NTNU
Norwegian University of
Science and Technology

The Smart Chair

Inducing dynamic seating through integration
of smart technology in chairs

Osmund Olav Bøe

Master of Science in Mechanical Engineering
Submission date: July 2017
Supervisor: Martin Steinert, MTP

Norwegian University of Science and Technology
Department of Mechanical and Industrial Engineering

Omnia in gloriam Dei fac.

Abstract

The world is becoming increasingly sedentary. Many jobs centered around physical work are being automated, and the time spent physically active is sinking. Various CPM (Continuous Passive Motion) chairs, and other chairs designed for increasing the activity-while-seated level have been made, and research on them are somewhat conflicted, with a majority not finding a statistically significant difference in the in the level of activity while seated (L. L. van Deursen et al. 1999; O'Sullivan, O'Sullivan, O'Keeffe, et al. 2013; O'Sullivan, O'Keeffe, et al. 2012; Aota et al. 2007; Groenesteijn et al. 2012).

Building on the previous work done the project thesis (appendix D) , and in TMM4245 The Fuzzy Front end 2016, two prototypes have been built capable of classifying sitting postures, estimating activity levels and altering tilt resistance. This thesis documents the building of the two prototype chairs, designed to test the effect of automated tilt resistance on a user's level of activity while seated. One chair is built to be immediate, and the tilt resistance alter the neutral angle of the chair, the other with a subtler change.

A small pilot test was performed ensuring that the prototypes were working as intended, but the limited testing did not yield enough data to determine the effects either prototype could have on seating behavior. Further field tests are strongly recommended. It is also recommended to implement a reinforcement learning algorithm to the tilt resistance setting. The result could be very interesting both commercially, and with regards to research on human computer interaction.

Sammendrag

Verden blir stadig mer stillesittende. Mange fysiske jobber blir automatisert, og daglig fysisk aktivitet synker. Forskjellige stoler med kontinuerlig passiv bevegelse, ekstra frihetsgrader, og andre stoler laget med hensikten å fremme fysisk aktivitet mens man sitter har blitt introdusert på markedet, og forskning på effektiviteten til slike produkt er noe motstridende, men majoriteten har ikke funnet statistisk signifikante forskjeller i aktivitetsnivå på disse, kontra «vanlige» stoler. (L. L. van Deursen et al. 1999; O'Sullivan, O'Sullivan, O'Keeffe, et al. 2013; O'Sullivan, O'Keeffe, et al. 2012; Aota et al. 2007; Groenesteijn et al. 2012).

Denne masteravhandlingen dokumenterer byggingen av to prototyp-stoler, designet for å teste effekten automatisering av vippemotstand-innstillingen har på aktivitetsnivået mens man sitter. Den ene prototypen endrer vippemotstand umiddelbart, og endrer også vinkelen på setet, den andre har en roligere, mindre direkte endring av vippemotstand.

Basert på tidligere arbeid i prosjektoppgaven «The Smart Chair» (appendix D) og faget TMM4245 The Fuzzy Front end er det bygget to prototyp-stoler, kapable til å klassifisere sittestillinger, å estimere aktivitetsnivå og å justere vippemotstand.

En pilot-test ble gjennomført for å forsikre om at prototypene virker som de var ment, men det begrensede omfanget til pilot-testen gav ikke nok data til å avgjøre effekten en slik stol kan ha på sittende oppførsel og ergonomi. Videre laboratorietester og felttester anbefales på det sterkeste. Videre anbefales det også å implementere en «reinforcement learning»-algoritme for å styre vippemotstanden, da resultatet kan bli interessant både i kommersiell setting, og med tanke på forskning på menneske-datamaskin interaksjon.

Preface

This master's thesis was written during the spring semester of 2017 at NTNU, Faculty of Engineering, Department of Mechanical and Industrial Engineering. Business sponsor was Flokk.

This master thesis is a continuation of the pre-master's project "The Smart Chair", beginning as a project in TMM4245 "The Fuzzy Front End".

I am deeply thankful to my counselor Martin Steinert, for taking time to answer questions despite a busy schedule. Flokk for sponsoring and supporting the project. I am also very thankful for all help from my co-counselor Andreas Wulvik. Even Jørs and Phillipp Anders for the cooperation in TMM4245. Last, but certainly not least, I want to thank my parents for their continuous support of all the choices that lead me here.

Osmund Olav Bøe

Trondheim, July 2017

Table of Contents

List of Figures

List of Tables

1. Introduction

Inducing dynamic seating

Various CPM (Continuous Passive Motion) chairs, and other chairs designed for increasing the activity-while-seated level have been made, and research on them are somewhat conflicted, with a majority not finding a statistically significant difference in the in the level of activity while seated (L. L. van Deursen et al. 1999; O'Sullivan, O'Sullivan, O'Keeffe, et al. 2013; O'Sullivan, O'Keeffe, et al. 2012; Aota et al. 2007; Groenesteijn et al. 2012).

To the best of the authors knowledge, the features from the smart, learning chairs, and the chairs aimed at increasing activity-while-seated levels have never been combined. That is, a chair that reacts to, and adapts to the anatomy and behavior of the sitter, therefore, the focus of the master's thesis was chosen to be the building and testing of a smart adapting chair.

There are two factors perceived by the author as most important to achieve active, dynamic seating. First, the height of the chair, which if set too high immobilizes the user to some degree, and reduce blood circulation to the legs, and if set too low, would leave the sitter leaning backwards, which severely inhibits activity-while-sitting. Secondly, the tilt resistance of the chair, which determines the angle of the seat and, or back, for a given torque, or the back-support at a given seat angle. Another important chair setting is seat depth, which is important for blood circulation to the legs, and in some chairs, will also affect the tilt dynamic.

Assignment

The aim of the master's project is to design, build, test and evaluate the commercial potential of a prototype smart chair. The prototype will feature automated height setting (largely in collaboration with a Fuzzy Front End team), to find appropriate height of the seat. Other settings vital to facilitate good sitting behavior, as perceived by seating experts, are to be measured, and guided or automated. We intent to measure the effect on seating behavior of: altering the tilt resistance, locking the tilt-functionality, and varying tilt resistance after prolonged sitting, and whether active seating can be induced by periodically altering the tilt functionality after prolonged sitting. Industry sponsor is SBS (Flokk) and contact person is Christian Lodgaard.

Reevaluated assignment

Well into the development it became clear that the nature of tilt mechanisms varies greatly between different chair models, a fact that should be better reflected in the task. A change in tilt resistance might affect behavior differently in a chair with a central pivot point, and one where the pivot point is placed in the front of the seat. In some tilt mechanisms, a change in tilt resistance is immediate, and will alter the seat angle, and in others, the change would not take effect until the chair is at a neutral angle, and not affect seat angle at all. The focus of the project was shifted to building two chairs, to accommodate research on the effect tilt resistance settings have on sitting behavior. One chair, HÅG Capisco, with immediate effect, and another, HÅG SoFi, with the "BalancedMovementMechanism™", leading to a subtler approach when the setting is automated.

Preceding Work

This master's thesis, along with the preceding premaster's project is a continuation of a project from the course TMM4245 The Fuzzy Front End, spring 2016. In the course, three students, including the author made a prototype chair that utilized force-sensitive resistors to measure pressure distribution in the seat and backrest of the chair. Based on the pressure data, sitting postures could be classified through machine learning in real time. Based on sitting data over time, feedback was generated, with a summary of the day's seating, as well as advice on how to sit in a healthier manner. Studies show that these messages have potential of influencing activity and behavior (Weering, Vollenbroek-Hutten, and Hermens 2012; O'Sullivan, O'Sullivan, O'Sullivan, et al. 2013).

Pre-master's project and further development

In the pre-master's project, possible applications of such a chair were evaluated further, with a focus on what others were doing in the field of smart seating, and research on sitting behavior and low back pain. The former found that data gathering, classification and data-generated feedback was the focus of several seating products and wearable electronic devices, like dorsaVi's ViMove™, sensomative office, and BMA's axia smart chair. The latter found that the research is conflicted as to whether there are significant links between sitting and LBP (Gupta et al. 2015; Hartvigsen J et al. 2000; Roffey et al. 2010; Vergara and Page 2002).

Fuzzy Front End 2017

A group in TMM4245 The Fuzzy Front End proposed a design for motorization of the height setting of chairs with a central pivot point. Due to time constraints, and the built prototype not being able to lift a seated person, the concept was not implemented in any of the prototypes in this thesis.

2. Theory

Low Back Pain (LBP)

In the following, theory and research on LBP is presented. Research on what can help prevent low back pain, mechanisms involved in LBP, rehabilitation, what products exist, and their usefulness has shaped the aim of this project. The main outcome being that there are chairs, and wearables marketed towards sufferers of LBP. Some who actively report on the users seated behavior, through general or custom advice, or through instant haptic feedback ("Axia Smart Chair – Your Personal Posture Coach" 2017, "Sensomative Office – Sensomative" 2017; Kent, Laird, and Haines 2015). Products like them showing some positive effects(Kent, Laird, and Haines 2015). Other, seat based products utilizing extra degrees of freedom or a motor causing continuous passive motion (CPM), showing little to no effect as a standalone measure (Ellegast et al. 2012; O'Sullivan, O'Keeffe, et al. 2012). To the best of the authors knowledge, no device has been made that effectively betters back health without involving the user's attention.

Causes

The cause(s) of LBP are often unknown. LBP can among others be caused by muscle strain or a vertebrae disk compressing a spinal nerve root, on rare occasion due to underlying disease(Goodman, Burke, and Livingston 2013). The reasons for developing low back pain are varied, and the mechanisms involved in the pain also varies greatly.

Prevention and rehabilitation of LBP

According to the European guidelines for prevention of LBP (Burton 2005) the research done on LBP concludes that physical exercise and information and education, from "back schools" are beneficial, and the research on most other areas have insufficient evidence to be recommended as a preventative measure for LBP. Including length correcting insoles for those with different length legs, the use of lifting belts, lumbar supports, specific chairs, or mattresses. A study on the effectiveness of educational materials to prevent occupational LBP (Shorthouse, Roffi, and Tack 2016) found that educational material alone did not seem to reduce sickness absence with manual labor workers, but could facilitate behavioral change. Overall, there are many different subcategories of LBP, with and the causes vary. Not all work-related cases can be prevented, but in general it becomes less likely for back pain to occur with people that exercise regularly, and are take time to evaluate the ergonomics of his/her work station. The guidelines are also relevant for rehabilitation, which usually takes between 2 and 12 weeks ("15 Things You Didn't Know about Back Pain - Independent.ie" 2017).

LBP and links to sedentary behavior

With regards to sitting, and sitting posture, physiotherapists mostly agree that a neutral spine, and what appears comfortable would be advantageous, though many disagree what constitutes a “neutral” spine, and what comfortable would entail(O’Sullivan, O’Sullivan, et al. 2012). The most common advice from experts is that “the best position is the next position”. Though it should be mentioned that increased variability in sitting postures was observed with pain developers in a test, and maladaptive sitting behavior might be a cause of the pain development. However, this test was conducted among people already suffering from NSCLBP (non-specific chronic low back pain), and regarded temporary discomfort (O’Sullivan, O’Sullivan, O’Sullivan, et al. 2013). It was also suggesting that customization of the seat to avoid the end range extension or flexion of the back might benefit as one component of rehabilitation (O’Sullivan, O’Sullivan, O’Sullivan, et al. 2013).

Research on seat based LBP prevention and rehabilitation tools

Over the course of the last two decades, several new products that aim to prevent, treat, or provide pain relief. Among them are chairs specifically trying to achieve “dynamic sitting”, defined as “’dynamic sitting’ (…) relates to the increased motion in sitting which is facilitated by the use of specific chairs or equipment” (O’Sullivan, O’Sullivan, O’Keeffe, et al. 2013). Such specific chairs range from chairs with extra degrees of freedom, to CPM chairs, where a motor continuously alters the seat in some way. The effectiveness of these chairs are somewhat debated, but for the most part, any benefits are statistically insignificant or of marginal value (Aota et al. 2007; D. L. van Deursen et al. 2000; Ellegast et al. 2012; O’Sullivan, O’Sullivan, O’Keeffe, et al. 2013; O’Sullivan, O’Keeffe, et al. 2012).

A newer study on the effect of the HÅG SoFi (Grooten et al. 2017)found that “Dynamic chairs may facilitate movements compared to conventional chairs.” with some inconsistencies between the outcome measures. Other findings were consistent with earlier research, in that the tasks performed have a greater impact on activity level than a chair does. They also found a difference in precision in a computer mouse task, with unlocked tilt mechanism being more efficient than with locked tilt mechanism. Locked tilt mechanism was in turn more efficient in the task than standing.

Machine Learning

Machine learning is a subset of A.I. regarding algorithms designed to give computers the ability to learn from data. Most often, machine learning is divided into three categories: Unsupervised learning, supervised learning, and reinforcement learning.

Supervised learning is used for predicting values based on previously labeled data. That is, prediction of discrete classes, continuous values through respectively classification and regression. The aim is to find a model that predict unseen data accurately based on the existing, labeled data.

Unsupervised learning deals with data that is not labeled, and can be used for dimensionality reduction, finding some common patterns, or hidden structures from unlabeled data, clustering data and so forth.

Reinforcement learning is somewhat similar to supervised learning, the difference being that there is no correct label or value, rather a reward function evaluating the action. Through exploring with trial and error or planning, actions that maximize the reward are gradually uncovered (see figure 2-1) (Raschka 2015). An example being a chess engine, learning by doing, and failing, with the indication of what are good moves given by "Win/loose" at the end of the game.

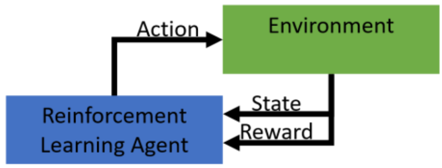


Figure 2-1: Reinforcement learning visualization

In this project, classification builds the foundation, unsupervised learning is used to test if there are sitting postures present in testing, that were not recorded and labeled. Reinforcement learning is not implemented, but is what would have been prioritized if more time was available.

3. Technology

Two prototypes with different tilt mechanisms were built. In the following, the technology used is presented, the product development processes are described, and a basic understanding of how the prototypes work is be provided.

Arduino

The Arduino is a microcontroller enabling easy prototyping of electronics and mechatronics. The Arduino is open source, based on the C++ programming language, and has a vast community of contributors, and has an extensive library of tutorials and example codes. Has ports for digital (binary) input/output at $5V$ logic, and a 10bit analog input (input of 0v-5v, translated to integers 0-1023). Can communicate through among others, USB-serial interface and the I^2C protocol. Arduinos come in different shapes and sizes, with different number of ports, and clock speeds.

For development purposes, and to work with a range of external devices, such as Windows, or a Mac, the Arduino platform is versatile, and flexible.

Raspberry Pi

Raspberry Pi Is a lightweight, compact computer with a footprint roughly the size of a credit card. Various operating systems can be booted to a Raspberry Pi, most notably a customized Debian distribution "Raspbian". The Rapberry Pi has multiple digital GPIO (General Purpose Input Output) pins, with $3.3V$ logic. It relies on a micro SD-card to work as its secondary memory, allowing for easy cloning of the entire secondary memory, significantly reducing marginal efforts of building more prototypes. The Raspberry Pi comes in several different models, with different clock speeds, wireless capabilities, and sizes.

With the addition of some analog to digital converters, all of the tasks performed by an Arduino in this project could be performed through the Raspberry Pi's native GPIO, along with two analog to digital converters.

Sensor and sensor data

Force Sensitive Resistor (FSR)

The FSR (figure 3-1) is utilized in both prototypes. The sensor consists of a “maze” of a conductive material on a less conductive foam. The conduciveness of the foam increases when compressed, resulting in the sensor’s resistance dropping.

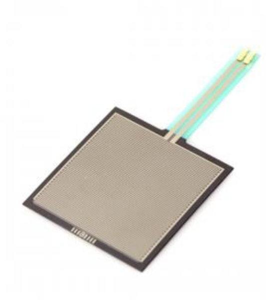


Figure 3-1:
Force Sensitive Resistor

Resistive divider

A special case of the more general voltage divider, where all inductances are purely resistive, as is the case with the FSR. A resistive divider is used to get a measure of the force exerted on the FSR as a voltage, rather than as a resistance, to accommodate for measurements on an Arduino. The voltage output of a resistive divider is given by equation the equation

$$V_{out} = V_{in} * \frac{R_2}{R_1 + R_2}$$

Where the FSR is represented as R_1,and R_2 represents a constant resistance of $3.3k\Omega$. The force-resistance ratio of the FSR is not constant, and the resistance of $3.3k\Omega$ was chosen empirically, as it gave the highest resolution in the force ranges sensed through a cushion when seated.

