[FeetLifter]

Final report Engineering Design (4WBB0)

[1 November 2021]

Group no: 276	
Name	Student ID
Stan Kobussen	1623559
Jyri Wagenaar	1590308
Mark te Lintelo	1564412
Bente Elst	1451189
Lars Schothuis	1600346
Amina Abdirahman	1587641

1. Group effectiveness

Strengths and weaknesses

For this course, the groups are formed by combining students from different majors. This inherently means that a group will have certain strengths as well as weaknesses. In this group, students are present from Industrial design, Building sciences, Industrial engineering, Automotive technology and Mechanical engineering. At the start of the project, students were asked to list their personal strengths and weaknesses as well as aspects of the project they could potentially contribute to. Without going into too much detail about each individual group member, a few strengths and weaknesses that were listed include:

Strengths:

- Sketching and AutoCAD
- Making of presentation videos and animations
- Making of cardboard designs
- Social skills
- Creative thinking
- Mechanical knowledge
- Experience in design projects

Weaknesses:

- Lack of experience in working with microprocessors such as Arduino combined with sensors and actuators
- Lack of contribution to group meetings
- Occasionally losing overview of the project as tasks get divided over the group members
- Occasionally waiting till the last few days to finish assigned tasks

This list seemed like a good starting point for the project. However, every member of this group is only a second-year student which means that the knowledge of each individual major will be limited. Therefore, it is important that the group is willing to learn and develop new skills. Luckily, all group members seemed to be enthusiastic to learn new things, expressed what tasks would fit them best and shared their knowledge with the group.

Development of group effectiveness

During the first couple weeks of the project, in the concepting phase, the skills of sketching and making cardboard designs proved to be helpful. These tasks were assigned to students with previous experience in such assignments. In general, the tasks were divided in a way that group members could use their skills and experience for the benefit of the whole group. This was also applied for the intermediate presentation. The student who has the best presentation skills was chosen as the presenter and the experience in making presentation videos and animations was used to make the design video. Considering the weaknesses, at the start of the project, some group members found it difficult to speak a lot during group meetings. Everyone in the group agrees that the input of each individual group member is valuable to make the project work. Therefore, the group encouraged individual group members to share their knowledge, speak up in discussions and ask questions. For example, during the meetings, the chairman often asked group members for their opinion to get the best of everyone's ideas. In the end, it was good to see that said group members became more confident and active in the meetings over time. As for the technical weakness of a lack of experience in working with electronics, multiple group members worked together in order to research relevant topics. By completing these tasks in a combined effort, everyone was able to learn something from it.

2. Design goal

For this assignment, an aid for personal in-house use must be created for elderly people or people with an impairment or disability. Before thinking of the product itself, it is important to answer some questions to create a clear view of the overall design goal.

What disability or impairment?

The product will be made for people who suffer from Parkinson's decease. The Parkinson's disease is a fairly frequently occurring degenerative illness. It is a disorder of unknown cause that affects as many as 1,4% of all individuals over age 55 (Swerdlow et al., 1996). Degenerative diseases, occur due to wear and tear aging of the body. These mean diseases have the characteristic where they are accompanied by a gradual decrease in one or more bodily functions. The Parkinson's disease effects the human central, peripheral, and enteric nervous systems (Braak & Braak, 2000). The underlying pathological process progresses slowly but relentlessly and involves multiple neuronal systems. The disease is the consequence of changes in the neuronal cytoskeleton developing in only a few susceptible types of nerve cells. Because the occurrence of Parkinson's disease follows no recognizable genetic pattern, it is thought to be a sporadic disorder. However, relatives of Parkinson's disease patients have a higher risk of developing the disorder than does the general population (Nimwegen et al., 2011).

What type of user?

In time, most people with the Parkinson's disease will not have much control over the muscles in their legs and therefore get trouble with walking. At a further stage of the disease, people will walk with a walker to stay in balance. Next to this, they won't be able to lift their feet anymore.

Our product is meant for those people with Parkinson's disease who have difficulty walking. According to experience from one of our group members (Bente), people who suffer from Parkinson's disease, have trouble walking. At a further stage of the disease, patients are not able to lift their feet anymore and will start to shuffle. This is due to trouble regulating the speed and size of their movements. These patients have trouble getting over small obstacles like a threshold or even a carpet. They will lose their balance and are at risk of falling. This problem occurred often when Bente was working in a nursing home for elderly. Next to this, according to a research (Jonasson et al., 2018), 33% of people with Parkinson's disease fall frequently. A lot of these people cannot get up by themselves after they fall. Also, 11% of participants in a research study, see 'Fear of Falling (FOF)' as the most stressful physical symptom. Our product is meant for those people with the Parkinson's disease who have trouble lifting their feet. If people with Parkinson's disease are Afraid of Falling, our product should help them to feel safer while walking in houses.

Which in-house activity?

The product is meant for people with the Parkinson's disease who have trouble walking. Walking is the in-house activity which of course can be in-house, in a care-house or another house. The problem people with PD have is that they cannot lift their legs properly. They often fall because their feet are stuck behind a bump or a step-up. The ability to walk is a freedom which some people with the Parkinson's disease cannot feel anymore. Therefore, is would be great to make a product which could help them walk more easily again and give them the ability to get around the house. Without needing the help of an assistant for example.

What kind of aid?

The problem is that people with the Parkinson's disease cannot lift their feet over bumps or obstacles. Therefore, people with the Parkinson's fall often or are not able to walk around the house on their own. Our product should help to get the person over the bump or step-up for example. First, while walking, it should detect a step-up or a bump. After that, the product should help the person to get over or on top of it. This can be seen as a walking assisting product. However, it can be seen as a falling prevention product as well because it will hopefully also reduce the chances of falling, and therefore reduce the overall risk of walking for those people with the Parkinson's disease. With the help of this device, they can move around the house autonomously. No further help of a caretaker will be needed, and the patient can stay at their own residence for a longer period. If not, the product could still help in the care-houses.

Why this idea?

We came up with this idea because Bente had experience in an elderly care. Here she experienced a lot of problems with people with the Parkinson's disease. The employees at the care house needed-to help these people often by lifting their legs. Also, people were falling over bumps and step-ups which sometimes caused them to break some bones.

Why is it innovative?

We think this is an innovative product because this kind of product is not yet in the market. There are of course many robot legs and arms for example which totally take over the body, but these can cost ten thousands of euros. Much cheaper products which can detect obstacles and help you step over it do not exist yet. This results in an extremely challenging and possibly complex project which we, as a group, are motivated to make a successful one.

3. Functional design and solutions

This chapter contains a list of functional specifications about the design. The specifications are ordered by the MoSCoW method. The solution-encyclopedia is given for each function.

