Universidad de Valladolid

ESCUELA DE INGENIERÍAS INDUSTRIALES

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería Mecánica

3D PRINTED ROBOT

Rodríguez Olivera, Carlos

Mª Isabel Sánchez Báscones

NOVIA University of Applied Sciences

Valladolid, Junio de 2018.

TFG REALIZADO EN PROGRAMA DE INTERCAMBIO

TÍTULO: OPEN-SOURCE PROJECT 3D PRINTED LIFE SIZE ROBOT

ALUMNO: Carlos Rodríguez Olivera

FECHA: 25/05/2018

CENTRO: NOVIA University of Applied Sciences

TUTOR: Camilla Moliis

RESUMEN

El presente documento describe el Trabajo de Final de Carrera realizado en un "European Project Semester" (EPS) en Vaasa, Finlandia.

El objetivo de este proyecto fue desarrollar y programar un robot de tamaño humano impreso en 3D, obteniendo así una réplica del proyecto de código abierto a nivel mundial InMoov. InMoov es un proyecto de colaboración entre universidades y de particulares en todo el mundo. A través de su página Web, se han obtenido los modelos para impresión 3D, y la guía de programación e instalaciones electrónicas. Se ha completado la construcción del torso inferior del robot, así como la programación de las tareas solicitadas para este proyecto como: presentación interactiva del Robot InMoov y sus funciones, demostración de interacción con personas mediante juegos y preguntas a través de cámaras de realidad virtual (Kinect) y programación de autoaprendizaje de gestos mediante tecnología Leap Motion.

PALABRAS CLAVE

Impresora 3D

InMoov

Robot humanoide

MyRobotLab

European Project Semester

NOVIA

OPEN-SOURCE PROJECT 3D PRINTED LIFE SIZE ROBOT

R3-DFin

Authors
Federico LORENZONI
Patricia HAHLBOHM
Carlos RODRÍQUEZ
Matthijs SOUILLJEE

Supervisor
Rayko TOSHEV

European Project Semester
Spring semester 2018
Vaasa, Finland
05.02.2018 - 25.05.2018

May 18, 2018

Acknowledgements

Acknowledgements to Novia for this wonderful opportunity and special thanks to Technobothnia for supplying all the resources that were needed. There are various people we want to thank, especially:

- Roger Nylund: for providing us with the knowledge of managing our project
- Rayko Toshev: for supervising this project
- Sulaymon Tajudeen: for providing technical assistance when needed (worked on the electronics of InMoov before the EPS group arrived and therefore had sufficient knowledge of the project)
- Hanna Latva: for teaching us the knowledge of English and academic writing

Besides these people, we also want to thank our fellow EPS students for an amazing time - during the working hours and our free time.

Thank you for the beautiful time in Vaasa!

Abstract

Combining printing and robotics was what this project was about. By using an 3D-printer such as the Ultimaker a robot was almost completely printed and ready to work. It was programmed to make him preform the tasks that where needed.

The robot was printed based on the OpenSource project that was called InMoov. The main goal was basically to build a humanoid robot by using the Ultimaker 3 Extended and Makerbot Replicator. By adding other technologies such as the Kinect and its camera system, VR, a tablet to control & interact and the Leap Motion to teach humanoid gestures. The robot will act as similar as possible to a human. In the end he should be able to support the people for example with teaching or researching.

The goal of this project has been achieved. Next to the main goal, also a couple of secondary where set such as VR InMoov, InMoov on Omron and beamer support. These goals where not achieved, because of a lack of time. By now complete functionally is realised and InMoov is running a script, that lets him represent technobothnia.

The future work on InMoov would be optimisation and improve his mobility by adding a secondary robot for movement or making the legs motorised. In addition a 12 volt battery would make him completely independent from wall power sockets and better wire management would, also benefit the aesthetics of the robot. Also, the script can be modified for other useful functions and can make him more versatile in use. To conclude, the main goal of this project has been achieved.

Abbreviations

3D	Three Dimensional
ABS	Acrylonitrile Butadiene Styrene
AC	Alternating current
AI	Artificial Intelligence
AUX	Auxiliary
COM	Communication
DC	Direct current
EPS	European Project Semester
FPS	Frames Per Second
HR	Human Ressources
IDE	Integrated development environment
IR	Infared
LED	Light Emission Device
PIR	Pyroelectric Infrared
PLA	Polylactic acid
PVA	Polyvinyl Alcohol
PM	Project Management
RACI	Responsible,Accountable,Consult,Informed
RGB	Red-Green-Blue
USB	Universal Serial Bus
VR	Virtual Reality
WBS	Work Breakdown Structure
Wi-Fi	Wireless Fidelity (artificial name)

Contents

List of Figures

List of Tables

1. Introduction

1.1. European Project Semester (EPS)

The first EPS project took part in 1995 and was organised by Dr. Arvid Anderson in Denmark. More than 20 years later, the European Project semester (EPS) is provided by 19 universities all over Europe. It was mainly organised for engineering students to practise their skills for their future work in companies and in multinational teams. Every student, who has finished at least two years of study, is encouraged to apply for the EPS. During the EPS, the students will be divided in multinational teams, which leads automatically to English as the spoken and written language. The focus lays on the teamwork. Besides that, the EPS covers many different fields of skills, which can be seen below [Jorgen Hansen, 2017].

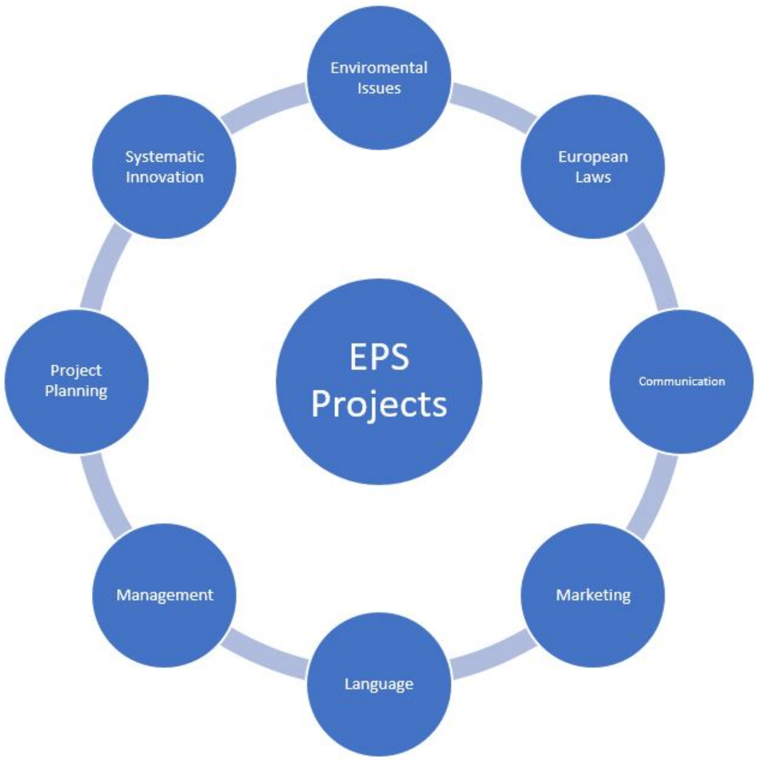


Figure 1.1: European project semester (EPS)

By being part of the EPS, the students have to grow together as a team to solve the various engineering problems. Students improve their English skills and work (mostly) on subjects they have not dealt with before. The support throughout the entire project is huge. Next to the project work, students have courses in the local language to keep also in touch with the real life in the country every day. They are followed by a supervisor, often a professor of the hosting university, that guarantees their perfect orientation, the achievement of the project and a complete improvement of the competences that are important for the jobs they will perform in the close future.