Accelerometer-Gyroscope

An accelerometer is a device that measures proper acceleration, which in practice means acceleration with respect to free fall. Some accelerometers measure proper acceleration in three axes. When stationary, an accelerometers output can be used to determine orientation of an object, though unable to distinguish between the earths gravitational pull, and acceleration. A gyroscope can be used to measure angular velocity. Using both, and processing the output, one can get accurate data on orientation and acceleration.

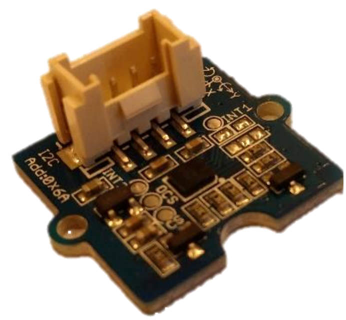


*Figure3-2*Accelerometer-Gyroscope module

4. Method

The design and building of a dynamic, adapting chair inquires the need for several engineering disciplines, including electronic design in circuit boards, mechanical and mechatronic aspects, computer science and programming, as well as the human centered aspects of ergonomics, psychology, and design involved in product development. In short, a complicated system to interact with a complex environment. To manage these different aspects, obtain valuable knowledge as early as possible, and work more efficiently, inspiration was taken from several product development methodologies, though adapted to better fit the scope of this project. The most relevant being agile software development, and the modular product architecture as presented in "Flexible Product Development" (Smith 2007).

The technical challenges were divided into two different categories, those involved in classification, only entailing small customization from the solutions in the preceding work, and those regarding automation of the tilt resistance setting. These two categories were again divided into critical functions.

Classification was achieved by adapting the solutions from previous work to best fit the chairs in question.

Automating tilt resistance was divided into three modules. M*echanism,* for mounting a motor, and converting the rotational movement of the motor to the appropriate form to alter tilt resistance. *Motor control,* for controlling motor digitally without damaging the components. *Positioning* for finding the current tilt resistance level. The end result of these modules being digital control of the motor position through an Arduino.

When prototyping, the approach is most like that found in a modular architecture. The modular architecture consists of modules, connecting strongly within themselves, but not much with other modules, or are completely isolated (Smith 2007). This approach was meant to reduce coupling between modules, and limit rework by ensuring flexibility between modules. A change in one of the modules was not necessarily to affect another module. For the physical prototyping, this meant motor control and positioning all were prototyped and tested separately. This approach is suitable for the earlier stages of product development, to learn, to communicate, to use for testing purposes. A redesign to an integrated product architecture is often advised before production to increase performance, reduce size, and to reduce interfaces, which often can cause problems. In physical prototypes, bolts can slip, in software development, they can cause unnecessary delays, and electromagnetic fields and loose wires can cause problems in electronic circuits (Smith 2007). In coding, this meant to reduce the code into several functions, and calling on them in turn in a script, rather than coding all in to one script. In each case, both with the physical prototypes, and in coding, a main body, being either the chair, or the main script.

On a lower level, a module was often built with a various number of iterations until working well both independently, and with the other modules. Each iteration was meant to test a concept, or increase fidelity, or to work better with the other modules leading up to a final integration on the chair. Earlier prototypes usually being made rapidly, to test one, or a few aspects, and later prototypes, being more complicated was developed more carefully.

There are not only technical aspects to consider in this project. The Human factor might be the greatest unknown. Testing a new feature, or the finished product with users might be what was most important for this project. This type of experience prototyping was used to test users reactions, especially to changed tilt resistance on the HÅG Capisco, as it would "lift them" slightly. Small tests like these would answer small questions, and elicit feedback. Larger scale tests are still needed to answer the more overlying question of if and how sitting behavior is affected by an automated tilt resistance.

5. SoFi

Description of chair

The HÅG SoFi base model is an advanced chair with regards to mechanisms, to provide adjustable chair height, seat depth, lumbar support, arm rests adjustable in three dimensions, variable tilt resistance, and tilt locking. The effectiveness of the tilt mechanism "BalancedMovementMechanism™" has been the subject of research papers (Grooten et al. 2017). It is worth mentioning that the seat depth is adjusted by moving the back rest, resulting in the center of pressure on the seat being moved backwards, affecting the tilt of the chair.

Tilt resistance

The HÅG SoFi has tilt resistance in both the backward leaning, and forward leaning directions. This tilt resistance can be adjusted with a slider mechanism underneath the chair to five discrete levels, and the tilt can be locked to the "neutral" angle with a similar slider. The neutral angle, being the angle of the seat is the same as if no one is sitting in the chair. One can adjust these sliders at any time, but the tilt resistance will only change while at neutral angle, either from sitting in the neutral angle, rocking the chair back and forth past the neutral angle, or leaving the seat. The tilting point is located centrally on the seat, along with the gas spring, as seen on figure 5-1.

Description of tilt mechanism

The tilt mechanism in HÅG SoFi, visualized in figure 5-1, consists of a tilting point, and rubber springs "k". The seat tilts around a point, thus compressing one of the rubber springs k, resulting in a tilt resistance. Tilt resistance works in a similar manner for forward and backward tilt. The tilt resistance is altered by changing the distances l_1 and l_2. Distance l_1 and l_2 are changed by rotating the rubber springs around a point eccentric to the center of the rubber spring. This rotation is controlled by a lever inside the tilt control unit. The rotation point of one rubber spring is accessible inside the tilt control unit.

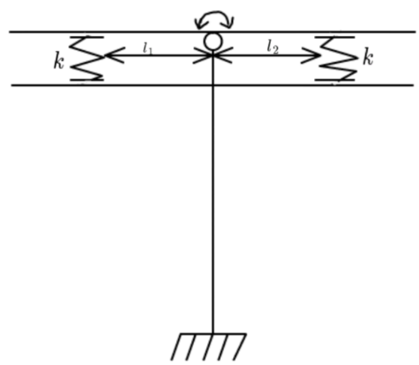


Figure 5-1 HÅG SoFi Tilt Mechanism Concept

To avoid problems regarding friction between the rubber spring, and either side of its enclosure, the mechanism is spring loaded to one of five different tilt resistances, with the actual change of tilt resistance being triggered when the chairs tilt reaches neutral angle. The neutral position being where the tilt mechanism is not exerting a torque.

Motorizing tilt mechanism

There are several concepts that could be used to alter tilt resistance with a motor. The two biggest questions are whether to control the rubber springs directly, or add a motor module to the existing tilt control, and what type of mechanism to use. Using the existing tilt resistance control, the five discrete levels of tilt resistance would provide repeatability when testing and gathering data, as well as low power requirements, due to the spring loading system. However, as mentioned the changes in tilt resistance will not come to effect until the seat is at the neutral angle. This could potentially be an unobtrusive, and nonimmediate intervention method for passive seating. The alternative approach of controlling the springs directly would likely be inferior in most ways. It could be useful for achieving continuous variability of tilt resistance, and smaller in size through elimination of the existing tilt control unit. It would however likely be much more difficult to achieve, due to friction between the rubber spring and its enclosure. This could be overcome through excessive force, likely to lead to wear on the rubber springs if not redesigning the tilt resistance mechanism entirely, or through only changing tilt resistance when the seat is in neutral position.

For testing the effect of varying tilt resistance has on seated activity, the approach of motorizing the existing tilt control unit was chosen. From the start, a lead screw mechanism seemed to be the best way to move the slider in the tilt resistance control unit. The reduction of the mechanism reduces the required torque of the motor.

Failing to find a proper anchor point for a lead screw mechanism, some other options were explored as well. The slider-crank-mechanism was rapidly prototyped with meccano parts (figure 5-2), which confirmed that the chosen motor was capable of delivering enough torque, as well as some inherent weaknesses to the slider-crank mechanism. The force delivered from the crank to the slider along the axis of the slider is sinusoidal, and no force is delivered when at the two end points. To deal with the issue of force in the end points, some slack could be introduced in the

Figure 5-2Crank-Slider Prototype

mechanism, however this would make it more difficult to control the position properly.

While exploring another, alternative approach, further examining of the sliders in the control unit yielded a proper mounting point for a lead screw mechanism. The

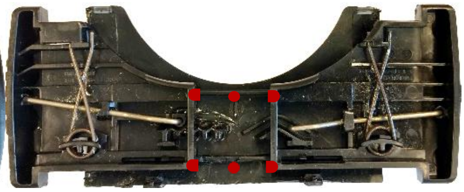


Figure 5-3a Control Unit Outside *Figure 5-3b Control Unit Inside*

sliders would still work if two holes were drilled along the center, through the sliders as indicated in figure5-3 a and b, where red marking indicates removal of

material.

A mounting module for the motor was 3d printed (figure 5-4), and an enclosure to the nut was mounted to the slider, consisting of three plates, as seen in figure5-5. The nut is kept from rotating along with the lead screw by plate number two, and plate number one and three transfers force in either direction along the axis of the lead screw. The prototype was made rapidly, but has proven robust enough to be kept.

Figure 5-4 Motor Mounting Module for HÅG SoFi

Positioning:

To precisely move the slider to the desired position, a positioning system was needed. The two best alternatives are to measure the rotation of the motor, or the translation of the slider. Measuring the position directly through a linear potentiometer was the least complicated, easiest to integrate and calibrate. The full system can be seen in figure5-5, and figure 5-8.

The motor is controlled through from an Arduino, using four relays(figure 5-8).. This motor driver is identical to the one used to control the motor on the HÅG Capisco-prototype. For the HÅG SoFi prototype, the card could be replaced with the motor driver seen on figure 5-6, with slightly altered codes on the Arduino. The codes used to control the motor can be found in appendix B, under the function “int Motor Control()”.

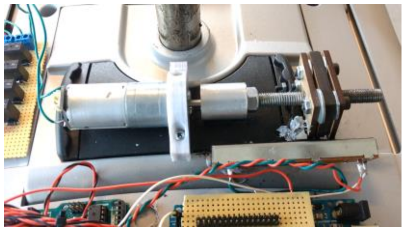


Figure 5-5 Tilt Resistance Setting Control

Sensors

An array of 12 FSRs are placed the seat, and 4 in the back rest, as seen in figure 5-7 a and b. Much thought and experimentation were given to the placement of the sensors, so that as much data as possible is captured, and every sensor is meant to capture something different. For instance, the four sensors in front capture data on how spread the user’s legs are, if the user is facing to the left, or right, give a good indication of whether the chair height is set appropriately, too high, or too low. After preliminary tests, a gyroscope-accelerometer module was also added, to help measure the seating activity.

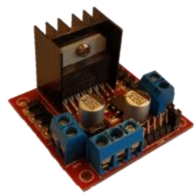


Figure 5-6 L298N-motor driver

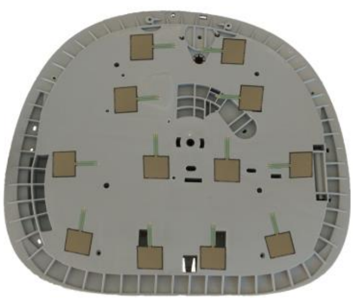


Figure 5-7a Sensor Placement in Seat

Figure 5-7b Sensor Placement in Backrest

Two Arduinos are used connected over the $1^{2}C$-protocol as shown in figure 5-8. An Arduino Mega ADK for pressure sensors, and power button, that acts as a master, and sends commands to, and requests data from the slave, as well as handling the serial communication to the computer. An Arduino Uno R3 acted as the slave Arduino, that controlling the motor. The separate accelerometer-gyroscope was also connected through $I^{2}C$. While preparing for tests, calculations, and data gathering were done on a Windows computer, and in testing those tasks were done on a Raspberry Pi.

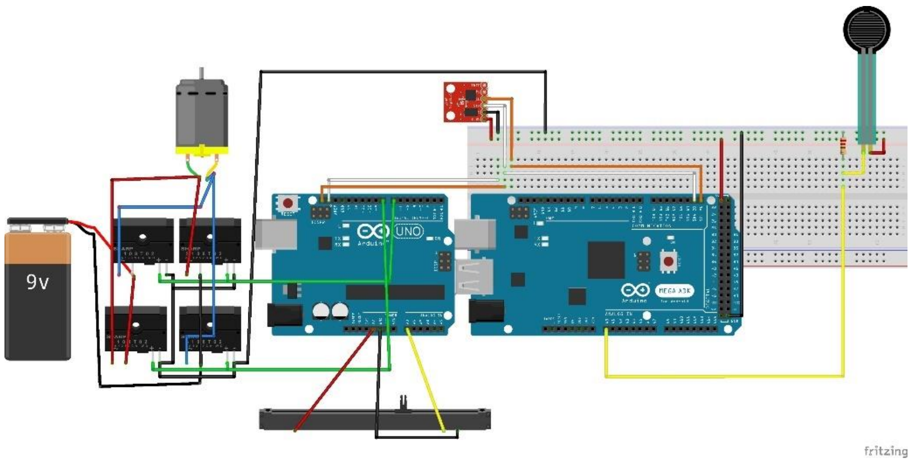


Figure 5-8 Wiring of Electronics on HÅG SoFi-prototype, with reduced FSRs

Comments:

Though the preferred end solution of a lead screw mechanism was known from the start, time and work could have been saved by more thoroughly exploring the chair from the start. Front loading the information gain could potentially reduce design time significantly. However, valuable knowledge was attained also by making the slider crank mechanism. Much of the work could be reused, including Motor control electronics, Arduino codes, and CAD of mounting system from second iteration of slider-crank mechanism.

Noise could be reduced by building a simple enclosure, one was not built to ease inspection.

The separation of functionality to two different Arduinos was due to port selection. There are no other technical reasons for separating the Arduinos. In building the chair, and testing the various functions, it was however beneficial, as the various modules could be tested and evaluated independently, and then later be connected.

6. Capisco

Description of chair

The saddle shape of the HÅG Capisco (8106) allows variation between sitting and standing working posture. The chair has adjustable seat height, seat depth, back support height, and adjustable/lockable tilt resistance. Adjusting the tilt mechanism affects the “neutral” angle of the seat. Neutral angle being the angle of the chair when no one is sitting in it. The tilt resistance is changed by turning a knob, and the effect is instantaneous, in sharp contrast to that of the SoFi. When motorized, a change in tilt resistance may be somewhat obtrusive, and the sitter might intuitively try to counteract the change of seat angle by pushing back.

Tilt resistance

The tilt mechanism of HÅG Capisco, visualized in figure 6-1, consists of a tilting point in the center of the seat, with a rubber spring offset from the tilting point, mounted to a beam with a permanent angle to the gas spring, slightly offset from the seat, and linked to the seat with a threaded rod. The threaded rod also acts as a lead screw, by which the tilt resistance of the chair can be tightened or loosened by rotation of the nut M.

A secondary effect of altering tilt resistance with such a mechanism is that the neutral angle of the chair, θ, also changes accordingly. Tightening the chair leads to a forward tilt, and vice versa.

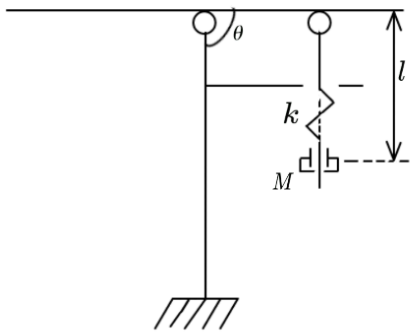


Figure 6-1: HÅG Capisco Tilt Mechanism Concept

Motorizing tilt mechanism

The tilt resistance mechanism of the HÅG Capisco consists of a lead screw tightening or loosening a rubber spring. The mechanism is simple, and coming up with the concept for motorization was straight-forward. However, given the pre-tension of the system when a person is seated, getting the necessary power, without a motor and power supply getting in the way of any other mechanism, or, even more importantly, getting in the way of the person, proved a more complicated task. To test the concept of how it would be easiest to control it, a smaller motor was mounted using a custom designed mount, as seen in figure 6-2. The concept being that the motor turns the nut of the existing lead screw. A consequence of this being that the motor moves linearly with the nut along the screw.

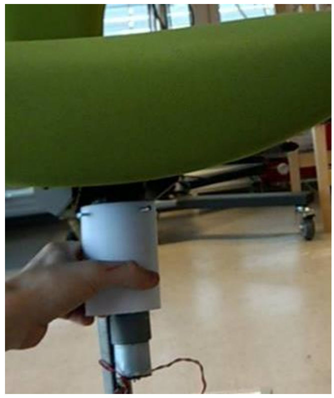


*Figure 6-2*Motor Mount, Capisco

This prototype was capable of tightening the tilt resistance to some degree. However, the motor could not deliver enough torque to tighten the mechanism enough, even while no one was seated. The motor was changed, and necessary mounting parts refined to occupy less space, and handle the larger torque delivered from the new motor. The motor was capable of tightening slightly less, or equal to the what the author could with the original Capisco mechanism.