Must have:	**Solution-encyclopedia**
Sensor and actuator	Solution 1: Distance sensor Solution 2: Accelerometer Solution 3: Sound meter Solution 4: LED Solution 5: Servomotor
Perform 1 complete step	Solution 1: Fixed rotation angle Solution 2: Fixed rotation time Solution 3: Measure when the obstacle is passed Solution 4: Use a button to stop the mechanism Solution 5: Use an accelerometer to find the position of the shoe in comparison with the obstacle
Should have:	**Solution-encyclopaedia**
Easy to use	Solution 1: The shoe should be balanced Solution 2: The user needs support to lean on, a walker for example Solution 3: Easy to attach to walker and feet Solution 4: Visual output, such as a display or light Solution 5: Light weight
Warning system for an incoming obstacle	Solution 1: LED Solution 2: Sound Solution 3: Vibration Solution 4: Display Solution 5: Message on phone
Not activate for walls or very high objects	Solution 1: Multiple distance sensors Solution 2: Coding to execute different tasks for different objects Solution 3: Camera to observe the object Solution 4: Human signal, such as pressing a button
Could have:	**Solution-encyclopaedia**
Wireless connection	Solution 1: Bluetooth Solution 2: Wi-Fi Solution 3: Frequency Solution 4: Light signal Solution 5: Sound signal
Universal fit	Solution 1: Adjustable in length -Attachable and detachable by: Solution 2: Magnetic force Solution 3: Velcro straps Solution 4: Buckles
Rods grip on ground while lifting	Solution 1: Slow movement Solution 2: Rubber contact area Solution 3: High contact area Solution 4: Strong components

Won't have:	Solution-encyclopedia
Lift a human's weight	Solution 1: Stronger motors Solution 2: Larger shoe, so there is place for stronger motors Solution 3: Higher quality materials which can carry more weight
Flexibility shoe box	Solution 1: Rubber sole for high grip and dampening Solution 2: Textile for flexibility of the shoe Solution 3: Leather for flexibility of the shoe Solution 4: Spring sole for more dampening
Lift over higher objects	Solution 1: Bigger shoe, so that the rods have the space to be bigger Solution 2: Longer rods Solution 3: Rods attached lower to the ground

4. Design concepts

This section will go into three possible concepts that fit the design goal.

Solution 1: Shoe lifter with rope

The first solution is a system in which the person's foot is lifted by the means of a rope. The sensors for obstacle detection are attached to the shoe in this case. Whenever the sensors detect that the person is standing in front of an obstacle, the processor gets a signal. At this moment the rope needs to be pulled in. This could for example be achieved by attaching a pulley with a small motor to the person's body (attached to a belt). Afterwards, the foot needs to be put back on the floor so the motion should be reversed. See Figure 1.

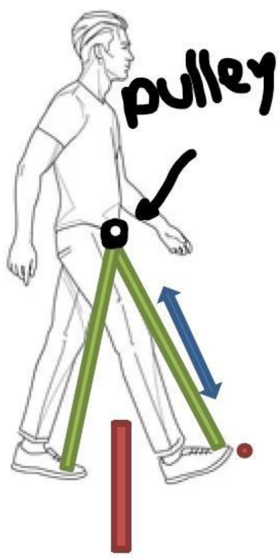

Figure 1: design concept rope

Positives:

- This system is quite compact since there is no frame or plating around it.

- With this sort of system, the amount that the foot is lifted can also be controlled.

Negatives:

- Whilst pulling on the shoe, the rope also pulls the person down to the floor. For people who already have trouble walking, this added instability may lead to dangerous and unwanted situations.

- The rope could get tangled up and potentially create a dangerous situation.

- The sensors are placed on the shoe which makes them vulnerable. Next to this,

small movements of the foot can cause errors or even completely wrong readings.

Solution 2: Shoe lifter standalone

Description: The second idea is that of a shoe that can lift itself up by the means of a rotating mechanism. In this case, as with the first solution, the distance sensors are placed on the shoe. Rods that are connected to electric motors are mounted on the side of a box that is placed under the shoe. Here, the processor that controls the movement needs to be placed somewhere on top of the shoe. Each side of the shoe has two rods to provide sufficient stability for the user, these two individual rods are connected through a chain or a belt. To exert a force onto the ground that is great enough, some sort of small gearbox may need to be used that increases the torque of the rotation. See Figure 2 and Figure 3.

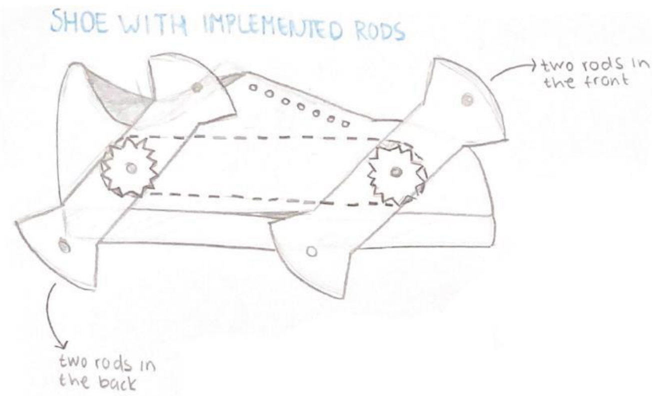

Figure 2: gear system with rods sketch

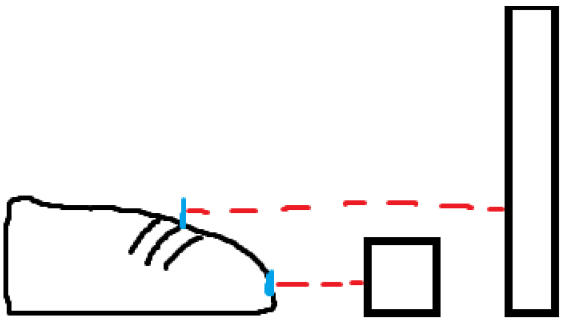
Figure 3: placement and working distance sensor shoe sketch

Positives:

- This sort of mechanism creates a movement that goes both up and forward. In this way, the user gets lifted and over the obstacle at the same time.

- With this idea, the force gets exerted onto the floor instead of onto a rope connected to the person. This only forces the user up a little which creates more stability than the first solution.

Negatives:

- People who have trouble walking will often use a walker. Because the sensors are mounted on top of the shoe, the walker may get in front of the sensors and could be wrongly detected as an obstacle.

- The box that is placed under the shoe will take away from the flexibility of the shoe sole.

- By placing the sensors on the shoe, small movements of the foot may cause inaccurate readings.

- The sensors and processor on the shoe are vulnerable.

Solution 3: Shoe lifter in combination with walker

Description: The third solution is more of an elaboration of our original idea that was the second solution. In this final solution, a walker is integrated into the system. The shoe with its mechanism and actuators remains the same as in the second solution. But in this case the sensors and the processor are mounted onto the walker. Contrary to the second solution, the walker will now detect the obstacle. The idea is that the walker will give off a small signal to the user upon detection of an object. This can be done in the form of a small sound for example. The processor will then start a delay of set time after which the electric motors on the side of the shoe will start rotating. During this delay, the user is supposed to get ready to get lifted over the obstacle. See Figure 4.

Figure 4: picture walker

Positives:

- This sort of mechanism creates a movement that goes both up and forward. In this way, the user gets lifted and over the obstacle at the same time.

- With this idea, the force gets exerted onto the floor instead of onto a rope connected to the person. This only forces the user up a little which creates more stability than the first solution.

- With this solution, the sensors are placed on the walker which keeps them parallel to the surface. This will make for more accurate readings.

- This solution considers the fact that people who have trouble walking will often use a walker.

- By placing the processor and sensors on the walker instead of on the shoe, they are better protected.

Negatives:

- The box that is placed under the shoe will take away from the flexibility of the shoe sole.

- For now, the connection between the processor on the walker and the actuators on the shoe needs to be hardwired.

5. Final design concept

Comparison

The first idea has as an advantage that it does not need anything under the shoe, comparing it to the other two ideas. However, this idea can create more danger because the ropes get stuck behind things, as well as the system failing easily. For people who already have trouble walking, the rope system will make it even harder. Therefore, this first idea can be dismissed because of the instability of this mechanism.

Now two solutions with the same concept of rotating rods are left. The concept of a platform under the shoe is exactly the same except one thing. In the 2nd solution, the sensor which detects if an obstacle is coming, is placed in the front of the shoe. While in the 3rd solution, this sensor is placed on a walker. When comparing the third solution to the second solution, the third solution adds a few more positives.

