpretation of spinning is less intuitive. Usually, spinning forward or backward is intuitively understood as moving forward or backwards, or increasing and decreasing something (cf. Section 2.4).

Especially a work about a low cost collision warning system seems to be of interest for my work [1]. In [1], the implemented system warns of vehicles coming from behind the cyclist. The system consists of two vibration motors being mounted on the handles of the bicycle. Instead of using IVC to warn of approaching vehicles, their approach is based on a "single-beam laser rangefinder and two ultrasonic sensors that detect oncoming vehicles from behind[...]" (cf.[1], p. 3731). The cyclist is warned when a vehicle is approaching. Depending whether the vehicle is more on the right of the vehicle or more on the left, the authors adjust the cyclist's course by having stronger vibrations on the corresponding side. To evaluate the effectiveness and usefulness of the system, test-subjects first had to take part in an experiment without training or knowledge of how the system works. The authors found out that the system improved the driving behavior of the cyclists intuitively without influencing the concentration during cycling. I also installed motors on the two bicycle handles to cause vibrations. However, in contrast to their paper [1], my focus is on dangers that are approaching from left and right in front of the cyclist and that build upon the most common crash scenarios mentioned before [16]. Alike to their approach, I use the frequencies of the vibrations to convey information about the distance of the obstacle. However, as their small experiment found that the cyclist's concentration and attention is not disturbed by haptic signals, I expect to have comparable results for my design, if this would be tested. Actually, I only examine the cyclist's reaction time and do not check the concentration and attention of the cyclist.

In order to implement my HCWS based on IVC, a human-in-loop simulation is needed for its extensive testing. By [8] and [10] it was investigated that the VCE being published by Buse, Sommer, and Dressler [11] is qualified for simulating VANET enabled ADAS. Heinovski et al. [10] developed the core VCE (cf. section 2.1) my cycling system is based upon. For example, the authors of [10] equipped a bicycle with hardware for Inter-Vehicle Communication to investigated the "benefits of networked assistance systems for road traffic safety" (cf. page 4 of [10]). Within the scope of my work it is important that their results "demonstrate the need for such an integrated framework for empirical studies" (extracted from [10], p. 1). Furthermore, the work of Stratmann [8] and his master thesis on the same subject "conclude[s] that the VCE is indeed suitable for conducting ADAS-testing experiments in a human-in-the-loop-setup" (extracted from [8], p. 10). Stratmann improved the VCE, e.g., he enhanced its 3D environment and added an application to measure the steering angle of the bicycle. Both works make it possible to build on a more realistic virtual environment that is necessary for testing my HCWS.

Chapter 3

Virtual Haptic Cycling Environment

My VHCE is built upon the existing VCE (cf. Section 2.1 on page 4). In this chapter, I describe the requirements for the implemented system, the installation on the bicycle, the Software Hardware Interface, and the chosen semantics for haptic signals. Additionally, I conduct a short evaluation of its performance.

3.1 Requirements

The task of the designed system is to warn cyclists of possible dangerous traffic situations. It should help to prevent collisions with other vehicles by giving directional cues. Additionally, I want to inform of general dangerous situations if vehicles are approaching from more than one direction at once. In general, it should be possible to control each vibration individually for left and right vibrations or have both motors vibrating at the same time. To include more details about a danger, I want to give further information of the distance to a critical point. It should be easy to vary how early the systems warns of possible collisions. Furthermore, the vibration pattern should be flexible to allow easy changes. My system should not only warn of vehicles which currently are on collision course with bicycle, but also of vehicles which are missing the bicycle by a short amount of time. Hence, I take a look at the *Post-Encroachement-Time* (PET) as described by Detzer et al. [4]. If two objects have intersecting driving routes, the PET is computed as the difference in time of the objects passing the intersection point [4]. Introducing the Post-Encroachment-Time (PET) to my collision warning algorithm integrates safety relevant buffer zones into my system. My haptic collision warning system should be intuitive and without causing much additional cognitive load.

In the following I describe the designed VHCE that is implemented to meet all the requirements above.

3.2 Implementations

The integration of haptic signals into the VHCE was structured into two different tasks. Firstly, I had to integrate the signals on the physical bicycle. Secondly I implemented the Software Hardware Interface triggering the vibration on the bicycle in case of arriving dangers. As indicated in Figure 3.1 on page 17, I modified both the Unity 3D component, and the interfaces of EVI and the Raspberry Pi. Additionally, I wrote test scenarios for SUMO. My main task, however, was to implement a new HCWS.

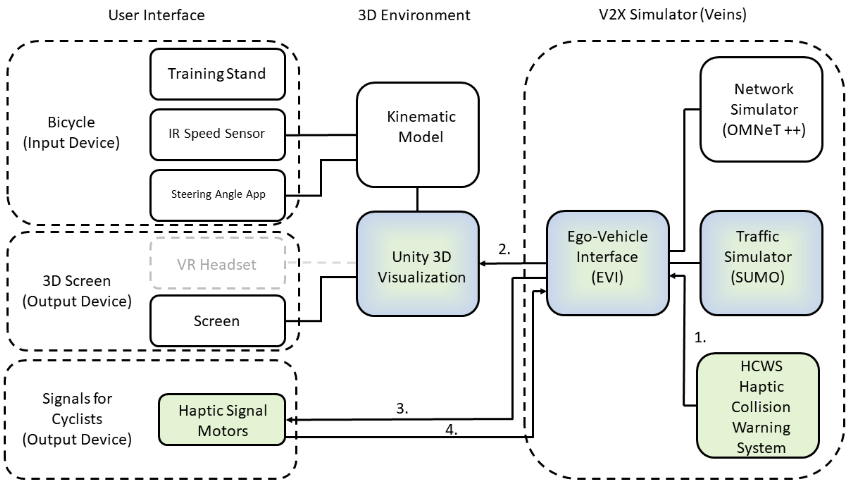


Figure 3.1 – Different steps of processing haptic signal messages based on Figure 2 by Heinovski et al. [10]. Green are the new developed components, blue-green are the extended components.

3.2.1 Bicycle-based Implementation of Haptic Signals

In the scope of my work I augmented a physical bicycle integrated in the VCE with haptic signals. For a first solution, Johannes Blobel and I decided to use small vibration motors to achieve an optimal cost-benefit ratio. As my work requires distinguishing between the left and right motor on the handlebar, we installed one of them onto each handle of the bicycle. Vibrations were caused by the two motors that were connected to the Raspberry Pi integrated in the VCE as discussed in Section 2.1. The motors needed an additional power supply because the Raspberry Pi did not offer sufficient current. They were controlled by an implemented control program which allows to trigger each motor individually by changing the value at two different General-Purpose Input/Output pins (GPIOs). Unfortunately, with this installation, it was impossible to differentiate between left or right as the vibrations were transferred to the other side of the handlebar. As a result, the whole handlebar was vibrating.

To solve this problem, we decided to move to smaller smartphone motors that

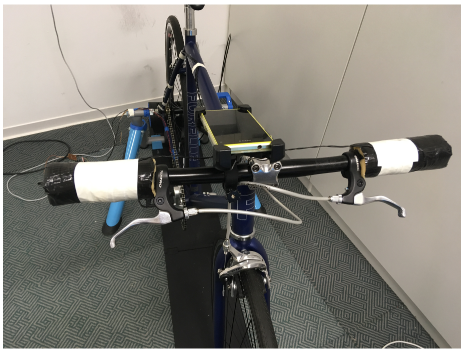


Figure 3.2 – Setup on bicycle with vibration motors and smartphone for steering angle.

also cause vibrations. The vibration amplitude cause by their shaking has a value of 0.75g and their frequency (for 3 V) is about 200 Hz[9]. This is in line with the psychophysical characteristics of the sense of touch as described in Section 2.4. The smartphone motors are shaftless, round vibration motors with a height of 3.4 mm and a width of 10 mm. As they are much smaller than the vibration motors selected first, different positions to install them on the handles were possible. In my implementation I installed them directly on top of each handle of the bicycle.

Before their installation I wrapped a layer of foam around each bicyle handle and secured it with tape (cf. Figure 3.2). In a next step, each motor was taped to the

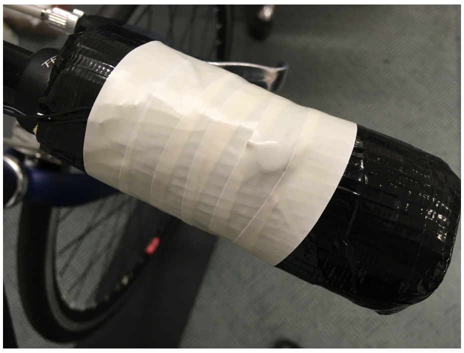


Figure 3.3 – Vibration motor taped to right handle of the bicycle

handle on each side by positioning it beneath the cyclist's hands (cf. Figure 3.3). The foam prevents the transmission of vibrations to the other side and allows the

[9] `https://www.pololu.com/product/1636`, 1 g = acceleration of earths gravity

cyclist to distinguish the two sides. The motors are controlled and run by the Pi in the same way as the first solution. In comparison to the first solution, these motors do not need an additional power supply as they have a recommended operating voltage of 2.5-3.5 V, only. This solution is used in my thesis.

3.2.2 Haptic Collision Warning System

My implementation objective was to implement the HCWS in Veins and to enable it to trigger vibration motors mounted on a physical bicycle. Within the scope of my work in the Safety4Bikes project[10] I already implemented an algorithm for simple collision detection in Veins as described in [10]. It checked whether two objects were colliding by examining if their graphical representations in the simulation were overlapping. Therefore, the shapes of vehicles were simplified to polygons. After that, the *separating axis theorem* was applied to efficiently check for overlaps. For my thesis, I decided against reusing this algorithm, as my target was to predict collisions, and not only to find them. Although it may be possible to extend the buffer zone around the vehicles to warn before an actual collision takes place, this approach does still not regard the driving direction and speed of the involved vehicles. Additionally, as I want to give warnings already several meters before the possible collision, the buffer zones would grow too large. This may result in many unnecessary warnings. As a result, ambiguous warnings may confuse the user and may damage confidence into the designed system. However, I reused some of the mathematical functions, e.g., how to normalize vectors and other geometrical computations.

To implement my HCWS in Veins, I initially designed the structure of the main algorithm. Algorithm 3.1 on page 20 is described by its pseudo code. Whenever a *mobilityStateChanged* signal is received in Veins, the algorithm is triggered. Firstly, the algorithm needs a list of vehicles currently existing in the simulation. Additionally, *earliness* and *range* are its arguments. The *radius* defines the area around the bicycle where the algorithm checks for approaching vehicles. The *earliness* defines the time a warning is given before a possible collision may take place. The variables *highestDanger* and *highestDanger* are used to determine the most dangerous message among all messages which could possibly be caused by approaching vehicles. By that, the cyclist is always aware of the most urgent danger. When a message is sent, *pattern* defines how the motors should react to the current danger, i.e., their vibration pattern. For my thesis, its value is always "STANDARD". The *modus* defines the side of vibration.

Next, it is assured that the algorithm runs only on the *ego-vehicle*, i.e., the vir-

[10] `https://www.safety4bikes.de`, accessed 2019-08-04

tual representation of the physical bicycle in the simulation. Basically, the algorithm compares the positions of all vehicles with the one the application is running on, the ego vehicle. If a given distance (*range*) between the ego vehicle and other vehicles is below a certain threshold, the collision probability increases. Therefore, a further look is taken at each of these vehicles. Based on their driving direction and speed, the vehicle's relative position to the *ego vehicle* is computed. For this purpose, I implemented Algorithm 3.4 discussed later. Depending on the vehicle's position, i.e., in front, on the left, or on the right of the ego vehicle, I used two other algorithms, the Algorithm 3.2 and Algorithm 3.3. For computations in these algorithms, basic

```
Require: earliness, radius
Get a list L of all vehicles v_i in simulation

if the vehicle v_c the application is running on == ego vehicle then
   highestDanger = 0
   highestDanger = NONE

   if |L| > 1 then
      for all v_i ∈ L do
         if v_i == v_c then
            break
         else
            Get coordinates curr_1(v_c), curr_2(v_i)
            Get angles ang_1(v_c), ang_2(v_i)
            distance = sqrt((curr_2.y − curr_1.y)^2 + (curr_2.x − curr_1.x)^2)

            if distance < radius then
               dangerDirection = computeDangerDirection(curr_1, curr_2, ang_1)
               if dangerDirection == "LEFT" or dangerDirection == "RIGHT" then
                  algorithmLeftRight(dangerDirection, earliness,
                                     highestDanger, dangerModus,
                                     curr_1(v_c), curr_2(v_i), ang_1(v_c), ang_2(v_i))
               end if
            else if dangerDirection == "FRONT" then
                  algorithmFront(earliness, distance, v_i, v_c, highestDanger,
                                 highestDanger)
            end if
         end if
      end for
      if a new warning was found then
         send message(dangerLevel, modus, pattern)
      end if
   end if
end if
```

Algorithm 3.1 – Structure of basic algorithm

Require: *earliness*, *distance*, v_i, v_c, *highestDanger*, *highestDanger*

```
1:  notMoving = "FALSE"
2:  if speed(v_i) == 0 then
3:    notMoving = "TRUE"
4:  end if
5:  if notMoving then
6:    ttc_1(v_c) = computeTTC(distance, speed(v_c))
7:    if ttc_1 < earliness then
8:      if dangerModus ≠ "BOTH" then
9:        dangerModus = "BOTH"
10:     end if
11:     if highestDanger < computeDangerLevel(ttc, earliness) then
12:       highestDanger = computeDangerLevel(ttc,earliness)
13:     end if
14:   end if
15: else
16:   if speed(v_c) > speed(v_i) then
17:     speedDif = speed(v_c) − speed(v_i)
18:     ttc = distance/speedDif
19:     if ttc < earliness then
20:       if dangerModus ≠ "BOTH" then
21:         dangerModus = "BOTH"
22:       end if
23:       if highestDanger < computeDangerLevel(ttc, earliness) then
24:         highestDanger = computeDangerLevel(ttc,earliness)
25:       end if
26:     end if
27:   end if
28: end if
29: return highestDanger, dangerModus
```

Algorithm 3.2 – Algorithm part for danger being in front of the bicycle (algorithmFront)

geometry is used to compute planes, lines, and intersections built on the direction the bicycle is moving.

If another vehicle is in front of the ego vehicle, the motors vibrate on both handles (*dangerModus* = "BOTH") if, and only if, the *earliness* is undercut. As described in Chapter 4, I interpret it to be a general dangerous situation. First I distinguish between a moving vehicle and a not moving one. This is described in Algorithm 3.2 on page 21. If the vehicle is non-moving, the time to collision (referred to as *ttc*) can be roughly computed from the distance to the other vehicle and the speed of the *ego vehicle*. If it is moving, the difference of speed is computed first (cf. line 17). If the other vehicle is faster than the ego vehicle, they will not collide. Otherwise, a warning is sent if the Time To Collision (TTC) is below a given value.

