Novel Design Approaches for a Multifunctional Steering Wheel

Teo Babič

MASTERARBEIT

eingereicht am
Fachhochschul-Masterstudiengang

Interactive Media

in Hagenberg

im Juni 2015

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere.

Hagenberg, June 29, 2015

Teo Babič

Contents

Kurzfassung

Seit Jahrzehnten steigt in Fahrzeugen die Zahl von Assistenzsystemen und die damit einhergende Zahl von notwendigen Interaktionen für den Fahrer. Seit jeher betreiben die Automobilkonzerne einen enormen Aufwand um die Interaktionen mit diesen Systemen für den Fahrer überschaubar zu halten. Ein wichtiger Aspekt dabei ist das Fahrzeuginteriör anzupassen und zu verändern, um Interaktionen für den Fahrer entsprechend zu vereinfachen. Beispielsweise können neuartige Display- und Sensortechnologien eingesetzt werden, um dem Fahrer nur jene Funktionalitäten anzubieten, die er in einer bestimmten Fahrsituation benötigt. Davon abgesehen können diese neuartigen Assistenzsysteme aber auch Aufgaben des Fahrers teilweise oder sogar vollständig übernehmen. Dabei ist es notwendig Systeme zu entwickleln, die zu jeder Zeit über den Zustand und das Verhalten des Fahrers informiert sind, um lebenswichtige Maßnahmen aktivieren oder deaktivieren zu können. Diese Arbeit beschreibt das Design und die Implementierung eines multifunktionalen Lenkrads, das die oben genannten Probleme addressiert. Das Lenkrad besteht aus einem Touchscreen mit taktilem Feedback, einer drucksensitiven Eingabefläche mit Vibrationsfeedback sowie aus einem kapaztiven Tracking der Handpositionen. Darüber hinaus wird eine Studie vorgestellt, die die Vor- und Nachteile eines Touchscreens mit taktilem Feedback an einem Lenkrad untersucht. Die Ergebnisse zeigen, dass verschieden stark ausgeprägtes taktiles Feedback die Fehlerrate nicht verändert und dass Fahrer in der Lage sind, ein solches Gerät während der Fahrt zu bedienen.

Abstract

Since decades the number of assistance systems in cars as well as the subsequent number of necessary interactions are constantly increasing. For almost the same period, automotive companies invest huge efforts to reduce the complexity of the interaction with those systems. The most promising approach is to change and adapt the car interior in order to simplify tasks for drivers. For example, novel display and sensor technologies can be used to reduce the number of devices in the driver's surrounding by only showing functionalities which are needed in a certain situation. However, modern driving assistants can also reduce the driving workload or even completely detach the driver from the driving task. As a consequence, it is also necessary to include systems into the interaction process, which constantly track the driver's attention level in order to enable or disable life-sustaining controls. This work presents the design and implementation of a multifunctional steering wheel prototype that addresses previously mentioned problems. The prototype multifunctional steering wheel features a touchscreen with tactile feedback, a pressure-sensitive interactive surface with vibration feedback, as well as a conductive hand tracking system. Additionally, an empirical study was conducted to assess the benefits and limitations of a touchscreen with tactile feedback mounted on a steering wheel. It was found that the different amounts of tactile feedback did not increase or decrease the error rate and that drivers can easily handle such an interaction device.

Chapter 1

Introduction

1.1 Motivation and Goals

Cars are getting smarter every year. Numerous of new systems are getting embedded into them from advanced driving assistants, mobile phone docks, camera tracking systems, hi-fi audio systems, advanced drivetrain controls, car-to-car communication controls, etc. This results to an increasing number of functionalities offered in a modern car, for example a 2008 BMW 7 series offers 700 different functionalities [4]. The usual way of controlling driving and infotainment system functionalities is with pushbutton switches that are mounted in the center console, on the steering wheel or on the handles behind the steering wheel. For a long time these interaction devices sufficiently controlled all functionalities offered by cars, but with the increasing number of functionalities, most of the interior parts are getting overloaded with new interaction devices. The center console changed dramatically in the last ten years where displays and touchscreens of different sizes, controller knobs, air gesture recognition systems, etc., where embedded into it. While on the other hand, steering wheels with controls placed on them, so called multifunctional steering wheels, mostly did not change since their first appearance.

In recent years not only cars got smarter and more powerful but also the wishes and expectations of drivers have changed. For example, today people strongly customize their devices, phones, tablets, computers, hi-fi systems, gaming consoles, etc. They use applications and settings that are best fitting to their needs. So while spending time in a car, they expect also the same flexibility and adaptability from the car systems. Additionally to that, modern cars strongly support the driver while driving in special scenarios and conditions. The cruise control system is adapting the car distance and speed according to other cars and objects on the motorway, driver performance tracking system recommends to take a break in case of fatigue, traffic jam assistant completely takes over the driving task from the driver and drives the car autonomously in case of a traffic jam, parking assistant parks a driverless car and picks up its driver outside of the garage.

All driving assistant systems reduce the driving workload and offer more time for the driver to relax or do something else while driving. Here by new interaction concepts could reduce the density of interaction devices from interior parts, would make the driver-car interaction more adaptable to the drivers needs and would support them in different driving workload scenarios. Multifunctional steering wheels show a big potential, because they did not evolve much in the past and have a excellent and unique interior position for the driver-car interaction.

1.2 Steering Wheel Interaction

Since multifunctional steering wheels started appearing they were quickly adapted by most of the car manufacturers and are today available in almost every cars with the basic equipment package. Multifunctional steering wheels differ between each other by the number, function mapping and layout of the controls placed on them. Usually there are two sets of these controls on a multifunctional steering wheel, placed on the right and left side of it. One set contain from two to eight controls, depending on the functionalities that the car offers. These controls can be normal pushbutton switches, rocker switches or rotating switches and are usually used as shortcuts to a certain function or as an extension of another interaction device. In most cases each control is mapped to one function of the car or infotainment system (e.g. menu navigation arrow buttons, cruise control enable or disable buttons, volume adjustment rotating switch) which results a very simple and inflexible interaction device.

In case of drivers who drive every day 20 km to work on a country road and for such a short trip they do not need the cruise control, do not pair their phone and do not use the voice control commands, all the most common multifunctional steering wheel controls are useless on a daily basis. A multifunctional steering wheel where the functions can be modified, could offer those drivers controls that would be more useful for their needs or maybe evan the most used ones.

In this thesis we will explain a new steering wheel interaction concept which uses modern technologies, e.g. touchscreens and smart materials, to solve problems mentioned above. The overall contributions of this work are as follows:

- Present a new multifunctional steering wheel interaction concept.
- Explain the process and technologies used for the prototype implementation.
- Reveal how tactile feedback influences the driving and task performance of the driver.

1.3 Related Work

In the previous chapter it was noticed that there are different areas (e.g. device position, function mapping, car autonomy, personalization) which can influence and contribute to the design of a new multifunctional steering wheel prototype. This chapter will present related work that was found addressing similar problems.

González [4] used a touchpad on a steering wheel to investigate seven methods for selecting a street name from a list. It was found out that a gestural text entry method is about 20% to 50% times faster than selection based or direct list-selection based methods. Angelini [1] presented a user election study for gestures performed on the steering wheel ring surface. Angelini derived a taxonomy of gestures performed on the steering wheel and offered suggestions for designing gestural interfaces based on that. The study showed that most participants prefer single hand gestures and thumb gestures. The analysis also showed that swipe and tap gestures were the most adapted ones by the participants. Döring [3] introduced a idea of a multi-touch steering wheel that allows gestural input as well as visual output. In the first part of the study the most typical gestures that the participants used for controlling the infotainment system were collected. In the second part this multi-touch steering wheel gestures were compared with traditional physical controls positioned in the middle console. The main finding was that gesture based interaction with thumb is well suited for driving. The multi-touch steering wheel reduced the visual demand for 58% to 77% compared to the traditional physical controls and that there was no significant differences between this two interaction possibilities regarding driving performance. Ulrich [10] demonstrated the potential of a touch-based gesture system on a non-functional steering wheel prototype, with a touchpad for the left hand and button-based module for the right hand. Ulrich proved that the participants were capable of remembering a set of 19 predefined gestures and physically completing those gestures with high accuracy. Additionally participants reported a high level of satisfaction and usability on subjective rating scales. For the right hand button-based module no results were presented because it was not included in the study. Kern [5] investigated which surface in the car interior has the best position for handwriting as a text input. Two touchscreen displays prototypes with visual feedback were created, one was placed on the steering wheel and the second one in the center console. Kern showed that handwritten text input using fingers on a touchscreen mounted in the middle of the steering wheel is well accepted by users and lead to 25% fewer corrections and remaining errors compared to text input in the center console. Kern also recommended that for the input surface on the steering wheel the visual feedback should be presented on the dashboard (behind the steering wheel), which improves input speeds. The importance of adapting to the drivers needs in a particular situation was investigated by Trösterer [9] in a long-term study, regarding driver experience and acceptance of a parking assistant system. It was shown that supporting drivers with an assistant system in a particular situation sounds trivial, but the results indicate that in case that this assistant system does not fulfill the expectations of the drivers in a particular scenario, they may turn off the system in the long term. That is way an advice was given, to give assistance really in situations where assistance is needed, otherwise drivers can be annoyed by the assistance system, for example, if the parking assistance system is activated while driving backwards or slow, although it was no parking situation or if the drivers are confronted with a situation, which they could easily handle (e.g. huge parking lot). That is way it is important to adapt the controls to the drivers behavior, preferences and let them decide what they want to use.

As shown, many researchers investigated the potential of gesture interfaces on the steering wheel and compared it to other interaction devices where additional benefits were discovered. Bringing tertiary tasks to the steering wheel has already proven to be a “best practice” in the design of many existing cars and gesturing with thumbs were found to be especially well suited for driving, where both hands should ideally remain on the steering wheel [3]. In the future, it is expected that the car interaction systems will be able to control drivers handheld devices via Bluetooth or similar protocols. Therefore, a single input device that can be used for both built-in and brought-in devices may be useful [4].

1.4 Thesis Structure

On the beginning of this work, Chapter 2 will explain the goals and conditions that the designed multifunctional steering wheel should solve and fulfill. The following Chapter 3 will evoke which technologies and devices were used to build various prototypes. These prototypes will then be compared and commented in Chapter 4. What follows is a user study presented in Chapter 5 which deepens the understanding of the prototype that was build and answers some questions that came up during the designing and implementation phase. Finally, Chapter 6 covers the conclusion of this work and presents future work.

Chapter 2

Concept

This chapter explains the design phase of a multifunctional steering wheel. Later this design will be used as a blueprint for the prototype implementation phase explained in Chapter 3. The multifunctional steering wheel consists out of four parts (see Figure 2.1), which will be explained in detailed later on:

- **Force sensing touchpad**: A touchpad mounted on the right side of the steering wheel that can recognize swipe and touch gestures, made on its surface.
- **Display buttons**: Buttons on the left side of the steering wheel with the possibility to change icons and customize their functionality.
- **Hand position tracking**: A hand tracking system by which the car can detect the hand positions of the driver.
- **User interface and cluster display**: A user interface that provides visual feedback and visualizes different scenarios of the prototype on a cluster display behind the steering wheel.

As already mentioned in the introduction chapter [1], steering wheels offer a unique interior position and the principle, how drivers interact with them, did not change since their adoption in the automotive industry. This reveals a big potential for new interaction possibilities, using the steering wheel in combination with modern sensing technologies.

2.1 Force Sensing Touchpad

The force sensing touchpad (from now FSTP) should be able to recognize swipe and touch gestures that the driver makes over its surface. These gestures will be then used to control the infotainment system. The regular control buttons from the multifunctional steering wheel will then be replaced with a more flexible and customizable interaction device. Additionally, haptic feedback should be provided for every action that the driver makes. Since

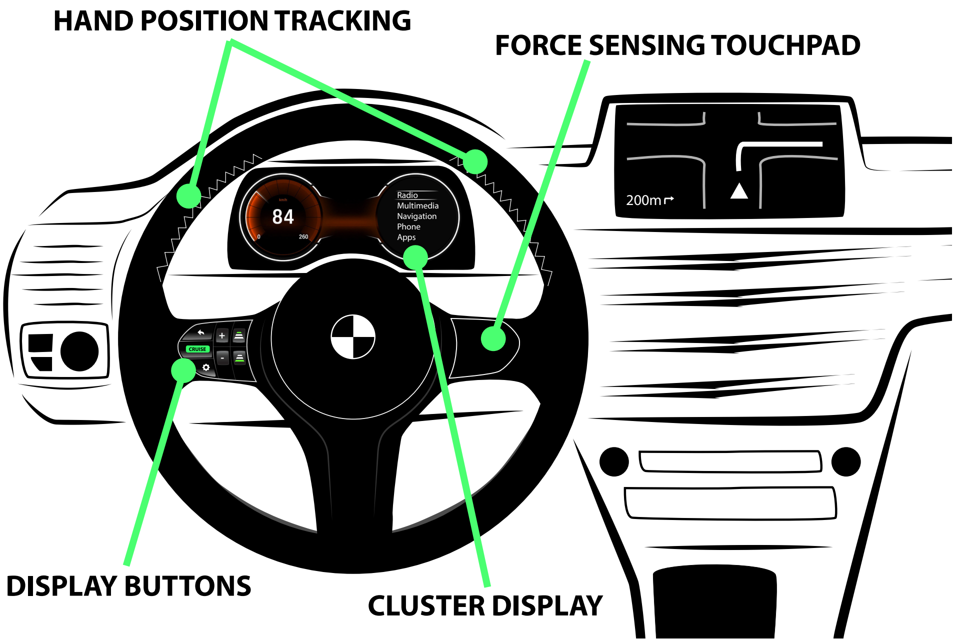

Figure 2.1: Conceptual multifunctional steering wheel sketch with a force sensing touchpad, display buttons, hand position tracking system on the steering wheel and a cluster display behind the steering wheel.

interior design is one of the most important aspects of the car, the whole touchpad should be hidden beneath leather, so that the technology used for it will not be visible. The FSTP will be placed on the right side of the steering wheel. The right side of the steering wheel was chosen after a research session, where currently available serial production cars were examined. It was found, that in cars where controls related to driving (e.g. cruise control, distance assistant, speed limiter controls) are directly on the steering wheel and not on the additional handle behind the steering wheel, in most cases these controls are positioned on the left side. Due to that, the infotainment controls (e.g. radio, phone, menu controls) are usually placed on the opposite (right) side (see Appendix A.1).