The first reason to come up with this 3rd solutions, was because people with Parkinson's disease who have difficulty walking, the user of our product, will already walk with a walker. This solves the instability problem of the second solution as well because a rotating shoe, without any assistant will probably cause the user to fall more often than not using the product.

Secondly, placing the sensor on the walker, will give more accurate measurements. A shoe will move up and down a bit, has different angles, which will make it much more difficult to make a reliable measurement out of the shoe. The walker is moving horizontally over the ground so this will make these measurements much more reliable. A sketch of how the sensor will fit onto the walker can be seen in figure 6.

The third solution only adds one negative, namely the wired connection between sensors on the walker to the shoe. This makes it hard to install the product, because the wires need to be attached to the walker with a tape or some tie-wraps, which also makes you attached to the walker while using it. Also, the sensors itself needs to be attached on the walker. Comparing it to the shoe only, the shoe has it all in one so only putting on the shoe platform properly should be enough to use the product.

All in all, the third solution seems to be the best for our problem statement because how user-friendly the product is, can be developed more in the future while the actual concepts of the product will stay the same. Therefore, this is the solution we will attempt to develop.

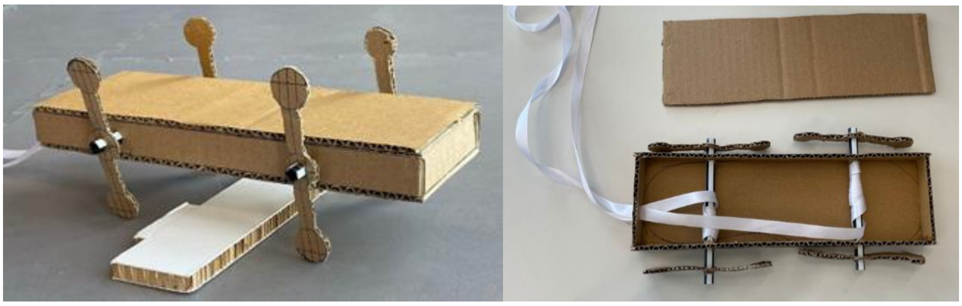

Figure 5: cardboard prototype of shoe with rods

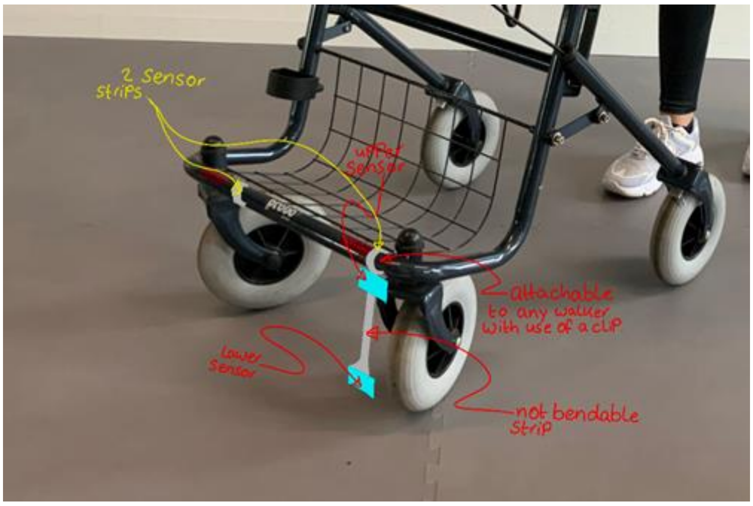

Figure 6: sketch of how sensors will fit on walker

6. Technical specification

Must have:	
Motor torque	The three motors must have enough torque to be able to lift the weight of the wooden box and a shoe which is around 1.8 kg. So, the motors must have a torque of over 7.6 kg cm.
Degree's rotation	The rods must make a rotation of exactly 180 degrees, to set one step. For this the code must let the rods turn for the exact amount of time to turn 180 degrees.
Motor timing	The Arduino code must make the three motors start working simultaneously to lift the shoe and keep it stable.
Rod's angle	The rods must all be mounted on the box at the exact same angle, which is parallel to the length of the box. So, the rods won't touch the box, shoe on top of the box, and lift in the same direction.
Rod's size	The rods all must have the same dimensions and these dimensions must be so that the rods will not touch each other or the ground when in a horizontal position.
Rods placing	On the side of one rod, the rod must be placed exactly in the middle of the box's length. On the side of two rods, the rods must be placed exactly on ¼ and ¾ of the box's length. All rods must be placed at the exact same height on the box.
Height	The rods must lift the box at least 5 centimeters heigh to get over most obstacles.
Forward distance	The box must lift the shoe at least 5 centimeters forward to get over or on top of most obstacles.

Supporting weight	The box must support the full weight of a human (80kg) to be able to walk/shuffle on the box.
Connection to power source	The two distance sensors should be connected parallelly to the 5v pin on the Arduino nano, so both sensors get 5 Volts. The three motors should be parallelly connected to the 6v battery, so all motors get 6 Volts.
Work autonomously	The rods must automatically start turning if the sensors sense an obstacle.
Should have:	
Delay before activation	The code should create a delay of at least 5 seconds between the time the sensors sense an obstacle, and the LED goes on and when the motors start rotating the rods.
sensors	The two distance sensors should sense the difference between an obstacle you can step over and a wall or obstacle you cannot step over.
Batteries	The design should work on two batteries, one 5v battery for powering the Arduino nano and one 6v battery for powering the three servo motors. With batteries the design is portable.
Sensor accuracy	The distance sensors should have an accuracy of at least 3 centimeters to measure obstacles correctly.
Motor power	The three motors should all turn at the same speed to keep the shoe stable while lifting.
walker to shoe cable	The cable from the walker (breadboard) to the shoe box (servo motors), should be connected to on the back of the shoe box, so it will be less in the way with walking.
Could have:	
Wireless connection	Motors could be wirelessly connected to the sensors. So, there won't be a wire between the walker and the shoe.
Universal fit	The design could be used on different walkers and shoes.
Rubber grip	The rods could have rubber on their ends, so they have more grip on the ground while lifting,
One microcontroller	The design could be made with only one microcontroller.
Won’t have:	
Lift a human	De motors won’t be strong enough to lift the weight of a human.
Flexibility	The shoe box will not be as flexible as a regular shoe.

7. Detailing

This section will go into the ideas and thoughts behind the detailing of the final design concept. The main focus here will be on three key components. First, the choice and placement of the electric motors will be discussed. Secondly, the design of the rotating rods is examined. Then, the final wiring scheme with its software will be explained. The detailing of components is not limited to these three subjects however, other parts that were detailed include the placement of sensors, Arduino and battery pack on the walker. The design of the box underneath the shoe was also optimized to save as much weight as possible whilst still providing sufficient strength to support the weight of a person.

Electric motors

Shortly after starting the design phase, it became evident that fully lifting the weight of a person would be difficult considering the available budget. That is why it was decided to first focus on lifting the weight of the device itself for this prototype. Through some rough estimations about the mass of the required components, the total mass of the device was estimated to be around 2 kg. To calculate the required torque of the electric motors to get the device off of the ground initially, Figure (7) is used.