If the danger is approaching from left or right, further computations take place as shown in Algorithm 3.3 on page 22. In a first step, I check whether the current driving lanes of the car and bicycle intersect at some point because otherwise there cannot be a collision. This is implemented in Algorithm 3.5 called *computePoint*. To decide whether a situation can lead to a collision, I take a look at the Post-Encroachment-Time (PET). The PET describes the amount of time two vehicles are missing each other at a possible collision point. The smaller the value is, the more likely is a collision between these objects. However, when one object is following the other, this value will automatically get really small and loses its significance for detection of possible collisions [4]. Therefore, I use this value only in situations where a vehicle is approaching from left or right. To be able to compute the PET, I need to compute the TTC for both vehicles involved. The TTC is based on the distance of two objects and their speed (difference) [4]. If a given PET value of 5

Require: $\mathit{dangerDirection}, \mathit{earliness}, \mathit{highestDange}, \mathit{dangerModus}, curr_1(v_c), curr_2(v_i), ang_1(v_c), ang_2(v_i)$

$\vec{s_1} = (curr_1.x, curr_1.y)$
$\vec{s_2} = (curr_2.x, curr_2.y)$
$\overrightarrow{dir_1} = (ang_1.x, ang_1.y)$
$\overrightarrow{dir_2} = (ang_2.x, ang_2.y)$
$\overrightarrow{dist} = \text{computePoint}(\vec{s_1}, \overrightarrow{dir_1}, \vec{s_2}, \overrightarrow{dir_2})$

if $dist.x > 0$ **and** $dist.y > 0$ **then**
  $ttc_1(v_c) = \text{computeTTC}(\mathit{distance}, \mathit{speed}(v_c))$
  $ttc_2(v_i) = \text{computeTTC}(\mathit{distance}, \mathit{speed}(v_i))$
  $pet = |ttc_1 - ttc_2|$

  if $pet < 5$ **then**
    if $\mathit{highestDanger} < \text{computeDangerLevel}(\mathit{ttc}, \mathit{earliness})$ **then**
      $\mathit{highestDanger} = \text{computeDangerLevel}(\mathit{ttc}, \mathit{earlines})$
    end if
    if $\mathit{dangerModus}$ == "NONE" **then**
      $\mathit{dangerModus} = \mathit{dangerDirection}$
    else if $\mathit{dangerModus} \neq$ "NONE" **then**
      $\mathit{dangerModus}$ = "BOTH"
    end if
  end if
end if
return $\mathit{highestDanger}, \mathit{dangerModus}$

Algorithm 3.3 – Algorithm part for danger approaching from the left or the right (algorithmLeftRight) is undercut, I will send a warning. To evaluate different possibilities of warning earliness, the *earliness* is changed after fifteen warning signals. I evaluate three different values of warning *earliness*, 3250 ms before collision, 3500 ms before collision, and 4250 ms before collision. In the beginning, I wanted to use values of 250, 500, and 1250 ms. As the cars can be seen quite early and the participants may need some time to adjust to the system, I decided to add an offset of 3000 ms.

In Algorithm 3.3 I distinguish between different severity levels of danger to adjust the temporal gap between two vibrations at the motors accordingly. I chose to distinguish between three levels, based on the distance to the point of a (possible) collision. Level 3 was defined as the most, level 1 as the least dangerous. Each level covered a third part of the distance to a possible collision point, based on the earliness a first warning should be given. For example, if a warning was sent 100 m before a possible collision point, the algorithm would compute level 1. If the distance was between 66 m and 33 m before a collision, it sets level 2. If the distance was between 33 m and 0 m it sets level 3. For my experiments, I decided to use only one level as the distances between a danger and the bicycle tended to be quite small. the reason is the small distances between intersections in cities and the chosen warning earliness. Furthermore, participants shall be able to adjust quickly to the system. As a result, the vibration motors either vibrate twice if a set of conditions takes place or not at all.

Additionally, I compute whether danger is coming from the left, from the right, or from in front of the cyclist. Algorithm 3.4 on page 24 shows the pseudo code of the implementation. Firstly, I define a corridor with a fixed width around the ego vehicle based on its driving direction (cf. line 1 of 3.4). I distinguish between points which lie in this corridor and those that do not. By defining the corridor as a plane with constraints for the width to both sides (a minimum and a maximum constraint) and using basic geometric computations, this can be done easily. If a point is in the plane, I have to decide whether it lies in front of the bicycle or behind. Any point behind is not relevant for my system as I do not warn of vehicles approaching from behind. The algorithm returns "NONE" (cf. line 6 of 3.4). The computations done in lines 9 to 19 are a little bit more complicated. They compute whether a point is positioned on the left side or on the right side of the corridor. I had to consider that the coordinate system of OMNet++ starts in the upper left corner. Consequently, I must rotate the vector $\vec{v_1}$ by α. α is the angle between the driving direction of the bicycle and the eastern coordinate axes, as this is defined as 0 in Veins. First, I compute a vector based on the position of the car and of the bicycle which is represented by a red arrow in Figure 3.4 on page 25. The green arrow visualizes the driving direction of the car which is given in radians in relation to east. Then, as shown in line 11 of 3.4, I compute the needed α and rotate the vector $\vec{v_1}$ by using a standard geometrical

Require: coordinates $\overrightarrow{p_1}(v_c), \overrightarrow{p_2}(v_i)$, driving direction $\overrightarrow{v_1}(v_c)$, normal $\overrightarrow{n_1}$ to $\overrightarrow{v_1}(v_c)$, cMin, cMax

Ensure: Correct detection of car position in dependency of bicycle position and driving direction

1: Solve $\overrightarrow{p_2} = \overrightarrow{p_1} + s \cdot \overrightarrow{v_1} + t \cdot \overrightarrow{n_1}$ with $s, t \in \mathbb{R}$

2: **if** cMin $\leq$ t **and** t $\leq$ cMax **then**
3:    **if** r ≥ 0 **then**
4:       return "FRONT"
5:    **else**
6:       return "NONE"
7:    **end if**
8: **end if**

9: Convert $\overrightarrow{v_1}$ to angle ang
10: $\overrightarrow{bc} = ((p_2.x - p_1.x), (p_2.y - v_1.y))$
11: $\overrightarrow{bc'} = ((\cos(v_1) \cdot bc.x - \sin(v_1) \cdot bc.y)(\sin(ang) \cdot bc.x + \cos(ang) \cdot bc.y))$

12: **if** $bc'.x > 0$ **then**
13:    **if** $bc'.y < 0$ **then**
14:       **return** "LEFT"
15:    **else**
16:       **return** "RIGHT"
17:    **end if**
18: **else**
19:    **return** "NONE"
20: **end if**

Algorithm 3.4 – Position Check

formula. After rotating, left and right could be easily computed by taking a look at the y-value of the vector as demonstrated by Figure 3.5 on page 25. If it is positive, the car is positioned on the right of the bicycle (cf. line 16), otherwise on the left (cf. line 14). Furthermore, the x-value has to be checked to see whether the car is also in front of the vehicle and not somewhere behind. This is guaranteed by checking if x has an positive value. If a vehicle is not important for further computations, "NONE" is returned.

If a vehicle is detected to be on the left or on the right side of the bicycle, their driving lanes have to be checked for intersection as shown in Algorithm 3.5 on page 26. In a first step, both direction vectors are normalized (cf. line 1 to 2 of 3.5) and compared. If they are equal, both lanes are parallel and no intersection takes place. The algorithm returns NONE. If not, I define the driving lanes as two lanes with a start vector s being the current position and a direction vector v based on the driving

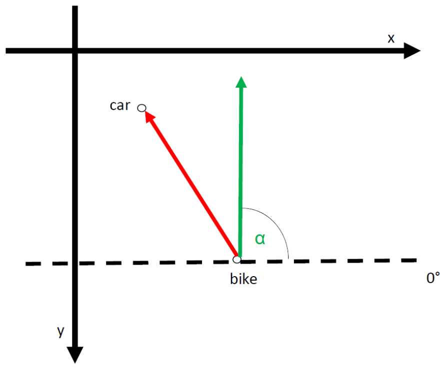


Figure 3.4 – Computing vector from bicycle to car position

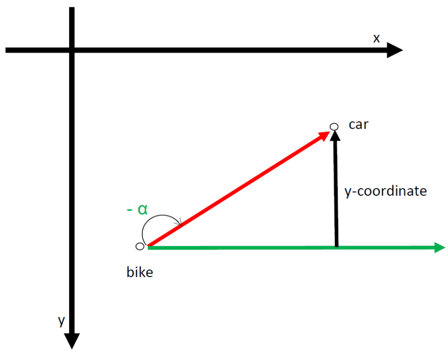


Figure 3.5 – Rotating vector clockwise by α in OMNet++

direction and equate them:

$$\vec{s_1} + r \cdot \vec{v_1} = \vec{s_2} + t \cdot \vec{v_2} \text{ with } r, t \in \mathbb{R} \tag{3.1}$$

In a next step, this equation has to be solved for r and t. It is solved by setting up two equations:

$$r = \frac{s_2.x + t \cdot (v_2.x) - s_1.x}{v_1.x} \tag{3.2}$$

$$t = \frac{s_1.y + r \cdot (v_1.y) - s_2.y}{v_2.y} \tag{3.3}$$

and computing one of the values by inserting one value into the other. If one value was found, the other can be computed by the equation shown above. To achieve that, I took a further look at possible exceptions. To prevent dividing by zero, two cases have to be considered separately.

Require: Position $\overrightarrow{s_1}(v_c), \overrightarrow{s_2}(v_i)$, driving direction $\overrightarrow{v_1}(v_c), \overrightarrow{v_2}(v_i)$

$\overrightarrow{v_1}$=normalizeVec($\overrightarrow{v_1}(v_c)$)
$\overrightarrow{v_2}$=normalizeVec($\overrightarrow{v_2}(v_i)$)
$\overrightarrow{v_{new}} = (0,0)$

if $\overrightarrow{v_1} == \overrightarrow{v_2}$ **then**
    return None
end if
double r, t
if $v_1.x == 0$ **then**
    $t = (s_1.x - s_2.x)/v_2.x$
    $r = (s_2.y + t \cdot (v_2.y) - s_1.y)/v_1.y$
else if $v_2.y == 0$ **then**
    $r = (s_2.y - s_1.y)/v_1.y$
    $t = (s_1.x + r \cdot (v_1.x) - s_2.x)/v_2.x$
else
    double $a = (s_2.x - s_1.x) \cdot (v_2.y) - (s_2.y) \cdot (v_2.x)$
    double $b = ((v_1.x) \cdot (v_2.y)) - ((v_2.x) \cdot (v_1.y))$
    $r = a/b$
    $t = (s_1.y + r \cdot (v_1.y) - s_2.y)/v_2.y$
end if

return $\overrightarrow{v_{vnew}} = r, t$

Algorithm 3.5 – Compute intersection of two lanes

The first case is

$$v_1.x = 0 \tag{3.4}$$

In this case,

$$v_2.x \neq 0 \tag{3.5}$$

as both vector were normalized and are not parallel. Now t can be easily computed by

$$t = \frac{(s_1.x - s_2.x)}{v_2.x} \tag{3.6}$$

and then r can be computed by

$$r = \frac{(s_2.y + t \cdot (v_2.y) - s_1.y)}{v_1.y} \tag{3.7}$$

with

$$v_1.y \neq 0 \tag{3.8}$$

as otherwise the bicycle would have no direction. The second case,

$$v_2.y = 0 \tag{3.9}$$

can be solved in the same way.

If both cases can be excluded, the equation for t can be inserted into the one for r and solved.

$$r = \frac{(s_2.x - s_1.x) \cdot (v_2.y) - (s_2.y) \cdot (v_2.x)}{((v_1.x) \cdot (v_2.y)) - ((v_2.x) \cdot (v_1.y))} \tag{3.10}$$

The dividend cannot be zero, as

$$((v_1.x) \cdot (v_2.y)) \neq 0 \;\; and \;\; ((v_2.x) \cdot (v_1.y) = \neq 0 \tag{3.11}$$

are guaranteed by the excluded cases.

$$((v_1.x) \cdot (v_2.y)) = ((v_2.x) \cdot (v_1.y)) \tag{3.12}$$

$$\Leftrightarrow \vec{v_1} = a, b \text{ and } \vec{v_2} = a, b \text{ with } a, b \in \mathbb{R} \tag{3.13}$$

$$\text{or } \vec{v_1} = a, a \text{ and } \vec{v_2} = b, b \text{ with } a, b \in \mathbb{R} \tag{3.14}$$

The result is a contradiction to the vectors being normalized and not parallel which was excluded by the cases before. Therefore in this case, the algorithm returns $\overrightarrow{v_{new}} = (0,0)$. All mathematical formulas I used and implemented are standard geometrical computations. To check mistakes and the implementation, I tested each function with a variety of values, especially values causing exceptions in the usual computation process, to ensure correctness. Furthermore, I tested a number of possible driving directions and points for collisions to evaluate the correctness of the program as a whole.

When the algorithm discovers a dangerous situation after its computations, it returns a warning. In this case a protocol buffer message is built. The message contains information regarding the current state of danger and the direction the danger is approaching from. Furthermore, I add the possibility to send a command for an arbitrary pattern as a string. For my experiments, I stick to a simple pattern. However in general, my implementation supports the research of different vibration patterns. The message is sent to the EVI as seen in step 1 of Figure 3.1 on page 17.

3.2.3 Interface Implementation

To build a protocol buffer message of the chosen form, I extended the protocol buffer code of the EVI and added a new class of protocol buffer messages, called *hapticsignals*. In the *hapticsignals* file, a message *Message* is defined to have an optional double value *time_s* and an optional HapticMessage *signals* (cf. line 12 to 14 of the code below).

```
1 syntax = "proto2";
2
3 package asmp.hapticsignals;
4
5 message HapticMessage {
6  optional uint32 entity_id = 1;
7  optional uint32 dangers = 2;
8  optional string vibrations = 3;
9  optional string pattern = 4;
10 }
11
12 message Message {
13  optional double time_s = 1;
14  optional HapticMessage signals = 2;
15 }
```

A HapticMessage *message* consists of an identification of the associated vehicle *entity_id*, an optional value *dangers* that describes the current danger level as an integer, and an optional value *vibrations* defining which motor(s) should vibrate. Additionally, a string value describing the desired pattern can be set. I extended the asmp.proto class with an optional field containing a *hapticsignals message* as shown above. When a message is received by the EVI, it has to be distributed to both the Raspberry Pi and Unity as seen in step 2 and 3 of Figure 3.1 on page 17. Therefore, I had to add code to the interface of Veins (of the core VCE) to read and save the messages into a buffer. When a message is received, a boolean variable is set *true* to cause further actions in the other interfaces. Furthermore, I extended the Veins interface to Unity to enable forwarding the new *HapticMessage*. The forwarded message triggers the logging of events in Unity whenever there is a warning.

To communicate with the Raspberry Pi, I had to extend the EVI by creating a new interface for the Raspberry Pi. Before implementing the communication, I decided which component would be the client and which component would be the server.

To reduce the amount of sent messages, the EVI acts as a client and the Raspberry Pi as a server. Therefore, when starting the EVI, the port has to be passed as an argument. To include the communication with the Pi in the regular EVI intervals responsible for updates, I extended the class of EVI that contains the main loop. Figure 3.6 demonstrates the client-server communication of Pi and EVI. Whenever a non empty *hapticsignals* message is received, the Pi interface of the EVI forwards it to the Pi and waits for a reply. The Pi then has to cause the corresponding vibrations and to confirm the received message.

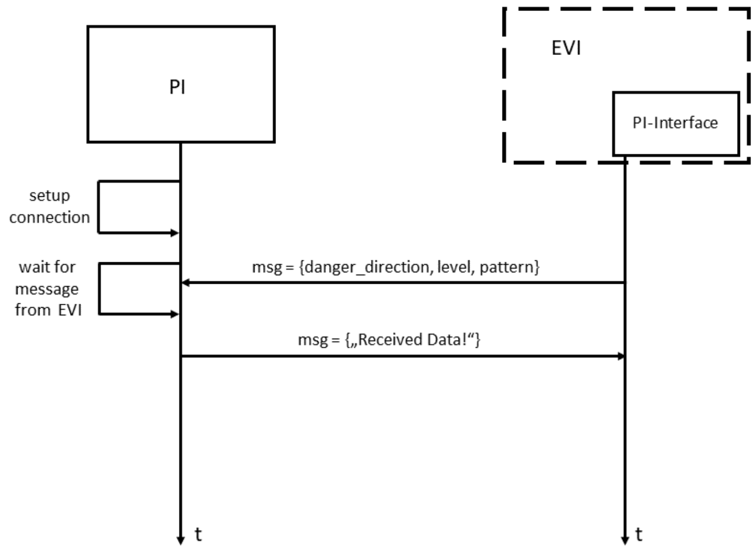


Figure 3.6 – Client-Server communication of Pi (server) and EVI (client)

3.2.4 Running the Raspberry Pi

To control the motors mounted on the bicycle, the Pi has to switch between two separate tasks. Algorithm 3.6 outlines the main program running on the Raspberry Pi. First of all, the Pi has to communicate with the EVI by receiving messages, saving them in a buffer, and sending a response. Secondly, the main program has to extract the data of the messages from the buffer and has to run the vibration according to the given orders. Each task is done by one thread. As a result, both threads are working in parallel.