1.2. Introduction (InMoov)

The InMoov project arose by an idea of the French sculptor and designer Gaël Langevin. Although he has more than 25 years' experience in his job, he started this project without any knowledge of robotics. He was tasked to design and build a prosthetic arm.

Figure 1.2: Gaël Langevin [1]

The build of the prosthetic arm did not succeed, but he kept on going to build a robot (2012) with the support of a worldwide community as the Open Source project InMoov. It is the bridge between the world of the 3D-printing enthusiasts and students, in engineering fields. It also offers the possibility to apply knowledge of students that usually have to study theoretical subjects. This is the reason, why "it is conceived as a development platform for Universities, Laboratories, Hobbyist, but first of all for Makers". At the same time, it is feasible with any 3D-printer which has basic capabilities and a printing area of just 12x12x12 cm (l x w x h), and it leads to the fulfilment of a complete humanoid robot. As every open source project, it is based on sharing with an online community and it opens the opportunity to anyone to reproduce the robot [Gaël Langevin, 2012].

At Novia, the project started as a free time project of Rayko Toshev and various other students. Now, after three and half years, the project is beginning to take shape.

This is also thanks to different groups and individuals, who worked on the InMoov project.

As a humanoid robotic project with various different technical components, InMoov is a real challenge. To give an indication how the building process of an InMoov robot is preformed, a broad building description is given. This manual will describe the building process in big lines.

The starting point for creating an InMoov robot is the manufacturing of the different parts. These parts can be taken of the STL library on the website (www.inmoov.fr). All the different parts are manufactured using an additive manufacturing method (3D printing). The material that is used for the manufacturing is ABS plastic. After manufacturing all the different parts, they can be assembled together. The same website gives detailed instructions how to assemble the parts and also an inventory of the different nuts, bolts, washers and rods that are needed for the assembly.

Next to completing the robot on a physical level with the manufacturing and assembling of all the different parts, it is also, of great importance, that the electronics of the robot are functioning correctly. These electronics consist of a couple of different components.

The first components are two Arduino AtMega's 2560. One of the Arduinos will be responsible for controlling the right arm and the right hand of the robot. And the other one will be responsible for controlling the left arm, left hand, the head and the torso/stomach. The Arduinos are also equipped with nervoboards, which allow the builder and the user to have a better overview of the functionality to control large amount of servos. These nervoboards are supplied by the official webstore of InMoov. Another electronic part of the electronics of InMoov are the servo actuators, which consist of a large range of different types of servos. These servos are used for every movement of InMoov.

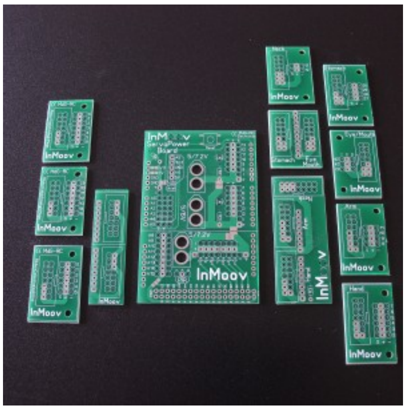


Figure 1.3: Nervoboards [2]

The last action to complete an InMoov project is to control InMoov. This can be done with a tablet, that can be mounted on the back. This can run its own special InMoov software. The software that is used for InMoov is called Myrobotlab. This software is especially made compatible with InMoov and has a large range of options. In this environment our InMoov will be programmed. All these steps in the building process will be explained in greater detail in the rest of the report. Next to these major components various other minor components are added for increasing the functionality of the robot.

Figure 1.4: Myrobotlab Logo [3]

Community

To successfully fulfil an open source project the knowledge from the community is of great importance. The InMoov community has a couple of ways to get information. These consist of three main websites/forums that all have their own purpose. The main sources to find information on are summed up below.

- http://inmoov.fr/ [Gaël Langevin, 2012] : This website gives all the general information that is need to be know to build your own InMoov and also has a small forum that handles some basic questions. The biggest inconvenience with it is that some parts are in the French language. But for general information this is the most common website.

- http://myrobotlab.org/ [Myrobotlab, 2011] : By using this website access to information over the software (Myrobotlab). This consists of a forum that goes into the deep questions of the use of Myrobotlab and editing the program itself. For programming InMoov this is the most informative website. Besides the general information and the forum, there is also a shout box that gives the opportunity to ask small questions with a rapid response.

- https://groups.google.com/forum/#!forum/inmoov [Google Group, 2013] : This is the biggest place to ask questions. The questions, that are asked, range from the whole building to the development process of InMoov. For more complex questions this is the website with the most information, because the size of the forum and the availability of information cover the whole building process.

Besides these opportunities to get the necessary support and help the community with more expertise, there is also the possibility to just Google a question. Most of the time it will lead to one of these websites, but sometimes helpful information can be found on third party websites. Including sites such as YouTube (video instructions on different subjects) or Thingiverse[1].

[1] modified parts to improve the capabilities of your InMoov

1.3. Starting point

In chapter 1.2 a broad building plan of an InMoov project is given. Now, it is important to know the starting point of this specific InMoov project. In the case of this project Rayko Toshev, Sulaymon Tajudeen, Aki Viitala, Thomas Höglund, Gerard Escribà and Alexandru Galben have been working on InMoov for the past three and half years as a free time and school project, which means that the robot from the waist up is already physically built and the electronics are connected. This part of the robot is not yet fully completed. There are still problems with the servos (e.g. jittering of servos and missing servos) and some parts still need to be printed or reprinted.

Now the starting condition of every part of InMoov will be discussed one by one to give a complete view of the starting point.

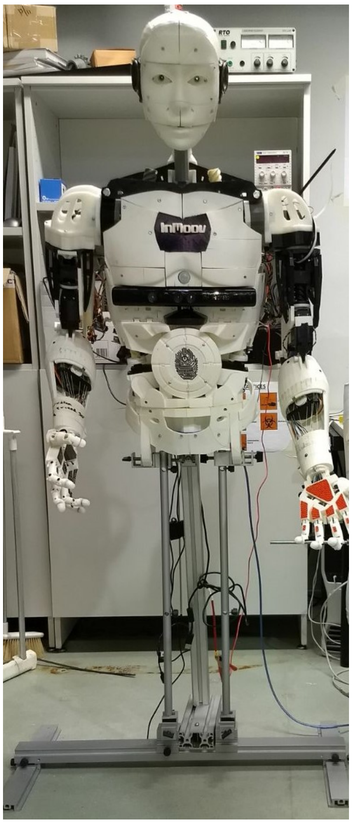


Figure 1.5: InMoov starting point

To give the starting situation in a concrete matter, the whole robot will be discussed from top to bottom. The different parts consist of the head, the hands, the torso, the legs. These parts will be discussed separately. The tasks, that needs to be performed on the different parts consist of problem solving (e.g. servo jittering and floating current), the replacement & reparations of a wide variety of different servos and the build of missing part. Also for every part of the robot a situation will be given. These situations consist of:

1. Build: This means that this part of InMoov still needs to be completely build
2. Complete rebuild: This means, that this part of InMoov is already build, but needs to be taken apart completely to make it work as expected
3. Partially rebuild: This means that this part of InMoov is already built and just smaller parts need to be rebuilt to get it working as expected