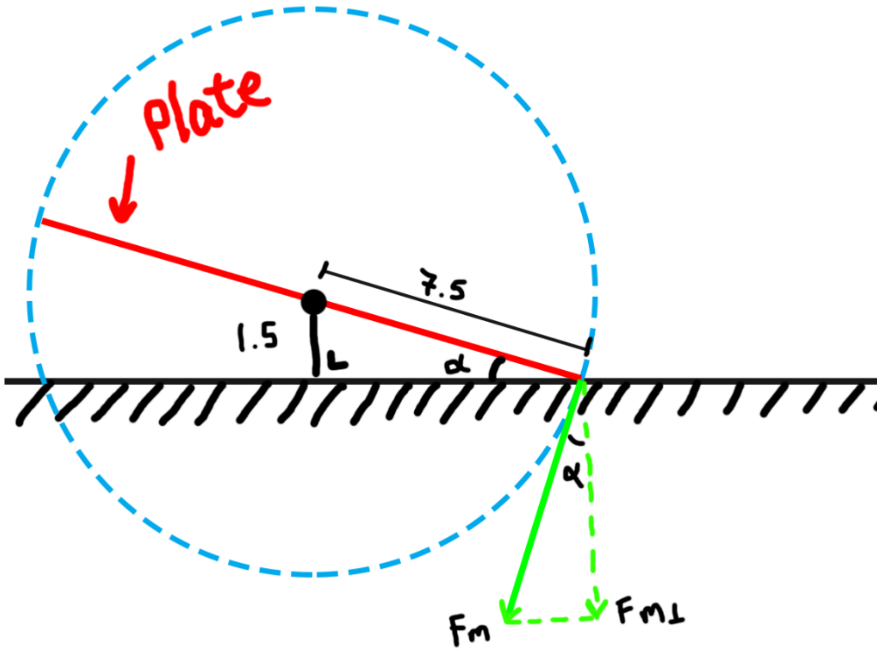

Figure 7: torque calculation sketch

The red line represents the rotating plate on the side of the shoe that is connected to the electric motor. The length of the plate is set at 15 cm and the plate is attached at a height of 1.5 cm. When the plate touches the ground, it exerts a force (F_m) onto the ground. The vertical component of this force (denoted as $F_{m\perp}$) needs to be greater than the weight of the device in order to create a motion that is partially upward. The weight of the device (F_w) is calculated using the following formula:

$$F_w = m * g. \quad \textbf{(1)}$$

Here, m denotes the mass of the device and $g(= 9.81)$ is the gravitational constant. This results in a weight of $19.62\ N$. Hence, $F_{m\perp}$ should be at least $19.62\ N$. In Figure (7) α can be calculated as follows:

$$\alpha = sin^{-1}(\frac{1.5}{7.5}). \quad \textbf{(2)}$$

So, $\alpha = 11.54°$. With this information, F_m can be calculated using the following formula:

$$F_m = \frac{F_{m\perp}}{\cos(\alpha)}. \quad \textbf{(3)}$$

This results in $F_m = 20.02\ N$. The required torque about the center of the plate can now be calculated using the following equation:

$$M = r * F_m. \quad \textbf{(4)}$$

Here, M is the torque (moment) about the center of the rod and r denotes the radius of the plate. Equation 4 yields $M = 1.50\ N \cdot m = 15.29\ kgf \cdot cm$. The design uses three electric motors, and the torque is equally divided over all motors. Therefore, the minimum torque that each individual motor should have turns out to be $5.10\ kgf \cdot cm$. Figure (8) shows a top view of the placement of the three motors inside of the box. Initially, the idea was to use only two motors, but this could cause the device to become unstable. Therefore, it was decided to use three motors instead. Perhaps four motors would be even better but in order to save on weight and cost it was decided to use three.

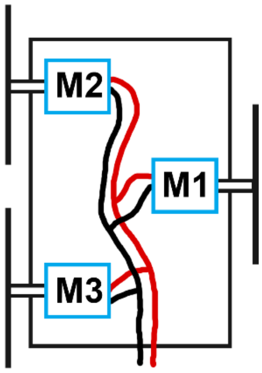

Figure 8: layout of motors in box

Rotating plates

In order to move the device on top of an object, a motion is required that has both a vertical and a horizontal component. To achieve this, rotating plates are connected to the electric motors and mounted on the side of the box. The design of these plates was a point of interest because they are quite a large part of the device and therefore contribute a lot to the total mass. So, one goal for the plates was to make them as light as possible.

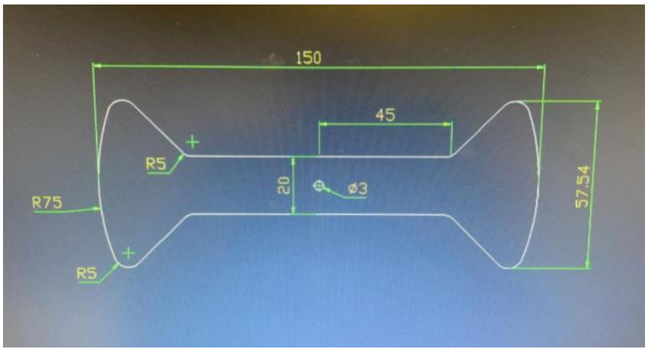

Figure 9: dimensions rods

However, they also need to be sufficiently strong in order to transfer the force from the motors onto the ground without bending or breaking. At first, the plan was to use steel plates as the base material for the rotating plates. Later, this was changed to aluminum because aluminum is about three times lighter than steel whilst still being relatively strong. The dimensions of the plates are also important with the length being the most relevant as this determines how far the box is lifted off of the ground and how far it is moved forward per rotation. Considering the height of common obstacles such as thresholds, the length of the plates was set at 15 cm. Next to this, both ends of the plates are rounded off to avoid them from cutting into the floor and to allow for a smooth motion. The final design is depicted in Figure (9), the plates ended up being cut out of a 3 mm aluminum plate with a laser cutter.

Wiring scheme and program

This final section of the design detailing will focus on the combination of electronics in the total system along with its code. The wiring schematic is shown in Figure (10). The main processing unit in the system is an Arduino Nano. This Arduino is ideal for this design because it is small and relatively cheap. The Arduino is connected to the three individual electric motors. The motors itself are powered by an external battery pack that contains four AA batteries. This was done because the Arduino cannot provide sufficient power to the motors. On the bottom right of Figure (10), two ultrasound sensors are depicted. These are connected to the Arduino and are used for the detection of the objects in front of the walker.