After testing the motor, and seeing that it was capable of tightening, the chair with someone seated, and hence actively tilting a person forward, a slight redesign of some parts was done to better fit the chair. One of these parts (figure 6-3) broke just prior to the planned start of testing. The part broke due to material settings in the 3D-print, which was set to 30% infill, and not to solid as with the similar part used during testing of the motor. The part was rapidly made later that same day from an aluminum plate

Figure 6-3

Sensors

An array of 6 FSRs are placed the seat, and 3 along the spine in the back rest. Much thought and experimentation were given to the placement of the sensors, so that as much data as possible is captured, and every sensor is

Figure 6-4Sensor placement in Seat

meant to capture something different. A gyroscope-accelerometer module was also added, to help measure the activity level.

Motor positioning

As with the HÅG SoFi, a linear potentiometer is used to determine the tilt resistance, by measuring the compression of spring k in figure 6-1. The linear potentiometer is attached to the moving motor, and measures the distance to the end of the lead screw. When the motor starts and stops, a small rotation relative to the lead screw may occur, The high force results in glue, or other adhesives not being a viable idea, as the connection, or linear potentiometer would likely be damaged with by such a rotation. A magnet was used to connect the slider on the linear potentiometer to the lead screw, as it allows for rotation. The solution is not perfectly robust, but allows for disassembly. A more permanent option would be to use a rotary encoder on the motor, and measure number of times the nut has turned.

Comments:

Noise could be reduced by using a ball screw mechanism. However, the increased pitch (linear distance per revolution of the nut) would require a larger motor, and the mechanism itself would need to be bulkier. Noise could in either case be further reduced by building an enclosure for the mechanism and motor.

Though no field testing was done with this chair, some preliminary testing of its reliability and durability was done, and the prototype is fully functional, and ready to be used in field testing.

7. Codes

The programs running on the Arduinos can be found in appendix B, and the ones Python programs running on Raspberry Pi are in appendix C they can also be found digita. A brief explanation on the functionality, and setup of the programs are provided in this chapter.

Arduino

The codes on the Arduino work reactively. It waits for a command either over the serial connection (USB), or from one of the buttons on the accompanying remote. When a command is sent over a serial connection, a corresponding action is performed on the Arduino.

On both the HÅG SoFi and the Capisco there are buttons that when pressed send a string over the serial connection. On the HÅG SoFi, these strings are "Position1" - "Position5", and "ShutDown", triggering actions on the computing device (Raspberry Pi or Windows machine) for changing tilt resistance setting, and closing the software and shutting down the Raspberry Pi.

"Pressure": Arduino sends all sensor values, including those of the gyroscope and accelerometer.

Motor control works differently on the HÅG SoFi an HÅG Capisco:

SoFi:

On the HÅG Sofi, the measuring of the position of the tilt resistance, as well as the change of tilt resistance is handled by a separate slave-Arduino. Using separate connections for the two Arduinos proved problematic with the Raspberry Pi. To best accommodate this, all messages and commands to the slave-Arduino go through the master Arduino.

Master: "X, Y": X determines the address of the slave-Arduino (in this case X=5), and Y is the command it sends on to the slave-Arduino.

Slave: "Y": The Slave receives an integer in the range of 0 to 5. If Y is 0, the current tilt resistance setting is returned as a number from 1-5, with 1 being the lowest tilt resistance level, and 5 being the highest. If Y is a number in the range 1 to 5, the tilt resistance is set changed to Y.

Capisco:

"Direction, Position": The Capisco motor control is not discrete, as on HÅG SoFi, but works has a continuous specter. That is, the range of appropriate values as measured on the potentiometer is from 150- loosest, to 350- tightest.

"Position": The value read from the linear potentiometer is returned.

Python programs:

Two different programs were made, one for Windows, and one for Raspberry Pi (appendix C). The Windows program was made with easy creation and evaluation of datasets in mind, and comes with a graphical user interface, giving an indication of what an end-user experience could be like. The Raspberry Pi program was written with testing the prototypes in mind.

The codes run similarly for both the HÅG SoFi and HÅG Capisco prototypes, adapting for the number of sensors, and the differences in motor control. Before elaborating the features of the different programs, the way supervised learning is implemented and used in the project is presented.

Machine Learning - classification

The most important functions used for prediction purposes will be presented in the following:

Pre-Processing

The Pandas library is used for most of the data management. A single data point is stored as a "series", with a list of sensor values, and an appropriate header labeling the sensors. Multiple series form a "data frame". All sensor values are stored with the label as a column header. In training data, there is also a separate column for the sitting posture.

Training

The training dataset is loaded as a Pandas data frame, and a classifier is fitted based on the data frame. That is, it learns how to predict sitting position based on sensor data. A single series, or an entire data frame can then be predicted by the classifier. For this purpose, a classifier called "Nearest Neighbor-BallTree", as it was shown to give consistently good results under testing for all postures.

Predictions:

After a classifier has been trained, a new data point can be loaded and predicted. A single data point can be loaded at the time, yielding a single prediction, or a data frame can be loaded, yielding several predictions. Data can be predicted in real time, and post-data gathering.

Evaluating models

To test how well a classifier works, the training data is split into a training part, and a validation part. The Training part is used to fit the classifier, and the validation part is used to test the performance of the classifier. The classifier takes in the sensor values, and predicts the sitting position. The predictions are compared to the actual This shows how well the classifier performs on "known data", as the data is from the same dataset as it was trained on. The performance on validation data (percent correct predictions) gives an indication of consistency within the dataset, but does not necessarily show how well the classifier is generalized, or whether the classifier is "overfitted". An overfitted dataset performs well on known data, but will perform worse on previously unseen data. To test the performance on unseen data, and the ability to generalize, a separate test dataset is used in the same manner as the validation data. Confusion matrices are used to visualize the performance, showing distribution of predictions given a position, as seen in figure7-4. A setup was also made to test 11 classifiers at a time on various datasets to help to choose a classifier that worked well consistently.

In use

When used with incoming data, as is the case for a real use situation, a single data point is evaluated at a time. Taking into account that the readings only represent an instantaneous position, and the user may be moving, a number of readings could be evaluated, and the most prominent prediction is used. For instance performing 10 readings a second, and setting a limit that if 6 or more predictions being similar, otherwise returning "moving" as the current posture. This was not implemented in the Raspberry Pi version, as the amount of data stored was deemed unnecessarily large when the entire datasets were to be analyzed later. Datasets can also be analyzed as a whole

Windows:

As the number of different tasks done by Python grew, the number of different programs became inconvenient. The programs were collected in a menu system.

A module called Pygame was used to create the menu system. Pygame enables real time graphical output, and can also track the keyboard and mouse in reak time. This tracking enables an easy and convenient way to collect, and label training and testing datasets. In the following, the user interface, and the associating functions in the software developed for use on Windows are presented.

The Program starts by prompting for a user name. This user name becomes the label for all the data stored, and to choose which training dataset to use, and also prompts if there are any special sitting postures that will be recorded.

User interface:

Main Menu (Figure 7-1): Choosing what sub-program to run, brings the user to other menus designed to e.g. create a personal dataset for classification purposes.

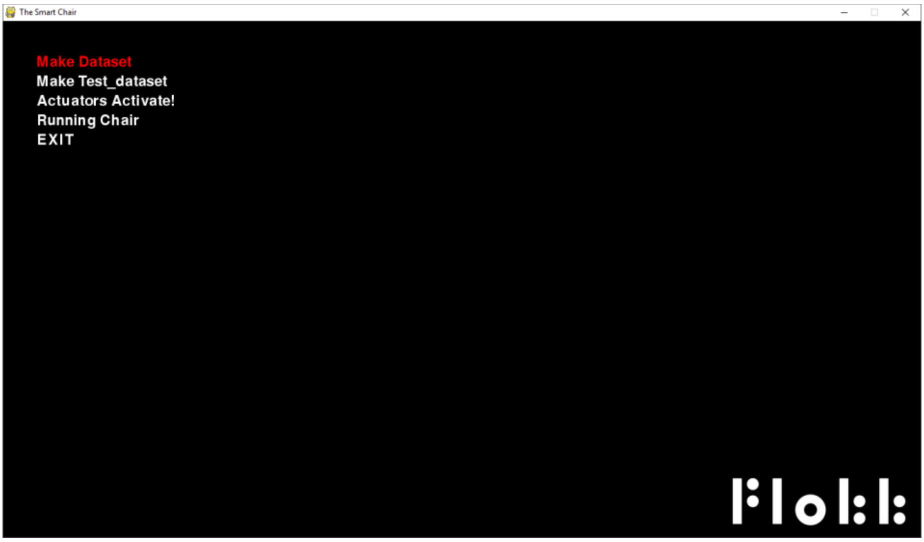


Figure 7-1 Main Menu

Make Dataset (Figure 7-2): The process of making a new dataset to work best possible with the test subject could be somewhat tedious in the past. To make it easier, the Pygame module was installed, allowing the monitoring of the keyboard. To record a position, one would simply hold a corresponding key as seen in figure 7-2. If recording more positions than ten, the next row of the keyboard would be used, "Q" for 11, "W" for 12 and so on. The sitting data is then stored to a .csv-file with the user id as the filename. For instance, "The Author.csv", or "Ola Nordmann.csv". The "Test Dataset" creates a confusion matrix as in the right half of figure 7-4. The confusion matrix indicates if there are conflicting, or ambiguous data points. Note that this does not test how well the machine learning algorithm work on unseen data, but gives a good indication of whether there are conflicting values in the dataset. The Diagonal indicates correct predictions. The "Make Test_dataset" leads to another menu.

Figure 7-2 Make Dataset-Menu

Make Test_dataset (Figure 7-3): To best test the performance of the classification, a separate dataset is recorded in a similar fashion as in “Make Dataset”. The “Test Dataset” option tests creates 14 different classifiers, and compares their effectiveness like in figure 7-5. Two confusion matrices are made for each classifier as seen in figure 7-4. One confusion matrix is made for the training dataset, as in the “Make Dataset”-menu, and one with the previously unseen data recorded in the “Make Test_dataset”-menu. The confusion matrices show what sitting posture has been predicted, and what the actual sitting posture was.

*Figure 7-3*Make Test_dataset - Menu

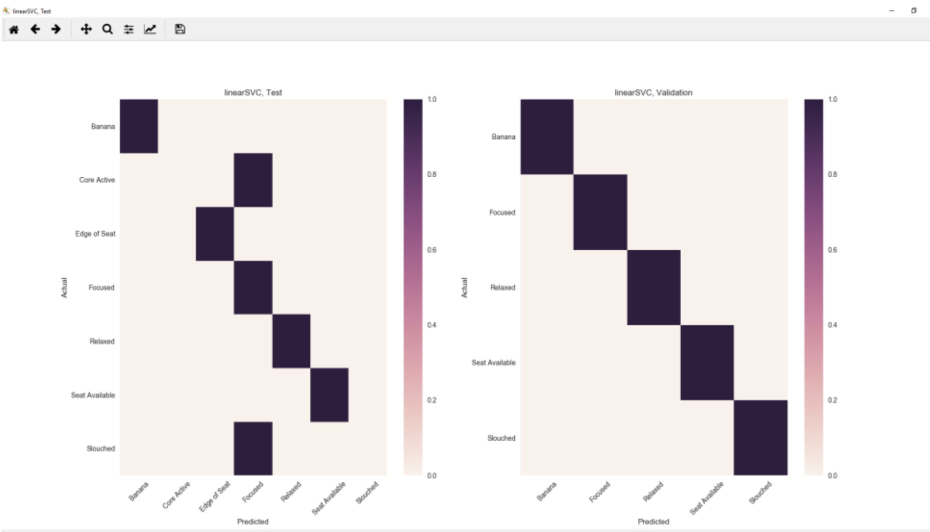


*Figure 7-4*Confusion Matrix Example

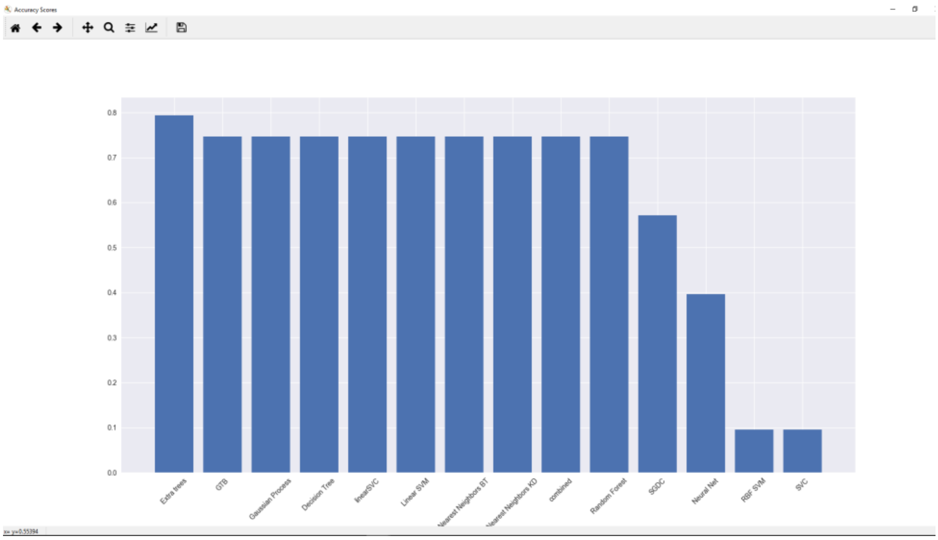


*Figure 7-5*Performance of Different Algorithms

Actuators Activate: allows the user to control tilt resistance from the software There is also a remote control mounted on the chair that does the same.

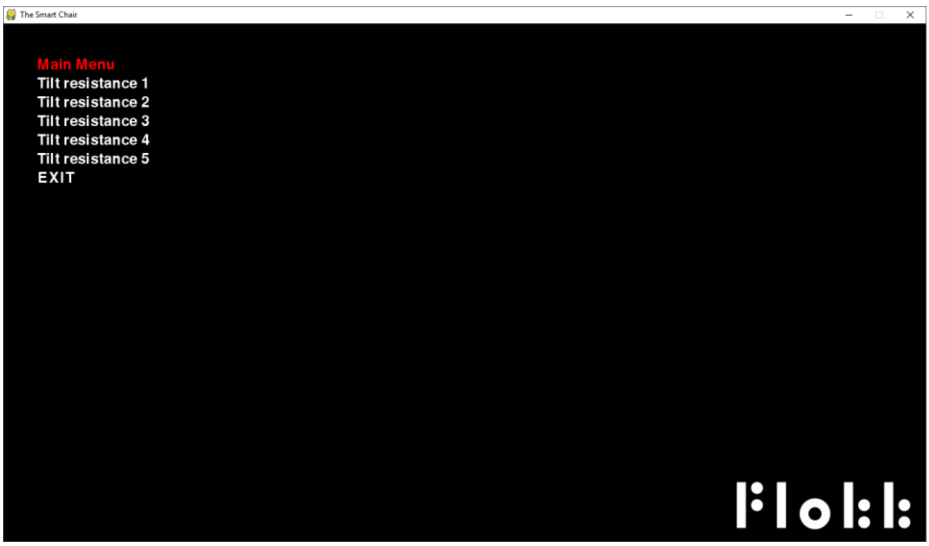


*Figure 7-6*Actuation Menu for HÅG SoFi

Running Chair: This is strictly speaking not another menu, but a state. In this state data is collected, sitting posture is predicted, and the tilt resistance setting is changed with a given condition. The condition at the moment being either every 15 minutes, or only on user prompt. This would be the place to implement reinforcement learning for one dimension (tilt resistance setting) to allow the prototype to learn how to choose appropriate tilt resistance, and when to change the tilt resistance setting.

Raspberry Pi:

The Python Codes on the Raspberry Pi were custom made for the testing, and to work without any input other than that from the Arduino(s). The Raspberry Pi is also capable of running the version made for Windows.

Startup: The Python program starts immediately after the Raspberry Pi has booted, and sends requests to the Arduino regarding the current tilt resistance setting.

Loop: The raspberry Pi sends a request every second to the Arduino, if no button on the remote control is pushed, the Arduino returns the sensor values. If a button is pushed, the Arduino sends back a string with a message correlating to the button pushed, either to change the tilt resistance, or to turn off the chair.

Every 15 minutes, an evaluation of whether to change the tilt resistance is undertaken. For testing purposes, this was limited to if the user was sitting for the majority of the 15 minutes.

Shutdown: When the Arduino sends the message “Shutdown” as a response to the power button being pushed, the data is saved, and if the Raspberry Pi is connected to the internet, the dataset is uploaded to the authors Google Drive, and then the Raspberry Pi then shuts down.

The program alternates between “test day 1”, and “test day 2” for each test subject. In test day 1, the tilt setting is only changed on user prompt. Upon reboot, test day 2 starts, and the tilt setting is randomly changed every 15 minutes.