For the first thread I had to set up a connection to the EVI in form of a socket.

Require: Port number of the EVI, port number of the Pi
Ensure: Corresponding vibrations according to received messages

```
pi_communication = communication.Communication(evi_port, pi_port)
msg_queue = queue.Queue()
Thread t1 = threading.Thread(target=communicationThread,
                             args=(pi_communication, msg_queue))
Thread t2 = threading.Thread(target=vibrations, args=(msg_queue))
t1.start()
t2.start()
```

Algorithm 3.6 – Main programm running on the Pi.

Therefore, I designed a communication class on the Pi. Since the EVI already used *ZeroMQ*[11] as a communication protocol, I decided to apply it, too. The implemented class has functions to receive and send messages to the EVI and is responsible for setting up a socket. Algorithm 3.7 demonstrates my implemented code. The socket is setup by initializing an object of the communication class. When receiving a message from the EVI, the protocol buffer message is read, processed, and returned as a self-defined message object. Additionally, the class has a function to send a short receipt back to the EVI which simply consists of a text "*Received data*" as seen in step four in Figure 3.1 on page 17. To ensure the ZeroMQ protocol, it is important to answer on each received message before another one can be received.

For the second thread, the buffer has to be checked regularly. Algorithm 3.8 on page 31 sketches its pseudo code. To avoid busy waiting, I included a timeout. If a message is in the buffer, it is taken out and processed. If the message includes an instruction for a vibration, the according vibration is caused by calling the motor control program as seen in Algorithm 3.9 on page 32.

[11] `http://zeromq.org/`, accessed 2019-08-02

Require: an object of the communication class pi_communication, a msq_queue
Ensure: ZeroMQ protocol and handling of messages

```
pi_communication.init()
msg = pi_communication.receive_signal_message()
if msg ≠ None then
    msg_queue.put(msg)
end if
pi_communication.sent_evi_reply()
```

Algorithm 3.7 – Task structure of the thread handling communication

Require: msq_queue
Ensure: vibration according to received messages
try msg = msg_queue.get(timeout=3)
except queue.Empty: print("queue timeout")
x = msg.getSide()
y = sg.getLevel()
z = msg.getPattern()
if x $\neq$None **then**
  gpio.vibration(x,y,z)
end if

Algorithm 3.8 – Task structure of the thread handling vibrations

To control the motors mounted on the bicycle, I wrote a small Python program running on the Raspberry Pi. Algorithm 3.9 on page 32 shows its python code. The program controls which motor vibrates how long and in which pattern by analyzing the information contained in the received protocol buffer message from the EVI. After two vibrations, the Pi stops the vibration if no new message is received. If a new message is received, the vibration pattern, duration, and side(s) are adjusted. In more detail, Algorithm 3.9 checks the requested vibration pattern, in my case called "STANDARD". It then causes two vibrations divided by a gap of 300 ms multiplied by the level. During the experiments, the level was always set to 1. A vibration is caused by setting the corresponding GPIO to one. For changing gaps due to distance to an obstacle, a shorter basic gap time would make sense. One vibration consists of 33 really short 10 ms vibrations with gaps of 10 ms. These short vibrations appear as one long vibration. This solution was necessary as the vibrations interfered with the speed sensor which is also connected to the Pi. Using long vibrations, led to nonsense values of the speed sensor and problems in the visualization of the simulation. Short vibrations avoid this problem. In future work, the problem may be solved by integrating speed measuring and vibrations into one application.

3.3 Semantics

After the installation of two motors on the left and right side of the bicycle, I had to determine how to trigger vibrations for a given dangerous situation. As described in Section 4 on pages 36ff., the design of my haptic signals builds on the following three crash scenarios [16] where a cyclist arrives at an intersection and:

- a car is coming from the left and driving straight ahead (cf. Figure 2.3 on page 9)
- a car is coming from the right and driving straight ahead (cf. Figure 2.5 on page 10)

Require: danger_level, pattern and side
Ensure: Vibration according to received data

```
side = x
level = y
if pattern == "STANDARD" then
  count = 66
  if level > 0 then
    while count > 0 do
      count = count-1
      if side =="LEFT" or side == "BOTH" then
        GPIO.output(26, GPIO.HIGH)
      else if side == "RIGHT" or side == "BOTH" then
        GPIO.output(19, GPIO.HIGH)
      end if
      time.sleep(0.01)
      if side == "LEFT" or side == "BOTH" then
        GPIO.output(26,GPIO.LOW)
      else if side == "RIGHT" or side == "BOTH" then
        GPIO.output(19,GPIO.LOW)
      end if
      if count == 33 or count == 1 then
        time_level = 0.3 · level
        time.sleep(time_level)
      else
        time.sleep(0.01)
      end if
    end while
  end if
end if
```

Algorithm 3.9 – Motor control for haptic signals

- a car is coming from the right and is turning right (cf. Figure 4.1 on page 38)

As the safety of cyclists is in the fore, the haptic signals have to warn of dangerous incidents, mostly approaching on the left or the right of the cyclist. Therefore, I decide that best approach would be to give the cyclist directional cues to indicate the direction of the danger. For example, when the cyclist arrives at an intersection, a left vibration warns of a danger approaching from the left. I came to this decision in accordance with the work of [7] discussed in detail in Section 2.5.

The situation gets more complicated if obstacles like other cars and bicycles appear from both sides of the cyclist. To keep the signal simple, I decided to treat this incident as a general, dangerous situation. I do not give individual information about each approaching vehicle on each handle, because this could result in a confusing, alternating sequence of vibrations (like left, right, left, etc.). In my design, both motors vibrate to inform the cyclist of a general dangerous situation. Through this, I intend to keep the necessary cognitive effort small and to prevent confusions. Once one of the vehicles has disappeared, the signals warn about the remaining danger.

Additionally, as I do not want to restrict my design to the above situations, I have to consider that obstacles can suddenly appear in front of the cyclist. I decide to warn with both sides vibrating, if an obstacle appears straight ahead. This is in line with my decision for dangerous situations. Finally, the most complex case appears if vehicles are arriving from the left, the right, and in front of the bicycle. As above, I interpret it to be a general dangerous situation. Summarized that means:

- left handle of bicycle is vibrating: danger is approaching from the left
- right handle of bicycle is vibrating: danger is approaching from the right
- both handles of bicycle are vibrating: be aware of a general dangerous situation
  - from straight ahead
  - from the left and the right
  - from the left or the right or from straight ahead

Research on haptic signals, e.g., done in [7], gives an important hint that directional cues provided by one-side vibrations are an intuitive matter for the cyclist and can therefore be understood without much cognitive effort. However, the chosen two-side vibration may lead to confusion about the information conveyed as three different reasons cause the same vibration of both handles. Firstly, it may warn of a danger in front of the cyclist. Secondly, it could be a warning about the simultaneous arrival of vehicles from both sides. Thirdly, a combination of the first two cases called a general dangerous situation is possible. Simply based on the vibration, a user cannot decide which event is occurring. Additional information is needed to realize which situation is taking place. Consequently, the cyclist's cognitive effort is increased. In the worst case, this information might be misunderstood, especially if the danger is not visible and the cyclist's interpretation is wrong. For example, the signal might cause the cyclist to focus on one already visible danger of a vehicle in front of the bicycle, and therefore to oversee another vehicle approaching from the right.

To gain further insight, I propose to do more experiments about two-side vibrations in future work. Using additional (maybe auditory or visual) signals like recommended in [22] could help to solve confusions about the current meaning. Obviously, decreasing the possible warnings and simply focusing on dangers either from the left or the right could be a feasible solution. However, I discarded this restriction because it would extremely restrict the application of my HCWS. Consequently, I decided to include both-side vibrations with multiple meanings and to collect feedback to such an implementation, although human errors may be possible.

Finally, I thought about the vibration pattern. As already mentioned, the frequency of the installed vibration motors is 200 Hz and therefore in line with the research on haptic stimuli as discussed in Section 2.4 on pages 10 ff.. In general, a warning signal consists of a repetition of vibrations, its vibration patterns. To convey information about the distance of an obstacle, I implemented different vibration pattern representing different levels of danger, i.e., the time-gap between two vibration sequences differs. The nearer a cyclist comes to a danger, the shorter the gaps will become. I give more details in Section 3.2 on pages 19ff.. However, for my experiments, I decided against the use of different levels of danger and simply reduced the warnings to one warning pattern being the same for each danger. One reason was the short distance between the different intersections. Furthermore, by simply giving one warning, I intended to reduce the confusion and time cyclists need to become acquainted with the system, especially since I also changed the time after which a warning is given, i.e., the warning earliness.

For a first try, I decided for three repetitions of the chosen vibration sequence as also used by [7]. After a few preliminary tests and resulting complaints of the test participants about this being too much, I reduced the number to two repetitions. Each of the two vibrations lasts for 660 ms. After the first vibration, a short break of 300 ms takes place. Even though Matviienko et al. [7] came to the conclusion that shorter vibrations would be better, I decided for these longer ones as the vibration caused by the motors is a little less strong than it could be due to technical reasons explained in Section 3.2. Instead of one long vibration, a vibration consists of many short 10ms vibrations. In the future, further patterns could be easily included by extending the vibration control on the Raspberry Pi. Especially when the duration of a gap is changed to convey more information about the approaching danger, a repetition may be unnecessary as one warning consists of multiple vibrations.

3.4 Performance Measurements

To evaluate the performance of the implemented haptic signals, I measured the latency between sending a message in Veins and the actual start of a vibration on the bicycle. As my system is a safety critical system, only a low latency is acceptable. To measure the concrete time, I recorded a video that I used to evaluate the latency of a pixel change on the screen and a vibration sound. The change of a pixel on a screen is quite fast and easy to implement. Videos taken by a simple modern smartphone support a frame rate of about at least 30 frames per second (in my case only 30) and have a good synchrony of picture and sound. As a result, this technique can be used to evaluate a value for the latency easily and quickly. If the measured latency is too high, other tools must be applied to measure the latency between each component separately to find the source of the decreased delay. To avoid a skewing on the latency by the EVI, I reduced its update rate from 100 ms to 10 ms.

To calculate the latency, I took fifteen measurements and rounded up for the start of sound to avoid errors in measurement that would lead to a false positive result. I measured a mean value of 90.47 ms with a standard deviation of 7.94 ms. No single measurement had a latency higher than 100 ms. Considering the short amount of time the motors need to start, the result is a reasonable value. Additionally, since this value is quite small, further investigations of each interface would probably not result in a much decreased latency. Therefore, no further look into the individual components was needed.

Chapter 4

Simulation Scenarios

For my psychological study, I designed scenarios for a psychological evaluation of the developed HCWS. Thereby, I restricted the field of traffic situations I wanted to investigate.

First of all, accidents caused by negligence or with aforethought by a cyclist who willingly violates traffic rules are out of scope. My scenarios exclude these cases as they either may be contrary to an authentic road behavior of many cyclists or hard to predict. Nevertheless, I include the misconduct of car drivers to generate a typical realistic experience where cyclists can be overlooked by cars as in real life. Additionally, this allows investigating whether the cyclist can prevent a collision with the help of the haptic signal or not. As 53% of fatal crashes happen in urban areas [5], I focus on such scenarios. Furthermore, urban areas are much more feasible as intersections tend to be closer together. Therefore, the cyclist does not have to cycle big distances. Otherwise, the experiments would go beyond the scope of a short study. Additionally, in 97.6% of all bicycle-vehicle crashes, the vehicle involved is a car (excluding buses) [23]. Therefore, I restrict the current simulation scenarios to collisions between bicycles and cars, but an extension to collision between bicycles is possible.

The study [5] declares seven factors being important to prevent accidents. Firstly, the light condition and obstruction of view, and by that the visibility of other road users, could influence possible sensors. Since the warnings of my haptic signals do not rely on IVC via light, light conditions are not relevant for my simulation scenarios. As my HCWS bases on IVC, the same argument is valid for the cyclist's age that enables to draw conclusions about the cyclist's size, which may be important for detection algorithms. Furthermore, the authors took a look at the speed limit at scenes of accidents. Speed limits make it possible to estimate the average speed of the vehicle and compute how early a warning should be given. Even though this point is not taken into consideration for my approach, it is interesting for planning scenarios as I have to determine a vehicle speed. For some typical dangerous situations the study states that the speed of the involved car was on average quite slow, only about 24 km/h [16]. This result supports my decision to limit my simulation scenarios to urban scenarios and to set up a low speed. However, the bicycle simulator allows a cyclist to gain a high speed easier than in real life. Therefore, I set the speed of a car a little higher. Another factor described in [5] is that a future ADAS has to consider bad weather conditions leading to less fast braking, especially for emergency brake conditions. In my approach, I do not take a further look into this aspect, but it may be an idea for future work to change forewarning depending on the weather. For cars, an emergency brake assistance is a useful support to save lives in dangerous situations. However, for bikes their automatic usage could easily cause an uncontrollable event, especially as there exists no seatbelts on bicycles. As a result, further factors, for example braking behavior of drivers during or shortly prior to an accident, are not included in my work.

Summarized the implemented simulation scenarios are:

- between bicycles and cars
- in urban traffic
- with low car speed of about 32.4 km/h

To keep the scenarios as manageable as possible and the experiment at an appropriate length, I focus on intersections and ignore situations happening driveways to parking lots or properties. Future work may take a look at other accident cases and include more scenarios and scenes (e.g., entries, parking lots) to achieve further improvements. Furthermore, I had to decide about the traffic situations suitable to test the HCWS in simulation. In Section 2.3 on pages 8ff. three different scenarios were identified. All these scenarios depend on the perspective of the car, so I have to reinterpret them from the perspective of the cyclist. In 2016 these three scenarios accounted for 42% of bicycle-car accidents. Preventing them could avoid nearly half of these accidents. Therefore, I initially planned to test my haptic signals exactly in these cases. However, one of the identified scenarios (cf. Figure 2.4 on page 10) assumes that the cyclist is driving on the left side of the road. Therefore, I did not include it to avoid confusion during the experiments. Simulating this scenario would require cyclists to change sides of a street between intersections which may reduce their concentration. Furthermore, a change of side would make the scenario at the next intersection predictably after some repetitions. Moreover, in Germany, this means a violation of traffic rules in most cases. Therefore, I chose another

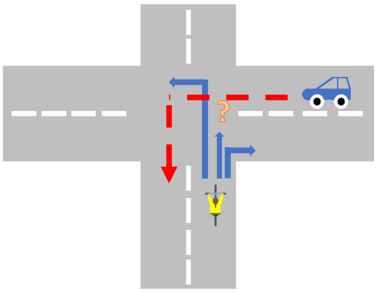


Figure 4.1 – Third scenario B1: car from the right is turning left, bicycle is coming from the left

scenario called B1 by [16] with a high likelihood for accidents. Figure 4.1 shows the chosen scenario. To add some more variety and make sure that a participant does not always expect a situation at each intersection, I included intersections where no other vehicle interferes with the driving path of the cyclist.

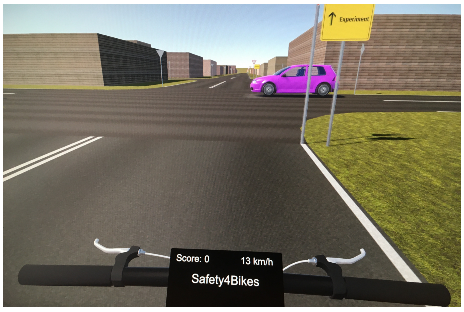


Figure 4.2 – Examplary situation in Unity 3D with car coming from the right

Summarized, one of the following situation randomly takes place if a cyclist arrives at an intersection:

- a car is coming from the left and driving straight ahead,
- a car is coming from the right and driving straight ahead,
- a car is coming from the right and turning left,
- or no vehicles approach the cyclist.

Figure 4.2 shows an exemplary scenario. The probability that a situation occurs depends on the construction of my experiment. Each scenario with cars can take place with and without vibrations, resulting in six possible combinations. If there is no car at all, there can never be a vibration. Therefore, seven combinations are possible. Nothing happens on (approximately) each seventh intersection, and one of the other three scenarios can be expected at each intersection with a probability of $\frac{2}{7}$.