Head

The head of InMoov needs to be completely disassembled. The reason therefore are the not-working and missing of servos. The problems with the servos occur as not functioning electronically and mechanically. Next to the servos, the speakers still need to be fitted with an amplifier and need to be wired correctly, so that both play the sound simultaneously. In figure 1.6 the current status (including problems and tasks) of the head can be seen:

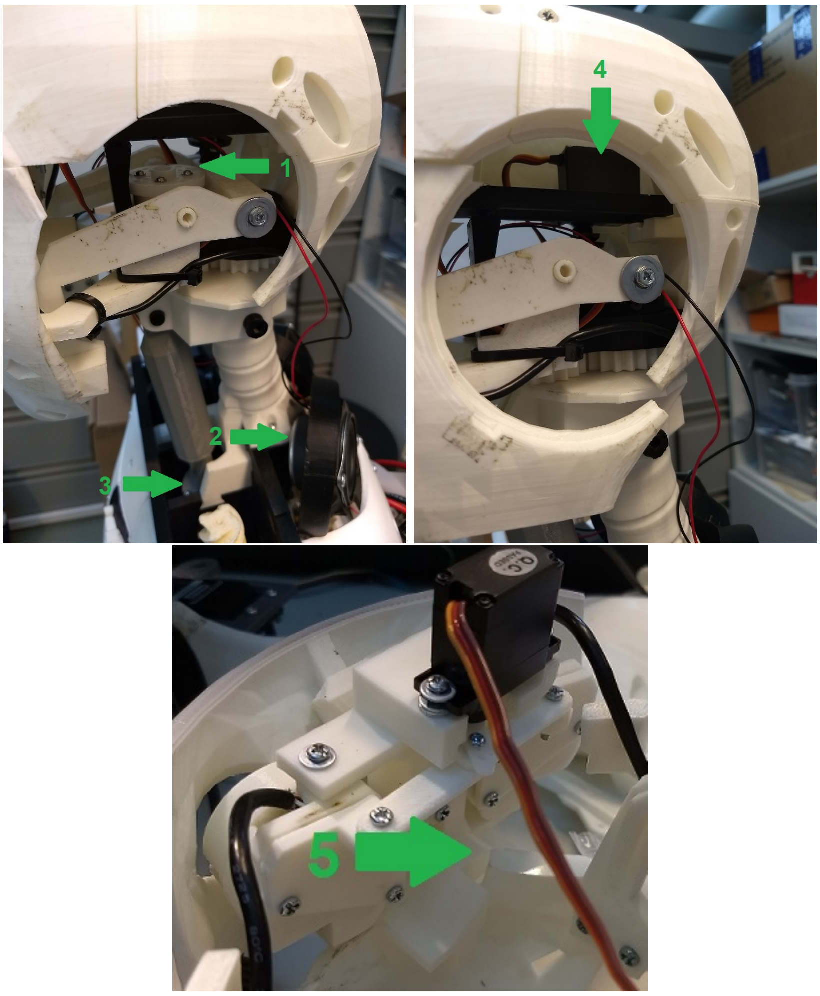


Figure 1.6: Starting point of the head: **1**: Missing servo for rotation; **2**: amplification of loudspeakers & soldering; **3**: servo of the neck installed wrongly (potentiometer taken out); **4**: jittering servo for the jaw; **5**: missing servo for y-movement of the eyes

The missing servos for the head rotation (1) and eye-y-movement (5) will be installed and the broken servos will be replaced by new ones (jaw(4) and neck(3)). After the complete rebuild of the head, will be tested and re-installed on the torso of InMoov.

Arms

The arms of InMoov need to be partially rebuilt. This will be done to replace jittering and broken servos and fix mechanical problems. The major task for the arms will be the tensioning of the fishing braids that control the fingers. Also, both wrists are not functioning and need to be fixed to use them. In the figure below the current status of the arms can be seen.

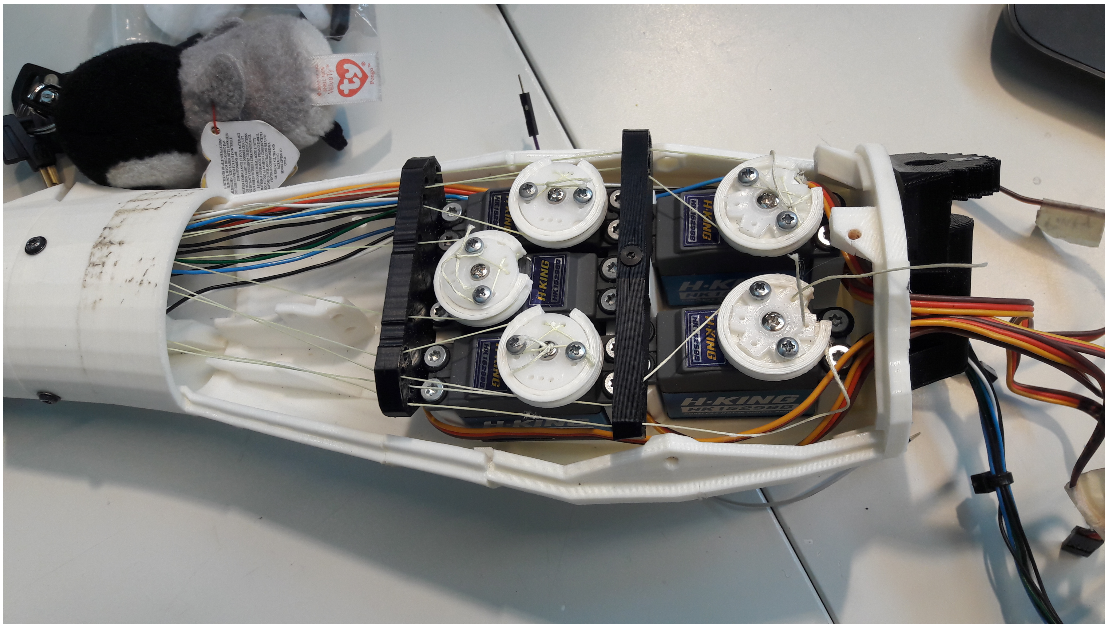


Figure 1.7: Arm starting point

After every adjustment on the arms, the complete arm will be tested and re-installed on the torso.

Torso

The upper torso and mid torso will be partially rebuilt with the exception of the lower stomach, that still needs to be built. The work on the upper torso will consist of replacing, fixing and testing of various parts. besides that, the cables need to be managed for aesthetic reasons. Since the robot is going to be completely disassembled, all cables need to be reconnected correctly in the final assembly. Next to the cables, a tablet needs to be chosen, bought and installed (and probably adjusted) to run the software of InMoov and to communicate with the arduinos. The tasks of the mid torso are the testing of one hacked pair of servos and repairing other parts if necessary. For the lower stomach a build is needed, because the two servos need to be hacked and installed. This process will take up a large amount of time. The current torso can be seen in the figure below.