8. Test Design

Intro: During the course of the testing of the HÅG SoFi-prototype, the test design, the dependent and independent variables have changed. To get an easy indication of movement, an accelerometer-gyroscope module is attached so that when seated, the z-axis points down, the y axis points forward, and the x axis to the side. As such, the most interesting variables when investigating the effect of tilt resistance are accelerometer in y and z axes, and gyroscope in x- axis, as they are all affected by tilting. Gyroscope in z-axis might also be interesting, indicating turning in the chair, or rocking from side to side. During testing, the chair is actuated to a random tilt resistance every 15 minutes. This 15-minute period is then split up in three equal parts. The independent variable being time since the last change in tilt resistance.

This last iteration of the experiment has only been run for one day, and no statistically significant results are obtained, but an interesting tendency has been evident throughout the dataset. The standard deviance in accelerometer axes x and y was higher in the periods following a change in tilt resistance. The dataset was split in three pieces, with three different test sizes. In split number one, the five first minutes after a change of tilt resistance was called 1, the following five minutes called 2, and the next five-minute period called 3. All instances of an empty chair were then removed. A similar approach was taken three times, split number two analyzing the respective time periods 0- 120s, 300- 420s and 600 – 720s. And data split three analyzing time periods 120- 240 s, 420- 540s and 720-840s. The respective mean values, and standard deviations from data split one, two and three are shown in table 8-1

Data split 1	μ_{Ay}	σ_{Ay}	μ_{Az}	σ_{Az}	μ_{Gx}	σ_{Gx}
1	0.1502	0.06023	-0.9902	0.01429	-0.4495	1.68011
2	0.152	0.0559	-0.9902	0.01186	-0.4701	1.58908
3	0.1474	0.05138	-0.9911	0.01111	-0.4396	1.61541
Data split 2						
1	0.1493	0.06459	-0.992	0.0134	-0.4476	1.40741
2	0.149	0.05542	-0.9901	0.0112	-0.4344	1.07833
3	0.1523	0.05384	-0.9905	0.01052	-0.4097	1.38901
Data split 3						
1	0.153	0.05788	-0.99	0.01611	-0.4518	1.80591
2	0.1558	0.05532	-0.9897	0.01242	-0.527	2.13571
3	0.1429	0.04645	-0.992	0.00881	-0.4491	1.12144

Table 8-1 Mean and std. Dev from accelerometer axes x and y, and gyroscope axis x

As mentioned, this is by no means a statistically significant finding, not enough data points are gathered for that to be the case. The result seems rather consistent over the course of that one day, but it could be this way by chance, movement could be caused by the user not liking the sound of the motor, or a number of other factors.

9. Conclusions, Limitations, and Further Work

Two prototype chairs have been made, capable of tracking instant sitting posture, and movement through the use of a series of force sensitive resistors, supervised machine learning and an accelerometer-gyroscope module. The chairs can automatically change tilt resistance, one chair in a subtle, and unobtrusive manner, another affecting the seating immediately and directly. A small pilot test was performed on the HÅG SoFi-prototype, to make sure that the chair was working as intended in a research situation. The accelerometer-gyroscope module was only added for the last test, but the results seemed promising. The variance in accelerometer axes y and z, respectively forward, and down when sitting in the chair, seemed consistently higher in the time following a change in tilt resistance. This is an indication that the activity-while-seated level increased in the time directly following a change in tilt resistance. This may be due to noise, or vibrations reminding the test subject to move. It might be random, as the test sample is only from one day of sitting, the limited testing did not yield enough data to determine the effects either prototype could have on seating behavior. Further field studies are strongly recommended.

It is also strongly recommended to implement a reinforcement learning algorithm to the tilt resistance setting control, the result could be very interesting both commercially, and with regards to research on human computer interaction.

Bibliography

"15 Things You Didn't Know about Back Pain - Independent.ie." 2017. Accessed June 18. http://www.independent.ie/life/health-wellbeing/15-things-you-didnt-know-about-back-pain-31367264.html.

Aota, Yoichi, Haruhiko Iizuka, Yusuke Ishige, Takashi Mochida, Takeshi Yoshihisa, Masaaki Uesugi, and Tomoyuki Saito. 2007. "Effectiveness of a Lumbar Support Continuous Passive Motion Device in the Prevention of Low Back Pain During Prolonged Sitting:" *Spine* 32 (23): E674–77. doi:10.1097/BRS.0b013e318158cf3e.

"Axia Smart Chair – Your Personal Posture Coach." 2017. *BMA Ergonomics UK*. Accessed June 22. https://www.bma-ergonomics.com/en/product/axia-smart-chair/.

Burton, A. Kim. 2005. "How to Prevent Low Back Pain." *Best Practice & Research Clinical Rheumatology*, Non-Specific Low Back Pain, 19 (4): 541–55. doi:10.1016/j.berh.2005.03.001.

Deursen, D. L. van, M. Lengsfeld, C. J. Snijders, J. J. M. Evers, and R. H. M. Goossens. 2000. "Mechanical Effects of Continuous Passive Motion on the Lumbar Spine in Seating." *Journal of Biomechanics* 33 (6): 695–99. doi:10.1016/S0021-9290(99)00231-6.

Deursen, Leo L. van, Jaap Patijn, John R. Durinck, Ruud Brouwer, Jeanne R. van Erven-Sommers, and Bernard J. Vortman. 1999. "Sitting and Low Back Pain: The Positive Effect of Rotatory Dynamic Stimuli during Prolonged Sitting." *European Spine Journal* 8 (3): 187–93. doi:10.1007/s005860050155.

Ellegast, Rolf P., Kathrin Kraft, Liesbeth Groenesteijn, Frank Krause, Helmut Berger, and Peter Vink. 2012. "Comparison of Four Specific Dynamic Office Chairs with a Conventional Office Chair: Impact upon Muscle Activation, Physical Activity and Posture." *Applied Ergonomics*, Special Section on Product Comfort, 43 (2): 296–307. doi:10.1016/j.apergo.2011.06.005.

Goodman, Denise M., Alison E. Burke, and Edward H. Livingston. 2013. "Low Back Pain." *JAMA* 309 (16): 1738–1738. doi:10.1001/jama.2013.3046.

Groenesteijn, Liesbeth, Rolf P. Ellegast, Kathrin Keller, Frank Krause, Helmut Berger, and Michiel P. de Looze. 2012. "Office Task Effects on Comfort and Body Dynamics in Five Dynamic Office Chairs." *Applied Ergonomics*, Special Section on Product Comfort, 43 (2): 320–28. doi:10.1016/j.apergo.2011.06.007.

Grooten, Wilhelmus J. A., Björn O. Äng, Maria Hagströmer, David Conradsson, Håkan Nero, and Erika Franzén. 2017. "Does a Dynamic Chair Increase Office Workers' Movements? – Results from a Combined Laboratory and Field Study." *Applied Ergonomics* 60 (April): 1–11. doi:10.1016/j.apergo.2016.10.006.

Gupta, Nidhi, Caroline Stordal Christiansen, David M. Hallman, Mette Korshøj, Isabella Gomes Carneiro, and Andreas Holtermann. 2015. "Is Objectively Measured Sitting Time Associated with Low Back Pain? A Cross-Sectional Investigation in the NOMAD Study." *PLoS ONE* 10 (3): 1–18. doi:10.1371/journal.pone.0121159.

Hartvigsen J, Leboeuf-Yde C, Lings S, and Corder EH. 2000. "Is Sitting-While-at-Work Associated with Low Back Pain? A Systematic, Critical Literature Review." *Scandinavian Journal of Public Health* 28 (3): 230–39.

Kent, Peter, Robert Laird, and Terry Haines. 2015. "The Effect of Changing Movement and Posture Using Motion-Sensor Biofeedback, versus Guidelines-Based Care, on the Clinical Outcomes of People with Sub-Acute or Chronic Low Back Pain-a Multicentre, Cluster-Randomised, Placebo-Controlled, Pilot Trial." *BMC Musculoskeletal Disorders* 16 (May). doi:10.1186/s12891-015-0591-5.

O'Sullivan, Kieran, Mary O'Keeffe, Leonard O'Sullivan, Peter O'Sullivan, and Wim Dankaerts. 2012. "The Effect of Dynamic Sitting on the Prevention and Management of Low Back Pain and Low Back Discomfort: A Systematic Review." *Ergonomics* 55 (8): 898–908. doi:10.1080/00140139.2012.676674.

O'Sullivan, Kieran, Leonard O'Sullivan, Peter O'Sullivan, and Wim Dankaerts. 2013. "Investigating the Effect of Real-Time Spinal Postural Biofeedback on Seated Discomfort in People with Non-Specific Chronic Low Back Pain." *Ergonomics* 56 (8): 1315–25. doi:10.1080/00140139.2013.812750.

O'Sullivan, Kieran, Peter O'Sullivan, Mary O'Keeffe, Leonard O'Sullivan, and Wim Dankaerts. 2013. "The Effect of Dynamic Sitting on Trunk Muscle Activation: A Systematic Review." *Applied Ergonomics* 44 (4): 628–35. doi:10.1016/j.apergo.2012.12.006.

O'Sullivan, Kieran, Peter O'Sullivan, Leonard O'Sullivan, and Wim Dankaerts. 2012. "What Do Physiotherapists Consider to Be the Best Sitting Spinal Posture?" *Manual Therapy* 17 (5): 432–37. doi:10.1016/j.math.2012.04.007.

Raschka, Sebastian. 2015. *Python Machine Learning*. Packt Publishing. http://proquestcombo.safaribooksonline.com/9781783555130.

Roffey, Darren M., Eugene K. Wai, Paul Bishop, Brian K. Kwon, and Simon Dagenais. 2010. "Causal Assessment of Awkward Occupational Postures and Low Back Pain: Results of a Systematic Review." *The Spine Journal* 10 (1): 89–99. doi:10.1016/j.spinee.2009.09.003.

"Sensomative Office – Sensomative." 2017. Accessed June 22. http://sensomative.com/en/products/sit-cat-office/.

Shorthouse, F. M., V. Roffi, and C. Tack. 2016. "Effectiveness of Educational Materials to Prevent Occupational Low Back Pain." *Occupational Medicine* 66 (8): 623–29. doi:10.1093/occmed/kqw072.

Smith, Preston G. 2007. *Flexible Product Development: Building Agility for Changing Markets*. John Wiley & Sons.

Vergara, Margarita, and Álvaro Page. 2002. “Relationship between Comfort and Back Posture and Mobility in Sitting-Posture.” *Applied Ergonomics* 33 (1): 1–8. doi:10.1016/S0003-6870(01)00056-4.

Weering, Marit G. H. Dekker-van, Miriam M. R. Vollenbroek-Hutten, and Hermie J. Hermens. 2012. “Do Personalized Feedback Messages about Activity Patterns Stimulate Patients with Chronic Low Back Pain to Change Their Activity Behavior on a Short Term Notice?” *Applied Psychophysiology and Biofeedback* 37 (2): 81–89. doi:10.1007/s10484-012-9181-6.

Appendix A Risk Assessment

NTNU HMS	Risikovurdering	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2601	22.03.2011
		Godkjent av		Erstatter
		Rektor		01.12.2006

Enhet: Institutt for maskinteknikk og produksjon **Dato:** 26/01/2017
Linjeleder: Deltakere ved kartleggingen
(m/ funksjon): Ansv. Veileder: Martin Steinert, Medveileder: Andreas Wulvik, Student: Osmund Olav Bøe
Risikovurderingen gjelder hovedaktivitet: Masteroppgave student Osmund Olav Bøe. Smart Chair.
Signaturer: *Ansvarlig veileder:* Student:

ID nr	Aktivitet fra kartleggings-skjemaet	Mulig uønsket hendelse/ belastning	Vurdering av sannsyn-lighet (1-5)	Vurdering av konsekvens: Menneske (A-E)	Ytre miljø (A-E)	Øk/ materiell (A-E)	Om-dømme (A-E)	Risiko-Verdi (menneske)	Kommentarer/status Forslag til tiltak
1	Soldering	Brenner fingertupper	2	A				A2	Tiltak ikke nødvendig
2	Laser cutting	Bearbeidingsobjekt tar fyr	4	A		A	A	A4	Det er satt ut CO^2-apparat, og opplæring om maskinen er gitt.
2	Laser cutting	Dannelse av klorgass	1	E				E1	Kun dersom en PVC-plate ved en feil blir kuttet i laserkutter. ikke aktuelt for min oppgave.
3	3D printing	Brenner fingertupper	1	A				A1	Tiltak ikke nødvendig
4	Hand tool usage	Skade ved bruk av kniv eller annet håndverktøy	2	A				A2	Tiltak ikke nødvendig
5	Test accidents	Fall eller klemfare ved bruk av prototypens aktuatorer	4	A				A4	Grundig forklaring med advarsel før testperson bruker prototype

NTNU HMS	Kartlegging av risikofylt aktivitet	Utarbeidet av	Nummer	Dato
		HMS-avd.	HMSRV2601	22.03.2011
		Godkjent av		Erstatter
		Rektor		01.12.2006

Enhet: Institutt for maskinteknikk og produksjon **Dato:** 26/01/2017
Linjeleder: Deltakere ved kartleggingen
(m/ funksjon): Ansv. Veileder: Martin Steinert, Medveileder: Andreas Wulvik, Student: Osmund
Kort beskrivelse av hovedaktivitet/hovedprosess: Masteroppgave, Osmund Olav Bøe. Smart Chair.
Er oppgaven rent teoretisk? : Nei
Signaturer: *Ansvarlig veileder:* Student:

ID nr.	Aktivitet/prosess	Ansvarlig	Eksisterende dokumentasjon	Eksisterende sikringstiltak	Lov, forskrift o.l.	Kommentar
1	Soldering					
2	Laser cunnting					
3	3D printing					
4	Hand tool usage					
5	Test accidents					

NTNU	Risikovurdering	Utarbeidet av	Nummer	Dato
HMS		HMS-avd.	HMSRV2601	22.03.2011
		Godkjent av		Erstatter
		Rektor		01.12.2006

Sannsynlighet vurderes etter følgende kriterier:

Svært liten 1	Liten 2	Middels 3	Stor 4	Svært stor 5
1 gang pr 50 år eller sjeldnere	1 gang pr 10 år eller sjeldnere	1 gang pr år eller sjeldnere	1 gang pr måned eller sjeldnere	Skjer ukentlig

Konsekvens vurderes etter følgende kriterier:

Gradering	Menneske	Ytre miljø Vann, jord og luft	Øk/materiell	Omdømme
E Svært Alvorlig	Død	Svært langvarig og ikke reversibel skade	Drifts- eller aktivitetsstans >1 år.	Troverdighet og respekt betydelig og varig svekket
D Alvorlig	Alvorlig personskade. Mulig uførhet.	Langvarig skade. Lang restitusjonstid	Driftsstans > ½ år Aktivitetsstans i opp til 1 år	Troverdighet og respekt betydelig svekket
C Moderat	Alvorlig personskade.	Mindre skade og lang restitusjonstid	Drifts- eller aktivitetsstans < 1 mnd	Troverdighet og respekt svekket
B Liten	Skade som krever medisinsk behandling	Mindre skade og kort restitusjonstid	Drifts- eller aktivitetsstans < 1uke	Negativ påvirkning på troverdighet og respekt
A Svært liten	Skade som krever førstehjelp	Ubetydelig skade og kort restitusjonstid	Drifts- eller aktivitetsstans < 1dag	Liten påvirkning på troverdighet og respekt

Risikoverdi = Sannsynlighet x Konsekvens
Beregn risikoverdi for Menneske. Enheten vurderer selv om de i tillegg vil beregne risikoverdi for Ytre miljø, Økonomi/materiell og Omdømme. I så fall beregnes disse hver for seg.

Til kolonnen "Kommentarer/status, forslag til forebyggende og korrigerende tiltak":
Tiltak kan påvirke både sannsynlighet og konsekvens. Prioriter tiltak som kan forhindre at hendelsen inntreffer, dvs. sannsynlighetsreduserende tiltak foran skjerpet beredskap, dvs. konsekvensreduserende tiltak.

NTNU	Risikomatrise	utarbeidet av	Nummer	Dato
HMS/KS		HMS-avd.	HMSRV2604	08.03.2010
		godkjent av		Erstatter
		Rektor		09.02.2010

MATRISE FOR RISIKOVURDERINGER ved NTNU

KONSEKVENS						
	Svært alvorlig	E1	E2	E3	E4	E5
	Alvorlig	D1	D2	D3	D4	D5
	Moderat	C1	C2	C3	C4	C5
	Liten	B1	B2	B3	B4	B5
	Svært liten	A1	A2	A3	A4	A5
		Svært liten	Liten	Middels	Stor	Svært stor
		SANNSYNLIGHET				

Prinsipp over akseptkriterium. Forklaring av fargene som er brukt i risikomatrisen.