2.1.1 Gestures

The FSTP should be able to detect different single touch gestures that drivers make with their thumbs over the interaction surface. The FSTP will operate with a basic gesture set, that most of the people consider natural and already know from interacting with their phones and tablets. These gestures are *swipe up*, *swipe down*, *swipe left*, *swipe right*, *drag* and *tap* (see Figure 2.2). These gestures will be then mapped to the user interface, so

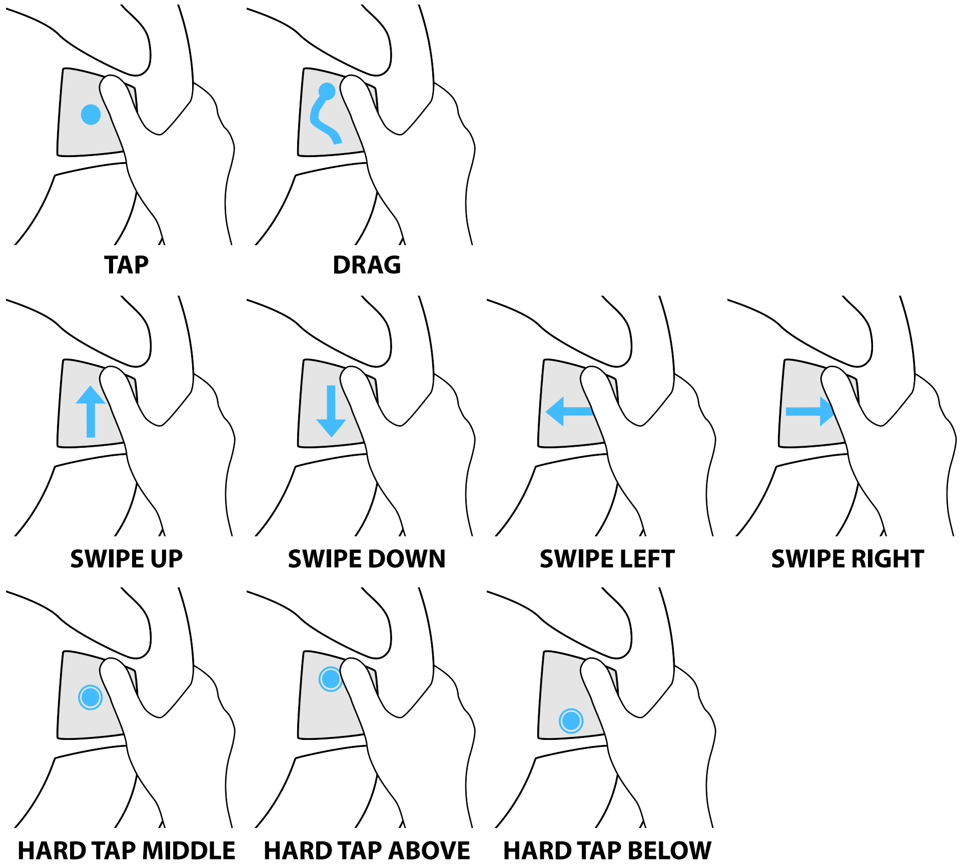

Figure 2.2: Gestures that should be detected by the force sensing touchpad.

that the driver can control all common functionalities usually offered in a modern infotainment system (see Table 2.1). The FSTP will additionally offer a new axis to the basic gestures set by using the force that will be applied to the surface by the thumb. This way three other gestures can be added to the basic gesture set, *hard tap up*, *hard tap down* and *hard tap mid*. These gestures can be used in all scenarios of the user interface and they always trigger the same actions, hard tap up and down increase and decrease the volume and hard tap mid always goes a step back (except when having a call). These three additional gestures operate on a different pressure level then the basic gesture set and work similar to the well known long tap. The difference is only that it is not time, that is used for separating of gesture sets between each other, but force applied to the FSTP.

Scenario	Swipe Up	Swipe Down	Swipe Left	Swipe Right	Tap
Main menu (List view)	Select menu below	Select menu above	/	/	Open selected menu
Multimadia - Movie player	/	/	Jump forward for 30s	Jump back for 30s	Play or Pause
Radio - Favorite station selection (Grid view)	Select station above	Select station below	Select station on the left	Select station on the right	Play or Pause
Radio - Manual station selection	/	/	Increase the FM frequency	Decrease the FM frequency	Play or Pause
Phone - Select contact (List view)	Select contact above	Select contact below	/	/	Call contact
Phone - Enter number (Keypad)	Select number above	Select number below	Move to column on the left	Move column on to the right	Select the number in the selection column
Phone - While in a call	/	/	/	Hang up	/
Navigation - Map view	Move map by drag				/
Navigation - Enter Address (Speller)	Select letter above	Select letter below	/	/	Confirm letter
Navigation - Address list	Select address above	Select address below	/	/	Confirm address
Apps (Grid view)	Select app above	Select app below	Select app on the left	Select app on the right	Open selected app
Internet browser	Move cursor by drag				Open hyperlink

Table 2.1: Functions that the gestures trigger in a certain infotainment system scenario.

2.1.2 Haptic Feedback

Swiping over an interactive surface does not provide any feedback about the actions we trigger and the current state of the system. In the automotive domain it is very important that drivers are always aware about their actions and that they do not trigger actions unintentionally. Considering this, the FSTP should also provide haptic feedback to the driver. Haptic feedback means that a mechanical stimulation as force, vibration, motion or enclosure, recreates the sense of touch. In the prototype haptic feedback

Figure 2.3: Vibration impulses, while scrolling through items in a list view.

will be provided by a vibration transferred to the drivers hand. A vibration impulse will be generated for every successfully recognized gesture. For example, while scrolling trough items in a list view, for every item that will be selected, a short vibration will additionally confirm the selection (see Figure 2.3).

2.2 Display Buttons

The control buttons opposite of the FSTP, on the left side of the steering wheel, will be replaced with display buttons. Display buttons closely mimic the pushbutton switches normally used on multifunctional steering wheels, but additionally to that, they offer more flexibility and customizability by changing their visual appearance and functionality. They will do that, with the help of a touchscreen embedded behind the buttons. These display buttons can then support the driver in different driving scenarios, for example they show cruise control controls while driving on the motorway and navigation, phone or voice controls in the city traffic.

2.2.1 Layout

Layout in which the display buttons will be positioned is following the trends of already existing layouts used by currently available serial production cars, with an average number of six buttons that are aligned in a grid (see Figure 2.4). The layout of the display buttons will be divided into two sections, the menu section and the function section. The menu section with three verti-

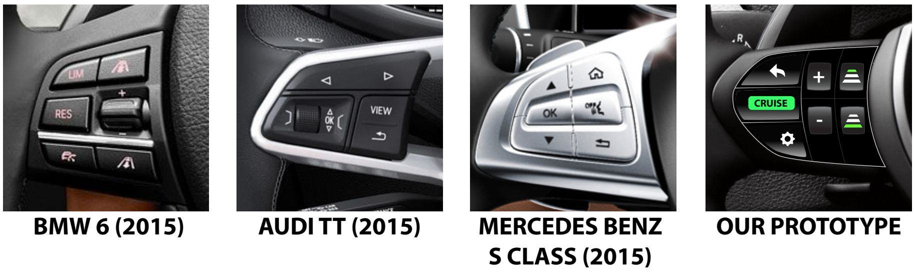

Figure 2.4: Button layouts on multifunctional steering wheels of available serial production cars and the layout of our display buttons prototype.

cally aligned buttons will be used as a selection menu for different driving modes. The function section with four buttons, which will be aligned in a grid, will show depending on the driving mode a set of functions with their icons (see Figure 2.4).

2.2.2 Tactile Feedback

Same as for the FSTP feedback should be provided to the driver. Tactile feedback will be provided in two different ways. The menu section simulates the look and feel of a real pushbutton switch. This means that the driver can feel the edges of each button and when pressing on it a click effect follows. In the function section, a plastic overlay will be mounted on top of the touchscreen, that will provide tactile feedback in a opposite form the menu section, so that the drivers will be able to feel the edges of the gaps which surround the icons shown on the touchscreen beneath (see Figure 2.5). These two different styles of tactile feedback will separate the two sections between each other, which will additionally help the driver to navigate between the buttons.

2.3 Hand Position Tracking

A evolving topic in automotive is also how to interact with an autonomous driving cars and consequently, how to monitor the driver and use this information later for the interaction. Passing the driving control to the car is relatively easy, but then regaining the control back to the driver is a more complex process. Monitoring the driver plays a crucial role. Therefore a hand position tracking system will be integrated into the steering wheel, which can detect, if the driver has both hands on the steering wheel. This will be then used, when the driver wants to regain the driving control back and disable the autonomous driving mode. When drivers will put both of their hands on the desired position on the steering wheel, the system will detect it and a five second countdown will be started. If the contact between the

Figure 2.5: 3D model of the display buttons control.

hands and the steering wheel will not be broken, the car will disable the autonomous driving mode and go into manual driving mode. The system should track the hands position on the upper side of the steering wheel, which means between the 9 and 3 o'clock position.

2.4 User Interface and Cluster Display

Today in almost every car we can find a dedicated infotainment system display placed in the center console. In most cases where a non-touch display is used, this display is controlled by an interaction device, usually a rotary knob, which is also placed in the center console. These rotary knobs are mainly meant to be used by the driver but also by the co-driver. Since the display is in the middle between them this makes sense, but recently there are more and more cars with a displays behind the steering wheel, so-called cluster displays. These cluster displays usually show content related to driving (e.g. speed, rpm, navigation map, fuel consumption) to the driver and free up space from the center console display. Since only the driver can control the multifunctional steering wheel controls we can conclude, that a cluster display behind the steering wheel that would visualize those actions makes perfect sense.

Kern [5] also concluded that for multifunctional steering wheel controls, visual feedback should be provided directly on the steering wheel or on the cluster display. Kern compared two touchscreen prototypes which where used for handwritten text input. One was placed on the steering wheel and another one in the center console. The two touchscreen prototypes provided visual feedback and additionally to that, the cluster display was used as well for visual feedback. It was shown that a touchscreen mounted on the steering wheel reduced the error rate for 25% and was well accepted by users, compared to the center console touchscreen.

For our prototype the cluster display should show the user interface that visualizes different driving and infotainment system scenarios shown in Table 2.2. This implemented cluster display user interface will also provide visual feedback for every actions the driver makes with the prototype steering wheel.

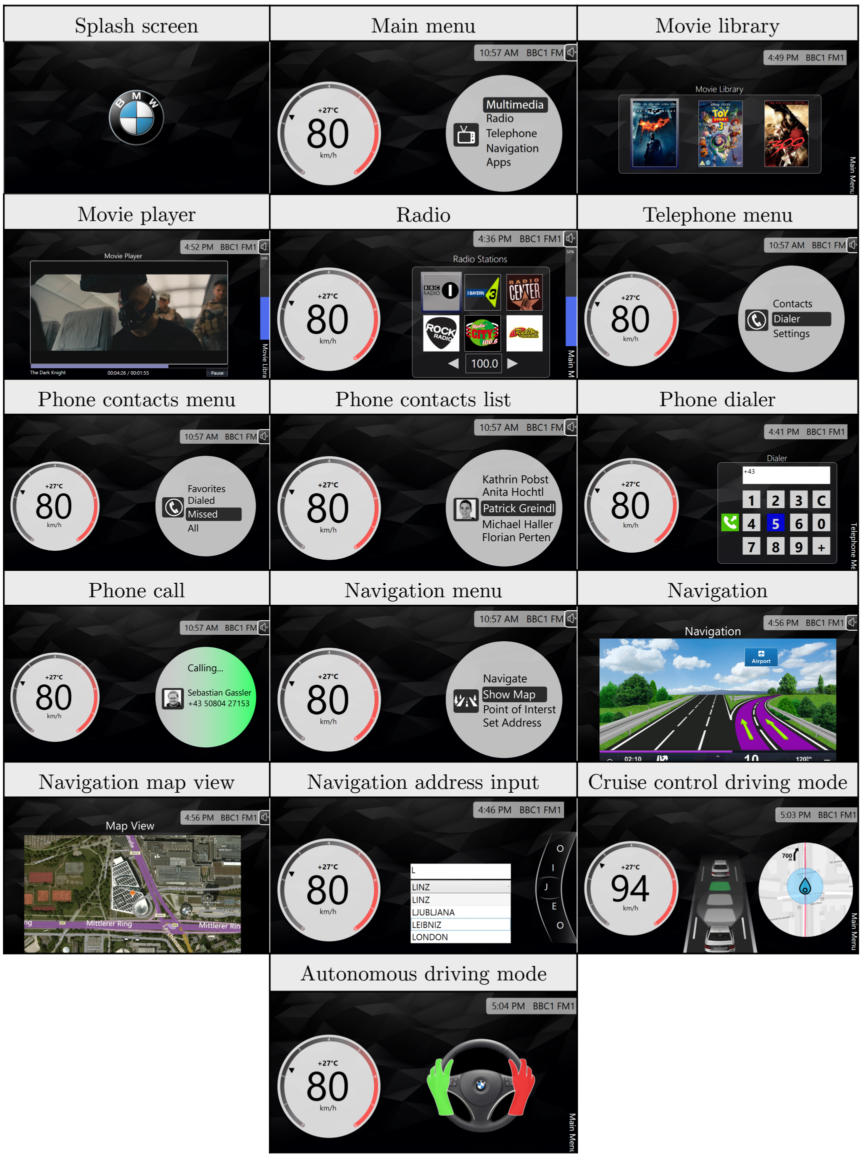

Table 2.2: Implemented cluster display user interface screens that visualize all infotainment system scenarios supported by the prototype.