Each scenario is triggered when the participant reaches a certain distance to the intersection. Since upcoming cars have to be registered and checked by the HCWS, cars must be spawned quite early. Therefore, it is not trivial to predict the participant's time of arrival at the critical intersection. Each time, a random number of cars between one and three cars is chosen. Each car is spawned with a probability of $\frac{1}{3}$, resulting in approximately two cars at each intersection. Multiple cars are spawned to compensate for changes in the participant's speed and thereby the cyclist's arrival at the intersection. As a result, the higher number of vehicles increases the probability for an interaction between the cyclist and a car. Furthermore, to increase the likelihood of interaction, the cars depart with a random offset that has an expected value of 6.5 seconds. This value in combination with a cyclist's driving speed of 16-21 km/h on average results in dangerous scenarios. Consequently, participants have to pay attention to more than one danger and the danger becomes less predictable. Predictability is a very important factor for the experiment as it can lead to uninterpretable results. If a participant can predict events at the intersection, it cannot be decided whether the implemented system helps preventing accidents, or whether the cyclist simply knew in advance what was about to happen.

My implementation of the simulation scenarios builds upon the work Stratmann did in his master thesis [24]. For example, cars are implemented to act according to priority signs at each intersection. Nevertheless, sometimes they ignore the priority of the cyclist, a problem already mentioned in [24]. For my experiments however, this is a benefit as in real life cars sometimes overlook cyclists, too. This introduces another aspect of surprise and makes situations at intersections less predictable for a participant. I implemented the city map as a grid demonstrated in Figure 4.3. For the scenarios and the visualization in Unity 3D, I was able to reuse a lot of code and ideas of Stratmann's work described in his master thesis [24]. Mostly, some parameters and names had to be changed and methods for the spawning of cars according to my scenarios had to be added. I also had to insert code for logging data needed later for the evaluation of the experiment. In my experiment, way-signs inform the participant whether to turn left, right, or to go straight ahead. Therefore, I had to generate them in the 3D visualization. The SUMO files necessary for this were automatically generated [24].

As mentioned in Section 3.2, I forward vibration messages to Unity to trigger a logging of events. The logging is needed to evaluate the participants' behavior during the experiments. To make that possible, I extended the class responsible for the connection to the EVI by implementing the recognizing of vibration messages.

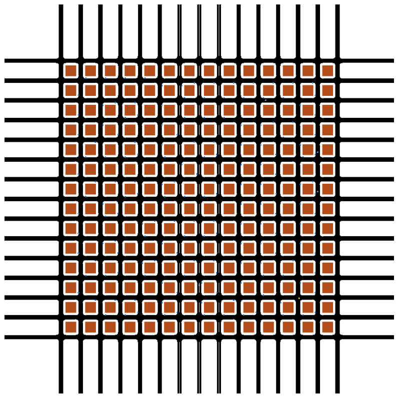


Figure 4.3 – Grid network as generated by SUMO. Blocks symbolize the houses, black lines the streets.

Whenever a vibration message is received, a log entry is written. To limit the amount of logged data, I log 7 times per second for 3 seconds altogether. Theoretically, 60 times per second would be possible. Since a human being needs some time to react and cannot change his or her speed each sixtieth of a second, such a frequency is unnecessary. For the actual logging, I was able to reuse and to extend some of Lukas Stratmanns implementation from his master thesis [8]. Each seventh second, the current time, the number of car crashes (up to then) and whether the signal was just received, is written in a CSV file. Furthermore, it is noted whether a log is the first one of a level to distinguish between levels with and without haptic signals. Additionally, participants lost 10 points on a score screen on the handlebar of the bicycle in the virtual environment if they collided with a car. For each block of the experiment a new logging file is created.

Chapter 5

Experiment

In this chapter, I describe the experimental study I set up to gain first insights into the usefulness of my implemented HCWS. I collected data from test participants when approaching an intersection where traffic is involved. Roughly speaking, I used the speed measured by the speed sensor of the physical bicycle to compare how they react on which time on dangerous situations with and without vibrations.

5.1 Method

The description of my experiment builds upon vocabulary commonly used in pyschological experiments [9] which is described in Table C.1 on page 85. For the experiment I installed two vibration motors on the physical bicycle. I used my implemented VHCE built on the VCE for the simulation environment (for more detail cf. Section 3.2). By using simulation as a laboratory experiment, I had total control of the surroundings and the experiment conditions. Unwanted confounders were minimized.

5.1.1 Participants

In total, 17 persons with unimpaired or corrected vision chose to participate in my experiment. Four of them were female, thirteen were male with their age ranging from 18 to 56 years which came to an average age of 24.9 years[12]. Except five people, all of them were Computer Science students at Paderborn University. All subjects participated voluntarily and without monetary salary. The experiment was signed off by the ethics committee of Paderborn University. The participants were recruited via the Distributed Embedded Systems group and the Internet. Furthermore, persons informed by buzz marketing contacted me by E-mail.

[12] with a standard deviation of 8.54 years leading to an asymmetrical skewness

5.1.2 Apparatus

As it can be seen in Figure 5.1, the setup of the experiment consisted of three slightly shifted monitors presenting the visual part of the VHCE as described in Chapter 3 on pages 16ff.. The monitor setup supports the immersiveness of the virtual environment by creating a wider field of view [25]. As a result, the participant gets an improved realistic overview of traffic situations by being able to look a bit to the left and to the right. Using monitors instead of 3D glasses reduces the risk of motion sickness during the experiment and eliminates additional cognitive effort for adjusting to them [24]. Using the same setup as Stratmann [24], the 24" 1920 × 1200 monitors were placed on a desk 1.5 m in front of the bicycle trainer. As before, they had a frame rate of 60 Hz. I installed two vibration motors on the physical bicycle. For the

Figure 5.1 – Experiment setup with start display

simulations environment I used my implemented VHCE as desribed above.

5.1.3 Design

I decided to use a within subject design with measurement repetitions. By having each test participant doing all parts of the experiment as a whole, especially with and without haptic, each person's change in behavior can be researched individually. Restrictions of a participants' health or mental feeling, i.e. their day-to-day performance, were as much as possible eliminated as they were nearly the same during the entire experiment. This helps to gain a more comparable impression about the benefits of haptic signals for collision warning and accident prevention. Moreover, the design allows to have a smaller number of participants of the experiment. For my experiments, I constructed eight different levels of simulation scenarios, each containing fifteen intersections. The eight levels were divided into two blocks composed of two levels with vibrations and two levels without vibrations. The vibrations were toggled on or off after each level. At the end of each level, or even during a level, it was possible to pause the simulation if a participant needed a break. After finishing the first block, there always was a short break. The whole experiment took about 45 minutes. At each intersection the participant was guided by way markers to follow a randomly generated route. Figure 5.2 gives an example. The way-signs

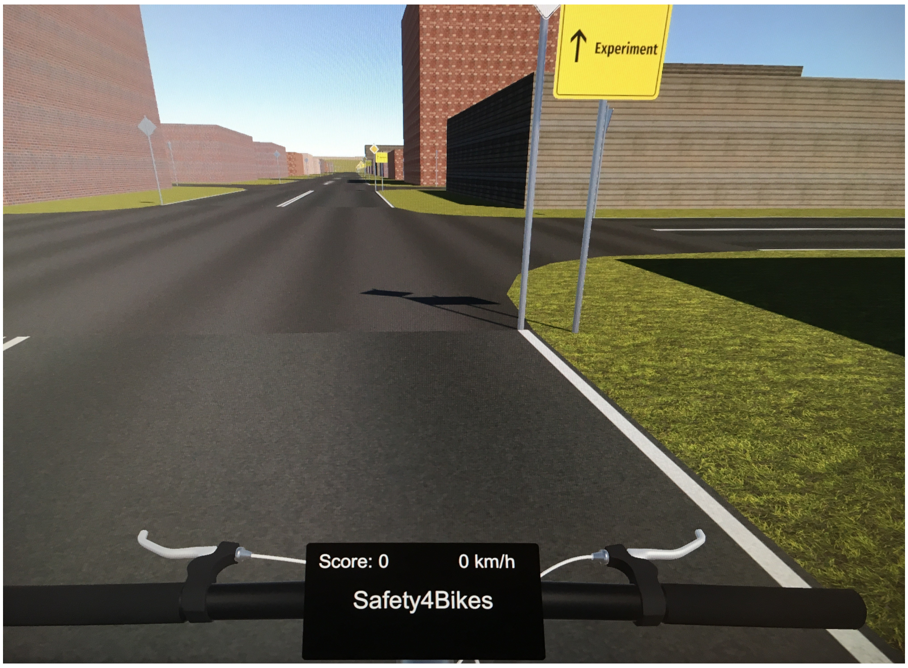


Figure 5.2 – Way-sign at an intersection in Unity3D

informed whether to turn left, right, or continue straight ahead. The participant could reach an empty intersection or face one of the three scenarios with cars as described in Chapter 4 on page 36.

To investigate whether the HCWS may help to prevent accidents, I chose a three factor design for my experiment. Figure 5.3 on page 44 sketches the dependencies between manipulated (green) variables and the measured variables (yellow) during the experiment. I decided to measure the *Speed* of the bicycle and the number of *Collisions* by manipulating three different test conditions. As I conducted a laboratory experiment, the stimuli of the environment or of the test participants themselves were limited as much as possible, and they were therefore not considered. The first independent variable *TrafficAtCrossroads* describes the different situations a partici-

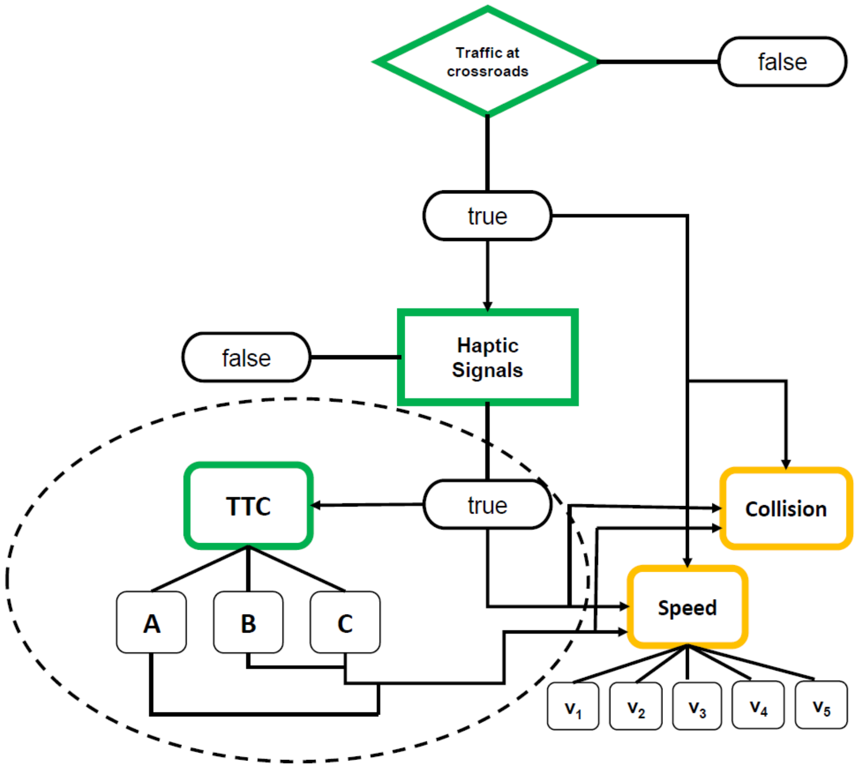


Figure 5.3 – Dependencies between dependent variables (Speed and Collision) (yellow) from the three independent variables (green) manipulated during experiment (with TTC being the "Time to Collision")

pant was confronted with at each intersection. Either the cyclist faced a constellation of cars or no traffic at all. Therefore *TrafficAtCrossroads* had two levels, true or false. Only, in case of traffic, the speed of the bicycle was measured and later analyzed. Furthermore, my study did not regard the effects on the cyclist being confronted with different traffic situations. The variety of traffic situations were only important to avoid predictability.

Secondly, half of the experiment was done with and half without haptic signals. This is described by my second independent variable *HapticSignals* of level two. If no vibrations were given, its value was false, and the measured speed and the number of collisions (mainly) depended only on the participant's reaction on the given traffic situation. Otherwise, I tried to analyze whether and how the vibrations effected the cyclists' speed and their number of collisions. Thirdly, I considered the Time To Collision (TTC) as a third independent variable of level three. During my experiment, I changed the point in time (A,B, or C) a warning was given before a possible collision. This variable also influenced the measured speed and number of collisions. In general, the TTC together with the given vibrations (*HapticSignal* true) manipulate both dependent variables. This is indicated by the broken oval in figure 5.3.

Whenever a collision warning was sent, data were logged for 3 seconds seven times a second. The recorded data include the current time, the cyclist's speed, which level the cyclist was in (to determine whether there was a haptic signal), and with how many cars he had collided up to that point. Based on these logged data, an investigation of the two dependent variables took place. It would also be possible to track the steering angle. However, this was beyond the scope of my thesis. I designed the dependent variable *Speed* to have five different levels each depending on the cyclist's reaction: stopping, accelerating, slowing down, doing nothing, or ambiguous. The other variable *collisions* measures the number of collisions up to a given moment in time.

5.1.4 Procedure

Before starting the experiment, each participant read and signed the declaration of consent. In a next step, the purpose of the haptic signals and the meanings of the different vibrations were explained. To get used to the feeling of my haptic signals, each possible vibration (both, left, right) was previously shown to the participant. After assuring the participants' understanding of the signal, the experiment structure was explained. Each participant was told to move like in real traffic and to react naturally. Moreover, they were told to avoid crashes with cars and follow the way-signs.

As described by Stratmann [24], I divided my experiment into two blocks to assure the concentration of the participants by allowing them to take short breaks. Furthermore, the first block has a tutorial at the beginning. The tutorial consists of six intersections without any cars to give a participant the opportunity to get familiarized with the simulation and the handling of the bicycle. By that, I hoped to reduce errors during the experiment caused by surprising behavior of the bicycle. Besides, the tutorial started with banners spanned across the street again informing the participant of his/her task. Before starting a new level, the participant was informed whether the next level would be with vibrations or without to avoid confusion. After finishing the physical part of the experiment, each participant was asked to fill out an online questionnaire. The whole experiment took about 45 minutes.

5.1.5 Questionnaire

To gain further knowledge about the participants' opinion of the signals, I designed a questionnaire the participants filled in after doing the experiments. The designed questionnaire consists of twelve questions and takes about five minutes to answer. In the questionnaire, six different topics are evaluated. This includes:

1. participant's information,
2. understandability,
3. acceptance,
4. distraction,
5. general questions,
6. and feedback.

Some of my topics and questions are chosen according to the interview questions done by Matviienko et al. [7]. They are shown in Table 5.1. However, as I could only refer to this information, I formulated all of the questions for my questionnaire on my own (for more detail, cf. questionnaire in Appendix C).

Section	Information
Named Topics	distraction, understandibility, participant's information
Scale	Likert Scale with 7 options
Named questions	"Which part(s) of the bicycle was (were) communicating?"(cf. [7], p. 15:5) "What do you think the bicycle was trying to 'say'?" (cf. [7] , p. 15:5)
Questions for Feedback	"[...] what they liked or disliked about the current implementation, any changes they would make [...]" (cf. [7], p. 15:5)
Questions on Acceptance and General Value	"[...] what they could imagine on their own bicycles, and the context and value of such signals" (cf. [7], p. 15:5)

Table 5.1 – Information given by Matviienko et al. [7] extracted from their verbal interviews

In my questionnaire, questions could either be answered by text or by choosing an answer on a Likert scale (cf. questionnaire in Appendix C, or Table 5.3). If the answer options were given in form of a Likert scale, participants had to choose among one of five options. By using an uneven number of options, it was possible to give neutral responses. I changed the order from most negative to most positive one time in between questions to insure the attention of a participant (cf. Question 6 and 7 of questionnaire in Appendix C, or Table 5.3 on page 55).