Farge		Beskrivelse
Rød		Uakseptabel risiko. Tiltak skal gjennomføres for å redusere risikoen.
Gul		Vurderingsområde. Tiltak skal vurderes.
Grønn		Akseptabel risiko. Tiltak kan vurderes ut fra andre hensyn.

Appendix B Arduino Codes

Can also be found in the attached zip file.

Capisco

```
#include <SparkFunLSM6DS3.h>
#include <Wire.h>
#include <SPI.h>
LSM6DS3 myIMU( I2C_MODE, 0x6A );  //accel-gyro

int in1 = 24;
int in2 = 26;
int Position;
int hijack1 = 8;
int hijack2 = 7;
int pushbutton = 10;
int LPot = 0;
long previousMillis = 0;
String state ;
String msg;
int buttonState;
//buttons

void setup() {
  // Open serial communications and wait for port to open:
  Wire.begin();
  Serial.begin(115200);
  pinMode (pushbutton, INPUT);
  pinMode(hijack1, INPUT);
  pinMode(hijack2, INPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  digitalWrite(in1, LOW);
```

```
  digitalWrite(in2, LOW);

  if ( myIMU.begin() != 0 )
  {
    Serial.println("Device error");
  }
  int Postition = analogRead(LPot);
}

void loop() {
  while ( Serial.available() == 0) {}
  String msg = "";
  while (Serial.available() > 0)
  {
    msg += char(Serial.read());
    delayMicroseconds(60);
  }
  if (msg == "Pressure") {
    String dataString = "";
    // read three sensors and append to the string:
    for (int analogPin = 5; analogPin < 15; analogPin++) {
      int sensor = analogRead(analogPin);
      dataString += String(sensor);
      if (analogPin < 14)
      {
        dataString += ",";
      }
    }
    dataString += myIMU.readFloatAccelX(), 10;
    dataString += ",";
```

```
    dataString += myIMU.readFloatAccelY(), 10;
    dataString += ",";
    dataString += myIMU.readFloatAccelZ(), 10;
    dataString += ",";
    dataString += myIMU.readFloatGyroX(), 4;
    dataString += ",";
    dataString += myIMU.readFloatGyroY(), 4;
    dataString += ",";
    dataString += myIMU.readFloatGyroZ(), 4;
    buttonState = digitalRead(pushbutton);
    if (buttonState == 1) {
      dataString = "ShutDown";
    }
    if (digitalRead(hijack1) == HIGH) {
      dataString = "Tighter";
    }
    if (digitalRead(hijack2) == HIGH) {
      dataString = "Looser";
    }
    Serial.println(dataString);
    previousMillis = millis();
  }
  else if (msg == "Startup")
  {
    software_Reset();
  }
  else if (msg == "Position") {
    Serial.println(analogRead(LPot));
  }
  else {
```

```
    //"adress of slave" + "," + "message for slave(0 if request)"

    String dir = getValue(msg, ',', 0);

    int variable = getValue(msg, ',', 1).toInt();

    motorControl(dir, variable);

  }

}

String getValue(String data, char separator, int index)

{

  int found = 0;

  int strIndex[] = { 0, -1 };

  int maxIndex = data.length() - 1;


  for (int i = 0; i <= maxIndex && found <= index; i++) {

    if (data.charAt(i) == separator || i == maxIndex) {

      found++;

      strIndex[0] = strIndex[1] + 1;

      strIndex[1] = (i == maxIndex) ? i + 1 : i;

    }

  }

  return found > index ? data.substring(strIndex[0], strIndex[1]) : "";

}

void software_Reset() // Restarts program from beginning but does not reset the peripherals and registers

{

  asm volatile ("  jmp 0");

}


void motorControl(String dir, int goal) {
```

```
  if (dir == "tighter" && goal > analogRead(LPot)) {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    while (analogRead(LPot) < goal) {
      delay(5);
    }
  }

  else if (dir == "looser" && goal < analogRead(LPot)) {
    digitalWrite(in2, HIGH);
    digitalWrite(in1, LOW);
    while (analogRead(LPot) > goal) {
      delay(5);
    }
  }

  digitalWrite(in2, LOW);
  digitalWrite(in1, LOW);
}
```

SoFi Master Arduino

```
#include "SparkFunLSM6DS3.h"

#include "Wire.h"

#include "SPI.h"

LSM6DS3 myIMU( I2C_MODE, 0x6A );  //accel-gyro

int pushbutton = 22;

int fake5v = 24;

int fGround = 26;

int hijack1 = 28;

int hijack2 = 30;

int hijack3 = 32;

int hijack4 = 34;

int hijack5 = 36;

long previousMillis = 0;

String state ;

String msg;

int Position;

int buttonState;

void setup() {

  // Open serial communications and wait for port to open:

  Wire.begin();

  Serial.begin(115200);

  pinMode (pushbutton, INPUT);

  pinMode (fake5v, OUTPUT);

  pinMode (fGround, OUTPUT);

  digitalWrite(fake5v, HIGH);

  digitalWrite(fGround, LOW);

  pinMode(hijack1, INPUT);

  pinMode(hijack2, INPUT);

  pinMode(hijack3, INPUT);
```

```
  pinMode(hijack4, INPUT);
  pinMode(hijack5, INPUT);

  if ( myIMU.begin() != 0 )
  {
    Serial.println("Device error");
  }
}

void loop() {
  while ( Serial.available() == 0) {}
  String msg = "";
  while (Serial.available() > 0)
  {
    msg += char(Serial.read());
    delayMicroseconds(60);
  }
  if (msg == "Pressure") {
    String dataString = "";
    for (int analogPin = 0; analogPin < 16; analogPin++) {
      int sensor = analogRead(analogPin);
      dataString += String(sensor);
      if (analogPin < 15)
      {
        dataString += ",";
      }
    }
    dataString += myIMU.readFloatAccelX(), 10;
    dataString += ",";
    dataString += myIMU.readFloatAccelY(), 10;
```

```
  dataString += ",";
  dataString += myIMU.readFloatAccelZ(), 10;
  dataString += ",";
  dataString += myIMU.readFloatGyroX(), 4;
  dataString += ",";
  dataString += myIMU.readFloatGyroY(), 4;
  dataString += ",";
  dataString += myIMU.readFloatGyroZ(), 4;
  buttonState = digitalRead(pushbutton);
  if (buttonState == 1) {
    dataString = "ShutDown";
  }
  if (digitalRead(hijack1) == 1) {
    dataString = "Position1";
  }
  if (digitalRead(hijack2) == 1) {
    dataString = "Position2";
  }
  if (digitalRead(hijack3) == 1) {
    dataString = "Position3";
  }
  if (digitalRead(hijack4) == 1) {
    dataString = "Position4";
  }
  if (digitalRead(hijack5) == 1) {
    dataString = "Position5";
  }
  Serial.println(dataString);
  previousMillis = millis();
}
```

```
  else if (msg == "Startup")
  {
    software_Reset();
  }
  else {
    //"adress of slave" + "," + "message for slave(0 if
request)"
    int address = getValue(msg, ',', 0).toInt();
    int command = getValue(msg, ',', 1).toInt();
    if (command == 0) {
      Wire.requestFrom(address, 1);
      while (Wire.available()) {
        Position = Wire.read();
      }
      Serial.println (Position);
    }
    else {
      Wire.beginTransmission(address);
      Wire.write(command);
      Wire.endTransmission();
    }
  }
}

String getValue(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = { 0, -1 };
  int maxIndex = data.length() - 1;
```

```
  for (int i = 0; i <= maxIndex && found <= index; i++) {
    if (data.charAt(i) == separator || i == maxIndex) {
      found++;
      strIndex[0] = strIndex[1] + 1;
      strIndex[1] = (i == maxIndex) ? i + 1 : i;
    }
  }
  return found > index ? data.substring(strIndex[0],
strIndex[1]) : "";
}


void software_Reset() // Restarts program from beginning but
does not reset the peripherals and registers
{
  asm volatile ("  jmp 0");
}
```

SoFi Slave:

```
#include <Wire.h>


int in1 = 8;
int in2 = 7;
int fakeGround = 4;
int LinearPot = A0;
int SecondGround = 4;


int UpLim1 = 50;
int LowLim2 = 125;
int UpLim2 = 141;
int LowLim3 = 220;
int UpLim3 = 240;
int LowLim4 = 310;
int UpLim4 = 345;
int LowLim5 = 400;
int oldPosition = 0;
int startPos;
void setup() {
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(fakeGround, OUTPUT);


  digitalWrite(fakeGround, LOW);
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  Serial.begin(115200);
  startPos = analogRead(LinearPot);
  Serial.println(startPos);
```

```
  if (startPos <= UpLim1) {
    oldPosition = 1;
  }
  else if (startPos < UpLim2 && startPos >= LowLim2) {
    oldPosition = 2;
  }
  else if (startPos < UpLim3 && startPos >= LowLim3) {
    oldPosition = 3;
  }
  else if (startPos < UpLim4 && startPos >= LowLim4) {
    oldPosition = 4;
  }
  else if (startPos > LowLim5) {
    oldPosition = 5;
  }
  Wire.begin(5);
  Wire.onReceive(receiveEvent);
  Wire.onRequest(requestEvent);
  Serial.println(oldPosition);
}

void loop() {
  while (Serial.available() == 0) {}
  Serial.println(analogRead(LinearPot));
  delay(500);
  // put your main code here, to run repeatedly:
  String SerialMsg = "";
  if (Serial.available() > 0)
  {
    while (Serial.available() > 0)
```

```
    {
      SerialMsg += char(Serial.read());
      delayMicroseconds(60);
    }
  }
  if (SerialMsg != ""); {
    int newPosition = SerialMsg.toInt();
    Serial.print ("Old Position:\t");
    Serial.println(oldPosition);
    oldPosition = MotorControl(newPosition);
    Serial.print ("New Position:\t");
    Serial.println(newPosition);
  }
}
int receiveEvent() {
  int newPosition = Wire.read();
  Serial.println (newPosition);
  int   oldPosition = MotorControl (newPosition);
  return oldPosition;
}
int requestEvent() {
  Wire.write(oldPosition);
}
int MotorControl(int newPosition) {
  if (newPosition != oldPosition) {
    switch (newPosition) {
      case 1:
        digitalWrite(in2, HIGH);
        while (analogRead(LinearPot) > UpLim1) {
        }
```

```
    digitalWrite(in2, LOW);
    oldPosition = newPosition;
    break;
  case 2:
    if (oldPosition < newPosition) {
      digitalWrite(in1, HIGH);
      digitalWrite(in2, LOW);
      Serial.println("Tightening");
      while ( analogRead(LinearPot) < LowLim2) {
      }
      digitalWrite(in1, LOW);
    }
    if (oldPosition > newPosition) {
      digitalWrite(in2, HIGH);
      while (analogRead(LinearPot) > UpLim2) {
      }
      digitalWrite(in2, LOW);
      oldPosition = newPosition;
      break;
    }
    oldPosition = newPosition;
    break;
  case 3:
    if (oldPosition < newPosition) {
      digitalWrite(in1, HIGH);
      Serial.println("Tightening");
      while ( analogRead(LinearPot) < LowLim3) {
      }
      digitalWrite(in1, LOW);
    }
```

```
    if (oldPosition > newPosition) {
      digitalWrite(in2, HIGH);
      Serial.println("Loosening");
      while (analogRead(LinearPot) > UpLim3) {
      }
    }
    digitalWrite(in2, LOW);
    oldPosition = newPosition;
    break;
  case 4:
    if (oldPosition < newPosition) {
      digitalWrite(in1, HIGH);
      Serial.println("Tightening");
      while ( analogRead(LinearPot) < LowLim4) {
      }
      digitalWrite(in1, LOW);
    }
    if (oldPosition > newPosition) {
      digitalWrite(in2, HIGH);
      Serial.println("Loosening");
      while (analogRead(LinearPot) > UpLim4) {
      }
    }
    digitalWrite(in2, LOW);
    oldPosition = newPosition;
    break;
  case 5:
    digitalWrite(in1, HIGH);
    Serial.println("Tightening");
    while ( analogRead(LinearPot) < LowLim5) {
```

```
        }
        digitalWrite(in1, LOW);
        oldPosition = newPosition;
        break;
    }
  }
  delay(100);
  return oldPosition;
}
```