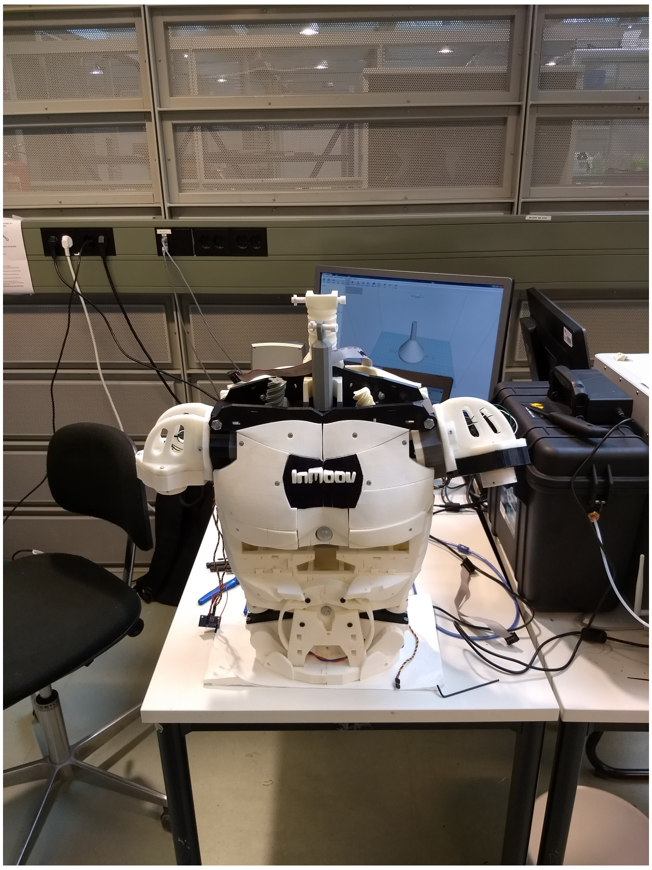


Figure 1.8: Torso starting point

Legs

The legs need to be built, various parts for the legs are already printed. The task for the EPS group is to print the missing parts as well as to assemble the legs and finally developing a setup how to move the robot securely.

Software

The software for InMoov is currently only installed and configured. The configuration will be preformed again, because of certain problems with the software and also in the end the software will run on the tablet. For now, the only way the software is used till now is to test individual servos and is never configured to run multiple servos.

The integration of the existing code and configuration of the software will be preformed. The custom code for the specific user case will be written. This custom code includes extra desired voice and gesture commands. The custom code takes a significant amount of time to develop and debug.

1.4. Project motivation

There are two inquiries to be answered at this point: "why are we here" and "why did we choose this project".

Vaasa is known as a great international city, gathering students from all over the world, in a country offering students a high-level education program. It is a city where it is possible to find out and improve your potential as engineers and co-workers. Meanwhile it is an enormous chance to improve one's level of English.

Once we got here, three different projects were offered, among them the InMoov Project. The choice was made to participate in the InMoov project, because of the technical challenging nature of the project. Also, the open source nature of the project.

The 3D printing is moving ahead at a staggering speed, since 1976 with the inkjet 3D printers to this day, being able to print all sorts of complex items like clothing, buildings, automobiles, food and in the near future, vital human organs. This joined with the robotics world, another fascinating technology which is aiding the humankind in all aspects of human life, providing us many chances in our future career as engineers.

1.5. The team

The project team consists of a multi-disciplinary and multi-cultural team. Members of the team are an industrial and management engineer, medical engineer, mechatronic engineer and mechanical engineer. Everybody of them will provide his or her own competences useful for the attainment of the main goal. The team will be supported by Professor Rayko Toshev, of the University of Vaasa.

Federico Lorenzoni, Ancona, Italy (President)

Student of Management Engineering at Università Politecnica delle Marche, Italy

"I consider myself a person with really good adaptation in environments where the main things are communication, team spirit and collaboration with each other for the realisation of goals. I think to have really good organising skills, and I was coordinator during the last years of many groups in different fields, firstly entertainment and charity events of great importance. My principal interest is to develop my competences in project management, and EPS is for sure the best way for the attainment of my goal."

Patricia Hahlbohm, Berlin, Germany (Secretary)

Applied Physics / Medical Engineering, Beuth University of Applied Sciences, Berlin

"Although my studies at home do not completely fit into the EPS, there are still some matching points in the courses such as programming and basically the entire physical part. But I am open and very curious about new fields, in which I can improve myself every day. Being able to learn together with people from all over Europe, it increases the motivation even more. I hope that in the end of the EPS, I will have learned more about industrial working and meanwhile grown together as a multinational team of Engineering students."

Matthijs Souilljee, Enschede, Netherlands

Mechatronic student at Saxion University of Applied Sciences, Enschede

“I am an ambitious third year Mechatronics student, who wants to develop himself besides the normal education program by studying abroad and living in a different culture. My special interest is in programming and embedded technology. Because of my background in electrical, mechanical and software engineering I hope to combine the different disciplines in the EPS semester. Besides my study I enjoy social activities, sports and working on my own projects. I am always willing to take on new challenges and get out of my own comfort zone to develop myself as a person and improve my knowledge on different technical and non-technical subjects.”

Carlos Rodríguez, Valladolid, Spain

Vocational trained in Development of Software Applications and student of Mechanical Engineering at Escuela de Ingenierías Industriales de Valladolid

"I have always loved to challenge myself and discover what I am able to overcome. That is why I have chosen an Engineering, to develop my own skills and knowledge inside of this type of ever-changing world. I like everything related to mechanics, programming and design. I have a great visual perception, creativity and ability with technology, as well as social competence and team leadership. Coming to Finland to join the EPS is the best way to prove my talents and start learning how the professional life works, within and outside of a company".

1.5.1. Belbin test

Another advantage of this energetic team consists of the different type of character of each member. Based on Belbin Questionnaire Results, strengths and weaknesses of everybody of us are enough balanced.