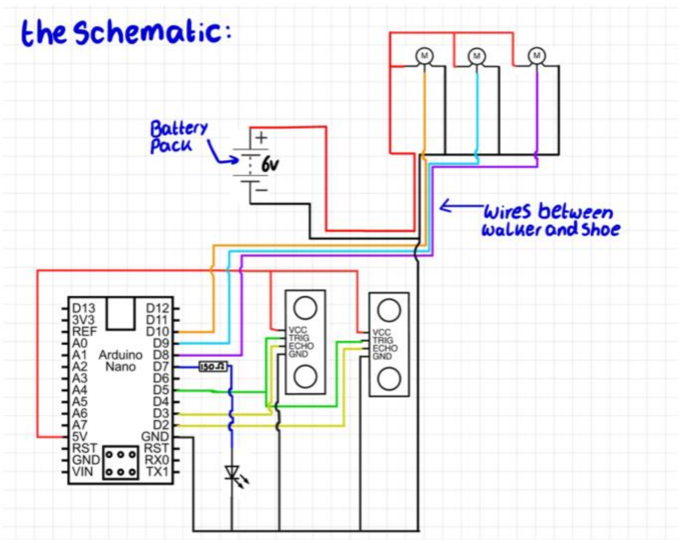

Figure 10: schematic electronic circuit

In Figure (11), an important part of the Arduino program is shown. This is a function that operates the electric motors under certain conditions. Outside of this function, the distance to the object is measured by the ultrasound sensors. At the start of the function that is depicted, it is checked whether or not there is an object in front of the feet of the person. If there is an object, a LED on the walker will turn on to notify the user that there is indeed an obstacle. After this notification, there is a delay before the motors are operated in order to give the user some time to get ready to be lifted.

```
void actuator() {
   if ((distance < 10) and (distance1 > 50)) {  // if object is detected -> execute if statement
  digitalWrite(LED_BUILTIN, HIGH);              // turn on led to notify user that object is detected

  delay(1000);                                  // time between led notification and operation of servos

  servo1.write(45);                             // set speeds of servos
  servo2.write(135);
  servo3.write(135);

  delay(1000);                                  // duration of rotation of the servos -> should match the speed of servo

  servo1.write(90);                             // stop moving the servos
  servo2.write(90);
  servo3.write(90);

  delay(5000);                                  // time before checking for another object
   }

   else{                                        // if no object is detected -> execute else statement
  servo1.write(90);                             // stop moving the servos
  servo2.write(90);
  servo3.write(90);
  digitalWrite(LED_BUILTIN, LOW);               // turn off the led
   }
}
```

Figure 11: Arduino code servo motors

8. Realization

In this following chapter here we elaborate on the assembling of the different parts as well as the assembling of the final product. The plan for production and the bill of materials will be discussed. Finally, the photographs of the final prototype can be observed.

Plan for production

The assembly of the box

The first thing that has been assembled was the box of the shoe. For this part there are six planes sawn from a 6 mm thick MDF plate. They are shaped so that the surfaces fit together, and the box is easily demountable for if any changes had to be made inside it. These shapes also make a sturdy surface. In figure 12 the planes as well as the assembled parts can be seen.

Figure 12: assembly of the box

Later in the process, all the planes of the box were glued down with wood glue and 4 nails were inserted on both sides except for the top part. This part is left open to make modifications of the motors possible.

Inside of the box two planes of Styrofoam have been placed, so that the motors are positioned higher and exactly aligned with the placement of the rods. To connect those rods to the box, holes have been drilled on the sides of the box. With a stapler the Velcro straps have been attached onto the sides of the box. To be able to distinguish the servo motors while coding they have been stickered with their names. The placement of the motors as well as the straps can be seen in figure 13.

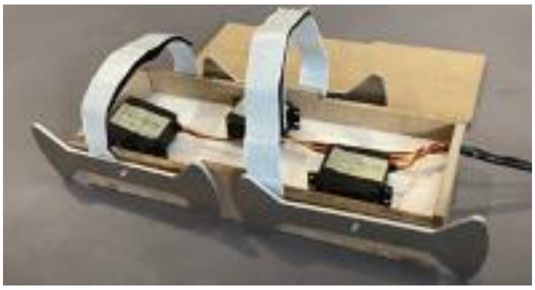

Figure 13: the box with lid off, rods, motors, Velcro straps and Styrofoam

The manufacturing techniques that have been used for the assembly of the box are sawing, drilling, gluing, nailing, stapling and cutting. All these actions were free of costs.

The assembly of the rods

The rods of the final product have been made of a 4 mm thick aluminum plate. The shape of the rods has been cut out by a laser cutter. In the following figures 15, and 9, the dimensions of the rod can be seen as well as the rods themselves. The manufacturing technique that has been used for the rods are laser cutting.

In the following figure 14, the assembly is visualized with an exploded view of the box and its connections.

Figure 14: rods connected to box

The assembly of the connection between the steel strip and distance sensors

For the distance sensors to be placed on the right spot, a steel stick is attached to the walker. The length of the steel stick can be adjusted by loosening the screw and tightening it at the length that must be maintained. In the following figures 16 this system is visible.

A small part of the stick has been sawn off to shorten it as the length of the stick was unnecessary long. Also, an MDF plate has been attached to the stick with super glue as well as a small layer of Styrofoam. Onto these

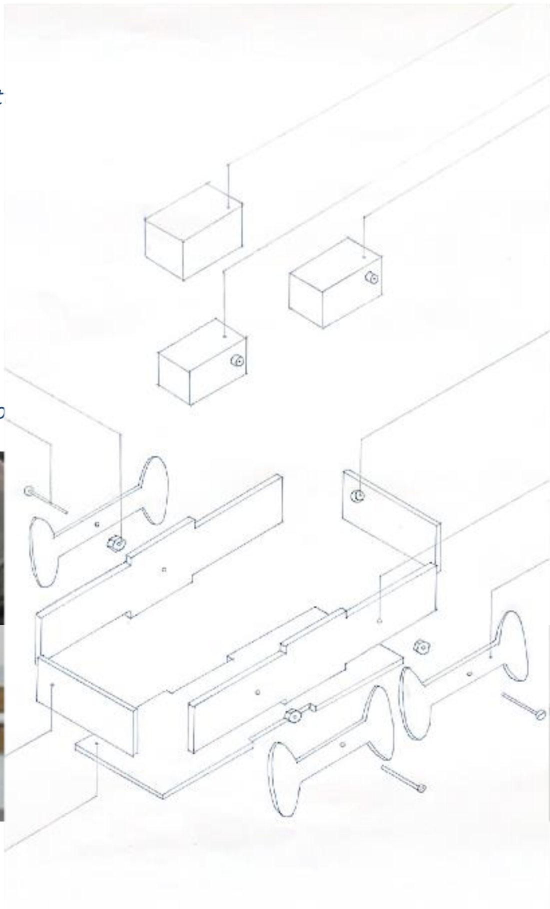

Figure16: sensors at

Figure 15: explo

layers the distance sensors are attached with super glue. The connection can be seen in the following figures 17.

Finally, the distance sensors have been marked with stickers according to their position on the plate and a hole has been drilled through the top part of the yellow mount of the steel strip to put a bolt through it. This way the strip can be connected to any walker as visible in figure 18.

Figure 18: sensor plate attached to a walker with screw trough yellow mount and sticker marking on sensors

The manufacturing techniques that have been used for this connection are sawing, gluing, cutting, nailing, drilling, sanding and polishing.

Blueprint of the steel strip and distance sensors

In figure 19 a drawing of the exploded view of the connection between the sensors and the stick is visible.

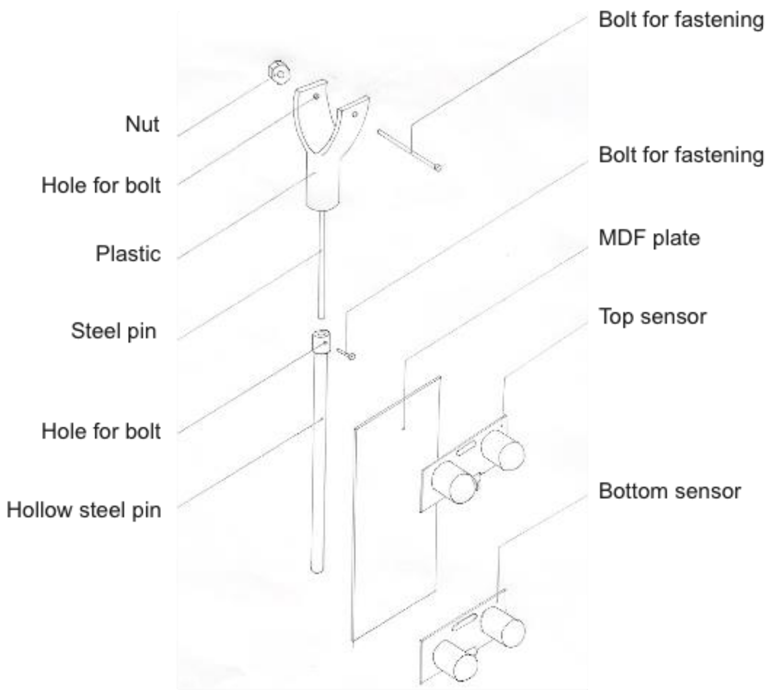

Figure 21: exploded view of connection between sensors and adjustable mount

Connection of the wires

The wires have been soldered onto a perfboard, connecting the led, sensors, the battery pack with batteries, servo motors, Arduino nano and its external power source to each other. In the following figures 21 and 22 is shown how these wires have been soldered onto the perfboard according to the schematic in Figure 10 visible in chapter 7. The soldering perfboard is made such that the wires of every component are close together, figure 20. All the wires are held together by heat shrinking tubes as visible in figure 24. The wire of the led has been made extra-long so it can be attached to the top part of the walker as seen in figure 23.