Appendix C Python Codes

```python
import DriveUploader2
import serial
import csv
import pandas as pd
import machineLearning
from time import sleep,time
import datetime
from os import system
import os
from random import choice
LowLimit=560
UpLimit=800
LastChange = time()
def buttonPressShutdown():
    myfile.close()
    try:
        DriveUploader2.uploadFile([filename])
    except:
        pass
    print 'shutting down in 5 sec'
    sleep(5)
    system("sudo shutdown -h now")

def analyseSeating(myfile,newPosition):
    myfile.close()
    analyzer =
pd.read_csv(filename).Label.tail(900)#last hour
    overview = analyzer.value_counts(normalize = True)

    myfile = open(filename,'ab')
    wr = csv.writer(myfile, quoting = csv.QUOTE_ALL)



    if  overview.index[0]!='Seat Available':
        newPosition= choice(range(150,350))
        new =  choice (alternatives)
        if newPosition>oldposition:

arduinoSens.write('tighter,{0}'.format(newPosition))
            oldposition=newPosition
        elif newPosition<oldposition:

arduinoSens.write('looser,{0}'.format(newPosition))
            oldposition = newPosition


        global LastChange
        LastChange = time()

    return  myfile, wr,newPosition
```

```
files = []
for (dirpath, dirnames,filenames) in
os.walk('/home/pi/Desktop/Tests'):
    files.append(filenames)
print files
filename = 'Capisco person1 day1.csv'
if filename  in files[0]:
    filename = 'Capisco person1 day2.csv'
if filename in files[0]:
    filename = 'Capisco person2 day1.csv'
if filename in files[0]:
    filename = 'Capisco person2 day2.csv'
filename = '/home/pi/Desktop/Tests/{}'.format(filename)
classifier = machineLearning.training()
arduinoSens = serial.Serial('/dev/ttyACM0',115200)

arduinoSens.write('Startup')
sleep(3)
while arduinoSens.inWaiting ():
    print arduinoSens.read(1)
arduinoSens.write('Position')
oldposition = arduinoSens.readline().rstrip('\r\n')

newPosition=oldposition
print int(oldposition)
try:
    pd.read_csv(filename)
    myfile = open(filename,'ab')
    wr = csv.writer(myfile, quoting = csv.QUOTE_ALL)
except:
    myfile = open(filename,'ab')
    wr = csv.writer(myfile, quoting = csv.QUOTE_ALL)
    arduinoSens.write('yo')



    wr.writerow(["Back1","Back2",
"Back 3","Right","Front Right","Front
Left","Left","Rear","Front","Accell X",
"Accell Y","Accell Z","Gyro X","Gyro Y",
"Gyro Z","Label","Time","Day","Month","Year",
"Tilt Resistance"])
print 'It begins! '


while True:
    while arduinoSens.inWaiting ():
        print arduinoSens.read(1)
    arduinoSens.write('Pressure')

    data = arduinoSens.readline()
    if 'ShutDown' in data:
```

```
        print 'shutdown'
        buttonPressShutdown()
    elif 'Tighter' in data :
        oldposition=str(int(oldposition)+30)
        if int (oldposition)>350:
            oldposition='350'

arduinoSens.write('tighter,{}'.format(oldposition))
        newPosition=oldposition
    elif 'Looser' in data:
        oldposition= str(int(oldposition)-30)
        if int(oldposition)<150:
            oldposition='150'

arduinoSens.write('looser,{}'.format(oldposition))
        newPosition = oldposition

    else:
        data = data.rstrip('\r\n')
        data = data.split(',')

        datapoint = pd.Series(data = data, index =
["Back1","Back2","Back 3","Right","Front Right",
"Front Left","Left","Rear","Front","Accell X",
"Accell Y","Accell Z","Gyro X","Gyro Y","Gyro Z"])
        prediction =
classifier.predict(datapoint.drop(["Accell X",
"Accell Y","Accell Z","Gyro X","Gyro Y","Gyro Z"]))
        data.append(prediction[0])
        datapoint.Prediction = prediction[0]
        sleep(.8)
        datapoint =
datapoint.append(pd.Series([prediction[0],datetime.date
time.now().time().replace(microsecond=0),datetime.datet
ime.now().day,datetime.datetime.now().month,datetime.da
tetime.now().year,newPosition     ],index =
["Label","Time","Day","Month","Year",
"Tilt Resistance"]))
        print datapoint

        wr.writerow(datapoint)
        if  time() - LastChange>900 and 'day2' in
filename:
            myfile, wr, oldPosition =
analyseSeating(myfile,newPosition)
```

```
import DriveUploader2
import serial
import csv
import pandas as pd
import machineLearning
from time import sleep,time
import datetime
from os import system
from random import choice
from os import walk

LastChange = time()
def buttonPressShutdown():
    myfile.close()
    DriveUploader2.uploadFile([filename])
    print 'shutting down in 5 sec'
    sleep(5)
    system("sudo shutdown -h now")

def analyseSeating(myfile,newPosition):
    myfile.close()
    analyzer =
pd.read_csv(filename).Label.tail(1800)#last hour
    overview = analyzer.value_counts(normalize = True)
    print overview
    myfile = open(filename,'ab')
    wr = csv.writer(myfile, quoting = csv.QUOTE_ALL)



    if  overview.index[0]!='Seat available':
        print 'Waaay too {0}... \n{1}%
actually!'.format(overview.index[0],overview[0])
        alternatives = range(1,int(newPosition))+range
(int(newPosition),6)
        new =  choice (alternatives)
        arduinoSens.write('5,{0}'.format(new))
        LastChange = time()
        newPosition = str(new)
        arduinoSens.write('Pressure')
    return  myfile, wr,newPosition,LastChange

TestingFiles = ['Person1,day1.csv','Person1,day2.csv',
        'Person2,day1.csv','Person2,day2.csv']
existingFiles=[]
for (dirpath,dirnames,filenames) in
walk('/home/pi/Desktop/Testing datasets'):
    existingFiles.extend(filenames)
```

```
print existingFiles
print TestingFiles
for i in [3,2,1,0]:
    print TestingFiles[i]
    if TestingFiles[i] not in existingFiles:
        currentFile = TestingFiles[i]

classifier = machineLearning.training()
arduinoSens = serial.Serial('/dev/ttyACM0',115200)

arduinoSens.write('Startup')
sleep(3)
while arduinoSens.inWaiting ():
    print arduinoSens.read(1)
arduinoSens.write('5,0')
sleep(.5)
newPosition = arduinoSens.readline().strip()

filename = '/home/pi/Desktop/Testing
datasets/{}'.format(currentFile)

try:
    pd.read_csv(filename)
    myfile = open(filename,'ab')
    wr = csv.writer(myfile, quoting = csv.QUOTE_ALL)
except:
    myfile = open(filename,'ab')
    wr = csv.writer(myfile, quoting = csv.QUOTE_ALL)
    arduinoSens.write('yo')
    preemptive = arduinoSens.readline()


    wr.writerow(["Front Left+","Front Left","Front
Right","Front Right+","Mid Left+","Mid Left","Mid
Right","Mid Rght+","Back Left","Back Right","Back+
Left","Back+ Right","Top Left","Top Right","Lumbar
Left","Lumbar Right","Accel X","Accel Y","Accel
Z","Gyro X","Gyro Y","Gyro Z","Raw Accel X","Raw Accel
Y","Raw Accel
Z","Label","Year","Month","Date","Time","Tilt
Resistance"])
print 'It begins! '


while True:
    while arduinoSens.inWaiting ():
        print arduinoSens.read(1)
    arduinoSens.write('Pressure')

    data = arduinoSens.readline()
    if 'ShutDown' in data:
        print 'shutdown'
```

```
        buttonPressShutdown()

    if 'Position' in data:

        if 'Position1' in data:
            print 'Something is crappening'
            newPosition = '1'
            arduinoSens.write('5,1')
        if 'Position2'in data:
            newPosition = '2'
            arduinoSens.write('5,2')
        if 'Position3'in data:
            newPosition = '3'
            arduinoSens.write('5,3')
        if 'Position4'in data:
            newPosition = '4'
            arduinoSens.write('5,4')
        if 'Position5'in data:
            newPosition = '5'
            arduinoSens.write('5,5')
        sleep(5)
        while arduinoSens.inWaiting ():
            print arduinoSens.read(1)
        arduinoSens.write('Pressure')
        data = arduinoSens.readline()

    data = data.rstrip('\r\n')
    data = data.split(',')

    datapoint = pd.Series(data = data, index = ["Front
Left+","Front Left","Front Right","Front Right+","Mid
Left+","Mid Left","Mid Right","Mid Rght+","Back
Left","Back Right","Back+ Left","Back+ Right","Top
Left","Top Right","Lumbar Left","Lumbar Right","Accel
X","Accel Y","Accel Z","Gyro X","Gyro Y","Gyro Z","Raw
Accel X","Raw Accel Y","Raw Accel Z"])
    temp = datapoint.drop(["Accel X","Accel Y","Accel
Z","Gyro X","Gyro Y","Gyro Z","Raw Accel X","Raw Accel
Y","Raw Accel Z"])
    prediction = classifier.predict(temp)
    data.append(prediction[0])
    datapoint.Prediction = prediction[0]
    sleep(.8)
    datapoint =
datapoint.append(pd.Series([prediction[0],datetime.date
time.now().time().replace(microsecond=0),datetime.datet
ime.now().day,datetime.datetime.now().month,datetime.da
tetime.now().year,newPosition     ],index =
["Label","Time","Day","Month","Year","Tilt
Resistance"]))
    print datapoint
    wr.writerow(datapoint)
```

```
    if 'day2' in filename:
        if  time() - LastChange>900:
            myfile, wr, newPosition, LastChange =
analyseSeating(myfile,newPosition)
```

Appendix D Pre-Master's Project

The Smart Chair

Osmund Olav Bøe

Fall 2016

PRE-MASTER'S THESIS

Department of Engineering and Materials

Norwegian University of Science and Technology

Supervisor: Professor Martin Steinert

Ass. Supervisor: Andreas Wulvik

Abstract

This thesis describes the processes involved, and choices made in the preliminary work in preparation of the master's project "The Smart Chair", a chair outfitted with sensors, used to classify seating behavior. Through designing, building and testing prototypes, the feasibility and potential of various concepts for a smart chair for research purposes, and commercial applications have been evaluated, and recommendations regarding future work was given. Focus has been on building a platform that can be easily modified to fit changing requirements, and easily implement new features, to best possible facilitate work done in the later stages of development.

Low back pain is the leading cause of disability in the world (Maniadakis and Gray 2000), and potential applications in this project includes research on back pain in relation to seating behavior, prevention of low back pain through and rehabilitation of patients struggling with low back pain through games, personalized feedback on seating behavior and dynamic, moving chairs.

This pre-master's project is a continuation of a project from the course "The Fuzzy Front End" at NTNU spring of 2016, and builds further upon the concepts and prototypes developed in the course.

Preface

This pre-master's project is for 15 credit points in the 9th semester of the master's degree program Mechanical Engineering (MTPROD), with the specialization Product Development and Materials Engineering. The Smart Chair project builds upon the work done by three students (including the author) in the course TMM4245 The Fuzzy Front End during the spring semester of 2016. The project was carried out at the Department of Engineering Design and Materials (IPM) in collaboration with Scandinavian Business Seating.

The course TMM4280, advanced product development was run in parallel with this project, and the module in machine learning was especially helpful, as the author's knowledge on the field was mostly limited to the pragmatic approach of the Fuzzy Front End, limited to how can we make it work, rather than why is the program set up like it is. The better understanding of the basic elements of machine learning will likely prove useful for future programming of features.

Acknowledgements

I would like to thank professor Martin Steinert, for granting me the assignment, and for giving valuable advice during both the Fuzzy Front End course, and the pre-master's project, also co-supervisor Andreas Wulvik, for guidance during this project, and for the help in TMM4245, introducing the team to machine learning, which made this project possible.

I also want to thank my fellow Fuzzy Front End students, Even Jørs and Phillip Anders, for the cooperation and their contributions in Fuzzy Front End course, and consequently to this project, as well as our coach Carlo Kriesi, Scandinavian Business Seating, and my industry contacts there, Erlend Weinholdt and Gaute Hovdal, and finally Paul Jarle Mork for insights to seating behavior, and back pain.

Contents

Chapter 1 - Introduction

The Smart Chair pre-masters-project is a continuation of a project started in the course TMM4245 Fuzzy Front End, and the task is to elaborate functional requirements and specifications for two chairs, one for research purposes, and one for commercial applications. The former with the goal of creating a platform for use in research such as the HUNT study (Helseundersøkelsen I Nord-Trøndelag), and the latter with the goal of testing and evaluating various functions, features and concepts for commercial applications.

Initial research

Research on seating behavior spans between health, productivity, interpersonal relations among others. The health aspect also has potential uses in commercial implementation. The socioeconomic cost of disability due to back pain on an individual and societal level are staggering, both directly through the cost of treatment, but much more so due to production losses and other indirect effects (Maniadakis and Gray 2000).

Research on the linkage of seating behavior and back pain is an ongoing field, and is subject to an increasing use of sensor technology. A literature review from 2000 found little evidence that increased sedentary work is linked to back pain (Hartvigsen J et al. 2000), whereas more recent quantitative studies indicate that the two are correlated (Astfalck et al. 2010; Dankaerts et al. 2006; Gupta et al. 2015).

Due to the socioeconomic cost of LBP, the health aspect of a chair and seating behavior was the main focus when starting the pre-masters-project, as it is highly relevant for both research and commercial purposes.

Similar work

Monitoring aspects of health through use of sensors is not a novel idea. For instance, blood pressure and cholesterol levels are used as a metric for risk of cardiovascular deceases. Indicators of risk of back pain are not as documented, or as widely accepted as the indicators of risk of cardiovascular deceases, but new products coming to, or already on the market aim to predict, prevent, and rehabilitate LBP. The company dorsaVi has made a wearable sensor called ViMove. A company called Sensomative has made a chair cover similar to– though more advanced than The Smart Chair prototype made in the Fuzzy Front End for classifying sitting positions, commenting on posture, and analyzing sitting behavior. Nintendo released a platform for exercise games, focusing on core strength and balance called Wii Fit

Structure of the report

The rest of the report is organized as follows:

Chapter 2 Introduces relevant product development methodology, theory of- and research on- lower back pain, including the related socioeconomic costs, its links to sitting, classification and rehabilitation. Necessary background to the technical aspects of this project, with respect to mechatronics and computer programming is also given in this chapter.

Chapter 3 Introduces some products of interest, that are somehow related to The Smart Chair-project.

Chapter 4 Highlights the work preceding this pre-master's-project.

Chapter 5 Presents some of the work done, the decisions made, and some reasoning as to why certain choices were made.

Chapter 6 Gives an overview of the current state of the project, and the current features and functions of the prototype.

Chapter 7 features the author's recommendations for further work in the upcoming master's thesis.

Chapter 8 Summarizes the project work.

Theory and Low Back Pain

Chapter 2 Theory and Low Back Pain

Low back pain (LBP)

Socioeconomic costs of LBP. Low back pain is the leading cause of disability in the world, and is costly to the society, businesses, and at a personal level, directly through treatment costs and indirectly because of e.g. disability leave (Maniadakis and Gray 2000). Though precise mechanisms causing back pain are not well understood, the reasons are more often than not mechanical, meaning muscle strain, spinal disk herniation, abnormal narrowing of spinal canal, osteoarthritis ("wear and tear") among others are the most prominent causes of LBP (Swezey and Calin 2012).

The increasingly sedentary lifestyle's effect on our back health is discussed in literature, and though prolonged sitting is generally thought of as a cause for back pain, a literature review on the prevalence of LBP in sedentary workers (Hartvigsen J et al. 2000) concluded that there is not enough evidence to claim that there is a higher risk of LBP in sedentary work positions, and that white-collar workers are among the least likely to be disabled by LBP. More recent studies are however seeing correlations between seating behavior and back pain. A study on objectively measured sitting time, where total sitting time, time seated at work, and time seated at leisure were compared to the participants perception of their intensity of LBP. (Gupta et al. 2015). Their findings were that total sitting time has a positive association with a sufferers' perception of high intensity LBP among blue-collar workers.

Two studies (Astfalck et al. 2010; Dankaerts et al. 2006) found that differences in sitting postures in patients with nonspecific chronic low back pain from pain free controls can only be classified when patients are subdivided. Certain conditions could be using various movement tests and posture analyses, highlighting the heterogeneity of nonspecific chronic low back pain, and inferring the possibility of diagnosing chronic LBP based on sitting posture, and analyzing sitting behavior.

Data-generated feedback. The increasing availability of cheap and reliable sensor technology yields opportunities for extensive quantitative research on back pain. A pilot study partially funded by dorsaVi utilized dorsaVi's ViMove motion sensor to compare the effectiveness of generating personal feedback based on sensor readings versus guideline-based care of LBP patients (Kent, Laird, and Haines 2015). The results indicated that there were significant advantages with utilizing data generated feedback. Similar results were found in another study, which suggested that personalized messages - on top of continuous visual feedback, had the potential of influencing activity levels (Weering, Vollenbroek-Hutten, and Hermens 2012).

Product development methodology

Prototype

The understanding of what a prototype is, and it’s role varies in different businesses and industries, and the word is likely to mean different things according to people with different backgrounds (Helander, Landauer, and Prabhu 1997). As this project is rather multidisciplinary, the rather general definition given Ulrich and Eppinger will be the basis for the understanding of a prototype: *“an approximation of the product along one or more dimensions of interest”*. The definition ranges from a storyboard, to a functioning standalone sensor, to a fully functioning product for user-testing.

Prototype-driven development. The reasons for prototyping can vary with the stage of a project, and culture of the business. Schrage suggested two vastly different approaches to prototyping, with specification-driven prototypes, and prototype-driven specifications and (Schrage 1993). Schrage claimed that “*small entrepreneurial companies built around a brilliant product concept tend to be prototype-driven”* whereas larger companies *“draw heavily from market research data before concepts are moved into the prototyping cycle”*.

Proof of concept. A focused prototype, testing one to a few dimensions of a project, or partial solution to a problem, for verifying the feasibility of the concept.

Hunter-Gatherer model. The “Hunter-Gatherer” model (Steinert and Leifer 2012) is a product development model for early stages, when requirements are not fixed, and the problem has a high degree of ambiguity, and the end goal is flexible. Built upon the duality of hunters and gatherers, the hunters with an exploration mindset, determining the aim, and next step of the project, and the gatherers with an optimization mindset. The model is built upon several probing cycles, where the “hunters” move towards the tentative goal, build a/some prototype(s), test, learn and reevaluate the situation, and target. The process of design-build-test is referred to as a probing cycle, or a probe. After several cycles, the “big idea” may become clearer, and at that time, the requirements are set. The model is meant as a tool for finding, and developing towards the next “big idea” during the early concept creation stage, and is helpful when dealing with unknown unknowns.

Product development tactic

Wayfaring and Hunter-Gatherer model. The Wayfaring model, highly related to the Hunting-Gatherer model, and through two case studies (Kriesi et al. 2016), the use of probing cycles for time efficient requirement generation. Wayfaring and Hunter-Gatherer are both cases of the aforementioned prototype-driven specifications philosophy. The approach of iteratively designing, building and testing has been the main philosophy of the work in both the Fuzzy Front End project, and this pre-master's project, using proof of concept prototypes to verify the feasibility of a concept, and to generate specifications for later prototypes.

Scrum. Originating from software development, scrum is used as a product development management strategy. Development work is done in cycles, or "sprints" with goals defined at the beginning of each sprint based upon a prioritized list of input from stakeholders. Usually daily meetings are held to update on status of projects. At the end of a sprint, an implementable product should be ready.

Technology

Arduino

Arduino is a programmable microcontroller based on the C++ language. Arduino is mainly used as a platform for reading sensor values and controlling actuators depending on the incoming voltage. The Arduino can read and send digital binary signals, and read analog signals from 0 to 5 V, returning integers from 0 to 1023. The Arduino has its own integrated development environment (IDE), and though other software can be used to program an Arduino, the Arduino IDE has a massive community, and lots of open source projects. This makes the Arduino a suitable platform for prototyping with sensors and mechatronics with various degree of complexity, as there are many example codes from other projects easily available from the vast community.

Sensors

Force sensitive resistor

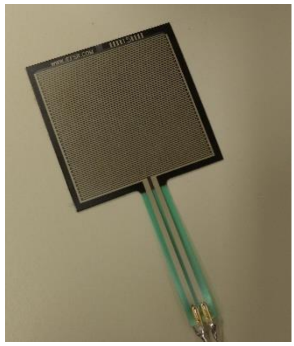


Figure 0-1 Force sensitive resistor

Force sensitive resistors (figure1) are used to measure the force exerted upon it. The sensors resistance is changed with applied force, where higher force yields lower resistance. Coupled with a normal passive resistor, the force sensitive resistor can be used as a voltage divider. The force working on the resistor can then be read as a function of the voltage output. The output varies with force and the passive resistors resistance, an appropriate resistance that has highest resolution in the force levels to be measured should be chosen. For this project a resistor of $10k\Omega$ was used.