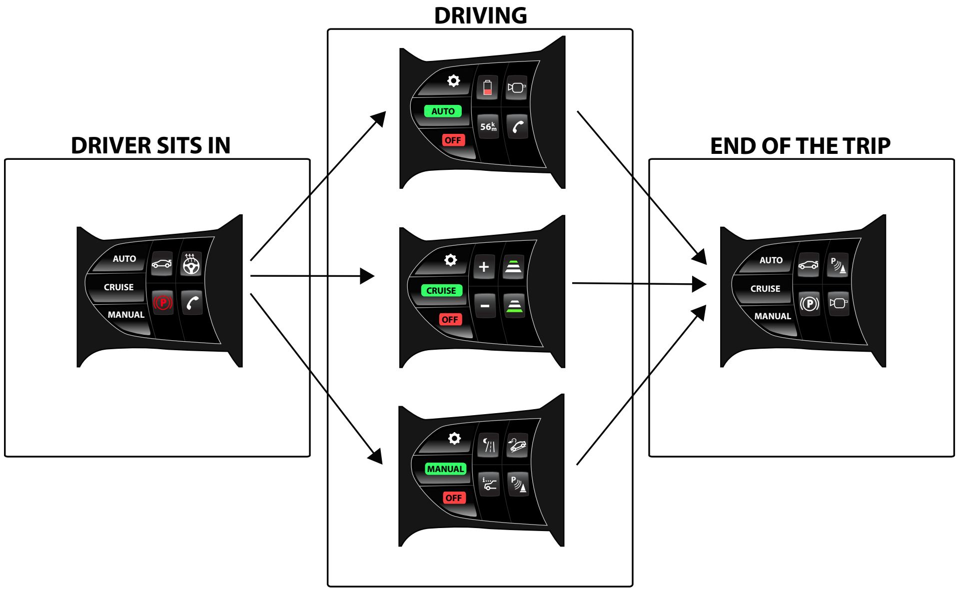

Figure 2.6: Implemented display buttons user interface, presented in three possible conditions, driver sits in, driving and end of the trip.

Also the display buttons will show a small user interface and provide visual feedback. The display button user interface will be made out of five screens (see Figure 2.6). These five screens will be separated to three condition, depending on the status in which the car is currently in. In condition one, the car was unlocked the driver sits in and the car offers steering wheel heating, trunk lid open, parking break off and phone controls on the display buttons. After that in the second condition, three driving modes can be enabled to start the trip, autonomous mode, cruise control mode, manual driving mode. According to the driving mode a new set of functions will be presented. In autonomous mode the battery capacity, battery range, 360 degree camera view and phone controls are presented. In cruise control mode the speed increase or decrease and distance to the car in-front increase or decrease controls are offered. In manual driving mode mostly driving assistant functions are shown, night vision, heel decline assistant, parking assistant controls and additionally, the head up display enable or disable control. Third condition is when the driver will be close to the goal of the trip, trunk lid open, parking assistant, parking break and 360 degree camera view controls are offered.

2.5 Safety and Hand Position

Due to the changes in steering wheel and airbag technology, according to the AAA Foundation for Traffic Safety and NHTSA, the recommended hand position on the steering wheel is at 3 and 9 o'clock. This position reduces the possibility of hand and head injuries during an airbag activation. Additionally, it makes it possible to do a 180 degree steering input without hands removal and increases the awareness of where the wheels are pointing. This recommendations will be taken into account while designing the prototype, so that no interaction devices will be placed on a potentially dangerous interior position.

Ulrich [10] conducted a simple anthropomorphic study where the appropriate size and placement of a touchpad on a steering wheel was determined. No exact values were presented, but it was shown how strained and unstrained thumb tap positions are spread over the touchpad (see Figure 2.7). In Figure 2.7 we can also see how this tap positions vary, so a relative gesture recognition system should be used for the FSTP prototype. That way the gesture recognition system could automatically adapt to the thumb length and position where the driver is making the gestures. Wang [11] measured the thumb lengths of his 16 user study participants and they ranged from 5.3 cm to 7.3 cm ($M = 6.5$, $SD = 1.4$). In our prototype this findings will be used, so that both the FSTP and display buttons controls, will not be more then 7 cm away from the edge of the steering wheel ring. Parhi and Karlson [7] found out that the optimal size of a square touch target is 9.2 mm to 9.6 mm on a touch device. They also found out that, when this size gets additionally increased the task completion times and error rates do not improve.

Figure 2.7: Figure from Ulrich's [10] anthropomorphic study, showing strained and unstrained thumb tap positions and various hand sizes of the participants.

Subsequently, we measured the sizes of the original BMW multifunctional steering wheel pushbutton switches, the average length was 18 mm (SD = 1.76 mm) and the average height was 10 mm (SD = 1.07 mm). Concluding from that, the menu section on the display buttons will have a 18 mm $\times$ 10 mm dimension and the function buttons a 10 mm $\times$ 10 mm dimension.

The conductive yarn, that will be explained later in the implementation chapter [3], is made out of a mixture of polyester and stainless steel, which is pliable, soft to touch and easy to from. For prototyping purposes there are no concerns about how healthy this material is, but for the automotive industry this information is very important, specially for materials which are in direct contact with the human skin. The techniques for mixing textile materials with metals are already known since the 7th century. Today this materials are used in various disciplines, for example fashion, upholstery, anti static environment, highly flammable environments, radiation environments clothing. There is also a wide range of this materials available with anti-bacterial, non-dissolving, toxicity, skin irritation and sensitivity certificates [17].

Chapter 3

Implementation

This chapter explains how the prototype was implemented and what technologies were used for it. The design decisions for the prototype were discussed in Chapter 2.

3.1 Interior Parts

For the prototype base an original BMW multifunctional steering wheel was disassembled and used. This prototype was then embedded into a BMW X1 clay mockup interior.

3.2 Force Sensing Touchpad

The FSTP was developed in two prototyping iterations. In each iteration a prototype was implemented with a different sensor technology. As explained in the Chapter 2, the FSTP should be mounted on the right side of the steering wheel and hidden under the steering wheel leather, so that no hardware parts are visible. For achieving that, the original steering wheel control buttons and airbag needed to the be removed. Instead of those parts, the FSTP hardware could be embedded into the steering wheel (see Figure 3.1).

3.2.1 Force Sensing with PyzoFlex

The first prototype was built on PyzoFlex sensor basis [8]. A ferroelectric material in the sensor uses pyro- and piezoelectric effects that can be used for sensing pressures on large or bended surfaces. It is constructed with a sandwich structure of four layers that can be printed easily on any material. The foil is bendable, energy-efficient, and it can be produced in a printing process. Even a hovering mode is feasible due to its pyroelectric effect. For the first FSTP prototype a 4×4 cell sensor matrix was printed and used. A specially designed microcontroller was placed in the airbag department

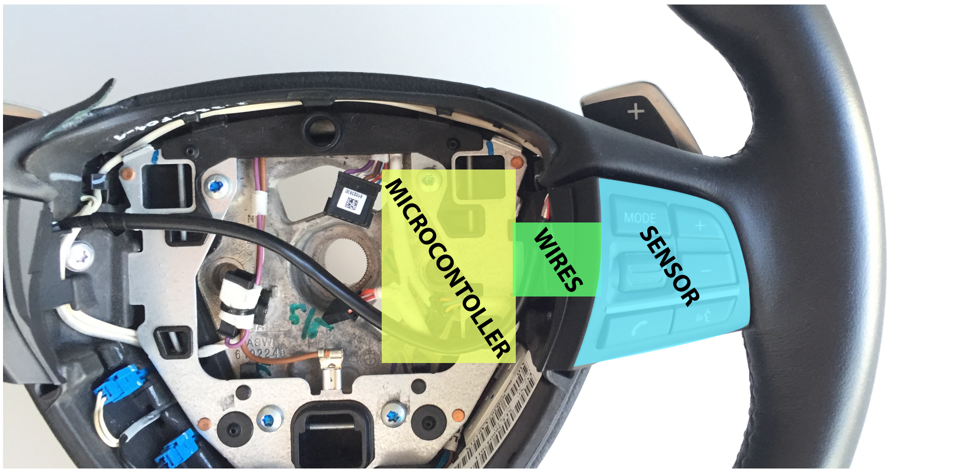

Figure 3.1: Location of hardware parts needed for the force sensing touchpad.

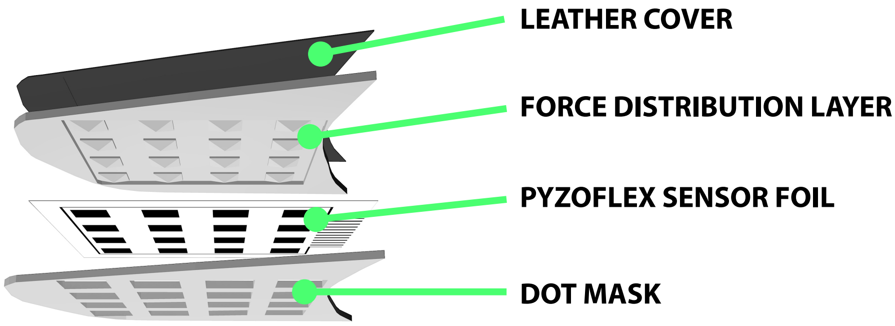

Figure 3.2: Layering of a force sensing touchpad with the PyzoFlex sensor matrix.

of the steering wheel. This microcontroller in combination with an Arduino Due was reading out forces applied of each cell of the matrix and was passing them to the user interface software. To improve the accuracy of the sensor, the PyzoFlex sensor foil was layered between two force distribution layers, which then maximized the force applied on each force cell and created a bigger gap between the force values (see Figure 3.2). Additionally to that, an algorithm was implemented which was calculating the force value of each cell and also the interpolation between them. This algorithm was able to programmatically increase the resolution of the matrix and consequently increased the accuracy of determining the exact position of where the force is applied.

3.2.2 Force Sensing with FSR

The second prototype was build on a force sensing resistor (FSR) basis [16]. Force sensing resistor consists of a conductive polymers, whose electrical resistance changes when force or pressure is applied. This change of resistance can then be detected with the help of a microcontroller. In the prototype a 16×10 FSR sensor matrix from Sensitronics was used. An Arduino Uno in combination with two 74HC595 shift registers and two 74HC4051 multiplexers was reading out the force value of each cell in the sensor matrix and passing them forward to the user interface software (for more detail see Appendix B.2). The resolution of the FSR sensor matrix was already high enough, so no additional distribution layers or algorithms were needed (see Figure 3.3).

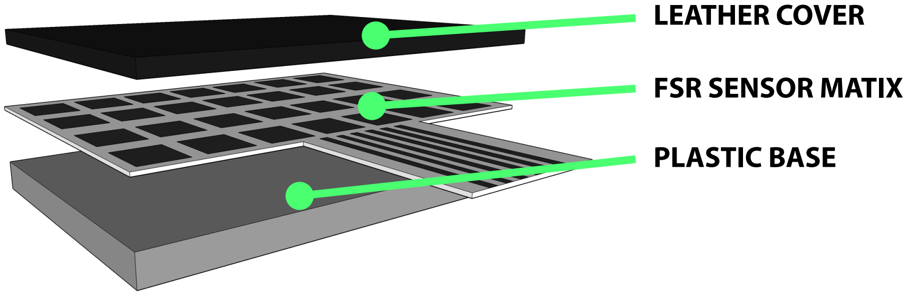

Figure 3.3: Layering of a force sensing touchpad with the FSR sensor matrix.

3.3 Haptic Feedback

To provide vibration feedback to the FSTP also two approaches were tested. First one was based on a linear resonant actuator (LRA) and the second one on a voice coil actuator. There are numerous types of vibration feedback, which mostly differ by what kind of an actuator is generating the vibration and how this vibration is distributed trough materials that surround the actuator. For analyzing and designing the vibration feedback a vibration analyzer was developed. The vibration analyzer used an accelerometer to measure the acceleration of the surface where the vibration was generated. Those accelerations were then visualized and presented in a graph. The optimal goal was to generate a vibration what is comparable to the clicking effect of the pushbutton switches used in modern BMW multifunctional steering wheels.

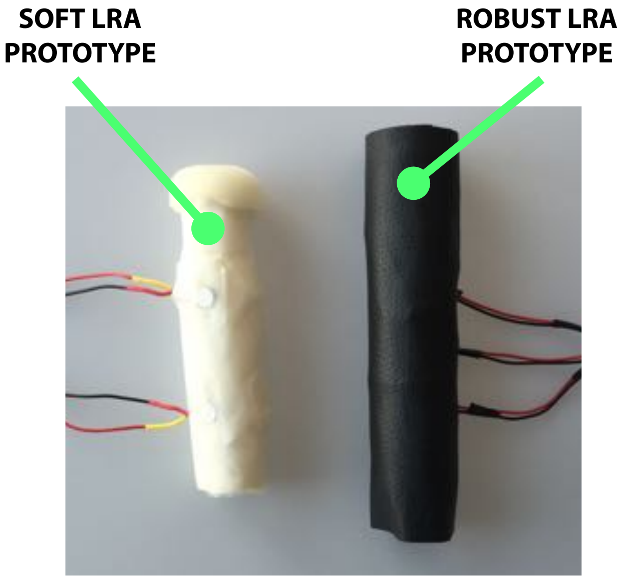

Figure 3.4: Both LRA prototypes, softly build prototype with two LRA's and the more robustly build prototype with three LRA's.

3.3.1 Vibration Feedback with LRA

LRA uses magnetic fields and electrical currents to create a force which moves a weight inside the actuator to its actuated position. A spring then returns this weight back to its starting position. This movement of the weight results a vibration. Two prototypes were made to test the feel of the vibration that a LRA can produce and the vibration distribution through the steering wheel materials. In the the first prototype three LRA's were used and in the second prototype only two (see Figure 3.4). The bigger difference between the two prototypes was, how the prototype steering wheel rings was assembled. The first prototype had a very robust and hard structure that simulates a real steering wheel ring. It was made out of a aluminum core which was then covered with thick paper and leather coating on top. The structure of the second prototype was much more softer and it is build out of a light aluminum core, foam and adhesive tape. Both prototypes were controlled by the user interface software with the help of an Arduino Uno controller and powered from the Arduino with 0 V to 5 V.