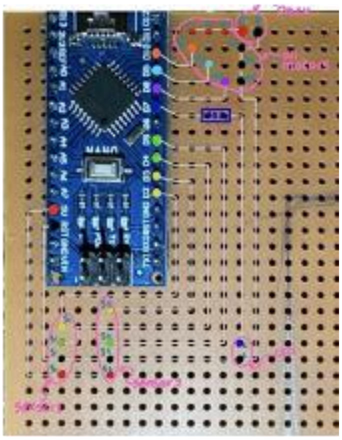

Figure 20: perfboard layout

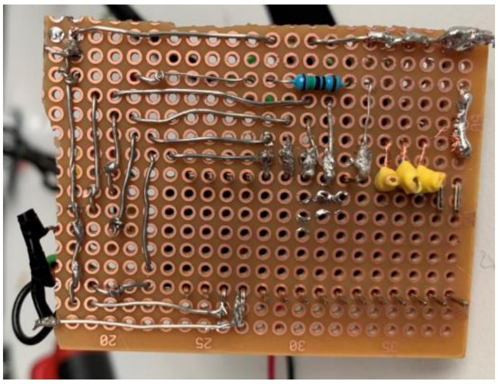

Figure 19: perfboard soldered

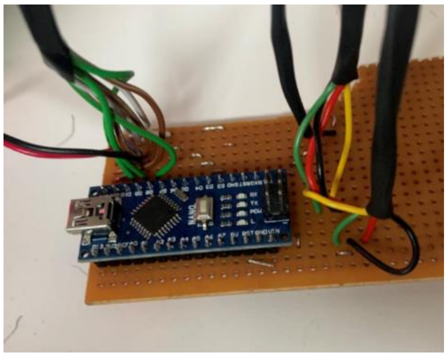

Figure 22: top view perfboard

Figure 24, heat shrinking tube around wires

Figure 23: led placement on walker

Bill of materials

Product	Quantity	Total price
MDF plate	2	1,00
Styrofoam	2	0,50
MG996R Servo motor – 10 kg continuous	3	21,00
Wires	2	7,00
HC-SR04 Distance sensor	2	6,00
Arduino nano	1	6,00
Usb to mini-USB cable	1	1,75
Steel stick	1	2,00
Aluminum rods	3	17,00
Led	1	0,20
Resistor for led	1	0,05
Velcro straps	1	1,50
Wires for led	1	0,80
Breadboard 400 points	1	3,00
Battery pack with loose cables	1	1,50
Total costs		69,30

Photographs of the material

In the following figures 25,26,27 the photographs of the final prototype can be seen.

Figure 25: box with shoe inside connected to walker with cord

Figure 26: attached sensor plate, perfboard and battery on walker

Figure 27: Box with shoe, horizontal rods and Velcro straps

9. Test plan and results (typically 2 pages)

This chapter includes the testing of the final system in detail. The results are compared with the functional and technical specifications of the prototype and are then set to be satisfied or not. The results are shown in the table below:

What is tested	Test method	Result	Satisfied
Test if all the components fit	Combine all the components and see if it fits accordingly.	All the components fit nicely, only the cable of the LED needed some extension and the sensor plate needed some adjustments to its shape.	Yes
See if the box can support a human of 80 kg	Keep adding more weight on the box until a weight of 80 kg in reached.	The box did not deform and held the 80 kg with comfort.	Yes
Distance sensors	Run the Arduino code of the distance sensors and see if an output in the serial monitor is shown.	The distance sensors are working, and an output is shown in the serial monitor.	Yes
Distance sensors accuracy	Run the code for the distance sensors and compare the displayed distances in the serial monitor with the real distance measured with measurement tape.	In smaller distances the sensor is accurate but in longer distances, such as 2 meters, it exceeds the 5 cm error limit.	Yes, since we do not measure further then 50 cm.
Motor	Run the code for the motors and see if they work.	The motors are working fine after some adjustments.	Yes

See if all the motors will start and stop at the same time	Create and run the code where all the motors will be rotating and then stopped at the same time.	The motors were working accordingly.	Yes
LED	Run the code for the LED and see if it can stay on for 5 seconds and then go off	At first the LED would not go off but with some adjustments to the code this problem was solved	Yes
Circuit	Test the connections made on the perfboard with the help of a multimeter measuring the voltage and current.	Everything was connected correctly except one thing, the ground was connected to the wrong Arduino pin, after changing this part everything was correctly connected.	Yes
Combine the codes	Upload the combined code and see if no errors are given.	Some pins did not line up since both codes used different pins, but after changing this everything was correctly without any errors	Yes
See if the sensor will sense an obstacle	Walk towards small obstacles with the walker and see if the LED goes on.	The sensor did not work all the time, so we changed the code form combined code 1 to combined code 2 (appendices), after this the sensors worked better but sometimes also went off when there was no obstacle in front.	No
Test if the sensors will sense a difference between and obstacle and wall from the plate on walker	Walk towards a wall and see if the LED will stay off.	the led will not turn on for a wall or high obstacle and will turn on for a low obstacle.	Yes
Test if the motors will start turning exactly when the LED goes off and 5 seconds after the LED went on how: watch when the motors start turning	Run the code and watch when the motors start turning.	the motors went on immediately after the sensor saw something, so we had to change the code, after changing the code form combined code 1 to combined code 2 (appendices) the motors started turning at the right time.	Yes
Test if the motors all start turning at the same time and	Run the code, change the code accordingly until everything is right	All motors started turning at the same time but the motor that was alone on one side	Not really

speed when in the box		could not keep up with the two motors on the other side, so we gave this motor more speed than the other two. This made the gap a bit less but after 2 hours of changing the code it was clear the motors would not run at the exact same speed. The chosen speed is close enough to make the product work. Because the rods do not hat the same speed the box will sadly not stay horizontally while lifting.	
Test if the motors and rods will stay in their place while rotating	Make the box lift, make it also lift with some weight on top.	When the top lid of the box is pushed down enough the motors will stay in place and the two rods that are together on one side will stay in place as well. However, the rod that is alone on one side turns in the direction that rotates the screw out of the motor. So, if there is too much weight on top of the box. This rod will rotate loose from the motor.	Not really
Test if the motors stop turning after 180 degrees when in the box	Put the weight of one shoe on top of the box, make it rotate and change the code while looking at the rotation of the rods.	After changing the delay in the code, a couple times, we found the right time to make the rods rotate 180 degrees. See delay in combined code 2	Yes
Test if the sensor plate does not touch the obstacle while walking over it	Walk over small obstacles, see if the plate touches the obstacle and change the position of the plate accordingly	the first time, the sensor plate touched the obstacle, but after moving the plate a bit towards the back, the plate did not touch the obstacle anymore. For higher obstacles then 2 centimeters the plate would still touch the obstacle though.	Yes
Test if you can still walk comfortably with the cord	Shuffle around on the ground with your foot on the box and see if the cord is in the way.	At the beginning the cord was too long, so you would walk over the cord	Yes

between walker and shoe		sometimes, but after shortening it, it was not in the way anymore.	
Test if the Velcro steps will secure the shoe in the box	put your shoe in the straps, start shuffling and moving your foot around and see if the box stays on.	When the straps are on tight enough, the shoe will be secure and not slip of the shoe box.	Yes