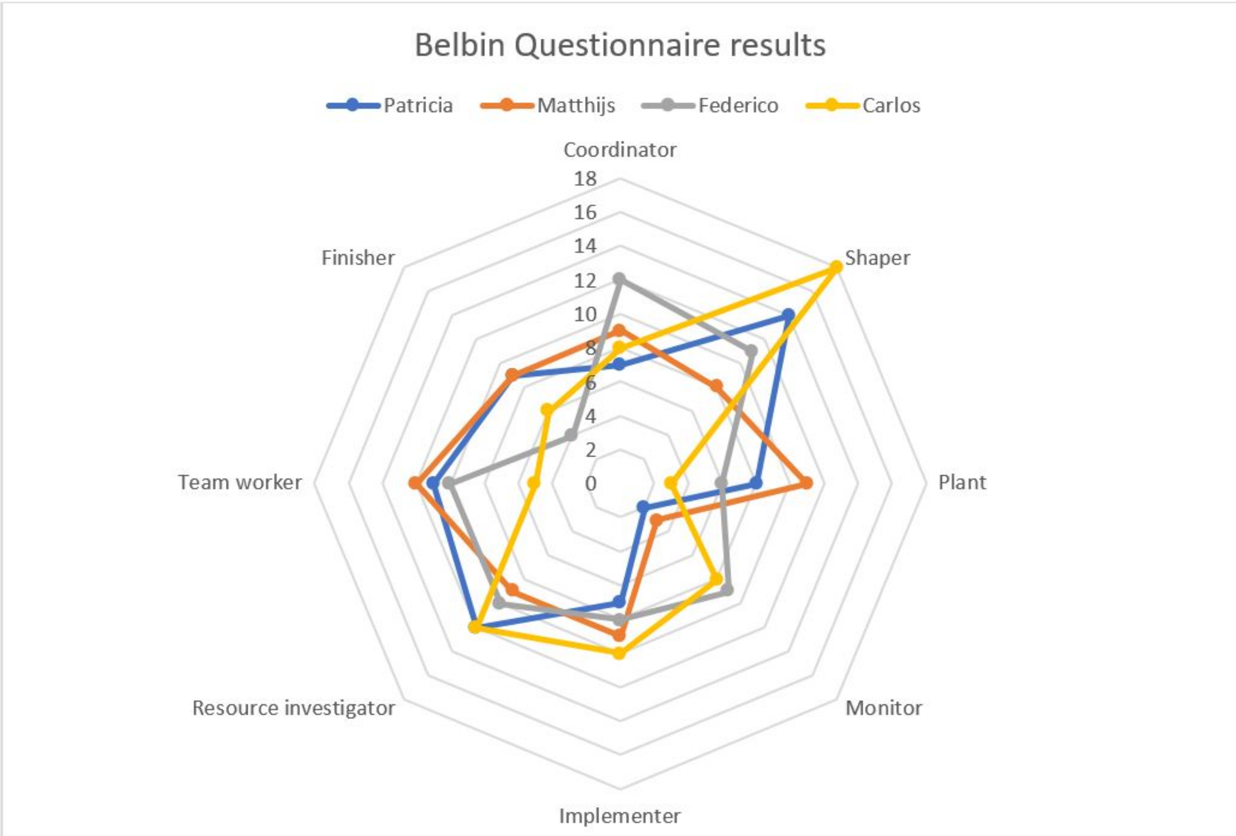


Figure 1.9: Belbin test results

A Belbin test takes information about the inclinations in behaviour and ideas of every member of a team, helping to find strong point and connected weaknesses. Every member of the group has been submitted to the Belbin test before the start of project working. The results draw attention to the tendency for each one. Federico especially has coordinating and monitoring skills, Matthijs stands out for his team worker and plant aspects, Patricia proves to be a really good resource investigator and ideas shaper, as well as Carlos. Generally, it is easy to get that the whole group has particular inclinations on shapers and resource investigation, and that the weak point

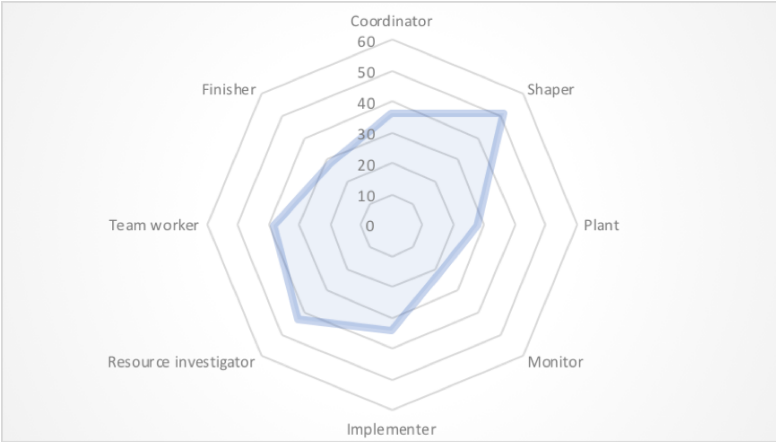


Figure 1.10: Total sum of Belbin test results

might be the lack of strong skills in finishing and monitoring. Otherwise the other characteristics look sufficiently balanced.

1.6. Brand Identity

The work that is being done throughout this EPS is closely related to the already existing project called InMoov. The basic idea about the logo was to pick the main object up as to print a robot in a common-sized 3D-printer.

R3-DFin

Figure 1.11: Logo of the EPS project

The name above is chosen because it combines the R-3D which stands for 3D printing, F which stands for Finland and as last in which stands for InMoov. This way all major components of the project are combined together.

1.7. Project target

The target for this project is defined as:

"**To complete InMoov and get it to an operational level, that it is ready for use in representing certain facilities (for example technobothnia). This will be done before the 22nd of May.**"

In order to reach the target stated above, smaller targets will help to fulfil the larger target. This project is no exception. The smaller targets are summed up below:

- To produce the still needed parts for InMoov and assemble these on the frame.
- Complete the electronics: this will consist of troubleshooting, (re)soldering, testing and configuring.
- To use the available written software to get the different functions of InMoov up and running. These functions consist of gesture support, leap motion support, voice commands, etc.
- Mounting and configuring of the back control tablet.

2. Research

InMoov is not the only humanoid robot in the world. These class of robots have all different goals, but all have in common that they boost the advancement of the humanoid robots. These type of robots all ways are the middle point of the discussion if robots should look like humans. The standard industrial robots do not look like real humans they reassemble more a arm. In the figure below a industrial robot can be found next to a humanoid robot.

Figure 2.1: Industrial robot [4]

Figure 2.2: Humanoid robot [5]

From a glance at both robots, can directly be seen what the discussion is about. Different people and companies focus towards making a robot look like a real human as close as possible. Others focus on making the movements as natural as possible. The goal of every humanoid robot can differ greatly but, the main goal of every humanoid robot can be defined to be:

represent a human as close as possible by looks, movements and behaviour

To give a better understanding of which humanoid robots are out there three examples will be given.

2.1. Sophia

Sophia is a humanoid robot that tries to reassemble a human face to her best abilities in many different ways. Sophia is created by Hanson Robotics, a Hong Kong-based company. She was activated for the first time on the 19th of April and made her first public appearance in Texas, USA during a festival on March 2016 [Harriet Taylor, 16/03/2016]. Sophia is able to do fifty different facial expressions.

The creators of Sophia are not backing down to give their robot some public appearances. So, Sophia is probably the best known robot by people around the world. She has been in many high profile interviews and also is the first robot to receive a citizenship. This citizenship was given by the country of Saudi Arabia and granted to her in October of 2017. In November 2017 Sophia booked another title for herself. She was named the United Nations Development Program first ever Innovation Champion, and the first non-human to be given any United Nations title. [Oscar Raymundo, 2016]

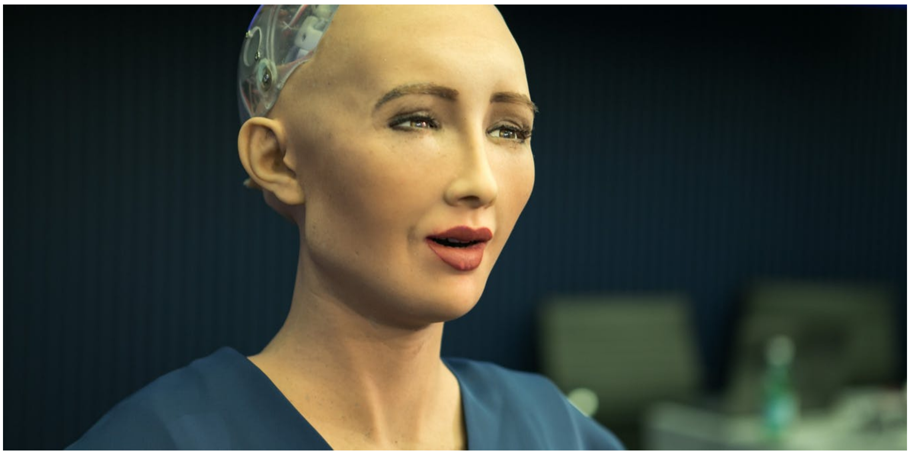


Figure 2.3: Sophia [6]

To conclude, Sophia is one of the humanoid robots with a big resume and got exceptional public attention. She is also a driving force in the discussion if robots can have a citizenship and can be treated like humans.

2.2. Atlas

Atlas is a humanoid robot which does not have the same goal as for example Sophia, but is a great innovation in its on class. Atlas is more focused on hard movements, like jumping.