3.3.2 Vibration Feedback with an Linear Voice Coil Actuator

As a alternative to the LRA prototypes a second prototype was build, which was providing vibration feedback with an linear voice coil actuator. The actuator used was a original BMW steering wheel actuator which is used

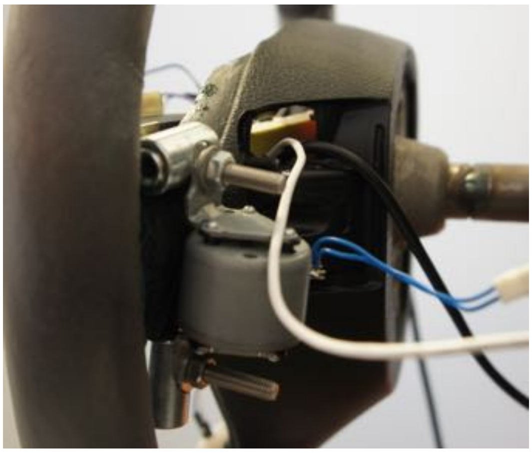

Figure 3.5: Linear voice coil actuator mounted behind the steering wheel ring.

for vibration alerts by the line keeping assistant. The principle how a linear voice coil actuators work is similar to LRA, they also use a weight which is then moved by a magnetic field and electrical current. The biggest difference between these two technologies is that this linear voice coil actuator operates on lower frequencies and is much bigger then a LRA. Because of its size the linear voice coil actuator could not be placed inside the steering wheel ring, so it was mounted behind it (see Figure 3.5). The actuator was controlled by the user interface software with an Arduino Mini and was externally powered with 0 V to 12 V.

3.4 Display Buttons

On the left side of the steering wheel a prototype of the display buttons was embedded. A 2.8 inch, 480×320 pixel, capacitive touchscreen display was mounted behind the overlay which expressed the look and positions of all the buttons. As designed the overlay consists out of two sections, the menu section with three transparent convex plexiglass pushbuttons and the function section with a 3D printed PLA plastic model (see Figure 3.6). The touchscreen was controlled by an Arduino Uno controller, which was displaying bitmap icons stored on a microSD card. Additionally, the three menu section pushbutton switches were control by an Arduino Micro (for more detail see Appendix B.1 and B.3).

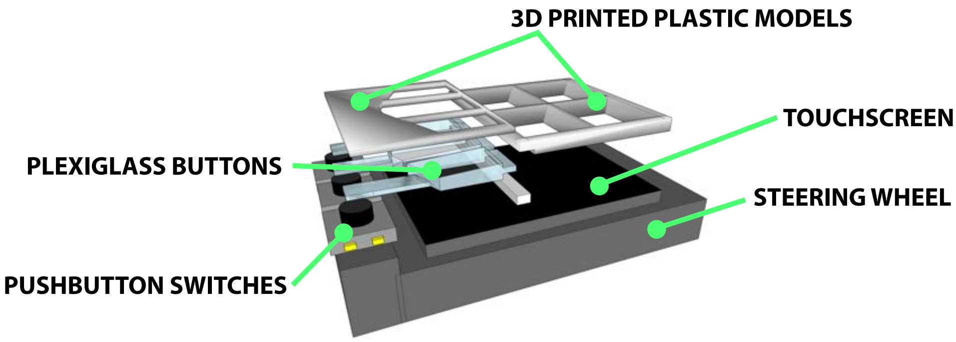

Figure 3.6: 3D model of the display buttons control.

3.5 Hand Position Tracking

For hand position tracking, the yarn which holds together the leather coating of the steering wheel ring was replaced with a conductive yarn (see Figure 3.7). This yarn is a mixture of 80% polyester and 20% stainless steel, which gives the yarn electrical conductive properties, it has a breaking load of 8094 g and has a surface resistance of less then 104 Ω. There are two conductive yarns sawn into both sides of the steering wheel ring so that each hand can be tracked separately. They spread from the 9 to 11 o'clock position on the left side and from 1 to 3 o'clock position on the right side of the steering wheel ring. These yarns were connected to a Adafruit MPR121 capacitive touch sensor which was then controlled by an Arduino Micro.

Figure 3.7: Conductive yarn used for fastening the leather on the steering wheel and tacking the drivers hand position.

3.6 Software

The implemented software was running on a desktop computer with a Intel XEON E3 3.85GHz processor, 8GB DDR3 RAM, 512MB ATI AMD FirePro 2270 graphics card and a Windows 8.1 operating system. The backend software implemented, scripts that were running on the microcontrollers, communication protocols and the control logic of all components used for the concept multifunctional steering wheel (see Figure 3.8). The backend logic was implemented in C and C++. Every Arduino microcontroller processed the raw data he was operating with on his own and prepared it for the communication with the computer (e.g. the raw sensor data was getting processed and compressed on the microcontroller processor and afterwards send to the computer, all bitmap icons were stored on a microSD card from which then the microcontroller displayed then on the touchscreen). That way the performance of the whole prototype was optimized. Through the whole process of developing different prototypes, the backend software was adapting to new protocols and technologies, which resulted to a backend where for example FSR, PyzoFlex, LRA, linear voice coil actuator, capacitive touchscreens or external power supply could be controlled. A WPF desktop application was developed, which then combined the backend with the user interface controlled by the frontend. Since the backend software was preparing all the data and states of the concept multifunctional steering wheel, the frontend software was responsible for displaying the user interface and making changes to it. There were two displays that needed to be controlled by the frontend, the cluster display and the touchscreen on the steering wheel. Additionally, the frontend managed all the resources that were used by the user interface (e.g. music files, movie files, animations, pictures, text data).

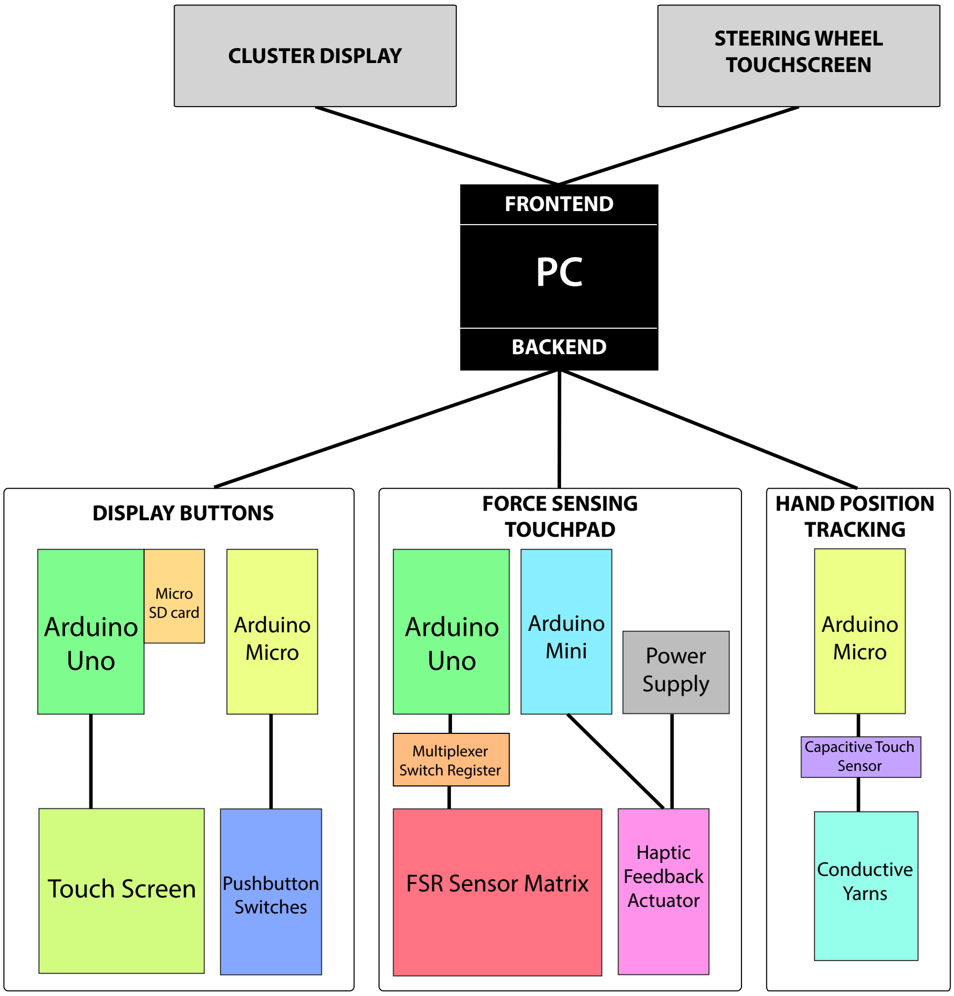

Figure 3.8: Architecture of all components that were used for the concept multifunctional steering wheel.

Chapter 4

Prototype Comparison

As described in Chapter 3 various FSTP prototypes were implemented which used different technologies or assembly methods. In this chapter these prototypes will be compared and analyzed. The best prototypes will be then implemented into the final multifunctional steering wheel prototype seen in Figure 4.3. Display buttons and hand position tracking are not mentioned in this chapter, since only one prototype of each was developed and directly embedded into the final multifunctional steering wheel prototype.

4.1 Gestures

After the FSTP was able to recognize all the gestures from the designed gesture set, the next step was to map them to the user interface. This step turned out to be more problematic that expected and needed some more planning. As already mentioned, most people know touch and swipe gesturing from their phones, tablets or other touchscreen devices. Gesturing over a touchscreen is pretty self explanatory because all the objects that the users can interact with are displayed directly under their fingers, they just need to press on them. In the FSTP prototype, the touch surface was separated from the screen and the direct visual connection to the objects was lost. This separation is well known from laptop touchpads or graphics tablets, but in the automotive this is a bit more critical, since the driver can not fully focus on the input device. For solving this problem a gesturing workflow was designed, that established this connection back with the help of visual and haptic feedback. The gesturing workflow starts with a visual clue that explains the current status of the user interface, for example which user interface elements are currently active and displayed on the cluster display. Next step is the actual user interaction, where the user can make a gesture on the FSTP. Shortly after the gesture is done, the user receives haptic feedback which notifies him that the gesture was recognized. The last step is the visual conformation, where the user interface visually responds to the gesture made, for example the screen changes, a user interface object is animated or music is played. This gesturing workflow was used for simple swipe gestures and also for the more complex continuos gestures, for example a drag gesture where the user receives haptic and visual feedback multiple times during the gesture. Every step of this gesturing workflow was considered essential and influenced all the other steps in it. Knowing that, the implementation of user interfaces was easier and the final user interface was more intuitive. This was noticed, while testing the FSTP prototype with different people, without explicitly explaining them how it works. Mostly they determined the gestures on their own, for example, when a list view of objects is show they automatically determined that a swipe up selects the object above and a swipe down the object below, when a timeline slider is show the user determined that swiping left or right turns the time forward or backward, if a grid of objects was show swipe gestures to all four directions could be used to select an object in the grid. Concluding from that, its was obvious how closely the user interface is tied to the gestures that control it and that a deeper understanding of every step of the gesturing workflow was needed to create a intuitive and self-explanatory interaction concept. Another problem noticed while mapping the gestures to the user interface, was the openness of the user interface itself. The reason for that is that the gesture sets are limited to a certain number, for example, in the radio station selection screen where a grid of radio stations is shown, immediately all five gestures (swipe up, swipe down, swipe left, swipe right and tap) from the basic gesture set are used and no gesture is left, for example, to move back to the main menu or change the volume. One way for solving this problem is to redesigning the user interface, for example to change the grid selection to a list view selection, which then only uses three gestures (swipe up, swipe down and tap). Redesigning the user interface is not a satisfactory solution, since there can be many different user interfaces screens that need to be changed and hierarchically splitted. For the FSTP prototype this problem was solved by redesigning the gesture set, where three additional unique gestures, that operate on the second pressure level, were added to the basic gesture set.

4.2 Sensor Types

The PyzoFlex sensor matrix with 16 sensor cells had a ten times lower resolution as the FSR sensor matrix with 160 sensor cells. For the designed gesture set both sensors performed sufficient and allowed gesture input with a low error rate. While experimenting with the FSR sensor matrix a new interaction possibility was discovered. Because of the bigger resolution it was possible to detect the force distribution of the thumbs fingerprint on the FSTP. Subsequently, the drivers could interact with vertically aligned

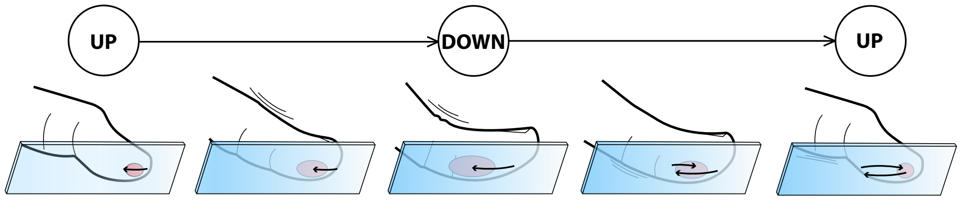

Figure 4.1: Micro-gesture using the force distribution of the thumb for vertical selection of user interface objects.

user interface objects (e.g. list views, grids) just by shifting the force from the tip to the second joint on their thumbs. This micro-gesture had couple of benefits, the driver could do the gestures without dragging his thumb over a surface, it maximizes the feeling of vibration feedback because the finger was always in a good contact with the surface and this method can be easily relocated to other, evan smaller interior parts. The only drawback to the regular swipe gestures is that it is a unknown interaction concept and the learning period can be a bit longer. Bonnet [2] also showed a similar interaction concept by developing an algorithm that used touchscreen data to enable the same kind of a micro-gesture. Due to the reasons mentioned above the FSR sensor matrix was embedded into the final FSTP prototype. Additionally, an algorithm was implemented that could simultaneously recognize and separate a swipe, drag or thumb micro-gesture, so the drivers could decide which gesturing technique they prefer for the user interface interaction.

4.3 Vibration Types

Two different vibration actuators, LRA and linear voice coil actuator, were tested while developing the prototype. The biggest difference between them was the size and how were they mechanically designed. The LRA had a diameter of 10 mm × 4 mm and weighed 5 g, because of its size it generated stronger vibrations in higher frequencies, for example 175 Hz. At lower frequencies, for example 50 Hz, the actuator moves the weight inside slower and because of the small weight the vibration is pretty weak. On the other hand the BMW linear voice coil actuator is much bigger, 40 mm × 40 mm and weighed 171 g, so its weight is much heavier and its vibration is much stronger than LRA's, specially at low frequencies. As mentioned in Chapter [2], the goal was to generate haptic feedback that is comparable to the haptic feedback of a BMW pushbutton switch. With the vibration analyzer we could measure the accelerations of surface to which the actuators were transmitting the vibration. We analyzed and compared different patterns that were measured by the accelerometer. Figure 4.2 shows, that a normal pushbutton switch generated a much stiffer click then the BMW pushbutton switch, which click effect is a bit softened. Comparing the two actuators between each other, it is noticeable that the LRA needs more time to develop and stop the vibration. So the obvious choice for the final prototype was the linear voice coil actuator that generates the strongest and most comparable haptic feedback, compared to the BMW pushbutton switch.