10. Design evaluation (typically 2 pages)

In conclusion, the end product works and is indeed able to assist people in need with getting over obstacles. There are still a lot of aspects in the design that could be improved upon. However, that is to be expected since this is just a first prototype and resources such as time and money were limited for this project. Looking back, the most important design step was the development of the rotating mechanism and its connection to the servo motors. This part is what basically brought the electrical and the mechanical side of the project together. The following section will go into several possible improvements for a second version of the device.

Wireless connection

In the current prototype, the connection between the motors in the box and the Arduino on the walker is hardwired. This is rather inconvenient because the wire could get in the way of the user and it could potentially lead to dangerous situations. A potential solution could be to use an additional Arduino inside of the box. The two Arduinos could then be set up to communicate with one another with an ethernet shield on top of the Arduino for example. The Arduino in the box could then be connected to the motors. In this way, there is no need for a wire that connects the box to the walker. Such a solution should be considered close to the start of the design process, when deciding which components to use.

Use of an additional motor

As of right now, there are three electric motors on the device. Two on the left side and one on the right. The decision to go for three motors instead of four was made because it would save on money and weight of the device. In the testing phase, it became evident that the side with only one motor had trouble keeping up with the other side. This resulted in an imbalance in the lateral direction and even with an adjusted code, this turned out to be difficult to fix. Therefore, the use of a fourth motor would probably be an improvement to the overall balance of the device. This idea should be implemented somewhere at the start of the design process, when determining the placement of the components on the device.

Use of stronger motors

The servo motors that are used help with lifting a person's leg, but they are not strong enough to lift the entire weight of the box plus a person's leg. The reason for this is the fact that stronger motors could not be bought within the budget of this project. To better work out this design and make it suitable for use in practice, stronger motors should be used that are able to lift a lot more weight. What type of motors should be used needs to be decided in the beginning of the design cycle as the dimensions of the box and the way the rods are attached depend on this.

Rubber profile on the rods

In the current product, the rods that are made of aluminum directly come in contact with the ground. Depending on the nature of the ground, the rods could lack the grip that is required to lift the box and the leg and slip. That is why a rubber profile should be added to the outer ends of the rods. With this profile the rods will be able to get a better grip on the ground which will prevent slipping. The addition of this rubber profiles could be done as soon as the rods have been made, but then it could even be done at the end of the design cycle when the whole product is finished.

Location of the sensors

The sensors in the current product are mounted to a plate that is in front of the wheels of the walker. This causes the sensors to be susceptible to damage as the obstacles in front of the walker will hit the sensors before the wheels can go over the obstacle. That is why the sensors should be attached somewhere behind the front of the wheels so the wheels will take the force of the incoming obstacle instead of the vulnerable sensors. The placement of the sensors should be decided upon in an early stage of the design cycle as the attaching mechanism of the demountable plate is location dependent.

Power supply in the box

In the current product, the power supply for the motors is attached on the walker. This means that a long cable must run from the box to the walker which will get in the way of the patient's shuffling. There is reason that this is done in this way is the lack of space in the box to fit in the battery pack. The better solution would be to increase the size of the box just so that the battery pack will fit. In combination with a wireless connection between the motors and the Arduino nano, the system would be wireless, and the box will not be restricted in its movement. The decision to put the power supply into the box should be made before the assembly phase in the design process. This is because the dimensions of the box depend on this and whether a hole for wires needs to be made in the box or not.

Different material for the box

For this project, the box was made of MDF wood. This could potentially be improved upon by using a different material such as aluminum. As of right now, the electric motors are not properly attached inside the box and that causes quite a bit of play in the rotating plates. Making a box out of aluminum instead would make it easier to mount the motors. Another advantage of using aluminum would be that it does not get destroyed as badly as MDF when it gets wet. Opting for a different material for the box would need to be done early in the design process because it could potentially change the way that components can be mounted.

11. References

APA style in alphabetical order

4x AA-batterijhouder met Losse Draden. 4x AA Batterij houder met losse draden - 4xAALEADSHOL. (n.d.). Retrieved November 1, 2021, from https://www.tinytronics.nl/shop/nl/power/batterijhouders-en-clips/aa/4x-aa-batterij-houder-met-losse-draden.

Braak, H., & Braak, E. (2000). Pathoanatomy of Parkinson's disease. *Journal of Neurology, 247*(S2), II3–II10. https://doi.org/10.1007/pl00007758

Breadboard 400 points. BB400P. (n.d.). Retrieved November 1, 2021, from https://www.tinytronics.nl/shop/nl/gereedschap-en-montage/prototyping-toebehoren/breadboards/breadboard-400-points.

Jonasson, S. B., Nilsson, M. H., Ledell, J., & Carlsson, G. (2018). Experiences of fear of falling in persons with Parkinson's disease – a qualitative study. *BMC Geriatrics, 18*(1). https://doi.org/10.1186/s12877-018-0735-1

MG996R servo - 10kg - continuous. MG996R-CONT. (n.d.). Retrieved November 1, 2021, from https://www.tinytronics.nl/shop/nl/mechanica-en-actuatoren/motoren/servomotoren/mg996r-servo-10kg-continuous.

Nano v3.0 - compatible. ARDNANOCH340. (n.d.). Retrieved November 1, 2021, from https://www.tinytronics.nl/shop/nl/development-boards/microcontroller-boards/arduino-compatible/nano-v3.0-compatible.

Ultrasonische sensor. HC-SR04. (n.d.). Retrieved November 1, 2021, from https://www.tinytronics.nl/shop/nl/sensoren/afstand/ultrasonische-sensor-hc-sr04.

Nimwegen, M., Speelman, A. D., Hofman-van Rossum, E. J. M., Overeem, S., Deeg, D. J. H., Borm, G. F., Horst, M. H. L., Bloem, B. R., & Munneke, M. (2011). Physical inactivity in Parkinson's disease. *Journal of Neurology, 258*(12), 2214–2221. https://doi.org/10.1007/s00415-011-6097-7

Swerdlow, R. H., Parks, J. K., Miller, S. W., Davis, R. E., Tuttle, J. B., Trimmer, P. A., Sheehan, J. P., Bennett, J. P., & Parker, W. D. (1996). Origin and functional consequences of the complex I defect in Parkinson's disease. *Annals of Neurology,* 40(4), 663–671. https://doi.org/10.1002/ana.410400417

12. Appendices

Appendix 1: Project planning

At the start of the project, a planning for the project was made. Figure (28) shows the division of the roles during the meetings. Figure (29) shows a rough planning for the design process over the weeks.