Atlas has an height of 1.5 meter, weighs 75kg, can carry 11 kg, is powered by a battery, uses hydraulics for its actuation, has 28 joints and uses LiDAR & Stereo Vision to see the world around him. All these things make Atlas a one of a kind robot. The creators of Atlas (Boston Dynamics a spin-off from the Massachusetts Institute of Technology) said this over there robot:

"Atlas is the latest in a line of advanced humanoid robots we are developing. Atlas' control system coordinates motions of the arms, torso and legs to achieve whole-body mobile manipulation, greatly expanding its reach and work space. Atlas' ability to balance while performing tasks allows it to work in a large volume while occupying only a small footprint.

The Atlas hardware takes advantage of 3D printing to save weight and space, resulting in a remarkable compact robot with high strength-to-weight ratio and a dramatically large workspace. Stereo vision, range sensing and other sensors give Atlas the ability to manipulate objects in its environment and to travel on rough terrain. Atlas keeps its balance when jostled or pushed and can get up if it tips over". [Boston Dynamics, 2018]

Figure 2.4: Atlas by Boston Dynamics [7]

2.3. ASIMO

ASIMO (Advanced Step in Innovative Mobility) is a humanoid robot created in 2000 by Honda. The development started in the 1980. This included various prototypes before they got to the final product. The main goal for creating ASIMO was to create a walking robot. Currently ASIMO is located in Miraikan museum in the Japanese capital of Tokyo as a display item. Out of the museum ASIMO made various public appearances all over the world.

ASIMO has a height of 130 centimetres and weighs 54 kilograms. Honda performed research to determine the optimal height for a mobility assistant robot. This research concluded that the optimal height was between 1.20 meters and the average adult, because of the operation of door knobs and light switches for example. ASIMO has also the main purpose to assist people during there daily tasks, or even to take daily tasks completely away from the people [Jason Ford, 22/11/2000], [Alok Jha, 2004].

Figure 2.5: ASIMO by Honda [8]

ASIMO has various abilities like:

1. Recognise moving objects
2. Posture
3. Gestures
4. Interaction with surrounding environment
5. Interaction with humans

The technology of ASIMO already impacted further development by pushing the research towards walking assistant. This research resulted in Stride Management Assist and the Body-weight Support Assist [Jason Ford, 22/11/2000].

3. Theory

3.1. Hardware

3.1.1. Arduino

Arduino was brought to life as an open source electronic platform. The first microcontroller board was invented in 2005 and since today, several microcontroller devices have been built and produced to control all types of different actions. Most of the different types of boards consist of an Atmel 8-bit AVR microcontroller [Wikipedia, 2018]. By choosing a type, the buyer meanwhile chose the amount of pins on the device. There are models for beginners (ARDUINO Uno) and some including enhanced features, such as the ARDUINO Mega (that is used for InMoov). The Arduino microcontroller was implemented "at the Ivrea Interaction Design Institute as an easy tool for fast prototyping, aimed at students without a background in electronics and programming" [Arduino Website, 2018]. The use of such tools became so popular, because it is an easy way to control everything that needs to be controlled technically. This could simply be LED's or even up to complex electrical circuits including servos or actuators. To work with the microcontroller, one needs the Arduino programming platform that consists of a so-called IDE, which means an integrated development environment. Further information on an IDE could be found here:
https://en.wikipedia.org/wiki/Integrated_development_environment.

The main surface uses two different parts, that are called **void setup ()**. In this everything that needs to be set up only once is written (e.g. declaring the used pins as OUTPUT. Or INPUT- pins). The **void loop()** contains the program, that will run for infinity. Below, there is an example program that if uploaded to the board, makes an LED blink if it is connected to Pin 10.

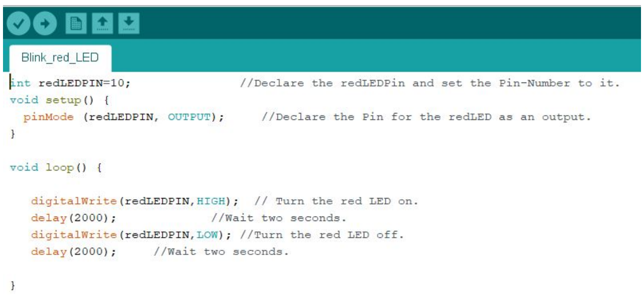


Figure 3.1: short program for a blinking LED

3.1.2. Arduinos for InMoov

The robot InMoov uses two ARDUINO Mega 2560 micro controllers to control the implemented servos for the movement of the robot. An example board can be seen below.

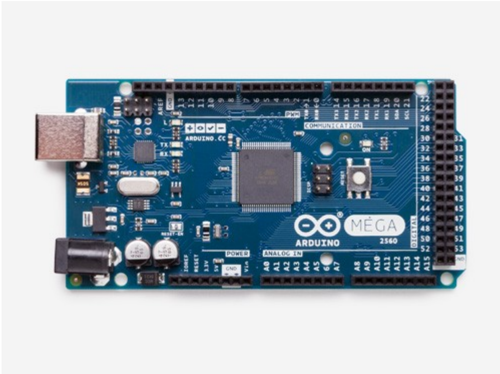


Figure 3.2: short program for a blinking LED [9]

The Arduino Mega board has 54 digital I/O-Ports and 16 analog Inputs and of course a serial interface for a usb connection. On the website, this device is the "recommended board for 3D printer and robotics projects" [Arduino Website, 2018]. The devices for the InMoov are specialized, which means that for a better overview, the related parts are hooked up together before connected to the board. One Arduino is the responsible controller for the right part, the other microcontroller for the left part, eyes, stomach and PIR-sensor. The mentioned connections and reunifications are shown in the following picture.

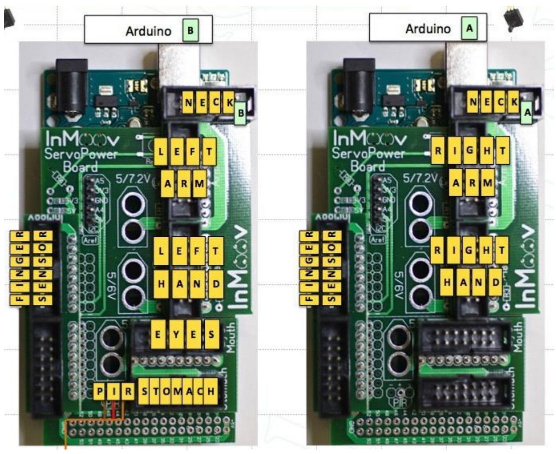


Figure 3.3: Micro controller for InMoov [10]

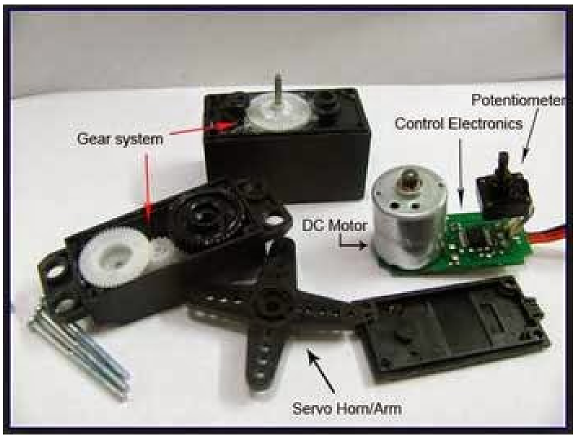
In order to be able to offer the pre-programmed software the servos in every single robot should be attached to the same pins as instructed. To achieve this goal, the inventor of InMoov uploaded a hardware map, in which the connection from servo to the attached pin is available.