An important factor while developing a vibration feedback prototype is how the actuator works in combination with other materials. Surrounding materials can dramatically change the vibration, because of the distribution of the vibration through different materials. Two prototypes, where more LRA's were placed into a steering wheel ring, were tested to find out, how the vibration gets distributed trough different materials and if it is possible to determine the origin of the vibration. The more solidly build prototype proves that, if the actuator is combined with more solid materials the strength of the vibration stays the same, but it distributes enormously through all materials close by. With this prototype it was impossible to determine which of the LRA's was vibrating, so directional vibration was impossible. The second prototype was build out of very soft materials and where the vibration was a minimally neutralized, but it was still very hard to determine where the vibration is coming from. So in the final prototype only a simple one shot vibrations were used.

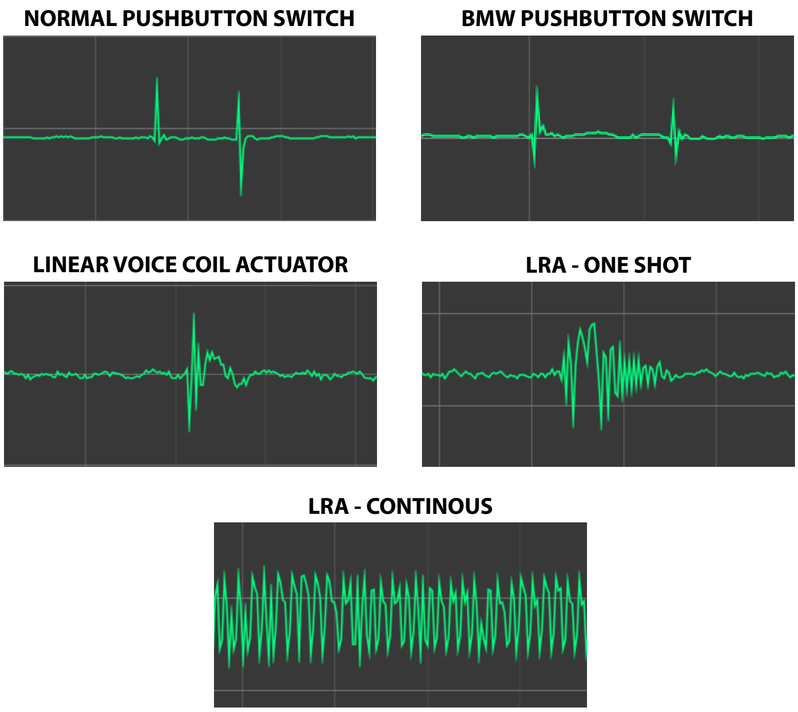

Figure 4.2: Surface vibration analysis of different actuators.

Figure 4.3: Multifunctional steering wheel prototype after the implementation phase.

Chapter 5

User Study

Due to distraction and safety reasons all driver interactions with the car should be as quick as possible and have a small error rate. In case of the steering wheel, each button expresses a unique feel that differentiates it from the other buttons. This feeling can be characterized with the physical form of the button, its position, how it operates mechanically and haptic feedback it provides while interacting. That kind of approach to characterizing buttons, has proven itself through the evolution of the multifunctional steering wheel. This brings up a question, if this characterized buttons will still perform equally good in the future, when we start reducing the driving workload or we change their visual appearance. In that case, a strongly characterized button could become redundant and would limit the interaction possibilities that could be offered to the driver in scenarios, where it is not required from him to fully focus on the driving task. To investigate this case we created a steering wheel prototype, where the usual control buttons were replaced with a touchscreen, that provided tactile feedback. Additionally to that we explored, if changing the visual appearance of the buttons can also improve the interaction.

5.1 Overlays

To investigate, how differently shaped buttons perform under different workloads, four overlays were implemented, which differ by the amount of tactile feedback they provide (see Figure 5.1). All overlays used the same seven button layout that was designed for the display buttons prototype described in Chapter 2. The four overlays were as follows:

- **Convex**: This overlay provides the most tactile feedback from all the overlays. It closely mimics the well known steering wheel pushbutton switches. Each button is made out of hard silicone and protrudes from the overlay surface. Having this convex buttons means that every shape that the driver feels under their finger is directly the button itself.

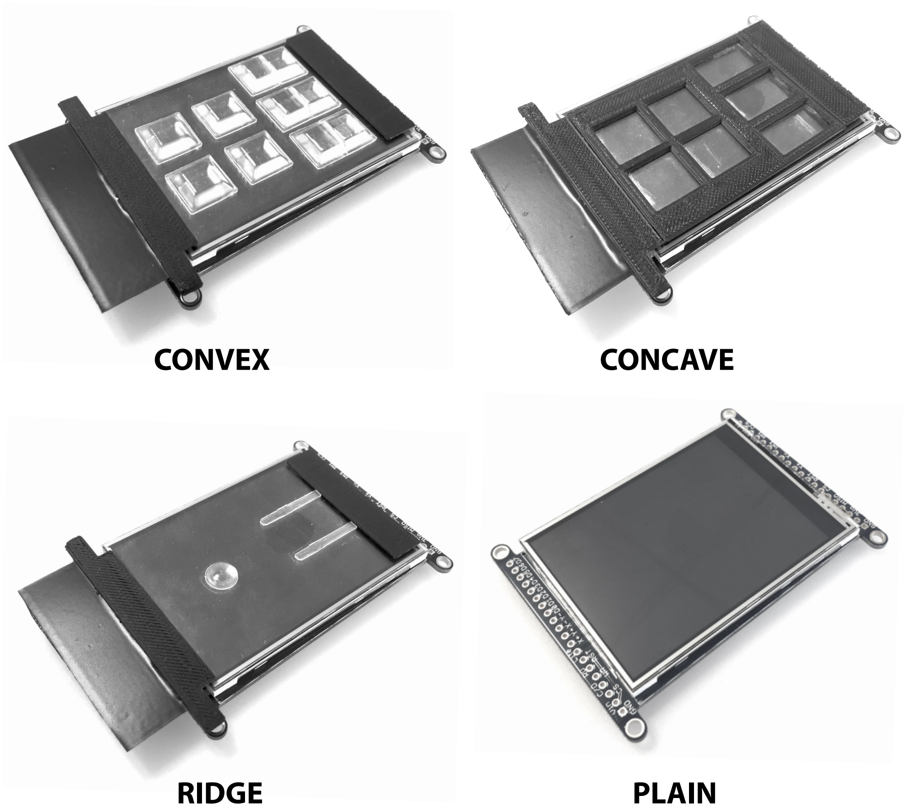

Figure 5.1: Prototype touchscreen with all four overlays.

- **Concave**: This overlay is a "negative" form of the convex overlay. The driver does not feel the buttons itself, but just the indentation of each button. These plastic borders are basically just tactile guides to the actual button displayed beneath the overlay on the touchscreen.
- **Ridge**: This overlay is a mid state overlay between the plain and concave overlay. It consists of a silicon bubble placed in the middle of the four function buttons and two silicon stripes that separate the three menu buttons. This overlay also presents tactile guides, which are comparable to the ridges on the F and J keys on a keyboard.
- **Plain**: This overlay represents the plain touchscreen surface, without any tactile feedback.

5.2 Hypothesis

We expected, that overlays with more tactile feedback will perform better then overlays with less feedback, in terms of error rates (*H1*) and task completion times (*H2*). That driving modes with higher workloads produce more error rates and longer completion times (*H3*). That the steering wheel touchscreen performs better as a task instruction source compared to the cluster display (*H4*) and that participants prefer overlays with more tactile feedback then overlays with less feedback (*H5*). The following, we explored the following hypothesis to address critical questions like:

- How do the different overlays affect the task performance?
- How does the workload influence the task completion times and error rates?
- What is the task performance of each overlay, when the workload gets reduced?
- How do the different instruction sources affect the task performance?
- What do the drivers prefer? Overlays with more tactile feedback or is an overlay with less feedback?

H1: Overlays with more tactile feedback generate fewer errors, then overlays with less tactile feedback. We wanted to find out whether the number of falsely pressed buttons gets reduced if more tactile feedback is provided. More tactile feedback should help the participants to increase their certainty of which button to press. Therefore, we measured the error rate.

H2: Overlays with more tactile feedback have shorter task completion times, then overlays with less tactile feedback. Very similar to the first hypothesis, we wanted to find out whether overlays with more tactile feedback reduce the time, that the participant needs for finding and pressing the correct button. Since the overlays with more feedback offer more tactile information for the participant to sense. Concluding from that, participants should be able to find and press the correct button faster. Therefore, we measured the task completion time.

H3: In driving modes where the driving workload is higher, also the task error rates are higher and task completion times are longer. We expect that when more work is required from participants to drive the car, their ability to complete the task fast and sufficient will decrease. Therefore, we measured the error rate, task completion time and counted the driving mistakes.

H4: Error rates are lower when the steering wheel touchscreen is used as an instruction source. We expect that if we use the touchscreen on the steering wheel as a task instruction source, it will result lower error rates compared to the approach, where the instructions are displayed on the cluster display behind the steering wheel. Therefore, we measured the error rate.

H5: Drivers prefer overlays with more tactile feedback then overlays with less tactile feedback. We expect that more tactile feedback will make the task completion easier. Concluding from that, participants will subjectively rate overlays with more tactile feedback higher then overlays with less feedback. Therefore, we measured the usability of each overlay, defined as rating in a survey.

5.3 Experimental Design

We recruited 12 graduate students from the local university to perform the user study. Participants were asked to drive a motorway driving simulator and solve primary tasks for each of the four overlays. For completing each primary task, they needed to press the correct buttons on the steering wheel, once they were instructed. To simulate different stages of driving workload, three driving modes were introduced, *manual driving*, *cruise control* and *autonomous driving*. Summarizing, the study had 3 driving modes × 4 overlays and 2 possible instruction sources, on average it took 40 minutes to complete all the tasks (13 minutes for each driving mode). In the study we measured the participants performance on the primary task in terms of task completion times and error rates. Performance of the driving task was measures in terms of driving mistakes.

5.3.1 Participants

In total, 12 graduate students (8 male, 4 female) from the local university participated in the study. Their age ranged from 23 to 39 years ($M = 27.2$, $SD = 3.8$). They were owning their driving license from 3 to 21 years ($M = 9.1$, $SD = 4.0$). When asked, 6 participants answered that they are driving regularly every day, 4 participants answered that they drive 1 to 3 times a week, and 2 participants 2 or 3 times per month. 9 of the participants also drive al least once a week on the motorway. 5 of the participants are also regularly interact with a multifunctional steering wheel, while 7 of them do not have it. The ones that use a multifunctional steering wheel were also asked, for what functions do they use it, the result was 59% volume control, 25% multimedia source change, 13% radio station change, 2% cruise control and 1% phone controls.

5.3.2 Primary Task

The primary task was conducted to explore the benefits and limitations of each touchscreen overlay. While researching, about the functions that are

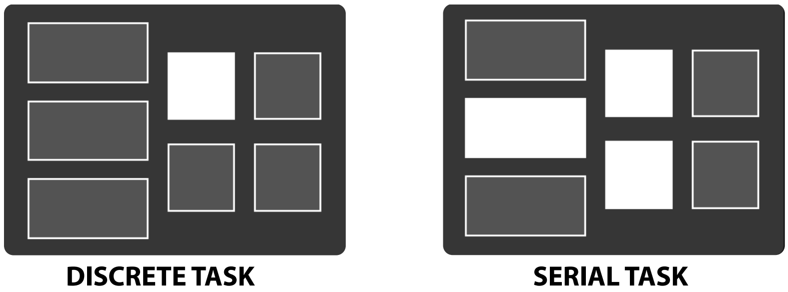

Figure 5.2: Visual instruction for a serial and discrete primary task.

usually placed on modern multifunctional steering wheels (see Appendix A.1), it was noticed that there are two different task types that control these functions. First task type represents buttons, which only control one discrete function (e.g. mute button, cruise control reset button, voice control activation button). The second task type represents buttons, that control a certain function in combination with other buttons (e.g. cruise control speed increase or decrease buttons, volume increase or decrease buttons, menu navigation arrows). From this two approaches, two task types were introduced, their visualization can be seen in Figure 5.2:

- **Discrete task**: Simulates a button with only *one* discrete function. When this button was successfully pressed another discrete button was instructed.
- **Serial task**: Simulates a combination of two or three buttons that need to be pressed, the order in which they are pressed does not matter. Once all the buttons of the combination were successfully pressed a new combination was instructed.

To further explore the potential of a touchscreen on the steering wheel, we changed the instruction source, after each set of discrete and serial tasks. At the beginning, the instructions were visualized on the cluster display behind the steering wheel and afterwards directly on the steering wheel touchscreen (see Figure 5.3).

5.3.3 Driving Task

As mentioned before, four driving modes were introduced:

- **Manual driving**: The driver had the full control over the car with an automatic gearbox.
- **Cruise control**: The driver was only responsible for the steering of the car and the car was maintaining the speed and distance to other cars automatically.

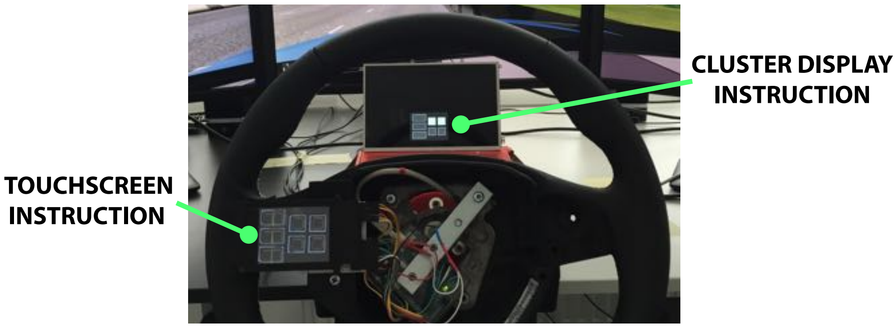

Figure 5.3: Two possible instruction sources. In the picture the instruction is visualized on the cluster display.