Date/role	Chairman	Minute taker	Board writer
9/10/2021	-	Amina	Bente
9/14/2021	Amina	Bente	Mark
9/17/2021	Bente	Mark	Stan
9/21/2021	Mark	Stan	Jyri
9/24/2021	Stan	Jyri	Lars
10/1/2021	Jyri	Lars	Amina
10/5/2021	Lars	Amina	Bente
10/8/2021	Amina	Bente	Mark
10/12/2021	Bente	Mark	Stan
10/19/2021	Mark	Stan	Jyri
10/22/2021	Stan	Jyri	Lars
10/26/2021	Jyri	Lars	Amina

Figure 28: role division

Figure29: rough planning

Week	Planning
1	- Introduction - Proposal of ideas
2	- Review more worked out ideas and settle on a final idea to use - Review RPC's - Come up with functional and technical specifications - Assign subtasks to group members
3	- Make a conceptualized version or sketch of the device - Prepare the intermediate presentation
4	- 28/09/2021: Intermediate presentation - Make a detailed design of the device - Make a list of all the required parts and labor to make the device - Order all necessary parts - 01/10/2021: Midterm GME
5	- Construct the device
6	- Continue building the device - Start writing the report - Start working on the final presentation
7	- Finish building the device - Test the device - Finish the final presentation
8	- 26/10/2021: Final GME - 29/10/2021: Final presentation - Finish writing the report
9	- 01/11/2021: Report

Appendix 2: Presentation videos

Intermediate presentation video: [Video 1]

Final presentation video: [Video 2]

Appendix 3: combined code 1

The first code that was proven incorrect during testing.

```
#include <Servo.h> //include servo motor library
 Servo servo1; // create servo objects to control the servo
 Servo servo2; // create second servo object
 Servo servo3; // create third servo object

const int trigPin = 4; //define pin number trigger sensor 1
const int echoPin = 2; // difine pin number echo sensor 1
const int trigPin1 = 5; //difine pin number trigger sensor 2
const int echoPin1 = 3; //define pin number echo sensor 2
const int ledpin = 7; // define pin number LED

// defines the sensors variables
long duration;
long duration1;
int distance;
int distance1;

void setup() { //read this once
  //set input and output pins
  pinMode(trigPin, OUTPUT);  // difine pin as output pin
  pinMode(echoPin, INPUT);   // define pin as input pin
  pinMode(trigPin1, OUTPUT); // define pin as output pin
  pinMode(echoPin1, INPUT);  // define pin as input pin
  pinMode(ledpin, OUTPUT);   // define pin as output pin
  Serial.begin(9600);
  servo1.attach(8);  // define pin servo 1 is attached to
  servo2.attach(9);  // define pin, servo 2 is attached to
  servo3.attach(10); // define pin servo 3 is attached to
}

void loop() { // read this continuously
  distancecalc();    // read void distancecalc
  distancecalc1();   // read void distancecalc1
  distanceprint();   // read void distanceprint
  led();             // read void led
  motor();           // read void motor
}

void distancecalc() {
//calculate the distance for bottom sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration =  pulseIn(echoPin, HIGH);  //define long duration
  distance  = (duration*0.034/2)/0.96; //define int distance with calculation to get distance in centimeters
}

  void distancecalc1() {
// calculate the distance for top sensor
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
```

```
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH); //define long duration1
  distance1 = (duration1*0.034/2)/0.96; //denfine int distance1 with calculation to get distance in centimeters
}

void distanceprint() {  //write measured distances in centimeters in serialmonitor
  Serial.println("distance bottom = "); // write the words: distance bottom
  Serial.println(distance); //write the measured and calculated distance of bottom sensor
  Serial.println("distance top = "); // write the words: distance top
  Serial.println(distance1); //write the measured and calculated distance of top sensor
}


void led() {
  if ((distance < 10) and (distance1 > 50)) { //if the sesors sense an obstacle
  digitalWrite(ledpin, HIGH); //set the LED on

  delayMicroseconds(1000);    //for 1 second
  }
  else {                      // if sensors do not sense a abstacle
  digitalWrite(ledpin, LOW);  // set the LED off

  }
}
void motor() {
   if ((distance < 10) and (distance1 > 50)) { //if the sensors sense an obstacle
    delayMicroseconds(5000); //wait five seconds
                            //servo's.write(0)= full speed clockwis,
                             //(90)=stand still,
                             //(180)=full speed anticlockwise
  servo1.write(0);           //rotate servo 1 full speed clockwise
  servo2.write(120);         //rotate servo 2 1/3 of full speed anticlockwise
  servo3.write(120);         //rotate servo 3  1/3 of full speed anticlockwise
  delayMicroseconds(2000);   //rotate for two seconds
   }
   else{                     //if sensors do not sense an obstacle
  servo1.write(90);          //do not rotate servo 1
  servo2.write(90);          //do not rotate servo 2
  servo3.write(90);          //do not rotate servo 3
  delayMicroseconds(2000);   //do not rotate for 2 seconds
   }
}
```

Appendix 4: combined code 2

This is the final code used in our end product

```
#include <Servo.h> //include servo motor library

 Servo servo1; // create servo objects to control the servo
 Servo servo2; // create second servo object
 Servo servo3; // create third servo object

const int trigPin = 4; //define pin number trigger sensor 1
const int echoPin = 2; // difine pin number echo sensor 1
const int trigPin1 = 5; //difine pin number trigger sensor 2
const int echoPin1 = 3; //define pin number echo sensor 2
const int ledpin = 7; // define pin number LED

//difines the sensors variables
long duration;
long duration1;
int distance;
int distance1;

void setup() { //read this once
  //set input and output pins
  pinMode(trigPin, OUTPUT);  // difine pin as output pin
  pinMode(echoPin, INPUT);   // define pin as input pin
  pinMode(trigPin1, OUTPUT); // define pin as output pin
  pinMode(echoPin1, INPUT);  // define pin as input pin
  pinMode(ledpin, OUTPUT);   // define pin as output pin
  Serial.begin(9600);
  servo1.attach(8);  // define pin servo 1 is attached to
  servo2.attach(9);  // define pin, servo 2 is attached to
  servo3.attach(10); // define pin servo 3 is attached to

}

void distancecalc() {
//calculate the distance for bottom sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration =  pulseIn(echoPin, HIGH);  //define long duration
  distance  = (duration*0.034/2)/0.96; //define int distance with calculation to get distance in centimeters
}

  void distancecalc1() {
// calculate the distance for top sensor
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH); //define long duration1
  distance1 = (duration1*0.034/2)/0.96; //denfine int distance1 with calculation to get distance in centimeters
}



void distanceprint() {
  Serial.println(distance);  //print the bottom sensors measured distance in cm
  Serial.println(distance1); //print the top sensors measured distance in cm
```

```
}

void actuator() {
   if ((distance < 10) and (distance1 > 50)) {  // if the sesors sense an obstacle
  digitalWrite(ledpin, HIGH);         // turn on led to notify user that object is detected

  delay(5000);                        // time between led notification and operation of motors
  digitalWrite(ledpin,LOW);           // turn led off
  servo1.write(0);                    // set servo 1 to full speed clockwise
  servo2.write(140);                  // set servo 2 to 5/9th of full speed anticlockwise
  servo3.write(140);                  // set servo 3 to 5/9th of full speed anticlockwise

  delay(1200);                        // duration of rotation of the servos to turn 180 degrees

  servo1.write(90);                   // stop moving the servos
  servo2.write(90);
  servo3.write(90);

  delay(5000);                        // time before checking for another object again
   }

   else{                              // if no object is detected
  servo1.write(90);                   // stop moving the servos
  servo2.write(90);
  servo3.write(90);
  digitalWrite(ledpin, LOW);          // turn off the led
   }
}
```