Capacitive sensor

The technology used in touch screens. A capacitive sensor can measure the proximity of a conductive object, or an object with a dielectric different from that of air. Used in combination with a compressible foam, pressure and pressure changes can be indicated by the proximity changes.

Figure 0-2Ultrasonic distance sensor

Distance Sensor

There are several available technologies for measuring distance, and many are compatible to the Arduino. In this project an ultrasonic distance sensor (figure 2) was used as it cheap and easily available, and works well enough for this project. The ultrasonic sensor has a transmitter and receiver of ultrasonic signals. The distance is calculated $D = \frac{c * \Delta t}{2}$ where D is distance, c is speed of sound, and Δt is the time from the signal is sent from the transmitter until received by the receiver. It can be noted that the ultrasonic distance sensor has an inherent weakness, as it can only correctly read distances to items perpendicular to the sensor. Is precise at short, to medium distances (few cm to about 3 m).

Alternatively, infrared sensors are also cheap, and easily available and programmable, also using a transmitter, receiver like the ultrasonic distance sensor, without the issue of "flat object not facing sensor" (figure 3) to the same degree as the ultrasonic sensor. Comes in both short range and long range sensors, however in this project, the distances to be measured spanned between the two, resulting in the use of ultrasonic distance sensor instead.

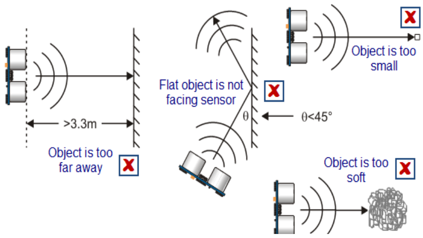


Figure 0-3Weaknesses of ultrasonic distance sensor http://gymlab.dk/robotter/2wd/

Other possibilities of measuring the height settings of the chair includes linear potentiometers among others. Should the height setting be motorized, the positioning system used in the motor could be used to measure the height of the chair.

Gyroscope/accelerometer

A gyroscope is used to indicate an objects orientation about the force vector, which usually is the gravitation direction. Under acceleration, the force vector changes direction in accordance with Newton's second law of motion, resulting in the wrong output when the gyroscope is under acceleration. An accelerometer measures non-gravitational acceleration. Uses include indication of movement level, and object orientation. The gyroscope has an inherent weakness in that it measures orientation When using the two sensors in combination, more accurate angles can be read, when correct filtering is applied, because the effect of acceleration on the gyroscope can be estimated from the accelerometer.

Making sense of data

To make best use of the data from the sensor, computational analysis is a necessity. The e share amount of data points, and dimensionality of the data points would be next to impossible to analyze and visualize without computational aid. There are many tutorials online regarding this subject, but the following paragraphs should provide enough information to understand the concepts introduced in the following chapters.

Python

Python is a high-level, cross-platform programming language, emphasizing code readability, with an extensive library. The Python programming language can be used for data handling purposes, has available libraries for communicating with Arduino, and is widely used for scientific computing. The open source machine learning library "scikit-learn" was used in this project for analyzing and predicting classes incoming sensor data from the Arduino. Python also has packages for emulating keyboard presses.

Machine Learning

Machine Learning is a subfield of artificial intelligence, where various algorithms turn data into knowledge for prediction-, regression- and data-driven decision- purposes without being explicitly programmed.

Supervised machine learning:

Supervised learning is a subfield of machine learning. Using data that is already labeled to predict new incoming, or otherwise unseen data. Used mostly for predicting class labels and regression where continuous values are predicted. In this project, supervised machine learning algorithms are used to classify sitting postures, and whether the chair height is too low, too high or set up properly based on recorded data.

Reinforcement learning

Reinforcement learning is a subfield of machine learning, where the model improves based on a reward function. Often used in interaction problems, it predicts an action for a given state, and through a form of feedback links whether the action was correct, or if another action would be better in the future. After some trial and error, actions are optimized with regards to the reward function.

Chapter 3 - Alternative Products

Products that keep track of activity levels and aim to increase activity level are seeing an all-time high. New advanced tools that analyze activity levels, muscle activity and posture in various ways, for the consumer market and for professional medical and research purposes. In the following paragraphs, certain products that are used in research on- and treatment of- LBP, or otherwise related to this project are presented.

Wearable

ViMove motion sensor system measures movement, muscle activity in the lower back and range of motion among others, and provides visual feedback of various features. ViMove also utilizes the gathered information to generate actionable data, seemingly to good effect

Figure 0-1ViMove by dorsaVi http://us.dorsavi.com/vimove/

(Kent, Laird, and Haines 2015). Primarily targeted at medical use, for diagnostics and rehabilitation, and insurance companies

Exergames

The combination of videogames and exercise, often referred to as "exergames". There are several commercialized exergames platforms, most noticeably, the very successful Wii Balance Board and accompanying Wii Fit games. The Wii Fit games are exercise and balance games controlled mainly through the "Wii Balance Board", and the Wii Remote Plus. The balance board consisting of a plane surface with four load cells used to track exerted pressure, and user acceleration as well as the center of gravity, and the Wii Remote Plus controller utilizing a gyroscope/accelerometer with 6 degrees of freedom. There is research showing it's potential as a tool for rehabilitation with low back pain sufferers (Seong-Sik Kim et al. 2014) and balance retraining with stroke victims through the "Wehab system"(Kennedy et al. 2011), though this was not expressed as a goal for Nintendo. Its value lie mostly in keeping people active through accessible and fun exercises at a low price. In 2012, the Wii Balance Board was awarded with the Guinness world record “The best-selling personal weighing device” with over 32 million devices sold.

Figure 0-2 Wii Fit U, Core Luge by Nintendo http://www.sidequesting.com/2012/06/e312-wii-fit-u-gets-to-the-core-of-subliminal-workouts/

Seat based products

Several products are built in to seats, or as accessories to chairs, with applications ranging from integrated massage capabilities to pressure analysis mats for design purposes, office solutions are integrated into chairs, as well as personal heating/cooling systems integrated in a chair. In the following paragraphs, the two products most related to this project are described, and other concepts are mentioned.

BMA, a chair company owned by SBS, has a product portfolio including smart chairs. One, more basic model that intend to make users aware of extended sedentary behavior through vibration of the chair and an app tracking sitting time, and advising when to take a break from sitting. The more advanced chair makes the user aware of how long he/she has been seated, and whether the sitting posture is unhealthy. Practical advice, and visual feedback and data from a person's seating behavior can be accessed by the user.

Sensomative is a start-up company making textiles with pressure sensors integrated for measuring pressure distribution on various surfaces, with a product specialized for office use, recording sitting behavior to give feedback for reducing or preventing musculoskeletal complaints. The author failed to find any information on if, or when their products will be released.

Other, non-sensor based, advanced or smart chair concepts such as Altwork (figure 6) or uChair, are also worth mentioning. Their approach to better ergonomics and work efficiency through integration of the workspace into a single furniture is more radical than that of this project, BMA's product and Sensomative's products. As user's temperature preferences differ, Tempronics has made a chair with integrated heating and cooling functions, proving that a chair can affect seemingly unrelated problems.

Figure0-3 Altwork http://altwork.com/our-gallery/

Chapter 4 - Preceding work

Fuzzy Front End Project

In the Fuzzy Front End project, after having chosen The Smart Chair project, the 3-person team (including the author) started by dismantling a chair to learn how the various mechanisms work. As two of the three had just finished the course TMM4220 Innovation by design thinking, some methods learned there was utilized for empathizing, learning and need finding. A stakeholder analysis was performed, and meetings with representatives from the chair industry (SBS), and health/research-personnel. Through these meetings, valuable insights regarding seating behavior was gained, most noticeably a quick guide on the most important settings of the chair, being the height of the chair and seat depth. The team was told that there is no such thing as a perfect sitting position, “the best position is the next one”. Optimal height was described as “feet firmly planted on the ground, with $90° - 95°$ angle from thighbone to shinbone.

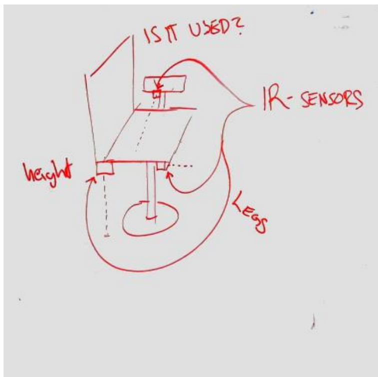


Figure 0-1 Concept 1, Fuzzy Front End

Within short time, the first prototype was built. Distance sensors were mounted to a chair, aimed right below the sitter’s knee to measure the needed seat depth. A concept was developed, but not built, where a set of distance sensors were to measure whether the chair was in use, the height of the chair, and the necessary seat depth (figure 7). That concept had some flaws, as for instance the height measured was absolute height, and did not give any indication about whether the set height was correct for the user or not. In a redevelopment of the concept two distance sensors were to measure distance to the calves, to calculate the angle of the shinbone, and measure necessary seat depth. The whole concept was however scrapped, as the output of the sensors were relying on seating behavior, clothing, and shape of the calves of the person sitting in the chair.

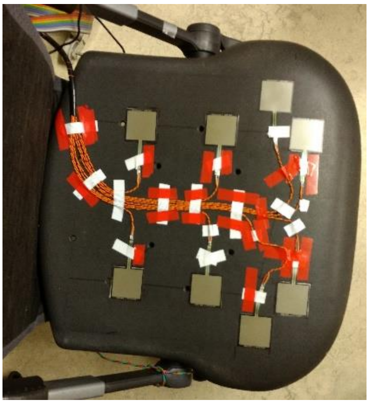


Figure 0-2 Force sensitive resistor matrix

The next iteration was based upon an advice from another group, using a force matrix consisting of a set of force sensitive resistors to recognize sitting positions. Using four force sensitive resistors in the seat, and one in the back of a chair, and hard coding some threshold values, some patterns emerged. The results were not very accurate, but the prototype worked as a proof of concept, showing pressure distribution can be used for classifying sitting positions. The group decided to continue in that direction, and refine the prototype further, with more, and more

accurate sensors (figure 8), and the implementation of machine learning. Though initially planning to implement more sensors, the group focused on learning to program in the Python programming language. The development from there on was mainly iterative. A meeting with Paul Mork gave some new ideas on what the chair could be used for, mainly the possibility of the chair being used to control an exergame, but the platform remained the same for the remainder of the project. For the final presentation, the technology was also built into a high-end office chair (figure 9), with some slight adjustments due to the new form factor of the chair. The Fuzzy Front End project utilized the principles of the Wayfaring- and Hunter-Gatherer -model. Probing until finding the “big idea”, and locking in on requirements from there.

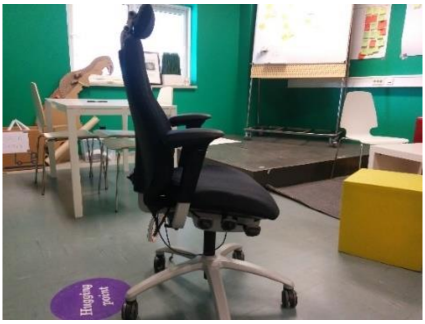


Figure 0-3 Final Prototype, Fuzzy Front End

Chapter 5 - Project work

Product development tactics

The early stage product development, and concept development of The Smart Chair has been the objective for the Fuzzy Front end -project, as well as for this thesis. As such, the focus has mainly been to build quick prototypes to test the feasibility of a concept, or to learn more about a technology, and identifying the unknown factors that may play a part in solving a problem (Gerstenberg et al. 2015). The development has shared many aspects with the methods suggested in wayfaring and Hunter-Gatherer model, with some key differences.

The project has been based on designing, building ant testing deigns in iterations, with goals that shift as new knowledge is attained. Some of the work was done in parallel, though mostly in the Fuzzy Front End course, work was done in cooperation with the whole team, and in this project, the work was obviously not done in parallel as the author was working alone. Though based around teams working together, the wayfaring and hunter-gatherer models have been the basis of the approach to prototyping and development of concepts through the entire project. The approach to product development has not followed a direct methodology, but rather utilized certain aspects from different methodologies that seems well suited at the time.

Task clarification.

The given task is to elaborate functional requirements and specifications for two chairs, one for research purposes, and one for commercial applications. The author's initial thoughts regarding development of a smart chair for research purposes were that the best possible research chair would be one that worked stably, delivers precise data, and measures the necessary aspects of what is to be studied, meaning a chair customized to the research purpose. Consequently, the focus of the pre-master's work on the research chair would entail increasing the stability of the code and making customization of sensor use, which sensors to use, the number of sensors etc. easily manageable.

The goal of the development of the smart commercial chair is to make something people would want to buy, meaning it must offer a feature that other alternatives cannot match, whether it be price, design, or another feature that the user is both aware of, and willing to pay for. Considering features not benefiting the user directly, but that are otherwise useful for SBS, or another part of the value chain could also be useful, but is not the focus of the author at this time. The focus at this time was to specify certain features that could be appreciated by the end users, or the business the end user works in, to consider the feasibility of implementing such a feature in a chair. How the feature could best be implemented has been granted some thought, but has not been prioritized, as it will be further elaborated in the master's thesis.

Alternative products

The products most similar to The Smart Chair prototypes are made by BMA, a Dutch company that has been acquired by SBS. There is not much benefit in making a product too much like their chairs, but should this prove to be the pinnacle of what can be achieved with technology in chairs, that information would also have some value. At the beginning of the project, the existence of a competing start-up company was brought to the author's attention. Sensomative, focusing on research, and expanding its portfolio to commercial applications, complicated the process of elaborating functional requirements of a smart chair for commercial applications, though ultimately not affecting the outcome of the project to a big degree. The awareness did however lead to an evaluation of the opportunities, and reevaluation of the direction of the project.

The effect of feedback and advice based on sensor-data is suggested in research (Kent, Laird, and Haines 2015; Weering, Vollenbroek-Hutten, and Hermens 2012), and the added bonus of not having to mount a sensor to the body could make BMA's and Sensomative's smart products helpful for many. While classifying sitting positions, and resulting advice could be beneficial, that task is perceived as done. Classification will be a vital part of almost all other features and functions, though as a means to achieve the goals, ad not the goal of the project in itself. The question "what advantages are there of having technology integrated in a chair?" arose. Resulting in the plan of making an automatically adjusting chair.

Exergaming. Exergaming for the commercial market is on the rise, games with the goal of activating, or exercising the user, such as the Wii Fit games, and more recently, Pokémon Go. Exercise may not be the goal of the player, but rather a side effect of playing a fun game, which is a goal to strive towards if making an exergame. The use of a Chair as a video game controller could be very interesting with regards to back pain. Precise movement of the trunk can be measured, learning to control posture through a game, activating and stretching key muscle groups among others may serve to severely reduce risk of LBP, and potentially as a rehabilitation tool as with Wii Fit (Seong-Sik Kim et al. 2014).

Other approaches to chairs with expanded functionality focusing on other aspects, like the Altwork's (figure 6) integrated office chair, and Tempronic's temperature control feature invites reflection on whether the development had been somewhat design fixated. The focus had been mostly on use of The Smart Chair as a chair, the approach focused on back pain, with posture control, and increasing the activity while seated as the main goals. There are likely has many other, undiscovered solutions and concepts that could prove useful in many ways.

Beginning the pre-master's project

At the start of this pre-master's project, an evaluation of the current state of the project, and inherent knowledge gaps that would slow down the project. The prototype was functional but for some reason unstable on Windows, which is the most-used computer operating system, and the operating system used by the author. The size of the team was reduced to one, meaning all aspects of the project needed to be well understood by the author.

The first weeks were consequently dedicated to learning more about the Python programming language, and optimizing the code with regards to efficiency, stability, and increased flexibility with regards to using new or different number of sensors, and how the code ran, focused on using The Smart Chair for various purposes in a sequence, as described in the following paragraph.

An issue with The Smart Chair is that the chair settings must be set appropriately for the classification of posture to be accurate, and the data points in the training dataset must be recorded from someone with somewhat similar bodily proportions (height and weight). The classification of whether the chair was set up too low, too high, or correctly was however more accurate for various body types. A code was written, that first helped set the chair to a correct height through classification of pressure data from force resistance sensors. When the chair was set to the correct height, an indication of weight was made, through the aggregate of the force resistance sensor values in the seat, excluding some sensors that are not activated in certain postures, and the height of the person was indicated from the correct seat height as measured by an ultrasonic distance sensor (figure 10). Appropriate data sets were then chosen based on height and weight.