- **Autonomous**: The car took over all driving controls and drove fully autonomous.

These three examples of driving modes were taken from the definitions of vehicle automation levels, presented by the National Highway Traffic Safety Administration NHTSA [13]. *Manual driving mode* represented the NTHSA - level 0, *cruise control mode* NTHSA - level 1 and *autonomous mode* NTHSA - level 4. In this study we did not investigate user distraction and traffic awareness, levels that involve a driver monitoring the autonomous system, were skipped. This resulted to three driving modes, with the biggest difference in terms of driving workload. The goal was to simulate a real world driving workload and not to overload the driver with more tasks, that would generate bigger workload shifts.

5.3.4 Apparatus

The studies were conducted in a quiet room, where a driving simulator cockpit was assembled. The driving simulator cockpit consisted out of a personal computer, three monitors and a Logitech G27 steering wheel game controller. The personal computer was running Windows 8.1 with an Inter Xeon 3.6 GHz processor, 16 GB DDR3 RAM and 4095MB Nvidia Quadro graphics card. The simulation was displayed on three Dell 24” 1920×1200 pixel LCD IPS monitors. The car driving simulator game City Car Driving [12] was used for simulating the motorway driving scenario. The original Logitech G27 controller steering wheel as then replaced with a BMW steering wheel prototype with an embedded 2.8” LCD TFT resistive touchscreen. To allow some contact with the touchscreen, without accidentally triggering touch events while participants slided their fingers over the overlay and searched for the correct button, a FSR sensor was placed behind the touchscreen. This FSR sensor was then measuring the pressure applied to the touchscreen and blocked touch events with a low force. Additionally a 9” LCD TFT display

Figure 5.4: User study setup.

was placed behind the steering wheel controller, which simulated a cluster display from a car. A Windows desktop application displayed task instructions and implemented custom logging mechanisms, to record performance measures in the background. Furthermore, participants were recorded with a video camera for a subsequent analysis. The whole user study setup can be seen in Figure 5.4

5.3.5 Procedure

At the beginning of the experiment, participants were welcomed, introduced to the purpose of the study and were given instructions on the tasks, they had to perform. The participants were told to do the tasks exercises as fast and accurate as possible (eyes-free interaction was not required). Subsequently, participants partook in a short driving practice to become accustomed to the driving task.

For each driving mode the participants had to test each overlay. For every overlay they completed a set of discrete and serial task, which were once instructed on the cluster display and once on the steering wheel touchscreen. Each participant completed a total of 1104 taps (3 driving modes $\times$ 4 overlays $\times$ 2 instruction sources $\times$ 15 discrete taps and 31 serial taps). In addition, qualitative feedback was collected, using questionnaires and short interviews. The whole test lasted for 50 minutes per participant. The presentation order for different overlays and driving modes was counterbalanced.

The driving simulation software offered full support for the *manual driving mode* with an automatic gearbox. For simulating the *cruise control mode*, where participants only had to steer, another person in the background was observing the simulation and manipulating the speed and distance to other vehicles. The same principle was used to simulate the *autonomous mode*, where the complete driving task was done by another person in the background.

To measure the participant's task performance, we logged the time that the participant needed for completing one task and the correctness of it. Additionally, driving mistakes were followed and logged. After completing all tasks, participants were given a follow-up questionnaire, specifically for comparing different overlay and visualization sources.

5.4 Quantitative Results

This section presents quantitative results of the user study evaluating overlays, driving modes, instruction sources and task types. For each of those variables, error rates and task completion times were collected and analyzed with an repeated measures ANOVA. Subsequently a post-hoc analyses was conducted, which consisted of paired-samples t-tests. For all statistical tests an alpha level of 0.05 was used. The task completion time was recorded and presented in milliseconds for precision.

5.4.1 Task Completion Times

Table A.2 shows the average task completion times and standard deviations, between different variables that were used in the study. It shows that reducing the driving workload also reduces task completion times, on average for 134 ms. A significant difference between driving modes can be reported, $F_{2,22} = 6.978$, $p = .004$. A post-hoc analysis showed that while driving in *manual driving* or *cruise control* mode, the task completion times are significantly higher compared to the *autonomous driving mode* (see Table 5.2). Furthermore, we noticed that using a overlay with more tactile feedback reduces the task completion times, on average for 61 ms. A significant main effect can be reported for the overlay comparison $F_{3,33} = 9.224$, $p < .001$. Additionally, the results show that the convex and concave overlays are significantly faster then the plain and ridge overlays (see Table 5.1). Compared to the cluster display, the steering wheel touchscreen resulted significantly faster task completion times $F_{1,11} = 30.583$, $p < .001$, which are on average faster for 111 ms. The biggest difference was between the two task types. Despite the significant difference $F_{1,11} = 78.428$, $p < .001$, this result needs to be excluded form the study, since there were more buttons then just one that needed to be pressed in a serial task, compared to a discrete task and therefore the task completion times were measured differently. Subsequently, the task completion times between correctly and incorrectly pressed buttons was analyzed and the result was not significant, $t(11) = -0.150$, $p = 0.883$.

Pair	F	p
Convex - Concave	$t(11)$ = -.967	0.354
Convex - Plain	$t(11)$ = -4.613	0.001
Convex - Ridge	$t(11)$ = -3.851	0.003
Concave - Plain	$t(11)$ = -5.231	0.000
Concave - Ridge	$t(11)$ = -2.413	0.034
Plain - Ridge	$t(11)$ = 1.386	0.193

Table 5.1: Significant mean differences between pairs of overlay types, in terms of task completion times.

Pair	F	p
Manual Driving - Cruise Control	$t(11)$ = 1.247	0.238
Manual Driving - Autonomous	$t(11)$ = 3.583	0.004
Cruise Control - Autonomous	$t(11)$ = 2.260	0.045

Table 5.2: Significant mean differences between different driving modes, in terms of task completion times.

Table A.2, shows that most of the participants solved the tasks with the same average speed.

5.4.2 Error Rates

In total 13,333 button presses were collected, of which 85 (0.64%) were incorrect. Once the error rate was divided to overlays it revealed that every overlay generated the same amount of errors, so no significant differences were discovered, convex = 21, concave = 23, ridge = 21, plain = 20 (see Figure 5.5). Figure 5.6 shows the error count divided to driving modes, it appears that reducing the driving workload also reduces the error rate, but no significant differences were discovered. Comparing the error rates between the two instruction sources revealed a significant difference $F_{1,11}$ = 5.792, p = .035, which shows that a steering wheel touchscreen, used as an instruction source, generates significantly less errors then the cluster display. Figure 5.7 shows, how the error rate gets reduced by the steering wheel touchscreen in reference to different driving modes, while the cluster display always performs the same.

Given such small error rates, no significant differences could be found for other variables of the user study (e.g. button position, serial task button combination, more parameters combine together).

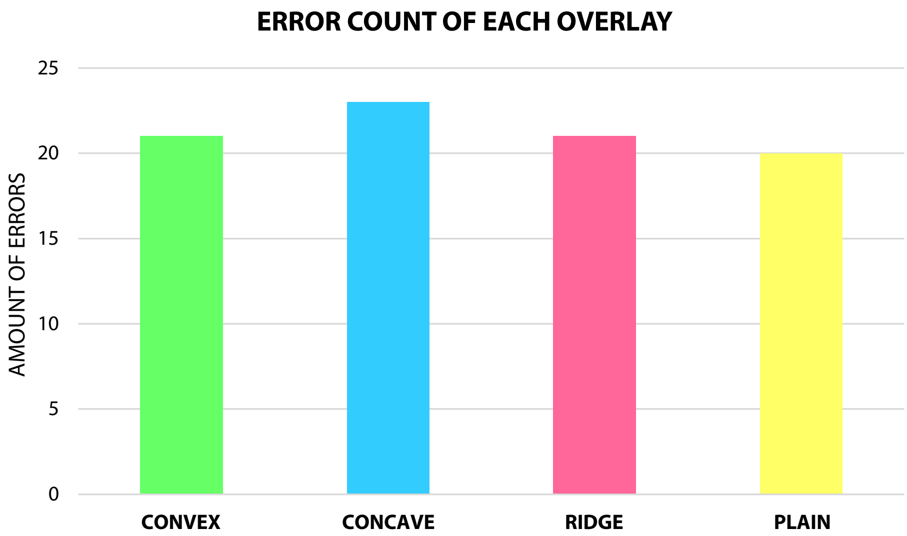

Figure 5.5: Amount of errors generated by each overlay.

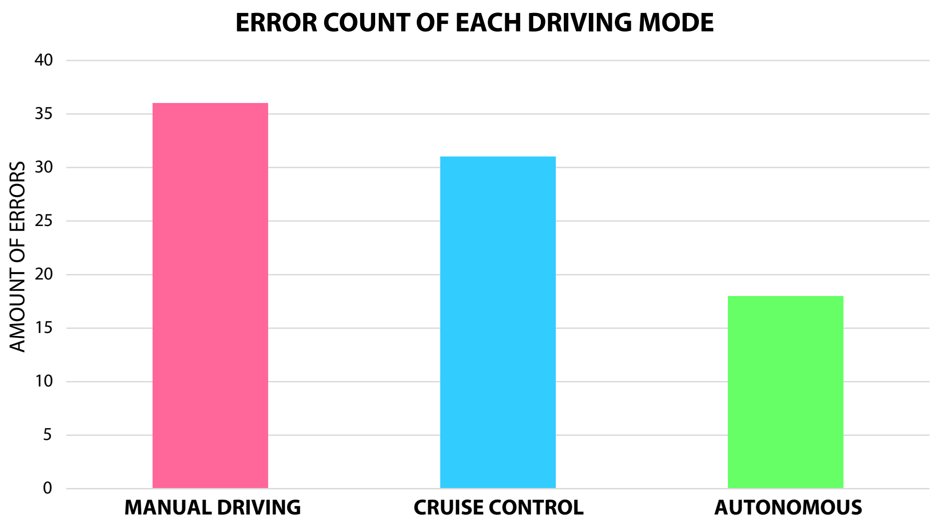

Figure 5.6: Amount of error generated by each driving mode.

5.5 Qualitative Results

In the final questionnaire, participants were asked to rate the overlays, based on a 5-point Likert scale (1 = strongly agree; 5 = strongly disagree). Overall, most of the participants found that overlays with more tactile feedback

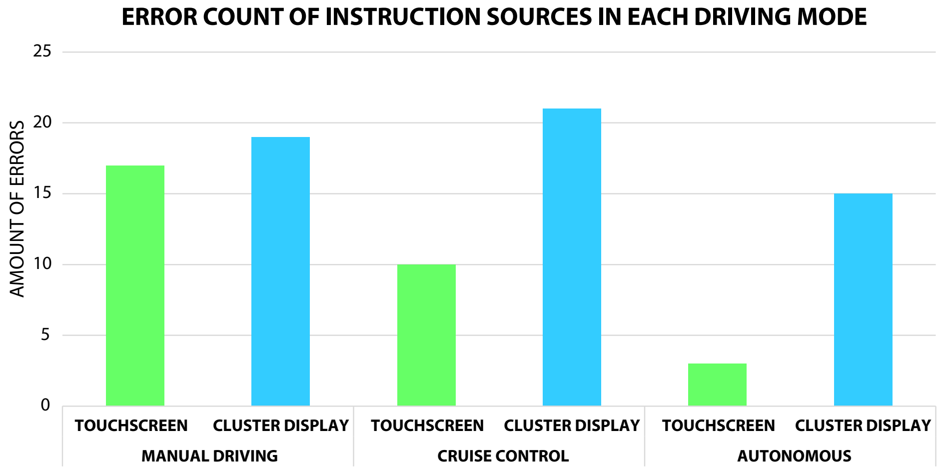

Figure 5.7: Error count of both instruction sources presented in reference to the driving mode.

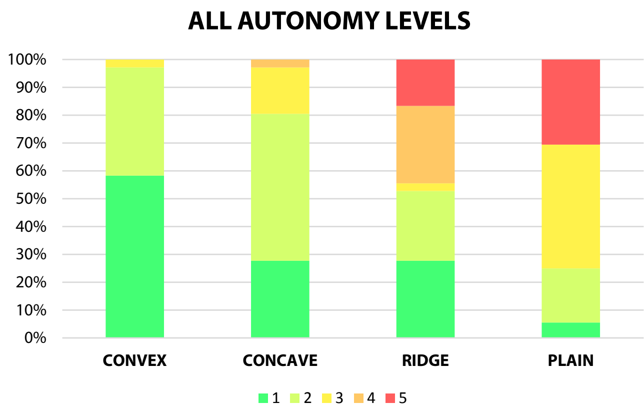

Figure 5.8: Summarized subjective ratings of overlays, through all driving modes.

helped them the more by solving tasks, so they rated them better compared to overlays with less feedback. Generally, participants ranked overlays by the amount of provided tactile feedback, from more to less, independent of the driving mode (see Figure 5.8). When the workload was reduced, in *cruise control mode* and *autonomous mode*, participants rated the overlays with less tactile with a better grade as in *manual driving mode* (see Figure 5.9).

Participants were also asked to rate the two instruction sources. The cluster display was rated better then the steering wheel touchscreen in *man-*

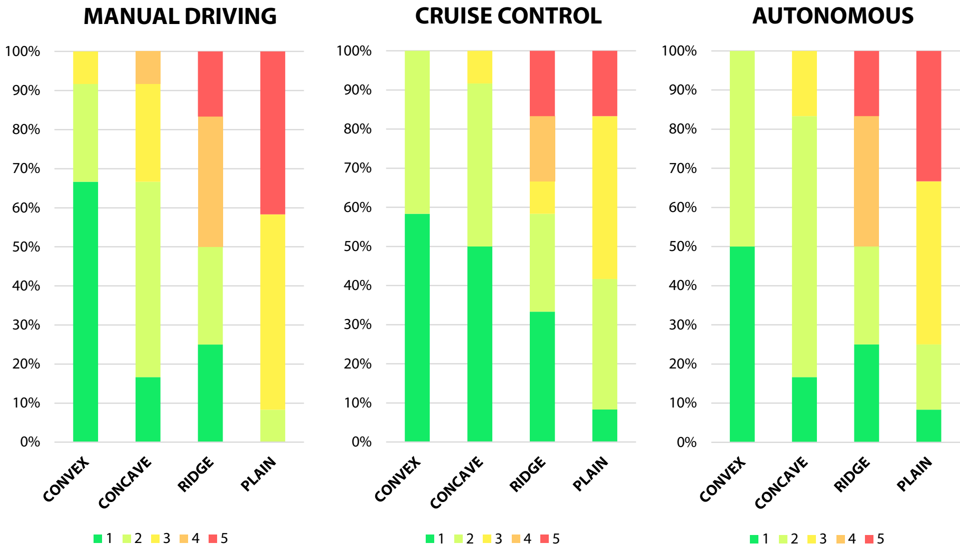

Figure 5.9: Subjective ratings of overlays for each driving mode separately.