	A	B	C	D	E	F	G	H
1	Group	Part Name	Arduino Uno	Arduino Mega	Pin Num	Position Rest	Position Min	Position Max
2								
3		thumb	X Right	X Left	2			
4		index	X Right	X Left	3			
5		majeure	X Right	X Left	4			
6	hand	ringFinger	X Right	X Left	5			
7		pinky	X Right	X Left	6			
8		wrist	X Right	X Left	7	90		
9								
10		bicep	X Right	X Left	8	0		90
11	arm	rotate	X Right	X Left	9	90	40	
12		shoulder	X Right	X Left	10	30		
13		omoplate	X Right	X Left	11	10	10	80
14								
15	head	neck		X	12	90	20	160
16		rothead		X	13	90	30	150
17								
18	mouth	jaw		X	26	10	10	25
19								
20	eyeX	eyeX		X	22	80	60	100
21	eyeY	eyeY		X	24	90	50	100
22								
23		topstom		X	27	90	60	120
24	stom	midstom		X	28	90	60	120
25		lowstom		X	29	?	?	?
26		led/neoR		X	30	?	?	?
27								

Figure 3.4: Micro controller for InMoov [11]

3.1.3. Servos

Servo motors are not an engine type. Rather, they serve as a control or drive unit in a closed circuit. They are used in model kits such as radio-controlled aircraft's. They are also used in large industrial robots, where they guarantee the precise positioning of, for example, robotic arm. All over the world there are providers for servomotors. But therefore, the prices could vary hugely. There are cheap versions available (approximately 18€). By buying these versions, there might be a slightly dependency in the limits which means that the horns on top of the motors cannot move the entire 180° as it is usually promised. Not only the limits vary but also the power of the engine depends on the price of course. Different servos can be for example some of mW or even up to 400kW. Servomotors can be used as DC or AC motors. Usually, when they are used with an AC power supply the forces of the motors are higher. A servo consists of five main parts, which are the motor (AC or DC), a potentiometer, a system of gears, the servo horn and the electrical control unit. The pictures below show the inner life of a servo motor (left) and an assembled motor that is implemented in the hand of InMoov.



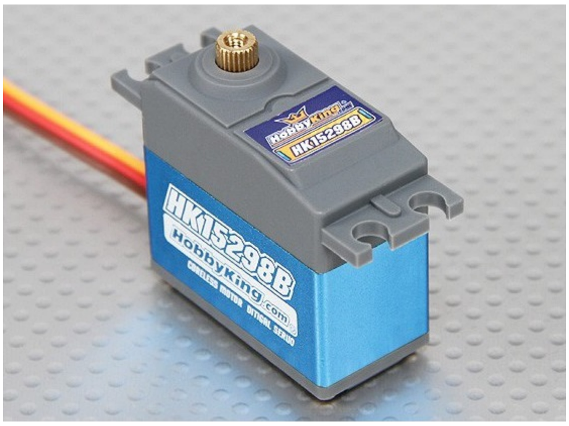


Figure 3.5: Elements of a servomotor (left) [12] and assembled servomotor (right) [13]

A comparison of the two most common types of servos (compared by their level of rotation of the horn) can be seen in the grid below.

Table 3.1: Types of servos [Howard Eglowstein, 2012]

	Types of servos	
category	*positional rotating*	*continous rotating*
frequency	most common	usually
limits	0 - 90 (or 180)°	360°
control signal	static positional point	direction (e.g. clockwise), speed
use	toys, radio controlled cars and aircrafts,...	as drive motor on a mobile robot

But, how does a servo work? The principle that is used to move a servo is rather simple. The horn of the servomotor is in a certain position. Since the potentiometer is directly connected to the gearbox, there is a voltage in the potentiometer for each position of the horn. If an electrical signal is sent mostly from a small microcontroller, it is transformed into an analog signal(Digital to Analog-converter (DAC)). The length of that incoming signal decides in which angle the servo of the horn is going to move to. This process is called pulse width modulation. An example is given in the picture below. Usually the microcontroller operates in a frequency of 50 Hz, which leads to an actual period of 20 ms.

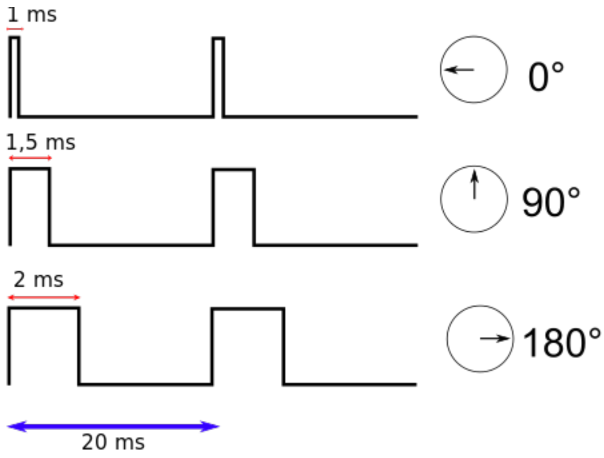


Figure 3.6: Pulse width modulation [14]

An electrical comparison element such as a comparator, compare the analog signal (reference input signal, see figure 3.7) with the signal that comes from the potentiometer (reference output signal, see figure 3.7). If there is a difference, the output of the comparator will be unequal to zero and will be amplified. This amplified signal will rotate the servo until the required position is reached. The described electrical circuit with its component are shown as followed:

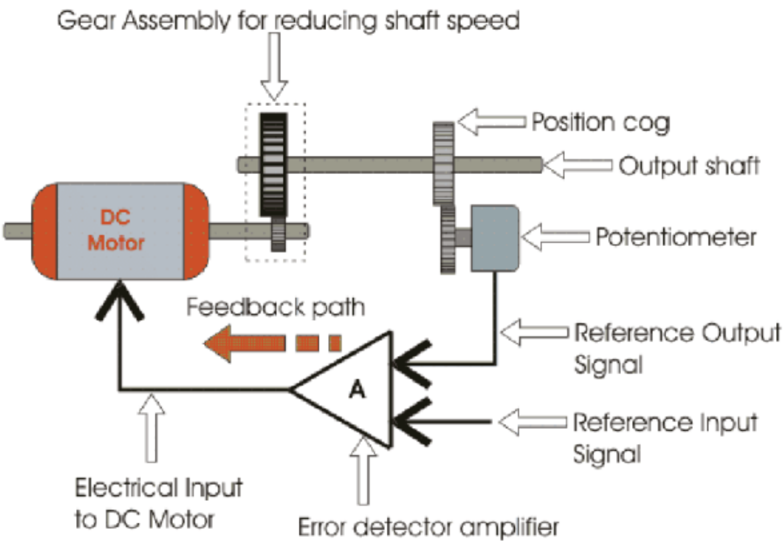


Figure 3.7: Working principle of a servo [15]

As described, the servo in combination with the control unit (microcontroller) are the main components, that move the part of the robot. They are mandatory as the robot needs to copy humanoid gestures. Therefore, many servos are integrated in the robot. In the following grid (Table 3.2) the number of servos in each part is listed based on the website of InMoov [Gaël Langevin, 2012].

In order to make the entire robot moveable, there are many servos mounted in InMoovs body.

Table 3.2: Amount of servos used in InMoov

Part	**Number of Servos**
hand & forearm	12
arms & shoulders	8
head & torso	5
stomach	4

Since the robot is not able to walk, no servos are needed in the legs. By now, not every servo is working the way it should. Electrical and mechanical problems, such as the limits or the zero-position (managed by a external potentiometer) of the servos need to be solved as the next step of work until the end of the project.