First of all, I collected personal data on the participants' age and gender. After that, the questionnaire was designed to evaluate the understandability of my haptic signals. During the first experiment of Matviienko et al. [7], participants were asked to stop and answer two questions whenever they received a signal. Since my procedure did not include a stop after each signal, I decided to include these questions in an adapted form under the topic "Understandability" into my questionnaire. The participants were requested to rate the understandability of the functionality, of the purpose of the haptic signal, and how distinguishable it was. In a third step, to evaluate the acceptance of my signal, I asked my participants, whether they would like such a system on their own bicycle. I added another question concerning their estimated acceptance of the system by their friends to gain further insight. In a fourth part, the participants had to estimate the distraction of the signal. Additionally, I inserted a question to rate how intuitive the signal was. The last two parts of my questionnaire contained questions regarding general information on the signal and a feedback. As I wanted to evaluate the benefit of haptic signals, I developed questions encouraging participants to think about the signal's value in dangerous situations and its helpfulness for preventing accidents. To allow further improvements, participants were also requested to rate the signal. Finally, they gave further feedback on the signal and their likes/dislikes of the implementation.

5.2 Results

Whenever a collision warning was sent by the HCWS, statistical and measured data were logged for three seconds with a frequency of seven times per second. Every data log started with the occurence of a haptic feedback, i.e., *HapticSignal* = true, and closes with the end of the last level. As described above, I collected the data of the protocol from two blocks each being composed of four levels, two with and two without vibrations. Based on the logged data, a first analysis of the dependent variables, namely the participants' speed, and number of collisions, took place. It would be also possible to track the steering angle. However, this was beyond the scope of this study. Furthermore, I evaluated a questionnaire to gain insight into the subjective opinions on my implemented haptic signals.

5.2.1 Preparation of Data

The logged data were stored as Comma-Separated Values (CSV) files. For further processing I converted them into Open Document Spreadsheets (ODS) which are comfortable for in-depth analysis with Microsoft® Excel®. In detail, the data were structured as follows:

- a column with the time stamps,
- a column with Boolean entries which indicate if a haptic signal was sent,
- speed (m/s),
- a counter for collisions so far, and
- a column with Boolean entries which indicate if the test drive changes from haptic feedback to no haptic feedback and vice versa (called level).

A first review of data showed that the log file obviously contained measurement errors, e.g., the bicycle accelerated by 6 km/h within $\frac{1}{7}$ second, and after another $\frac{1}{7}$ second it went down by nearly 6 km/h. Since this was unlikely with a bicycle, I decided to use a formula which auto-corrected these effects. I applied the following method:

$$\text{If } v_i - v_{i-1} > 3 \text{ km/h } \wedge \ v_i - v_{i+1} > 3 \text{ km/h , then } v_i = \frac{v_{i-1} + v_{i+1}}{2} \quad (5.1)$$

$$\text{with } v_i \text{ speed at time i}$$

If such a drastic aberration consisted of two consecutive values, a similar method was used. The used smoothing is a commonly known method (cf. [9], p.149).

After the preparation of received data for further analysis, I decided to focus first on the comparison of the difference between the participants' reactions with and without haptic signals. More precisely, in the first analysis, I did not differentiate between the different values of the TTC, but only distinguished between vibrations being activated and vibrations being turned off. The analysis of the influence of TTC was done in a second step. In my study I defined five different actions how a cyclist can react to a dangerous traffic situation. Therefore, the dependent variable *Speed* of my experiment had five levels representing them (cf. Figure 5.4):

- Nothing: The cyclist does not react within the three seconds to be analyzed. The speed within the measured range does not change significantly in the given range.
- Brake: The cyclist stops in the given time range. The speed goes down and becomes slower than 1 km/h (due to the simulator, if you brake, the speed goes down to near zero).
- Slow Down: The cyclist slows down. The speed goes down significantly but does not fulfill the rules for a brake.

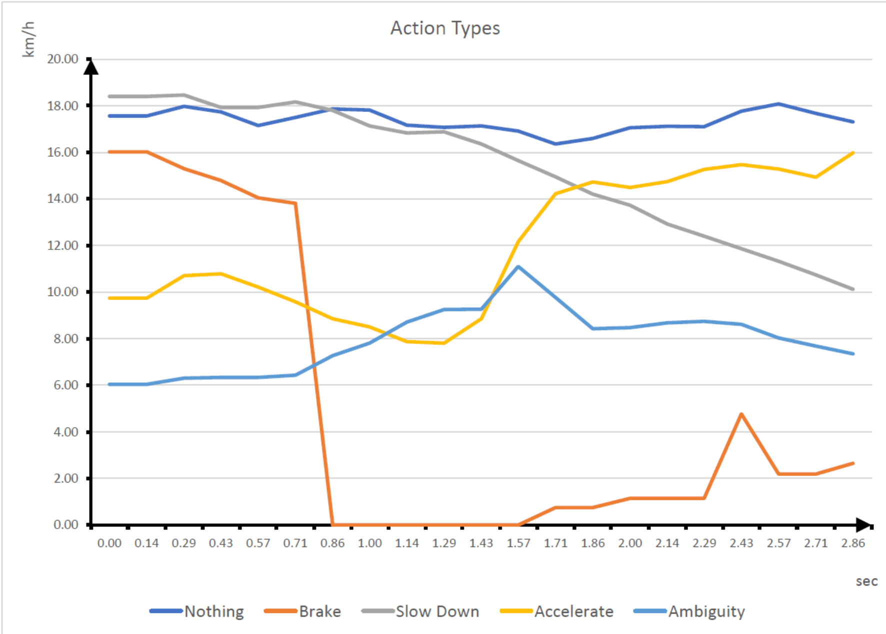


Figure 5.4 – Example for different action types describing how cyclists can react on dangerous situations, interpreted from data logged for three seconds with a frequency of seven times per second. The Y-Axis shows the measured speed (in km/h) of the bicycle and the X-Axis the time progress (in sec).

- Accelerate: The cyclist accelerates. The speed goes up significantly within the measured range.
- Ambiguity: The speed changes significantly but shows neither an acceleration nor a slow down.

To determine the cyclist's reaction within a given time interval, I had to define first my understanding of a reaction. I measured the (reaction) time if and only if the speed has changed between two measured points (with a difference of $\frac{1}{7}$ seconds) by a minimal value of two km/h. Otherwise, I considered it to be a simple fluctuation in speed. Only if I computed such a significant difference, I noted if the value was descending or ascending and continued my analysis. Finally, at the end of the data interval I determined which kind of action was executed and the corresponding reaction time (not for ambiguous or nothing condition). A reaction time of zero was decided to be a non realistic value and was hence excluded. Further refinement of possible actions was beyond the scope of my thesis.

For each cyclist taking part in the experiment, I evaluated the statistical data as shown in Table 5.2 on page 52 (or in the fact sheets being inserted into Appendix B). In detail, for each participant, I compute

- the number of test records (grand total of 34,514 records),
- the speed adjustment due to measurement errors (speed sensor is sensible to light fluctuation) (over all it was 1.71%),
- the number of events (caused by an initial haptic signal, grand total app. 1,494 events, one event may contain more than one haptic signal, e.g., a second warning),
- the number of collisions (grand total 25),
- and the different reactions (participant brakes, slows down, accelerates, does nothing, or reacts ambiguous).

Afterwards, I computed the mean reaction time of a participant for all given events (warnings) depending on the action which took place. I distinguished between time for reactions with haptic signals or without haptic signals. Without haptic signals meant, that no haptic signal was sent to the participant of the test although a vehicle appeared at the intersection. The test participants had to react on their own without any haptic support. Thereby, I was able to analyze their reaction times without haptic. As most data was evenly distributed, I also computed the standard deviation. I applied the common formula for the standard deviation (with A representing the action type):

$$s^2(A) = \sum_{i=1}^{n} \frac{(v_i - M)^2}{n} \tag{5.2}$$

$$SD = \sqrt{s^2(A)} \tag{5.3}$$

If the minimal reaction time of a single event was zero, I decided to count it as ambiguous, as it could be neither a reaction on the signal nor the detection of a vehicle. Summarized, for each action done by the participant I reported the following numeric data structured into with and without haptic:

- minimal reaction time (during the whole experiment);
- maximal reaction time (during the whole experiment);
- M (mean value) (of the whole experiment);
- SD (standard deviation) (of the whole experiment)

Most of the time, comparisons of minimal or maximal reaction times between haptic and no haptic were useless, however not always. Therefore, I noted them.

Finally, I reported an overall sheet for all participants together I called *Master Sheet* (cf. Table 5.2 on page 52). After investigating the data of each participant, I came to the conclusion that the calculation of mean value and standard deviation for all participants together was useful. Hence, *M* and *SD* were also computed over all calculated values. To get precise results, I worked individually with each record of all participants of the experiment. As the number of data per participant varied, I calculated the values over all single data. That led to much more precise results without giving more weight to some data than to others. Summarized, I calculated:

$$M_{\text{all}} = \frac{\sum\limits_{i=1}^{n} \sum\limits_{j=1}^{m_{c_i}} v_{ij}}{\sum\limits_{i=1}^{n} \# c_i} \tag{5.4}$$

where n is the number of participants of the experiment and m_{c_i} the number of relevant values in a certain category, e.g., subject brakes. v_i is the speed in a certain category for participant i. The similar method was applied to calculate *SD* in the Master Sheet.

Taking a further look at the overall results in Table 5.2, one will see that 48.53 % of all 1,494 logged actions based on 34,514 records (cf. Figure B.5 on page 80) were braking processes. Therefore, the data on braking are the most detailed ones and can be investigated with more reliability than the others. Only about 5.09 % of all reactions resulted in slowing down, in accelerating at least about 9.71 %. In about fourth of all cases, 23.02 %, the participants showed no reaction at all. 13.65 % of all collected reactions were ambiguous and could not be investigated in depth. Among the seventeen participants, the data of one participant could not be analysed due to a recording error. The speed adjustment done before analysis due to measurement errors was 1.71 %, only (cf. Figure B.5 on 80). In total, 25 accidents took place during the experiment, 8 with and 17 without haptic.

When computing the absolute mean values (according to the different reaction times) over all collected single data together, haptic in general seemed to improve the reaction time for all different action types (cf. Table 5.2). For example, the absolute mean time measured for braking $M_{\text{allbraking}}$ is 1.2 seconds with haptic and 1.48 seconds without haptic. Focusing on the single sheets for each participant, the absolute values measured for braking with haptic support were better for 12 participants. For comparing the absolute values, only differences of a minimum of 100 ms were counted. Slowing down was better for six out of seven persons, the other 10 participants had no values for with or without haptic so a comparison was not possible. However, for a more differentiated consideration, I had to examine the connections between the mean values and standard deviations (SD error bars) of

	Total	with haptic	w/o haptic
no. of events (total)	1,494	781	713
no. of collision	25	8	17
subject brakes	725	412	313
subject slows down	76	48	28
subject accelerates	145	74	71
subject does nothing	344	157	187
subject ambiguous	204	90	114

	no. of collision	**with haptic** min time	max time	M	SD
no. of collisions	8				
subject brakes	5	0.15	4.35	1.20	0.66
subject slows down	0	0.15	5.85	1.28	0.68
subject accelerates	1	0.15	4.65	0.95	0.72
subject does nothing	2				

	no. of collision	**w/o haptic** min time	max time	M	SD
no. of collisions	17				
subject brakes	5	0.15	4.2	1.48	0.78
subject slows down	2	0.30	4.65	1.55	1.00
subject accelerates	1	0.15	4.65	1.33	0.95
subject does nothing	9				

Table 5.2 – Fact Master Sheet for all 16 participants together (data of one participant were invalid). The first table lists up some statistical data. The second and third table show the computed values with and without haptic. Time is measured by seconds. For “nothing actions” no time is noted.

each of the 16 participants (cf. Figure A.1, Figure A.2 and Figure A.3 on pages 71 to 73). As the standard deviations are an important factor of uncertainty, I had to regard the relationship between the associated SD_{Haptic} and SD_{noHaptic} for interpreting whether the difference of the two mean values M_{Haptic} and M_{noHaptic} was of practical importance. This is done in the discussion.

Additionally, I evaluated the second independent variable, the changing *TTC*, which defines the earliest moment in time a warning signal is sent before a possible collision. I implemented three different values of warning *earliness*, 3250 ms before collision

(C), 3500 ms before collision (B), and 4250 ms (A) before collision [13]. For each participant who got a haptic warning I reported the given warning distance A, B, or C. For each distance, I structured the data into the different reactions and computed the mean reaction time and standard deviation. In a first step, these computations were done for six participants in total randomly selected. Due to the low amount of data on slowing down and accelerating, no comparable data could be computed

[13] In the beginning, I wanted to use values of 250 ms, 500 ms, and 1250 ms. As the cars can be seen quite early and the participants may need some time to adjust to the system, I decided to add an offset of 3000 ms.

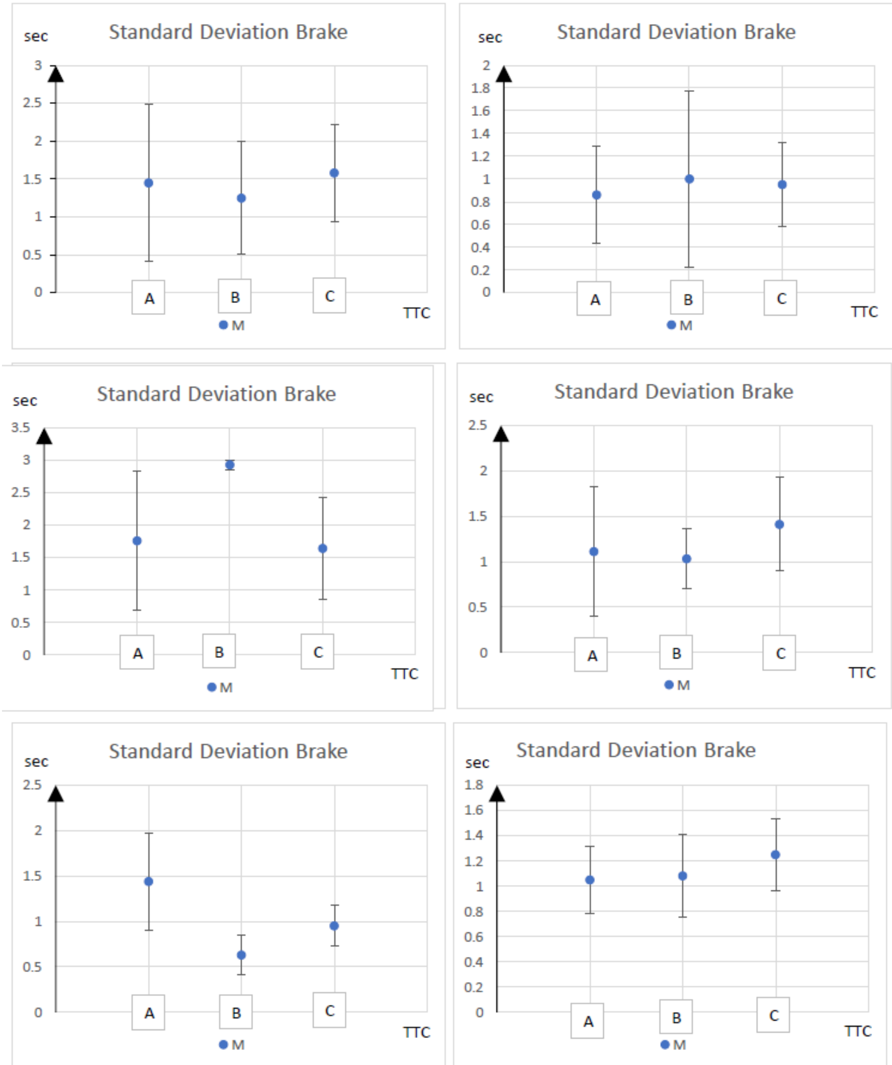


Figure 5.5 – SD error bars of six participants when sending haptic signals for different values of TTC (A, B, C), M is the mean value, time is measured by seconds.

for these action types. For braking, more data were available. Figure 5.5 on page 53 describes the SD error bar of the six participants for braking divided into the different TTCs. As it can be seen, these data do not show a general tendency.