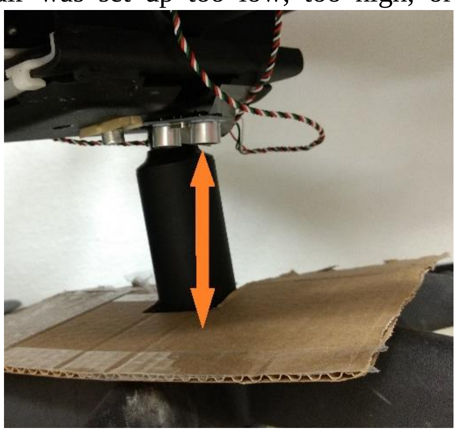


Figure 0-1 Height measurement system

This solution was tested in small scale, and could correctly differentiate between several people. This was however a quick prototype, and improvements such as implementation of machine learning in the choosing of data sets, rather than manually segmenting based on aggregate values, as the posture could influence the indicated weight, and an alternative means of height measurement. At a later stage, the height function will likely be motorized, and height measurement could be done through motor control. For schematics of this prototype, see appendix A

This entire process could be done by the means of having personal sitting data for each chair, or differentiating between who is sitting in the chair through a user profile and a “sign in” sequence, Bluetooth, NFC, or other technology. As the chair, regardless of chosen recognition technology would need to be set up correctly to accurately predict sitting positions and posture, the chosen method seemed preferable due to fewer steps involved from the user, and less technology needed to be prototyped. This decision should however be reevaluated before a potential commercial release, as user preferences vary, and whether only one person uses the chair, the chair is used in shift-based work, or in a “hot desking” environment.

Prototyping the dependencies of a function

The goal of making a chair that can automatically adjust its settings correctly was articulated early in the project from the author, as well as from other stakeholders. With a working classification system for whether the height setting was set up properly or not, making an Arduino act upon that would be the next step towards reaching the previously mentioned goal.

To build and test output control through Arduino, several small iterations were done. First, setting up the Arduino with control through the serial port, meaning opening communication to the Arduino, and associating certain incoming signals with an action. In this case, the signals controlled the color and intensity of an RGB LED. As the ability of sending signals to the Arduino was already implemented in the code, control of Arduino output based on seating classification was now made possible. Though not reaching the goal of automating chair adjustments, some of the aspects needed were implemented successfully. The addition of a motor to control the setting of the chair will be evaluated further, and likely implemented in the master’s project.

Unforeseen improvement potential. The successful implementation of Arduino control in Python led to new ideas for improving stability, and performance of the code. Though the code was made more stable on Windows previously, the startup sequence was not, as the first of initiation resulted in some wrong predictions, caused by Arduino signals sent before the Python program was fully initiated. Adding a feature to the Arduino, to not start sending values until the Python program sends a signal indicating full initialization, eliminated the problem of wrong readings in the initialization, and the resulting noise in the seating data.

Though not a distinguishing feature in itself, the control of Arduino output from Python means the opportunity of controlling motors among others, it enables many opportunities. Another resulting possibility is changing between different Arduino codes, controlling which sensors values are to be read and the frequency of sensor data readings from within the Python code, or even sending sensor data on demand. The sensor data requirements vary with the different applications of The Smart Chair. Controlling a videogame, requiring as little lag as possible, recording training data requires the possibility of labeling the data, and actively choosing when data is recorded, motor control requires initialization of motor control etc. Usually the functioning of a Python script would require uploading an appropriate Arduino code, but with some tweaking, Arduino output could be optimized for each of the varying functions from within the Python script.

The implementation of an automated height adjustment is likely to have some yet undiscovered problems. The feature can however likely be implemented in the manner described in the following, though some deviations due to unforeseen challenges are expected. Finding a proper height first, and recording pressure data as the height varies, recording and labeling pressure data with respect to deviation from proper height, and the direction the motor is running. The recorded data can then be used to recognize which way, and how far to adjust the seat to have a properly adjusted chair.

Ideas not prototyped:

Some ideas and concepts that have not been tested, or been mentioned in the preceding chapters, but are worth noting, and could be interesting to revisit are mentioned in the following:

Alternative height adjustment. Seeing that almost all office chairs are used in combination with a desk, or a table of some sort, having a chair set up optimally is also dependent upon the rest of the work space. With this in mind, the best approach to height adjustment could be to change the "ground level". The chair height could be optimized to give the best possible work position for the accompanying desk, as a footrest as seen on figure 11, or a similar platform was elevated or lowered to adjust best to the user's body.

Figure 0-2HÅG Foot Ring, URL: http://www.haginc.com/products/hag-foot-ring/

This could e.g. be used with a standing desk, without having to change the height of the desk when changing between working seated and standing.

Work station. Somehow implementing a small mouse pad and a keyboard into the chair, partially addressing the previous issue with improper desk height, and as a means to have a computer station without needing to take up space with a desk. Could be useful for office workers that only need a digital workspace, i.e. a monitor, and input devices. As there are other solutions addressing this issue, such as couch gaming lapboards, and the Altwork station (figure 6), the idea has not been further pursued.

Docking station. The idea of having the chair capable of charging phones and computers, seems somewhat practical. However, as the market for docking stations seem oversaturated, and the concept seems more suited for a product on a desk than integrated to a chair, little time was spent analyzing and pursuing this concept.

Chapter 6 - Current state

Though somewhat touched upon in the preceding chapters, a full summary of the functions and features, as well as recommendations of further work will be exclaimed in this chapter.

Basis platform

The product built is a chair, with sensors mounted, that can record data on seating behavior, and classify sitting posture. At the end of this pre-master's project, the chair is outfitted with an ultrasonic distance sensor, measuring the height setting of the chair, and 10 force sensitive resistors mounted on the seat and back-rest, measuring pressure distribution. This sensor data is interpreted by a supervised machine learning algorithm, called "Random Forest", to classify whether the chair adjusted to an appropriate height, and classify seating postures. The format, and frequency of the sensor data, and other signals can be controlled from the Python script, to ensure best possible fit between sensor data retrieval and application.

Information output

The platform can provide the user with personal feedback on how the users seating behavior could be improved, either through instant feedback through a LED, or as a more detailed report with additional advice.

Physical output

The Arduino can be controlled from the Python script, and thus, physical output can be controlled by seating postures, and positions. This enables the control of motors, LEDs, or any of the devices controllable by an Arduino, from the chair. At the end of this project, the applied use of this is instant feedback, and guidance for chair height setup.

Digital output

One of the Python scripts can emulate keyboard presses, resulting in some simple games being controllable with body movements in the chair. Emulation of quick presses, and pressing then holding buttons are possible, and can be utilized appropriately according to which game is to be controlled.

Chapter 7 - Further work

For the Master's project, the author recommends the following approach, for achieving higher precision data readings and prediction accuracy, as well as implementing new features for commercial applications.

Chairs. To test the potential of various chairs to be rebuilt into Smart Chairs is advised, particularly RH Mereo, HÅG SoFi and HÅG Capisco.

Actuators. The addition of a motor controlling the height of the chair, as well as an actuator adjusting the tilt mechanisms of The Smart Chair should be added, to automate chair adjustments, and in other ways affect seating behavior.

New sensors. The addition of new sensor types, including capacitive sensing with compressible foam as a position and pressure indicator, and a gyroscope-accelerometer combination. The height ultrasonic distance sensor should be replaced with a linear potentiometer or if height is controlled by a linear servo, the built-in positioning system can be used.

Machine Learning. Initial testing of other supervised machine learning algorithms seemed promising. Additional testing and evaluation of various machine learning algorithms should be done after the addition of new sensors. The feasibility of using reinforcement learning to make a chair that somehow reacts to certain seating behavior, nudging towards better seating behavior without being intrusive, should be evaluated.

Raspberry Pi. For The Smart Chair to be commercially viable, the requirement of having the chair connected to a laptop or desktop computer must be dealt with. In the Fuzzy Front End, some work was dedicated to porting the Python script to a Raspberry Pi, some of the features ran OK, but further work is needed to integrate all computation into the chair.

Exergaming. It would be interesting to further test the gaming capabilities, and potential as a rehabilitation tool in cooperation with a back-health expert. Finding a game with a suitable control scheme to build upon could prove difficult. The best solution might entail making a custom game for this application. Consequently, the author advices keeping the idea in mind further, though not prioritizing it for the master's thesis.

Chapter 8 - Summary

During the pre-master study, concepts for a smart chair for research purposes, and a smart chair for commercial applications have been generated and evaluated. Several prototypes were built based upon a prototype chair, capable of classifying sitting postures. These prototypes were used to test feasibility and evaluate various concepts, and recommendations to give recommendations for further work. Initial planning for various concepts regarding motorized chair adjustments, seated exergaming for prevention of- and rehabilitation from- low back pain, implementation of new sensors in The Smart Chair and changes in the accompanying software has been undertaken. The main bulk of the work has gone into planning for the master project, and building a platform that can be easily modified to fit changing requirements, and easily implement new features.

References

Astfalck, Roslyn G., Peter B. O'Sullivan, Leon M. Straker, Anne J. Smith, Angus Burnett, Joao Paulo Caneiro, and Wim Dankaerts. 2010. "Sitting Postures and Trunk Muscle Activity in Adolescents With and Without Nonspecific Chronic Low Back Pain: An Analysis Based on Subclassification." *Spine* 35 (14): 1387–95. doi:10.1097/BRS.0b013e3181bd3ea6.

Dankaerts, Wim, Peter O'Sullivan, Angus Burnett, and Leon Straker. 2006. "Differences in Sitting Postures Are Associated With Nonspecific Chronic Low Back Pain Disorders When Patients Are Subclassified:" *Spine* 31 (6): 698–704. doi:10.1097/01.brs.0000202532.76925.d2.

Gupta, Nidhi, Caroline Stordal Christiansen, David M. Hallman, Mette Korshøj, Isabella Gomes Carneiro, and Andreas Holtermann. 2015. "Is Objectively Measured Sitting Time Associated with Low Back Pain? A Cross-Sectional Investigation in the NOMAD Study." *PLoS ONE* 10 (3): 1–18. doi:10.1371/journal.pone.0121159.

Hartvigsen J, Leboeuf-Yde C, Lings S, and Corder EH. 2000. "Is Sitting-While-at-Work Associated with Low Back Pain? A Systematic, Critical Literature Review." *Scandinavian Journal of Public Health* 28 (3): 230–39.

Helander, M. G., T. K. Landauer, and P. V. Prabhu. 1997. "What Do Prototypes Prototype?" In *Handbook of Human-Computer Interaction, 2nd Edition*, 2nded. North Holland. http://proquestcombo.safaribooksonline.com/book/electrical-engineering/computer-engineering/9780444818621/part-ii-design-and-development-of-software-systems/chapter_16_what_do_prototypes.

Kennedy, M. W., J. P. Schmiedeler, C. R. Crowell, M. Villano, A. D. Striegel, and J. Kuitse. 2011. "Enhanced Feedback in Balance Rehabilitation Using the Nintendo Wii Balance Board." In *2011 IEEE 13th International Conference on E-Health Networking, Applications and Services*, 162–68. doi:10.1109/HEALTH.2011.6026735.

Kent, Peter, Robert Laird, and Terry Haines. 2015. "The Effect of Changing Movement and Posture Using Motion-Sensor Biofeedback, versus Guidelines-Based Care, on the Clinical Outcomes of People with Sub-Acute or Chronic Low Back Pain-a Multicentre, Cluster-Randomised, Placebo-Controlled, Pilot Trial." *BMC Musculoskeletal Disorders* 16 (May). doi:10.1186/s12891-015-0591-5.

Kriesi, Carlo, Jørgen Blindheim, Øystein Bjelland, and Martin Steinert. 2016. "Creating Dynamic Requirements through Iteratively Prototyping Critical Functionalities." *Procedia CIRP*, 26th CIRP Design Conference, 50: 790–95. doi:10.1016/j.procir.2016.04.122.

Maniadakis, Nikolaos, and Alastair Gray. 2000. "The Economic Burden of Back Pain in the UK." *Pain* 84 (1): 95–103. doi:10.1016/S0304-3959(99)00187-6.

Schrage, Michael. 1993. "The Culture(s) of PROTOTYPING." *Design Management Journal (Former Series)* 4 (1): 55–65. doi:10.1111/j.1948-7169.1993.tb00128.x.

"Scrum for Mechanical Product Development Teams - 191951.pdf." 2016. Accessed December 14. http://publications.lib.chalmers.se/records/fulltext/191951/191951.pdf.

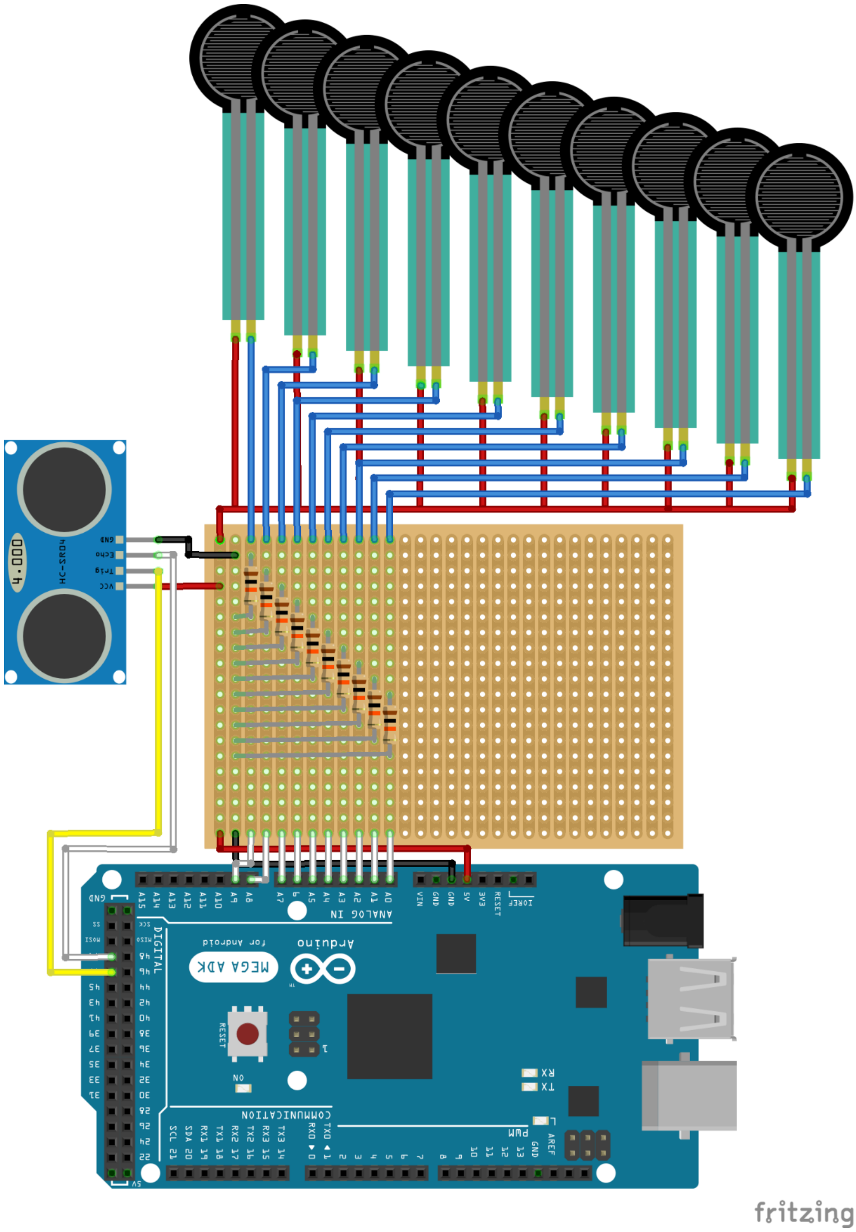
Seong-Sik Kim, Won-Kyu Min, Jung-Hee Kim, and Byoung-Hee Lee. 2014. "The Effects of VR-Based Wii Fit Yoga on Physical Function in Middle-Aged Female LBP Patients." *Journal of Physical Therapy Science* 26 (4): 549–52. doi:10.1589/jpts.26.549.

Steinert, Martin, and Larry J. Leifer. 2012. "'Finding One's Way': Re-Discovering a Hunter-Gatherer Model Based on Wayfaring*." *International Journal of Engineering Education* 28 (2): 251–52.

Swezey, Robert L., and Andrei Calin. 2012. *Fast Facts: Low Back Pain*. Vol. 2nd. Fast Facts. Oxford: Health Press. http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=439683&site=eds-live.

Weering, Marit G. H. Dekker-van, Miriam M. R. Vollenbroek-Hutten, and Hermie J. Hermens. 2012. "Do Personalized Feedback Messages about Activity Patterns Stimulate Patients with Chronic Low Back Pain to Change Their Activity Behavior on a Short Term Notice?" *Applied Psychophysiology and Biofeedback* 37 (2): 81–89. doi:10.1007/s10484-012-9181-6.

Appendix A – Schematics



Appendix B - Product URLs:

ViMove: http://us.dorsavi.com/vimove/

Sensomative: http://sensomative.com/en/

Wii Fit U: http://wiifitu.nintendo.com/

Tempronics: http://www.tempronics.com/products.html

Altworks: http://altwork.com/