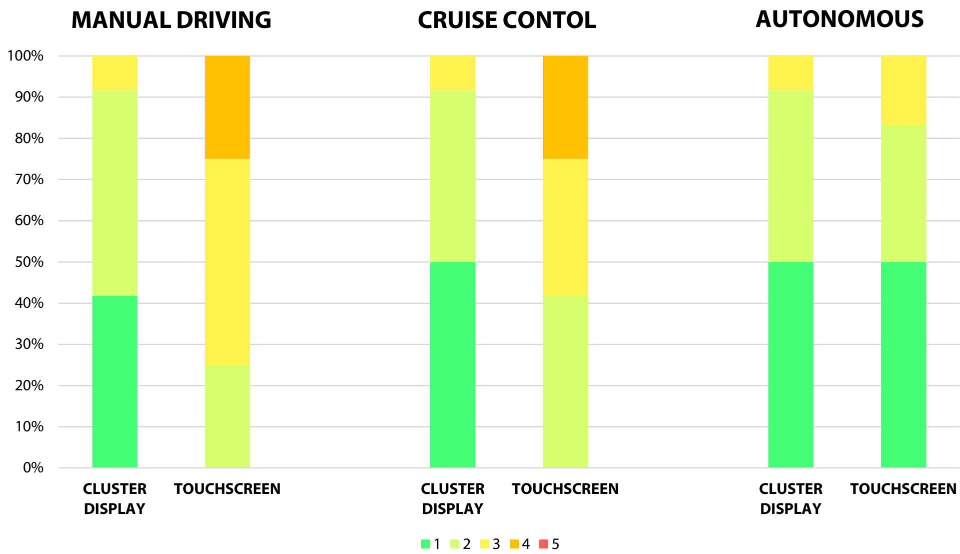

Figure 5.10: Subjective ratings of instruction sources for each driving mode separately.

ual driving mode and *cruise control mode*, where in *autonomous mode* they were ranked equally good (see Figure 5.10). Generally through all driving modes the cluster display was more preferred from the participants (see Figure 5.11).

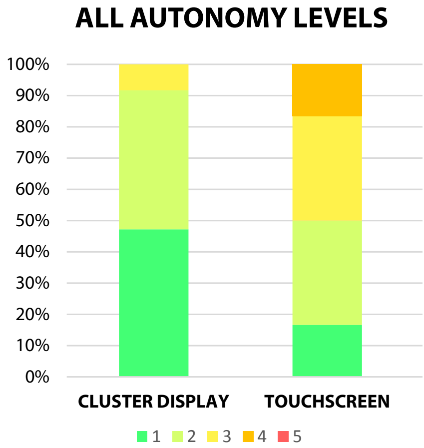

Figure 5.11: Subjective ratings of both instruction sources, through all driving modes.

5.6 Interviews and Observations

From the video footage analysis, we found that four participants were completing the tasks eyes-free by using the convex, concave or ridge overlay, while driving in the *manual driving mode* and while the instruction were presented on the cluster display. Two of these participants were also completing the tasks eyes-free with the plain overlay, which does not provide any tactile feedback. One of the participants completed the tasks eyes-free with the convex overlay, when the instructions were shown on the steering wheel touchscreen and while driving in *manual driving mode*. In other words, the participant was always looking on the road, used the tactile feedback provided by the overlay and read the displayed instructions from the steering wheel touchscreen out of his field of view. In cases where the tasks were not solved eyes-free, participants glanced to the cluster display, then to the overlay, solved the task and then looked back to the road. So they glanced to three different positions before they solved the task. If the instruction was presented directly on the steering wheel touchscreen, they glanced only to two positions, road and overlay. It was noticed that for both task types, discrete and serial, they glanced only once to the overlay, although they needed to press multiple buttons in a serial task. While driving in *autonomous mode*, six participants glanced to the road every three to five seconds, where other participants fully focused their view on the cluster display or touchscreen and were completely unaware of the motorway environment.

Additionally to that, some other behavioral habits were discovered from the video analysis. Five participants used the overlay as their initial position of the thumb, while others removed their thumb from the overlay, after every task and placed them on the steering wheel ring. Resting the thumb directly on the overlay (e.g. on the bubble in the middle of the ridge overlay or in the middle of the four squares on the convex or concave overlay) facilitated the process of continuously searching for his initial position on the overlay, before completing a new task.

During the whole study participants caused six car crashes, three crashes occurred while using the convex overlay. One crash occurred while using the concave, ridge and plain overlay.

At the end of the study participants were asked, if they would accept an interaction device on the steering wheel, even though they do not need to steer with it any more (e.g. *autonomous mode*), most of them agree that this is acceptable (5 participants = strongly agree, 4 = agree, 3 = neutral, 0 = disagree and strongly disagree). When asked, how detached from the driving experience they felt, while driving in *autonomous mode*, the majority answered with the answer completely detached (8 participants = completely detached, 3 = detached, 1 = neutral, 0 = focused completely focused). They were also asked to share some thoughts, while filling out the questionnaires:

- "Convex provides great feedback because I can feel what is a button and what is not a button."
- "Convex and concave are absolutely not the same."
- "A plain display is actually not that bad."
- "Convex is not as good as concave, because the feedback areas are interactive, so I am afraid that I will press something unintentionally."
- "Ridge is the best, I feel everything I need to know to interact with the system fast and precise."
- "On the beginning, I thought ridge is a really clever idea, but later it didn't help much."
- "Especially in the *autonomous mode* all overlays perform the same."
- "Cluster display instruction source is good in the *manual driving mode*, but for everything else is the steering wheel touchscreen better."
- "Steering wheel touchscreen instructions are more handy while leaning back and enjoying the ride."
- "I like the cluster display as an instruction source more evan while driving autonomously, because I like to look outside and in that case its closer to the windshield."

5.7 Discussion

The results of this study have shown, that participants completed the tasks with an unexpectedly low error rates and generally short task completion times. Due to the small error rate of 0.64%, no significant differences were found for the different overlays, driving modes and task types. Exceptionally, it was found out that displaying the instructions directly on the steering wheel touchscreen performed significantly better compared to the cluster display in terms of error rates and task completion times. Significant task completion time differences were also found between different driving modes, overlays and correctly or incorrectly pressed buttons.

In detail, changing the overlay did not effect the error rate. Therefore *Hypothesis 1*, which says that overlays with more tactile feedback generate fewer errors then overlays with less feedback, must be rejected. It was noticed that the accuracy of the single thumb interaction is very high on the steering wheel, since the hand has a fixed and stable hold on the steering wheel ring. This compared to the interaction in a car with a phone, tablet or center console touchscreen, which mostly do not provide tactile feedback or stable hold while drivers still manage to control them, is a great benefit and probably one of the main reasons for those results. Subsequently, the task completion times difference between overlays was significant. It was shown that convex and concave overlays with more tactile feedback result shorter task completion times, compared to the ridge and plain overlay. Therefore we can confirm *Hypothesis 2*.

Completing a serial task did not differ from completing a discrete task in terms of workload, gaze time and error rates. For example, to complete a serial task participants glanced to the overlay, found the reference from their thumb to one of the instructed button, pressed it, looked back to the road and pressed the rest of the buttons eyes-free. While driving in *cruise control mode* or *autonomous mode*, it was noticed that this principle slightly changed, participants felt the lower driving workload, so they spent more time looking at the overlay.

Changing the driving mode to reduce the driving workload, did not reveal a significant difference in error rates and driving mistakes, although the task completion times had a significant difference. Consequently *Hypothesis 3* can be partly confirmed, since the driving workload did not influence the error rates and the task completion times were significantly reduced by lowering of the driving workload. These findings are similar to Mizobuchi's [6] study results, where a personal digital assistant device was used to input text with different sizes of keyboards, while standing and walking on different speeds. It was shown that the workload of walking did not effect the input difficulty. Mizobuchi concluded that walking itself is not a suitable secondary task for assessing the mental workload associated with different text input tasks. Likewise, Kern [5] compared two touchscreen prototypes which where used for handwritten text input. One was placed on the steering wheel and another one in center console, no significant differences in driving performance were noticed between this two conditions. Concluding from that, we can say that driving manually is not a difficult task, which would be difficult to manage, while interacting with a touchscreen on the steering wheel. Even when a more critical scenario was simulated (more traffic, unexpected behavior of other cars, emergency vehicle), participants automatically ignored the primary task and were fully focused on the driving task, until the danger was over and then continued. From the participants observations and questionnaires it was found, that most participants completely disengage from the driving experience, when the car starts to drive on it own. Some of them also did not respond when some extremely dangerous scenarios were simulated (e.g. driving on the other side of the motorway, stopping on the motorway, crashing). A real world simulation would result different results, since acceleration and movement of the car could not be simulated, to additionally inform the driver about the current car actions. Additionally, it was noticed, how the sitting position changed, to more relaxing when driving in *autonomous mode* or *cruise control mode*, this leaned-back position made the steering wheel touchscreen evan more accessible and visible.

A significant difference in error rates and task completion times was noticed while changing the instruction source. The steering wheel touchscreen performed much better then the cluster display. Therefore *Hypothesis 4* can be confirmed. It was shown that driving in the *manual driving mode*, where the driver needs to look at the road, both instruction sources performed equally good, but as soon as the driving mode was changed and driving workload reduced the steering wheel touchscreen started to perform significantly better. We can conclude that the visual instruction directly on the button is very self-explanatory and that it is hard to press an incorrect button, while having this visual feedback.

Generally, it was found out that there is a significant difference between many of the study variables, in terms of task completion time. Although, a trend of time change was discovered and proven significant, the task completions times deferred on average for 102 ms (driving modes = 134 ms, overlays = 61 ms, instruction sources = 111 ms). If we compare this time difference between variables to the time of an average human eye blink (100 to 400 ms [15]), we can argue that these differences are negligible for an interaction process. The average time for all times measured in the study was 1027 ms (SD = 898 ms), which is clearly sufficient for the usage in automotive, according to the NHTSA Driver Distraction Guidelines for In-Vehicle Electronic Devices, where it is recommended, that a single glance duration should not exceed 2 seconds [14]. Additionally, it was noticed that falsely and correctly pressed buttons occur at the same time stamp. This shows that the falsely pressed buttons are not delayed because of some external parameters (e.g. driver distraction, workload change, driving environment change) and are truly falsely pressed buttons.

Overlays with more tactile feedback were rated slightly better by the participants in the questionnaire and *Hypothesis 5* can be confirmed, although it was not noticed, that these ratings would be resembled in the task performance. For *driving modes* where driving was still required (*manual driving* and *cruise control mode*), cluster display was rated better as an instruction source compared to the steering wheel touchscreen. Subsequently, it was shown that the steering wheel touchscreen significantly reduces the error rates and task completion times. We can argue, that participants ranked the overlays and instruction sources by their already existing knowledge and experience. It is very common to glance at a cluster display or an analog dashboard, while displays on the steering do not exist and are very extraordinary. The same goes for buttons with less tactile feedback compared to the well-known pushbutton switches. A similar phenomenon could be noticed in the past, after the removal of physical keyboards from mobile phones. In *autonomous driving mode*, where driving was not required and a new driving experience was presented, both overlays and instruction sources were rated better and were very well expected.

Chapter 6

Conclusion and Future Work

In this work, we presented a new driver-car interaction concept. A multifunctional steering wheel prototype with an embedded touchscreen, interactive surface and hand position tracking was developed. The touchscreen was covered with transparent physical buttons and plastic overlay buttons which provided tactile feedback to the driver. The functionality and visual appearance of these buttons could be personalized according to the drivers needs or adapt to the current driving mode of the car. The interactive leather surface could recognize various swipe or tap gestures and provided haptic feedback in form of vibration impulses. With the interactive surface the driver could control all the functionalities offered by a modern infotainment system. The hand position tracking system could detect the positions of the drivers hand and based on that the drivers awareness level could be determined. Our multifunctional steering wheel prototype addressed three problems that occurred in the automotive industry. The first problem is the increasing number of interaction devices in the car interior. The second one is the customization and personalization of the interior. The third one is the importance of a tracking system by which a cars can determine the awareness level of drivers and then allow them the control of life-sustaining functions.

In the implementation phase we found out that FSR sensor matrix was much easier to embed and implement compared to the PyzoFlex sensor matrix. Linear voice coil actuators provided much stronger and preciser vibration feedback impulses compared to LRA actuators. Furthermore, conductive yarns turned out to be a reliable solution for tracking drivers hand positions on the steering wheel. The final prototype can be seen in Figure 6.1.

After the implementation a user study was conducted, exploring the importance of providing tactile feedback for a steering wheel touchscreen. Four tactile feedback overlays were compared under different driving workloads. Additionally, the cluster display and the steering wheel touchscreen used as visual instruction sources were compared between each other. It was found out that overlays with more tactile feedback did not reduce the error rate compared to the overlays with less feedback or evan no feedback. Reducing the driving workload by changing the driving mode (*manual driving*, *cruise control*, *autonomous*) also did not reduce the error rate. It was shown that the touchscreen on the steering wheel used for visual instructions significantly reduced the error rate compared to the cluster display. All task completion times for different overlays, instruction sources, task types used in the study were in a automotive sufficient range (less then two seconds). Participants subjectively rated overlays with more tactile feedback better then overlays with less feedback, although this ratings were not reflected by the performance measures.