5.2.2 Results of Questionnaire

After having conducted the experiment, I evaluated the collected information of seventeen filled out questionnaires. In Table 5.3, mean value *M* and standard deviation *SD* are visualized for questions answered by Likert scale values. In most cases, the mean value is in favor of my HCWS or at least neutral. Most of the time, it fluctuates between "neutral" and the next positive answer option. One exception is the helpfulness of the signal during the experiment. It was rated approximately "neutral". Furthermore, the mean value of the signal's timing rates between "a little too late" and "neutral". The first question on the understandability of the functionality and purpose of the signal has an asymmetric skewed towards the right side.

Additionally, 15 out of 17 participants gave feedback in textual form. Fourteen people recommended changes for improving the signal. Three people would like a shorter duration of vibration and to reduce the number of repetitions. Seven people requested a higher strength of vibrations to improve the distinguishability, especially since the handlebar sometimes transferred the vibrations to the other side. About half of the people (6) criticized the timing of the signal. As they discovered the cars quite early, the signal tended to be too late for assistance. One person asked to turn the signal off when the cars were approaching from the direction he/she was turning to (especially right when there is no danger of collision). Another person asked for the signal not to appear under a given speed.

Furthermore, 14 people gave feedback on what they liked about the implementation. Out of 14 answers, five praised the driving behavior of the bicycle and the intuitive design of the simulation. Four people mentioned that the vibration pattern was very clear and the direction detection worked flawlessly. Three participants liked the realistic behavior of cars which sometimes ignored traffic rules. One comment approved of the design not forcing people to look down at some screen. Two participants liked that the experiment included a real bicycle in a virtual environment. Furthermore, the immersiveness and the provided overview of situations were positively mentioned once. Someone else mentioned the increased awareness of road signs, but also that it took time to get used to the system (like in real life to an ADAS).

When asked about their dislikes of the implementation, opposite to the opinion above, four out of fifteen answers criticized the behavior of cars which tended to

Question	0 5	*M*	*SD*
1. How understandable was the function and purpose of the haptic signal to you? compl. incomprehensible ... compl. understand.		4.47	0.80
2. How easily distinguishable were the signals that the bike was giving? very difficult ... very easy		3.47	1.07
3. Would you like to have such a system on your on bicycle? not at all ... absolutely		3.29	1.05
4. What do you think, would your friends like to have such a system on their bicycle? not at all ... absolutely		3.18	0.73
5. I estimate the haptic signals as: very distractive ... not distractive at all		3.94	0.77
6. I estimate the haptic signals as: very intuitive ... not intuitive at all		1.82	0.81
7. How would you rate the timing of the signal? too late ... too early		2.53	0.8
8. Did those signals help you in identifying dangerous situations? not at all ... absolutely		2.71	1.05
9. Do you think these signals could help people to prevent accidents in real life? not at all ... absolutely		3.88	0.86

Table 5.3 – Responses to questionnaire with Likert Scale from 0 to 5

ignore priority. A further dislike by four people was the simplicity of situations. They requested more cars and higher traffic. Besides, other distractions like playing children and pedestrians could be included. Two participants would prefer the vibration stopping once the danger was over. One person disliked the simple design of houses. Someone else mentioned that the cars spawned too late. Another participant pointed out, that he/she had problems to differentiate between vibrations in the beginning. Furthermore, the early visibility of cars was leading to a reduced need for signals (1). Another participant criticized the design of way-signs as he/she tended to get inattentive and wished for more "fun made-up directions to spice them up a little". One comment mentioned a participant's problem to keep a certain speed to reach intersections at the right moment.

Additionally, some participants (4) mentioned verbally that the signals improved their awareness of traffic and way-signs as they were able to rely on the system for approaching danger. Five of them said they disliked the regular change of warning earliness as they found it confusing.

5.3 Discussion

To discuss the results, I first describe criteria for comparing the reaction time with and without signal support. Based on this, I draw some conclusions relating to the usefulness of my implemented haptic signals.

5.3.1 Analysis of Reaction Time

To draw conclusions about the influence of haptic, all reactions of an action type caused with haptic feedback had to be compared to their counterpart without haptic feedback. They were evaluated based on whether they are better, worse, or equal (i.e., no difference of practical importance) than the same reactions done without haptic signals. For deciding that, I had to decide whether the two mean values M_{Haptic} and $M_{noHaptic}$ were meaningfully different. Only in this case, I was able to call a mean value being "really" better or worse than the other one.

As inferential statistics was beyond the scope of my bachelor thesis, I decided to use a simplified procedure. Firstly, I computed the difference of the two mean values as shown in Equation (5.5). To decide whether the mean reaction time was meaningful better, equal, or worse, I calculated the mean value of SD_{Haptic} and $SD_{noHaptic}$ in Equation (5.6). Then I compared the difference of both mean value with 1/3 of the average of both deviations. This is done in Equation (5.7), Equation (5.8), and Equation (5.9).

$$|M_{Haptic} - M_{noHaptic}| = D \tag{5.5}$$

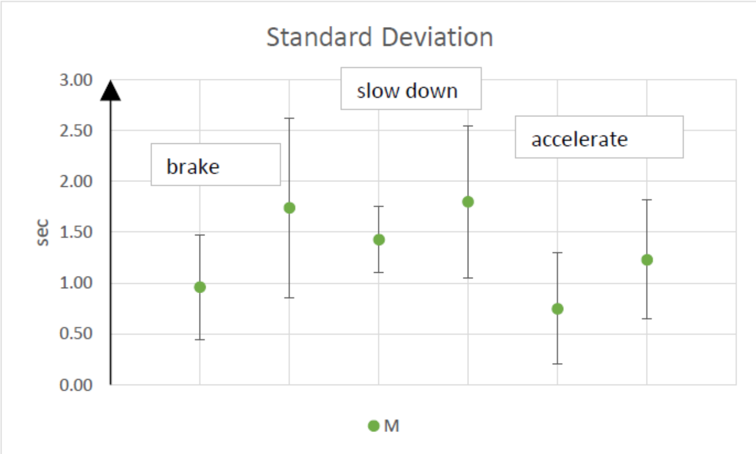
$$SD_{mean} = \frac{SD_{Haptic} + SD_{noHaptic}}{2} \tag{5.6}$$

$$(D \geq \frac{1}{3} \cdot SD_{mean}) \land (M_{Haptic} < M_{noHaptic}) \Rightarrow \text{better} \tag{5.7}$$

$$(D \geq \frac{1}{3} \cdot SD_{Haptic}) \land (M_{Haptic} > M_{noHaptic}) \Rightarrow \text{worse} \tag{5.8}$$

$$(D < \frac{1}{3} \cdot SD_{mean}) \Rightarrow \text{equal} \tag{5.9}$$

Comparing the two participants given in Figure 5.6 on page 57, including the computations above avoids drawing wrong conclusions. The upper participant P2 was always better on average with haptic signals than without haptic. This can be easily seen in Figure 5.6 (cf. Figures B.1 and B.2 on pages 76 and 77). Having a first look at participant P14, it seems for a moment as if he/she was better at accelerating without signals. However, the above computations show that this difference is probably not meaningful and therefore I consider it to be only equal to the reaction time measured for acceleration with haptic signals. In contrast, the difference for "slow down" is



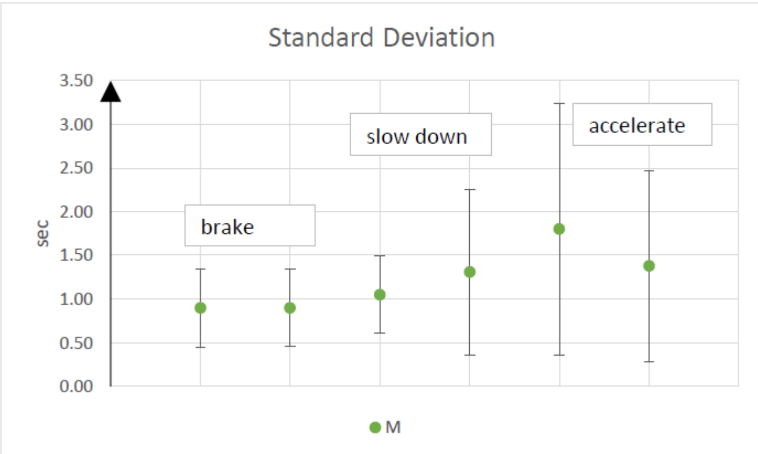


Figure 5.6 – *SD* error bars of participant P2 (upper graphic) and P14 (lower graphic) divided into braking, slowing down and accelerating, for each action the first SD error bar refers to *SDs* and *Ms* measured with haptic support, the second to *SDs* and *Ms* measured without haptic.

.

of practical importance (because of its smaller SD). In this case, haptic signals are better than reactions without haptic. The whole sheet of Participant 14 is shown in Figures B.3 and B.4 on pages 78 and 79.

Using the computations (cf. Equation (5.5) till Equation (5.9)) above, nine participants from a group of sixteen persons had a better reaction time with haptic signals for braking (in opposite to 12 when regarding only absolute values), six had an equal reaction time, and one participant was better without haptic support. This is described in Table 5.4 on page 58. Looking at slowing down, only five participants were better with haptic, one was equal, and two were worse. For the other

Comparison	Better	Equal	Worse
subject brakes	9	6	1
subject slows down	5	1	2
subject accelerates	7	3	4

Table 5.4 – Comparison of mean reaction times for all 16 participants with help of the above computations, better/equal/worse means that the mean reaction time with haptic support tends to be better/equal/worse for 9/6/1 participants, and so on.

participants, the needed minimum of data was not available. But, as mentioned above, the data on slowing down were quite limited. The same applied to accelerating, where seven participants were better, three equal and four worse with haptic.

Interpreting the different TTC using the computations described in Equation (5.5)

Comparison	Better	Equal	Worse
A vs. B	1	4	1
A vs. C	2	3	1
B vs. C	4	1	1

Table 5.5 – Comparison of different TTCs A, B, and C done with the above computations

until Equation (5.9), I draw the same conclusion as by comparison of the absolute mean values of A, B, and C. I did not find a meaningful general tendency (cf. also Figure A.4 on page 74). As described in Table 5.5 on page 58, the interpretation was inconsistent. On the one side, TTC B seemed to be better than TTC C, and TTC A a little bit better than C. But on the other hand, A was equal to B. This conclusion is confirmed by Figure 5.5 on page 53. Therefore, no further evaluations took place.

5.3.2 General Discussion

After my experiment, I investigated the collected data describing the participants' reactions to haptic signals. Since I restricted to 16 (17) test participants, I only gained a short insight on the signal's effects. That means, that the generalization of my results must be considered carefully. The results of my experiment highlight that the signal improves the reaction time of the majority of participants during the braking process. Furthermore, even if the number of accidents was low during the experiment, there were still more than twice as many accidents without the signal. In most cases, the signal seemed to improve the safety of the cyclist as it often reduced the reaction time. Possibly, a higher improvement of safety may be observed by increasing the simulated traffic and restricting the view on intersections. Some people reacted slower with vibrations during accelerating or slowing down. In these cases, one must remember that the number of records for these actions was very low and probably not representative. If a person has a low number of records for one special action, single outliers change the result a lot. All in all, the signal shows a promising tendency of increasing the reaction time in average about 280 ms during braking actions. Accelerating is improved even further, but due to the low amount of data this has to be considered carefully. The same applies to slowing down which seems to be equally good without haptic signals.

After analyzing the data for the different TTCs, it is still not possible to come to a conclusion about the different signal timings. This corresponds to the results in the questionnaire where participants were of different opinions on the timing. Some found it fitting, some too early, some too late. Verbally, people complained about the regular change after each fifteenth warning. Therefore, changing the experiment, e.g., the design of one experiment just with haptic and focusing one different TTCs may help to draw better conclusions. Participants of the experiment were mostly computer science students. Some of them showed a very reckless driving behavior during the simulation even though they were repeatedly told to behave like in real life. When talking to the participants after the experiment, some of them mentioned they felt like in a computer game and accidentally started to act accordingly. As a result, these people inclined to ignore the warnings and rushed through intersections. Maybe having a wider variation of participants would help to obtain more representative data. Of further interest are people accustomed to ADASs (for cars). One participant with daily contact to such systems seemed to rely much more on the signals and trusted them completely. Additionally, my group of participants consisted of far more male participants than female ones. The group should be more demographically diverse and consist of a higher number of participants for extended future studies.

Based on the questionnaire, the signals seem to be clearly recognizable. Participants of the study rate the signal as mostly intuitive and non distractive. Hence, there seems to be a tendency of the signal to support a cyclist in simulated traffic without increasing cognitive load. As a result, a cyclist can pay attention to the surrounding traffic without getting distracted by the designed signal. As criticized in the questionnaire, the intersections were quite wide and allowed a good overview of approaching cars. By that, dangers were visible early. Hence, some participants felt that the support did not really help to avoid accidents during the simulation. Nevertheless, they thought that the signals might prevent accidents in real life. Interesting was the fact that people were mostly neutral on having such a system on their own bicycle. Increased traffic may be useful to change the participants' perceived quality of the signals during experiments. Especially parking lots or house exits are also scenes of accident. Therefore, including them in future experiments may allow further insight into an optimal design for haptic signals. Additionally, reducing the field of view by adding obstacles like trees or including elements of surprise can make situations less predictable and increase the perceived support of the signals. Participants had diverging opinions on the behavior of the other vehicles: some liked their ignorance of traffic rules as they felt realistically, some felt their behavior too often to unpredictable. However, cars ignoring traffic rules also adds realism to the simulation. Many accidents in traffic are caused by inattentiveness. Even though people can theoretically see an approaching danger, they simply overlook it. Hence, including dull parts in a simulation can be used to reconstruct such situations.

Extending the tutorial by including some first intersections with traffic and increasing its length, may reduce the number of accidents during the actual experiment. As mentioned by one participant, the system needs some time to get adjusted to, like ADASs in cars in real life. Maybe conducting a longer experiment would increase the understandability of the signal.

Chapter 6

Conclusion

6.1 Summary

In my thesis, I developed a haptic collision warning system, my HCWS. The task of the designed system is to warn cyclists of possible dangerous traffic situations. Hence, my system does not only warn of vehicles which currently are on collision course with the bicycle, but also of vehicles which are missing it by a short amount of time. It is adjustable how early the systems warns of possible collisions. In detail, I mounted a smartphone vibration motor on top of each handle of a physical bicycle. For controlling the motors, I implemented a control program on a Raspberry Pi and connected it to the rest of the VCE system. The program causes vibrations according to detection of my implemented collision warning algorithm in Veins. Whenever a warning is received, two vibration sequences are caused. The signals convey mainly directional cues informing about approaching danger. Additionally, I inform of general dangerous situations if vehicles are simultaneously approaching from more directions. It is possible to control each vibration individually for left and right vibrations or have both motors vibrating at the same time. Furthermore, the vibration patterns are variable to allow easy changes. To evaluate the HCWS, I designed test scenarios simulating typical dangerous situations in urban traffic. Furthermore, I implemented a suitable logging of data and developed a questionnaire. Next, a laboratory experiment in within subject design was conducted with 17 participants. During the experiment, the traffic situations, the TTC at which the earliest warning was sent, and whether haptic signals were given or not, were changed. Measurements included the number of car collisions and the reaction time. Thereby, I distinguished five different actions: braking, slowing down, accelerating, doing nothing, or ambiguous data.
Based on the questionnaire, the participants of my experiment rated the developed HCWS as understandable, intuitive and non distractive. For the simulation, however,

most of them requested more complex scenarios, but expected an improvement of safety in real life traffic. This corresponds to the participants' comments that the simulated cars were visibly too early and reduced the need for additional warnings. The logged data highlight that the signal improved the reaction time of the majority of test cyclists. Additionally, the number of collisions with other were more than halved when the HCWS was activated. Summarized, the system improved the safety and reduced the reaction time of most participants of my experimental study.