Our user study has shown how the tactile feedback of a steering wheel touchscreen is related to the driving workload and visual instruction positions. This is only one of the questions that appeared during the planning phase of the study, in the future more questions should be investigated in this context (e.g how many functions can be shown on a multifunctional steering without confusing the driver, is there a need to provide additional haptic or visual feedback, can this prototype be improved in terms of interior design). Subsequently, it should be investigated if these technologies used in the prototype, display buttons, interactive surfaces, conductive yarns, can also be used in other interior parts (e.g. center console, doors, seats, floor).

Figure 6.1: Final version of our multifunctional steering wheel prototype.

Appendix A

Study Data

	BMW	Mercedes Benz	Audi	VW	Opel	Volvo	Ford	Toyota	Lexus	Honda	Citroen	Fiat	Kia	Mazda	Renault	Skoda	Seat	Alfa Romeo
Cruise control - On/Off	Left			Left	Left	Left	Left			Right	Left		Right	Right	Left			
Cruise control - Reset	Left			Left	Left	Left	Left			Right	Left		Right	Right	Right			
Cruise control - Set	Left			Left	Left	Left	Left			Right	Left		Right	Right	Right			
Cruise control - Speed Up	Left (R)			Left	Left (R)	Left	Left			Right	Left		Right	Right	Left			
Cruise control - Speed Down	Left (R)			Left	Left (R)	Left	Left			Right	Left		Right	Right	Left			
Driving assistant - Increase distance	Left			Left	Left		Left			Right								
Driving assistant - Reduce distance	Left			Left	Left		Left			Right								
Driving assistant - Limiter - On/Off	Left			Left	Left					Right								
Traffic jam assist - On/Off	Left																	
Return to main menu		Left								Right	Right							
Menu go in			Left	Right													Right	
Menu go out			Left	Right					Right	Left			Right			Left	Right	
Back		Left	Left			Right		Left	Right	Left	Both		Right			Right		
OK		Left	Left (S)	Right			Both	Left	Right									
Arrow up		Left	Left (S)				Both	Left	Right		Both		Left	Left		Right (S)	Right (S)	
Arrow down		Left	Left (S)				Both	Left	Right		Both		Left	Left		Right (S)	Right (S)	
Arrow left			Left	Right			Both	Left	Right	Left						Left		
Arrow right			Left	Right			Both	Left	Right	Left						Left		
Entertaiment source	Right				Right			Left	Left	Left		Right	Left					
Media arrow left			Right			Right	Right				Right	Right		Left			Left	
Media arrow right			Right			Right	Right				Right	Right		Left			Left	
Media arrow up	Right (R)			Right	Right	Right (S)			Left									Right
Media arrrow down	Right (R)			Right	Right	Right (S)			Left									Right
Volume up	Right	Right	Right (S)	Left	Right	Right	Right	Left	Left	Left	Right	Left (S)	Left	Left		Left (S)	Left (S)	Left
Volume down	Right	Right	Right (S)	Left	Right	Right	Right	Left	Left	Left	Right	Left (S)	Left	Left		Left (S)	Left (S)	Left
Mute		Right	Right (S)		Right	Right	Right				Right	Left					Left	Left
Voice activation On	Right	Right	Right	Right	Right		Right	Right	Left	Left			Left	Left			Right	Left
Voice activation Off		Right			Right													
Phone On	Right	Right	Right	Right	Right		Right	Right	Left	Left		Left	Left	Left		Right	Left	Right
Phone Off		Left					Right	Right	Left	Left		Left	Left	Left				Right
Navigation comand repeat			Right															
Custom button			Right															
Play / Pause			Right									Right						
Change custer view/template			Left					Right										

Right side | Left side | Both sides | R = Rocker switch | S = Scroll wheel

Table A.1: Functions and their positions on modern multifunctional steering wheels.

Driving Mode	M	SD
Manual Driving	1081	647
Cruise Control	1034	597
Autonomous	880	424
Overlay Type	M	SD
Convex	949	535
Concave	967	543
Plain	1052	609
Ridge	1025	586
Task Type	M	SD
Discrete	807	506
Serial	1091	577
Instruction Source	M	SD
Buttons	943	583
Cluster Display	1054	553
Task Correctness	M	SD
Correct	999	571
Incorrect	935	473
User	M	SD
1	931	531
2	909	481
3	1044	654
4	1097	504
5	890	475
6	1172	631
7	900	363
8	847	476
9	1365	920
10	835	444
11	991	402
12	1000	516

Table A.2: Task completion times and its standard deviations presented in milliseconds for each variable used in the user study.

Appendix B

Technical Information

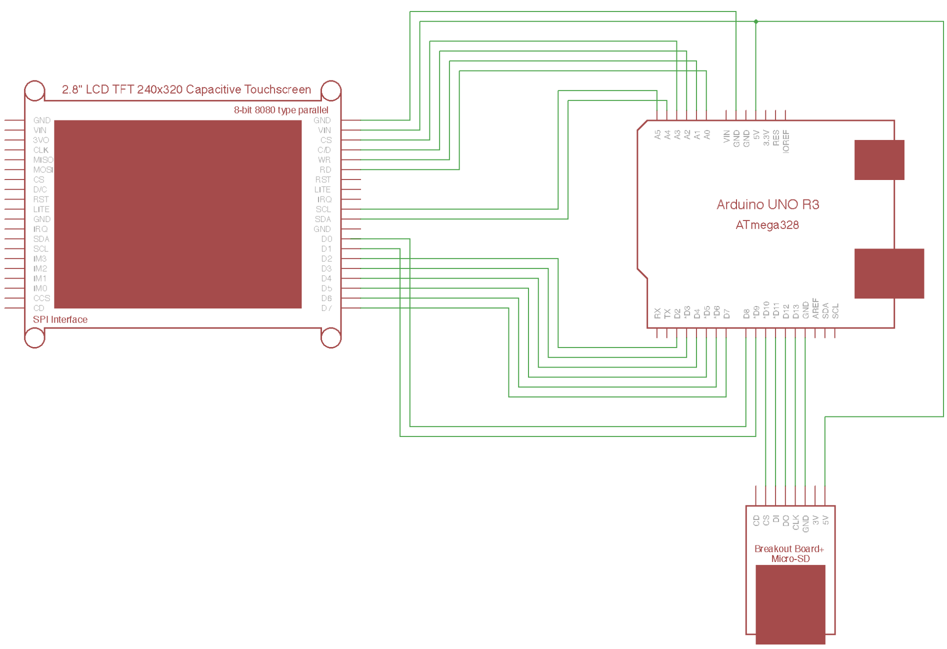

Figure B.1: Schematic of the touchscreen, microSD card reader and Arduino Uno used for the display buttons prototype.

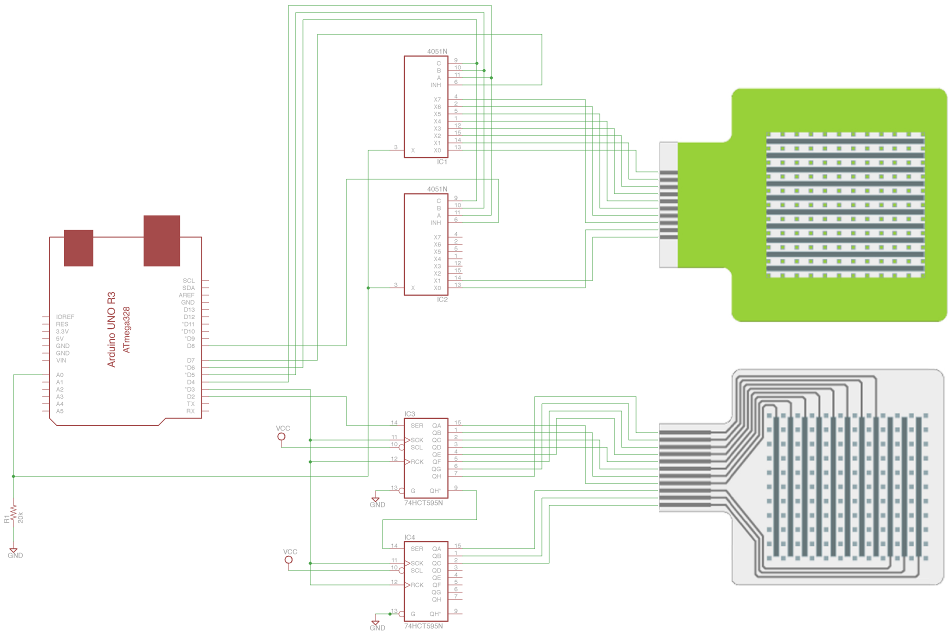

Figure B.2: Schematic of the FSR sensor matrix, multiplexers, switch registers and Arduino Uno used for the FSTP prototype.

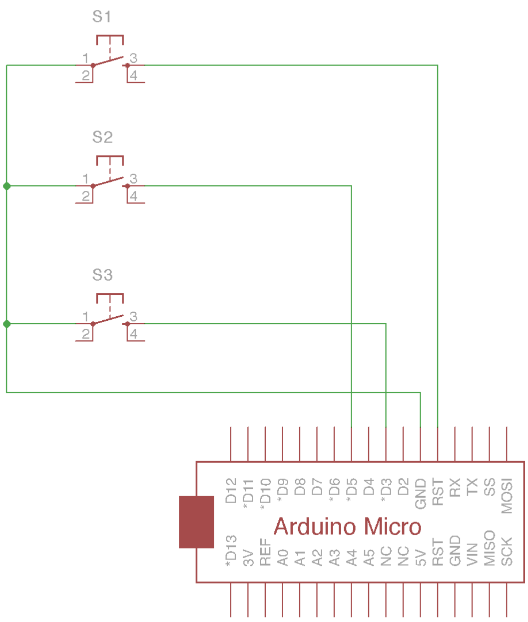

Figure B.3: Schematic of the three pushbutton switches used for the menu section in the display buttons prototype.

References

Literature

[1] Leonardo Angelini et al. "Gesturing on the Steering Wheel: A User-elicited Taxonomy". In: *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. AutomotiveUI '14. Seattle, WA, USA: ACM, 2014, 31:1–31:8 (cit. on p. 3).

[2] David Bonnet, Caroline Appert, and Michel Beaudouin-Lafon. "Extending the Vocabulary of Touch Events with ThumbRock". In: *Proceedings of Graphics Interface*. GI '13. Regina, Sascatchewan, Canada: Canadian Information Processing Society, 2013, pp. 221–228 (cit. on p. 26).

[3] Tanja Döring et al. "Gestural Interaction on the Steering Wheel: Reducing the Visual Demand". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '11. Vancouver, BC, Canada: ACM, 2011, pp. 483–492 (cit. on pp. 3, 4).

[4] Iván E. González et al. "Eyes on the Road, Hands on the Wheel: Thumb-based Interaction Techniques for Input on Steering Wheels". In: *Proceedings of Graphics Interface 2007*. GI '07. Montreal, Canada: ACM, 2007, pp. 95–102 (cit. on pp. 1, 3, 4).

[5] Dagmar Kern et al. "Writing to Your Car: Handwritten Text Input While Driving". In: *CHI '09 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '09. Boston, MA, USA: ACM, 2009, pp. 4705–4710 (cit. on pp. 3, 11, 43).

[6] Sachi Mizobuchi, Mark Chignell, and David Newton. "Mobile Text Entry: Relationship Between Walking Speed and Text Input Task Difficulty". In: *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices &Amp; Services*. MobileHCI '05. Salzburg, Austria: ACM, 2005, pp. 122–128 (cit. on p. 43).

[7] Pekka Parhi, Amy K. Karlson, and Benjamin B. Bederson. "Target Size Study for One-handed Thumb Use on Small Touchscreen Devices". In: *Proceedings of the 8th Conference on Human-computer In-*

teraction with Mobile Devices and Services. MobileHCI '06. Helsinki, Finland: ACM, 2006, pp. 203–210 (cit. on p. 14).

[8] Christian Rendl et al. "PyzoFlex: Printed Piezoelectric Pressure Sensing Foil". In: *UIST '12: Proceedings of the 25th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM, Oct. 2012, pp. 509–518 (cit. on p. 16).

[9] Sandra Trösterer et al. "Using a Parking Assist System Over Time: Insights on Acceptance and Experiences". In: *Proceedings of the 6th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. AutomotiveUI '14. Seattle, WA, USA: ACM, 2014, 22:1–22:8 (cit. on p. 4).

[10] Thomas A Ulrich et al. "Playing charades with your car–The potential of free-form and contact-based gestural interfaces for human vehicle interaction". In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. Vol. 57. 1. SAGE Publications. 2013, pp. 1643–1647 (cit. on pp. 3, 14).

[11] Yuntao Wang et al. "Understanding Performance of Eyes-free, Absolute Position Control on Touchable Mobile Phones". In: *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services*. MobileHCI '13. Munich, Germany: ACM, 2013, pp. 79–88 (cit. on p. 14).

Online sources

[12] City Car Driving. June 2015. URL: http://citycardriving.com/ (cit. on p. 34).

[13] NHTSA. Preliminary Statement of Policy Concerning Automated Vehicle. May 2013. URL: http://www.nhtsa.gov/staticfiles/rulemaking/pdf/Automated_Vehicles_Policy.pdf (cit. on p. 34).

[14] NHTSA. Visual-Manual NHTSA Driver Distraction Guidelines for In-Vehicle Electronic Devices. Feb. 2012. URL: www.nhtsa.gov/staticfiles/rulemaking/pdf/Distraction_NPFG-02162012.pdf (cit. on p. 44).

[15] Wikipedia. *Blink — Wikipedia, The Free Encyclopedia*. [Online; accessed 10-June-2015]. 2015. URL: http://en.wikipedia.org/w/index.php?title=Blink&oldid=660008964 (cit. on p. 44).

[16] Wikipedia. *Force-sensing resistor — Wikipedia, The Free Encyclopedia*. [Online; accessed 18-June-2015]. 2015 (cit. on p. 18).

[17] Wikipedia. *Metallic fiber — Wikipedia, The Free Encyclopedia*. [Online; accessed 17-June-2015]. 2015. URL: https://en.wikipedia.org/w/index.php?title=Metallic_fiber&oldid=641537836 (cit. on p. 15).

Messbox zur Druckkontrolle

— Druckgröße kontrollieren! —

Breite = 100 mm
Höhe = 50 mm

— Diese Seite nach dem Druck entfernen! —