6.2 Future Work

In future work, changes in the scenarios may help to improve measured data. Reducing the view on intersections could increase the perceived need for warnings. Expanding the surrounding traffic, or adding more situations and details may improve the realism of the simulation. To support further research on different warning earliness, an experiment only focusing on changing timing for haptic signals could be conducted. In some point in the future, field tests might be included to further research the effect of the designed system in real life.

The developed HCWS could be improved by refining conditions in the detection algorithm and adding further computations. Excluding cases could limit the number of given warnings. Increasing the reliability of the system the HCWS is integrated in and reducing the number of system crashes would improve the external conditions for future experiments. Additionally, the interference of speed sensor and vibration motors should be investigated and solved. Further experiments could help to examine a more fitting duration of vibration. Improving the strength of vibrations without making them indistinguishable may help users to easier understand the conveyed information. Finally, I propose to do more experiments about two-side vibrations in future work.

List of Abbreviations

ADAS	Advanced Driver Assistance System
CSV	Comma-Separated Values
EVI	Ego Vehicle Interface
GPIO	General-Purpose Input/Output pin
HCWS	Haptic Collison Warning System
IVC	Inter-Vehicle Communication
ODS	Open Document Spreadsheets
PET	Post-Encroachment-Time
SUMO	Simulator of Urban MObility
TraCI	Traffic Control Interface
TTC	Time To Collision
VANET	Vehicular Ad-hoc NETworks
VCE	Virtual Cycling Environment
Veins	VEhIcular NetworkS
VHCE	Virtual Haptic Cycling Environment

List of Figures

List of Tables

Bibliography

[1] J. Van Brummelen, B. Emran, K. Yesilcimen, and H. Najjaran, “Reliable and Low-Cost Cyclist Collision Warning System for Safer Commute on Urban Roads,” in *IEEE International Conference on Systems, Man, and Cybernetics (SMC 2016)*, Budapest, Hungary: IEEE, Oct. 2016. DOI: 10.1109/SMC.2016.7844814.

[2] C. Spence and C. Ho, “Multisensory Warning Signals for Event Perception and Safe Driving,” *Theoretical Issues in Ergonomics Science*, vol. 9, no. 6, pp. 523–554, Oct. 2008. DOI: 10.1080/14639220701816765.

[3] F. Meng and C. Spence, “Tactile Warning Signals for In-Vehicle Systems,” *Accident Analysis & Prevention*, vol. 75, pp. 333–346, Feb. 2015. DOI: 10.1016/j.aap.2014.12.013.

[4] S. Detzer, M. Junghans, K. Kozempel, and H. Saul, “Analysis of Traffic Safety for Cyclists: the Automatic Detection of Critical Traffic Situations for Cyclists,” *WIT Transactions on The Built Environment*, vol. 138, pp. 491–502, May 2014. DOI: 10.2495/UT140411.

[5] N. Chaurand and P. Delhomme, “Cyclists and Drivers in Road Interactions: A Comparison of Perceived Crash Risk,” *Accident Analysis & Prevention*, vol. 50, pp. 1176–1184, Jan. 2013. DOI: 10.1016/j.aap.2012.09.005.

[6] S. Céspedes, J. Salamanca, A. Yañez, C. Rivera, and J. C. Sacanamboy, “Platoon-based Cyclists Cooperative System,” in *7th IEEE Vehicular Networking Conference (VNC 2015)*, Kyoto, Japan, Dec. 2015, pp. 112–118. DOI: 10.1109/VNC.2015.7385556.

[7] A. Matviienko, S. Ananthanarayan, S. S. Borojeni, Y. Feld, W. Heuten, and S. Boll, “Augmenting Bicycles and Helmets with Multimodal Warnings for Children,” in *MobileHCI ’– Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2018)*, Barcelona, Spain: ACM New York, Sep. 2018. DOI: 10.1145/3229434.3229479.

[8] L. Stratmann, D. S. Buse, J. Heinovski, F. Klingler, C. Sommer, J. Tünnermann, I. Scharlau, and F. Dressler, "Psychological Feasibility of a Virtual Cycling Environment for Human-in-the-Loop Experiments," in *Jahrestagung der Gesellschaft für Informatik (INFORMATIK 2019), 1st Workshop on ICT based Collision Avoidance for VRUs (ICT4VRU 2019)*, to appear, Kassel, Germany, Sep. 2019.

[9] S. Reiß and V. Sarris, Experimentelle Psychologie - Von der Theorie zur Praxis. Pearson Deutschland GmbH, Apr. 2012.

[10] J. Heinovski, L. Stratmann, D. S. Buse, F. Klingler, M. Franke, M.-C. H. Oczko, C. Sommer, I. Scharlau, and F. Dressler, "Modeling Cycling Behavior to Improve Bicyclists' Safety at Intersections – A Networking Perspective," in *20th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2019)*, Washington, DC: IEEE, Jun. 2019.

[11] D. S. Buse, C. Sommer, and F. Dressler, "Integrating a Driving Simulator with City-Scale VANET Simulation for the Development of Next Generation ADAS Systems," in *37th IEEE Conference on Computer Communications (INFOCOM 2018), Demo Session*, Honolulu, HI, USA, Apr. 2018. DOI: 10.1109/INFCOMW.2018.8406997.

[12] D. S. Buse, M. Schettler, N. Kothe, P. Reinold, C. Sommer, and F. Dressler, "Bridging Worlds: Integrating Hardware-in-the-Loop Testing with Large-Scale VANET Simulation," in *14th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2018)*, Isola 2000, France: IEEE, Feb. 2018, pp. 33–36. DOI: 10.23919/WONS.2018.8311659.

[13] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, Jan. 2011. DOI: 10.1109/TMC.2010.133.

[14] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent Development and Applications of SUMO - Simulation of Urban MObility," *IARIA International Journal On Advances in Systems and Measurements*, vol. 5, no. 3 4, pp. 128–138, Dec. 2012.

[15] A. Varga, Modeling and Tools for Network Simulation, K. Wehrle, M. Güneş, and J. Gross, Eds. Springer-Verlag, Berlin, Heidelberg, 2010, ch. OMNeT++, pp. 35–59. DOI: 10.1007/978-3-642-12331-3_3.

[16] M. Kuehn, T. Hummel, and A. Lang, "Cyclist-Car Accidents - Their Consequences for Cyclists and Typical Accident Scenarios," in *Proceedings of the 24th International Technical Conference on the Enhanced Safety of Vehicles (ESV*

2015): Traffic Safety Through Integrated Technologies, Gothenburg , Sweden, Jun. 2015.

[17] E. B. Goldstein, Sensation and Perception, Eight Edition. Wadsworth Cengage Learning, 2010.

[18] ——, Wahrnehmungspsychologie, 4th ed. Spektrum Akademischer Verlag, 1996.

[19] B. Poppinga, M. Pielot, and S. Boll, "Tacticycle: a Tactile Display for Supporting Tourists on a Bicycle Trip," in *Proceedings of the 11th Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI 2009)*, Bonn, Germany, Sep. 2009. DOI: 10.1145/1613858.1613911.

[20] H. Steltenpohl and A. Bouwer, "Vibrobelt: Tactile Navigation Support for Cyclists," in *Proceedings of the 2013 international conference on Intelligent user interfaces (IUI 2013)*, Mar. 2013, pp. 417–426. DOI: 10.1145/2449396.2449450.

[21] M. Camila Escobar Alarcon and F. Ferrise, "Design of a Wearable Haptic Navigation Tool for Cyclists," in *Proceedings of the International Conference on Innovative Design and Manufacturing (ICIDM 2017)*, Milan, Italy, Jul. 2017.

[22] D. A. Norman, The Design of Everyday Things. Basic Books, 2013, pp. 177–179.

[23] A. MacAlister and D. S. Zuby, "Cyclist Crash Scenarios and Factors Relevant to the Design of Cyclist Detection Systems," *Proceedings of the International Research Council on the Biomechanics of Impact (IRCOBI 2015)*, vol. 50, pp. 373–384, 2015.

[24] L. Stratmann, "Measuring Cyclists' Visual Attention and Interaction with an Interactive Real-Time 3D Simulation," Master's thesis, Paderborn University, Institute of Computer Science, May 2019. DOI: 10.17619/UNIPB/1-761.

[25] James J. Cummings and Jeremy N. Bailenson, "How Immersive Is Enough? A Meta-Analysis of the Effect of Immersive Technology on User Presence," *Media Psychology*, vol. 19, no. 2, pp. 272–309, May 2016. DOI: 10.1080/15213269.2015.1015740.

Appendix A

Standard Deviations

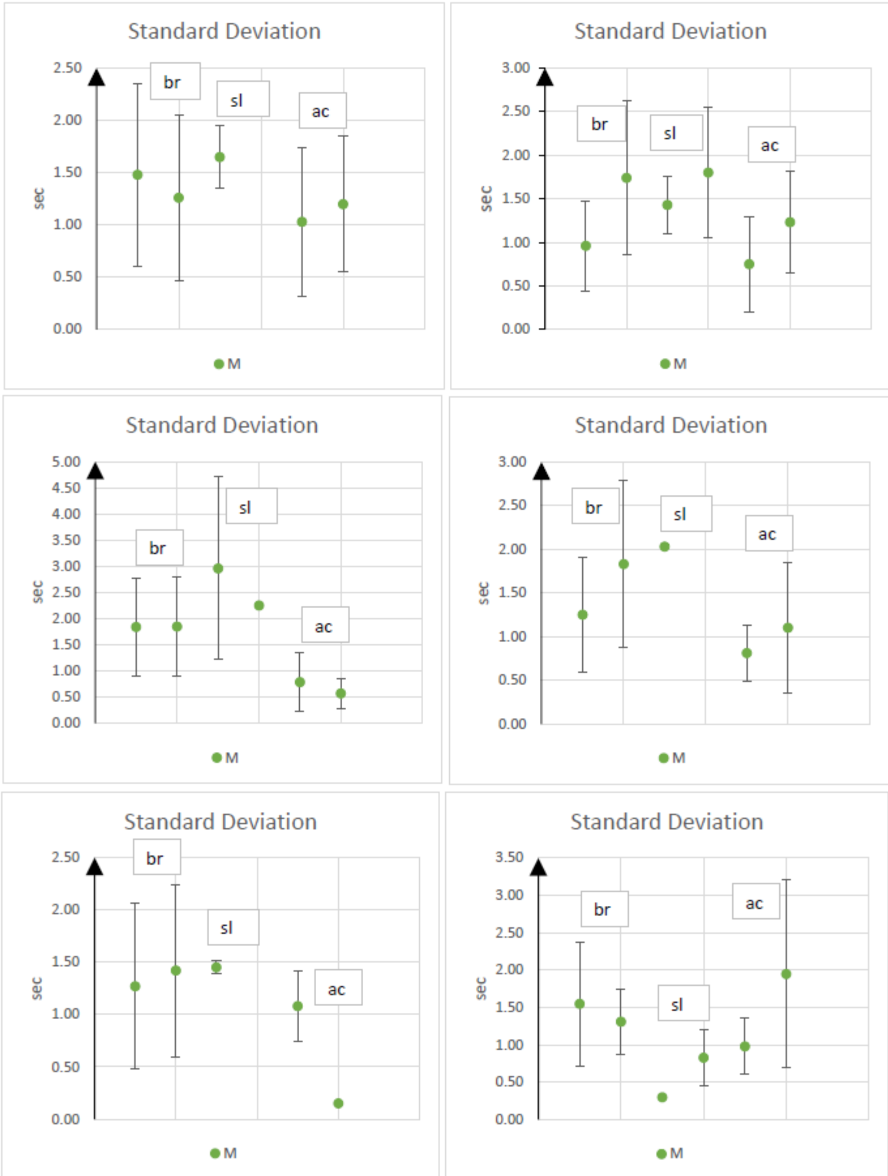


Figure A.1 – SD error bars for the first 6 participants: P1 until P7 (data of P6 were invalid). The values are structured by actions brake (br), slow down (sl), and accelerate (ac). For each action the values of the first error bar are always calculated with measurements with haptic support, the second one without signal. The mean value is the green point, and the SD is drawn as a black error bar around the mean value. Sometimes an error bar is missing as there were not enough data available.

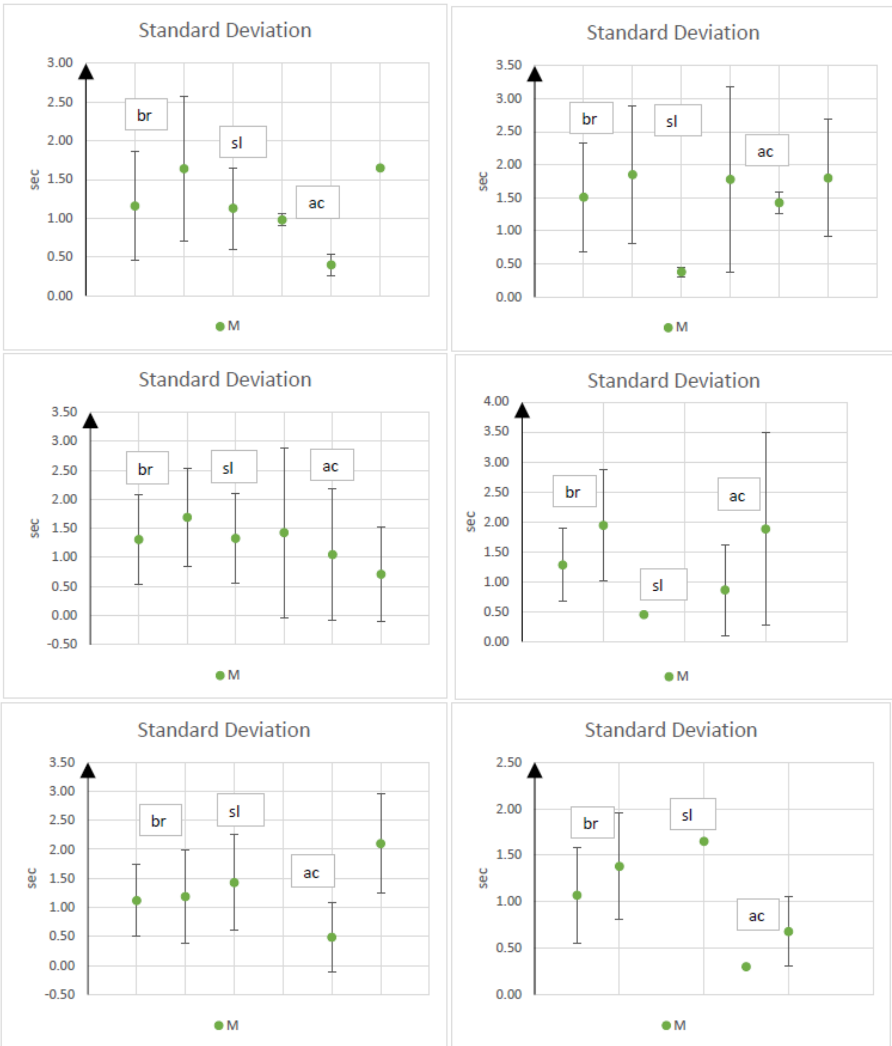


Figure A.2 – SD error bars for P8 to P13, representation of SD and M as described in Figure A.1

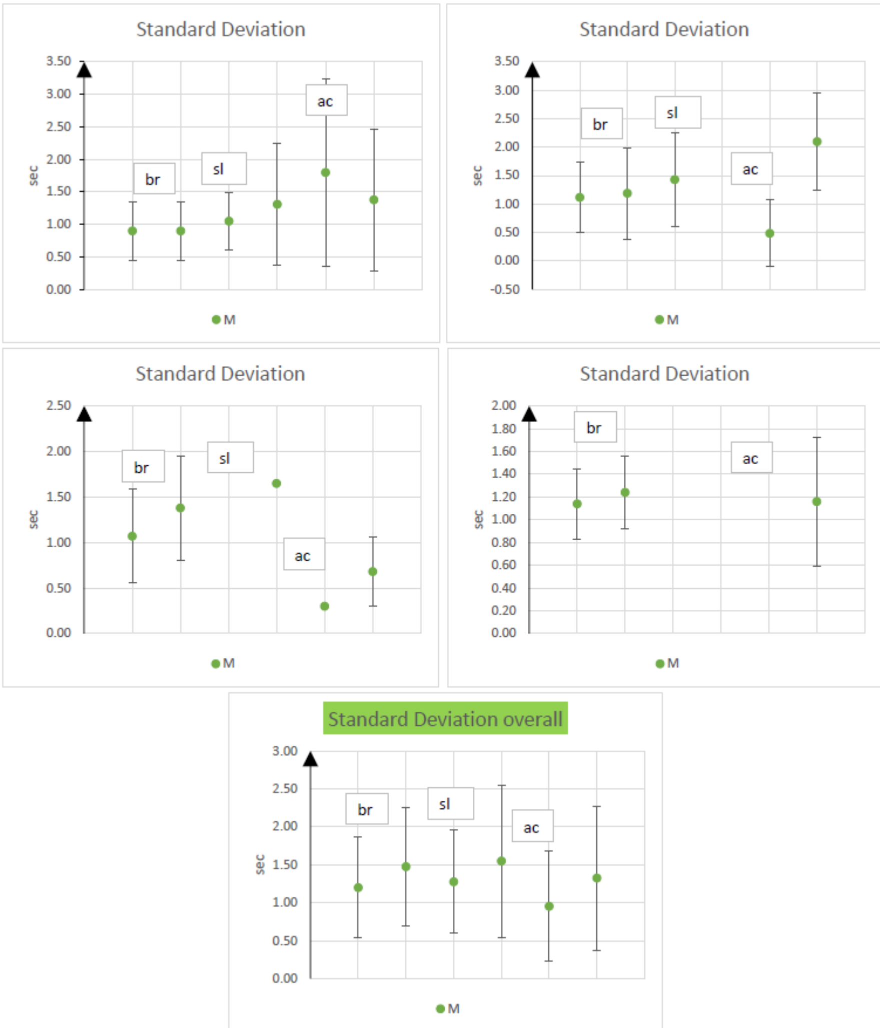


Figure A.3 – SD error bars for P14 to P17 and overall 16 participants (SD and M of the Master Sheet itself), representation of SD and M as described in Figure A.1

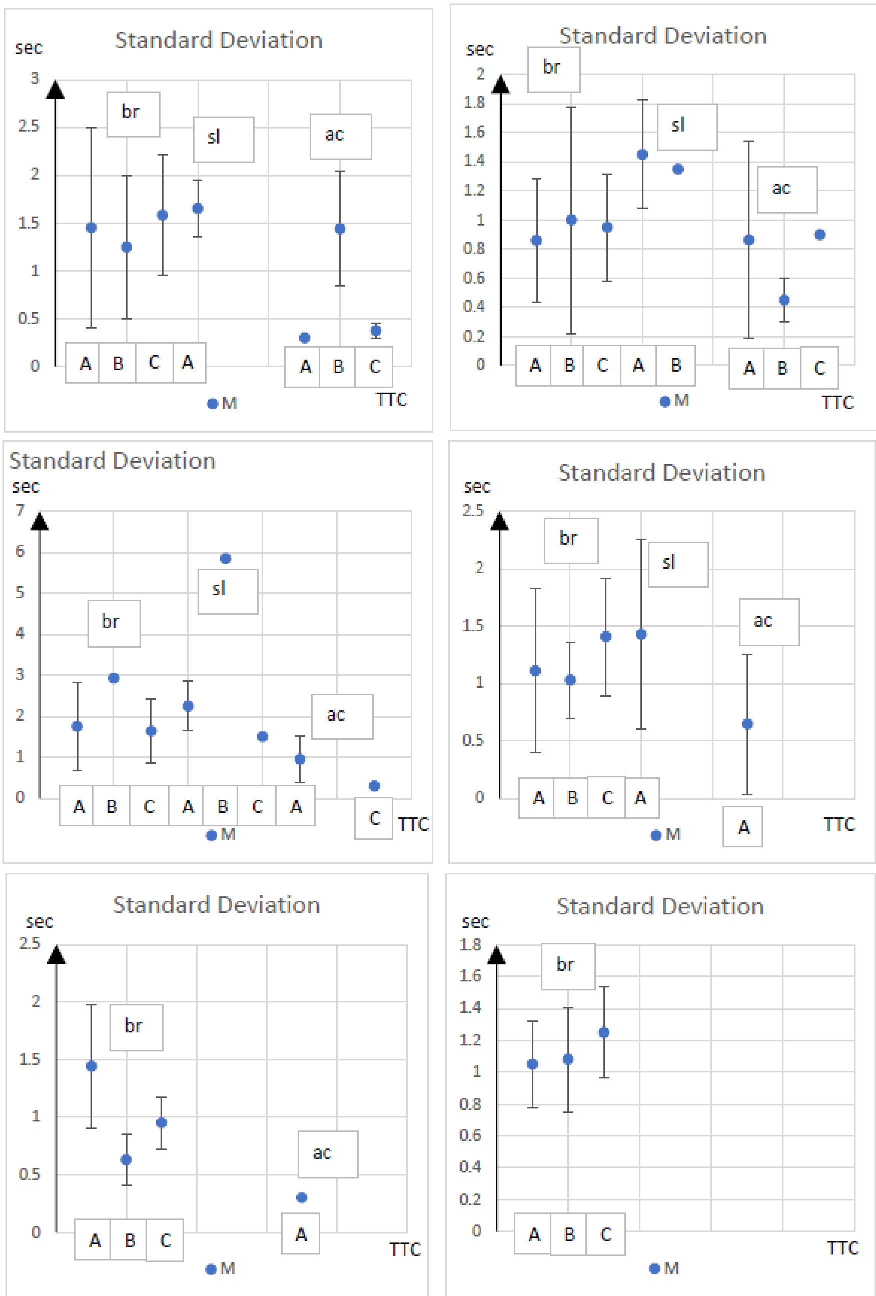
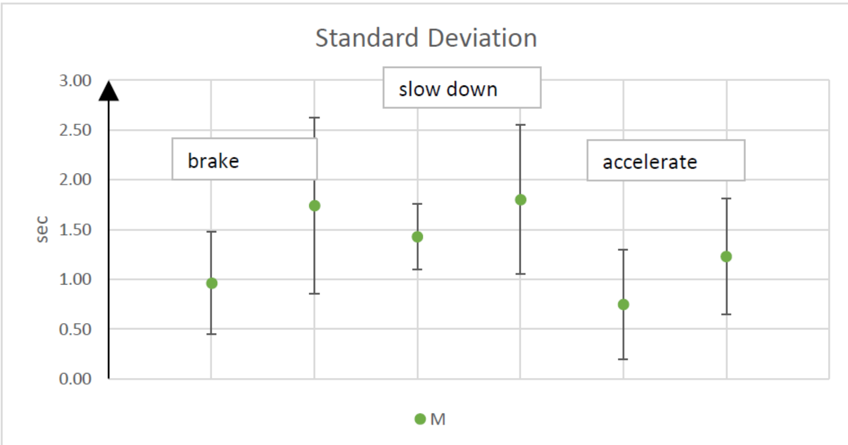


Figure A.4 – SD and M for Warning Distances A, B and C of 6 participants, separated into all three actions, SD and M as described in Figure A.1. SD error bars are missing if not enough data were available.

Appendix B

Fact Sheets

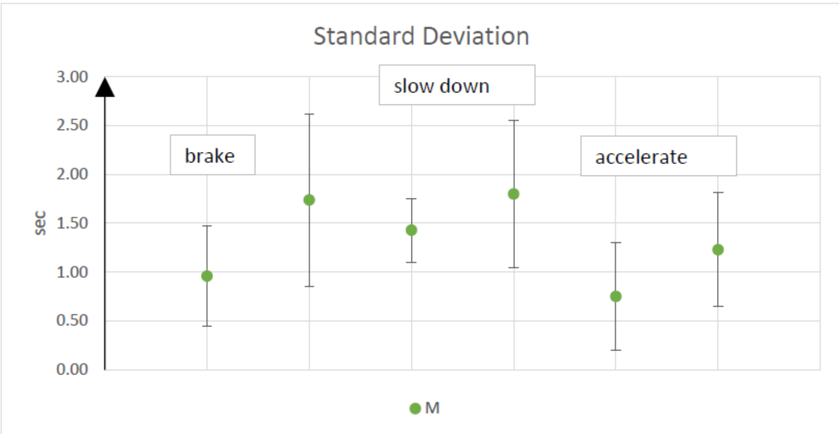
Subject: 2
speed adjusted 1.19%
records: 2,260



		with haptic mean	
	better	**equal**	**worse**
subject brakes	1	0	0
subject slows down	1	0	0
subject accelerates	1	0	0

Figure B.1 – SD, M and comparison table of Participant 2, SD and M as described in Figure A.1. If the mean values are compared with regard to their standard deviations and the compution of 5.3, the table indicates that Participant2 reacts faster for all three actions when getting haptic warnings.

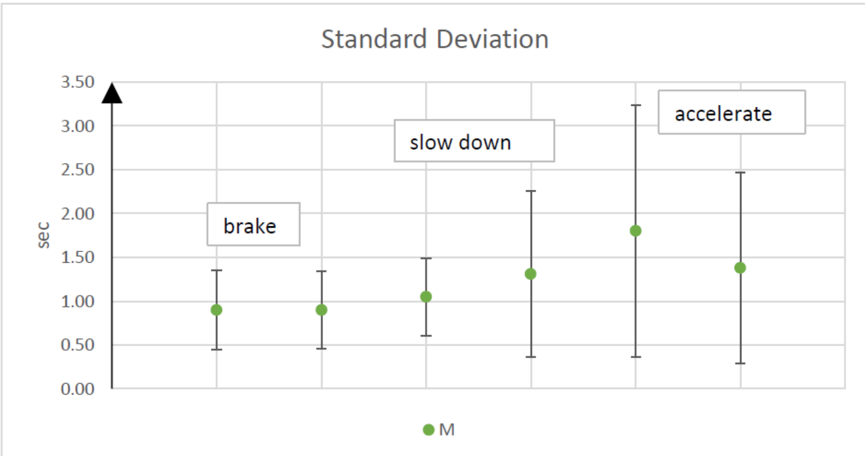
Subject:	2
speed adjusted	1.19%
records:	2,260



	with haptic mean		
	better	**equal**	**worse**
subject brakes	1	0	0
subject slows down	1	0	0
subject accelerates	1	0	0

Figure B.2 – Fact Sheet for Participant 2, containing all numeric and statistical data

Subject:	14
speed adjusted	2.78%
records:	2,267



	with haptic mean		
	better	**equal**	**worse**
subject brakes	0	1	0
subject slows down	1	0	0
subject accelerates	0	1	0

Figure B.3 – SD, M and comparison table of Participant 14, as described in B.1 and Section 5.3

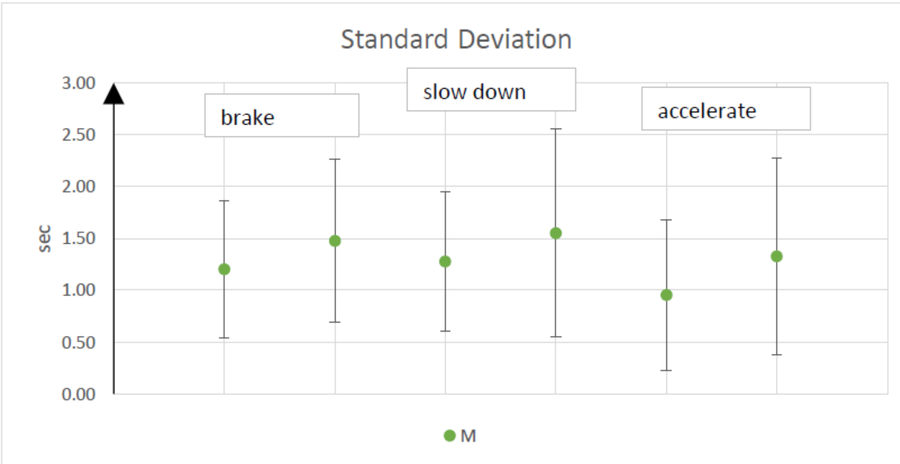
	Total	with haptic	w/o haptic
no. of events (total):	96	49	47
no. of collisions	0	0	0
subject brakes	39	18	21
subject slows down	10	6	4
subject accelerates	11	6	5
subject does nothing	25	14	11
subject ambiguous	11	5	6

	with haptic				
	no. of collisions	min. reaction time	max. reaction time	M	SD
no. of collisions	0				
subject brakes	0	0.30	2.1	0.9	0.45
subject slows down	0	0.30	1.8	1.05	0.44
subject accelerates	0	0.30	4.65	1.8	1.44
subject does nothing	0				

	w/o haptic				
	no. of collisions	min. reaction time	max. reaction time	M	SD
no. of collisions	0				
subject brakes	0	0.30	1.8	0.90	0.45
subject slows down	0	0.30	2.85	1.31	0.94
subject accelerates	0	0.30	3.45	1.38	1.09
subject does nothing	0				

Figure B.4 – Fact Sheet for Participant 14, containing all numeric and statistical data

no. test persons	16
speed adjusted	1.71%
records:	34,514



		with haptic mean	
	better	**equal**	**worse**
subject brakes	1	0	0
subject slows down	0	1	0
subject accelerates	1	0	0

Figure B.5 – Master Sheet, SD error bars computed overall 16 participants together structured by actions types, left SD error bar was measured with haptic and right SD error bar without haptic support. Furthermore (in the first part) some statistical data like number of speed adjustments and data records are presented. The last table "evaluates" the mean reaction time of all participants together computed for haptic signals to be better/equal/worse than without haptic signals. Its significance is low and must be considered carefully.

Appendix C

Questionnaire

Thank you for participating in the cycling experiment. In the following, I would kindly ask you to invest 5 minutes of your time to answer different questions on the experiment.

Section A: Participant's information

A1. What is your gender?

male	female	other
☐	☐	☐

A2. What is your age?

Section B: Understandability

B1. How understandable was the function and purpose of the haptic signal to you?

completely incomprehensible	mostly incomprehensible	neutral	mostly understandable	completely understandable
☐	☐	☐	☐	☐

B2. How easily distinguishable were the signals that the bike was giving?

very difficult	mostly difficult	neutral	mostly easy	very easy
☐	☐	☐	☐	☐

Section C: Acceptance

C1. Would you like to have such a system on your on bicycle?

not at all	mostly not	neutral	mostly yes	absolutely
☐	☐	☐	☐	☐

C2. What do you think, would your friends like to have such a system on their bicycle?

not at all	mostly not	neutral	mostly yes	absolutely
☐	☐	☐	☐	☐

Section D: Distraction

D1. **I estimate the haptic signals as...**

very distractive	slightly distractive	neutral	mostly nondistracti ve	not distractive at all
☐	☐	☐	☐	☐

D2. **I estimate the haptic signals as...**

very intuitive	slightly intuitive	neutral	not that much intuitive	not intuitive at all
☐	☐	☐	☐	☐

Section E: General questions about the signals

E1. **How would you rate the timing of the signal?**

too late	a little to late	fitting	a little too early	too early
☐	☐	☐	☐	☐

E2. **Did those signals help you in identifying dangerous situations?**

not at all	mostly not	maybe	mostly yes	absolutely
☐	☐	☐	☐	☐

E3. **Do you think these signals could help people to prevent accidents in real life?**

not at all	mostly not	maybe	mostly yes	absolutely
☐	☐	☐	☐	☐

Section F: Feedback and Suggestions

F1. **What would you like to change about the signals?**

F2. **What did you like about the implementation?**

F3. **What did you dislike about the implementation?**

Thank you for your participitation!

Term	Meaning
Independent variable or Factor	variable is controlled/modified by the experimenter
Dependent variable	variable changes as response to independent variables
Within subject design or Repeated measure design	one single group of test persons, all persons are tested under all test conditions, repeated measurements to compare the effect of experimental variations (independent variables)
Factor with n levels	the experiment compares (allows) n types (variations) of the factor
SD	standard deviation is a measure which quantifies the amount of dispersion of a set of data values, measurement of uncertainty. If data is distributed normally, 68% of data are in the range of M +/- SD
M	mean value, sum of all values divided by the number of values

Table C.1 – Commonly used vocabulary definitions in experiments