3.1.4. Kinect

The Kinect device is a product that maps the area in front of the sensor. It was introduced into the market in November 2010, provided by the company Microsoft. It is mainly used in the Xbox 360 game console where the user interacts in the game without a controller [Wikipedia, 2018].

Figure 3.8: Kinect device from the Xbox 360 (used in InMoov) [19]

As already mentioned, the Kinect is able to create a map of everything that is located in front of the sensors. In that way, it can construct 3D- objects. In order to do that, the Kinect consists of three sensors, four microphones and a motor. These elements are integrated into the device as followed:

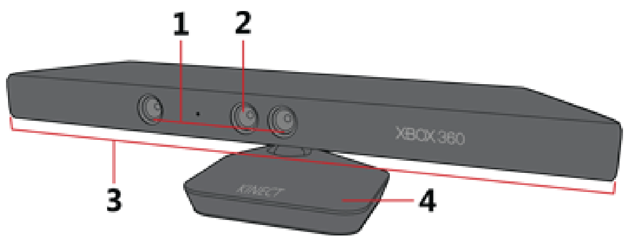


Figure 3.9: Integrated hardware including 1- IR transmitter and camera, 2 – RGB camera, 3 – 4 microphones, 4 - motor) [20]

The secret behind its ability is the IR transmitter. With this light, a cloud of infrared dots are layered over the scenery in front of the Kinect. The reflected dots from an object can be seen with the IR camera which is located on the right side of the device. Due to the calibration of the Kinect, it knows the distance of these object and with that information, it generates the depth image. The closer the object is to the sensor, the brighter it gets. White parts indicate that the distance could not be calculated from the Kinect.

Figure 3.10: Depth image of a person made by the Kinect [21]

Furthermore, there is a RGB-camera integrated in the Kinect, which is attached in the same calibrated distance as the other two sensors. It takes a coloured picture and merges it with the depth image. This leads to its skill to do a 3D – scan.
Next, there are four microphones located around it. These can capture sounds and locate the source of the noise as well at the same time. It is a unique technique with which multiple players or noise sounds can be separated from each other.
The last part is a small motor on the foot of the device which is the reason for its ability to tilt the camera up and down for about 30° [Greg Borenstein, 2012].

All of these sensors included in the Kinect make it interesting and important that the device is a part of InMoov. It can register objects in front of the body of the robot and realize its surroundings. Even movements of people can be detected and in the best case copied by the robot.

3.2. Software

The software that will be used to operate InMoov consists of different languages:

- Java
- Arduino
- Python

Java will be running in the background and will work in the background. The end user of Myrobotlab has not that much interaction with the Java code. The Arduino code will be supplied by the InMoov website. This code will deal the translation from the Python code to the Arduino. This code can be edited, but a working program is given. So if the given program is used the focus can be put on the Python program. Python will be the coding language that is used to program InMoov. Because of the large amount of available function in python by Myrobotlab, there will be a large emphasis on the already existing functions to control InMoov. These functions are available for every part of InMoov ranging from voice commands to leap motion control. A summary of the different parts that are already integrated in Myrobotlab can be seen in Appendix A. In addition to the functions of InMoov the flowchart for Myrobotlab is available in Appendix B

An example for the integration of a function into the program is:

i01.moveHand("left",0,0,0,0,0)

In this example a couple of different things can be seen. First of all i01 (the object) is a reference to an object made in Python. The definition of an object is: "in computer science, an object can be a variable, a data structure, a function, or a method, and as such, is a location in memory having a value and referenced by an identifier".

So, in this case i01 has a couple of different functions that can be accessed inside this object. The function that is called in this example is the function "moveHand".

This function also gives the possibility to attach different arguments to it these arguments define which hand (left or right) has to be moved and to which angle each servo of the hand has to be moved. This can be seen as: **moveHand ("which hand left or right","angle of the thumb","angle of the index finger","angle of the middle finger","angle of the ring finger","angle of the pink","angle of the wrist")**. Using this way each part of the robot can be moved and also various things such as gestures can be implemented.

Next to a large amount of functions that Myrobotlab possess. There are also example programs that give all the information that is needed to write your own program. These example programs go into different parts of InMoov for example one of the programs goes into how to move the hand of InMoov this example goes through commanding the hand with the use of voice commands. The most important example program is the Full program. This program has all the different functionalities of InMoov in it and is a good starting point for writing your own program with the functionalities that are wanted from InMoov.

3.2.1. Myrobotlab

The mode that InMoov can be programmed consists of three different modes, which change the way InMoov is controlled. These environments are:

- Virtual mode
- Rightside
- Full

To give a good idea on what these modes are, the interface of Myrobotlab will firstly be explained. This will give an insight in how to different environments work.

Interface

The interface of InMoov has various different tabs which can be accessed and that have their own functionality. In the figure below the different tabs can be seen. Not every of them has an interface, so do not have any functionality.

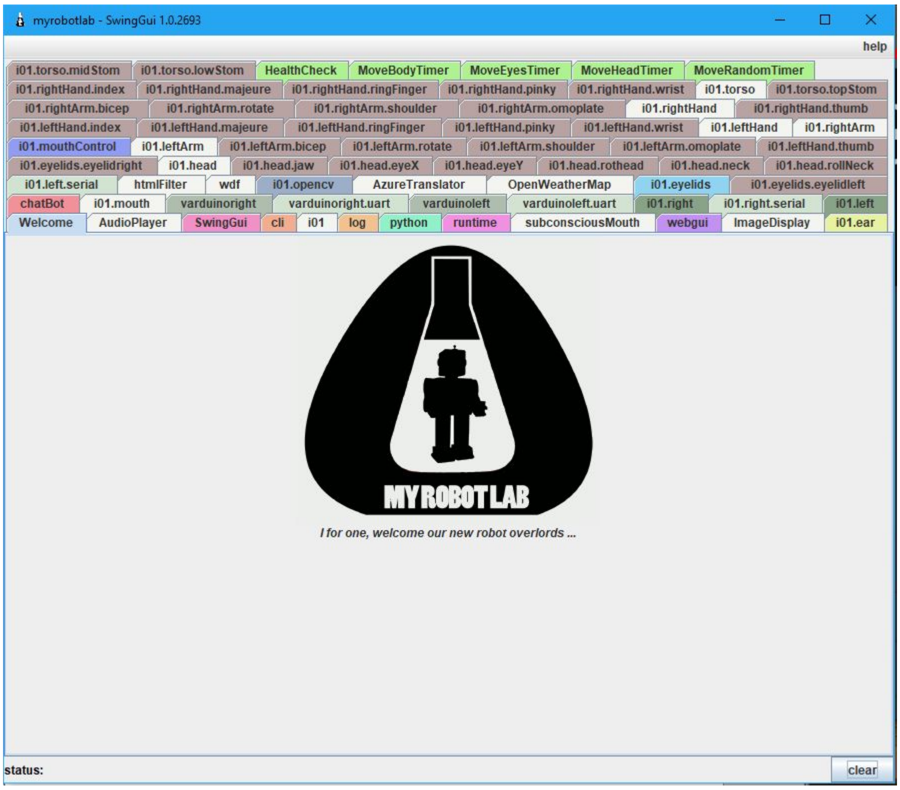


Figure 3.11: Intro screen Myrobotlab

In the picture 3.12 below, an empty Python tab is shown, the most important one. With the execution button, the written code will be compiled and executed. It is an important window where most of the time will be spent to develop and modify the entire program.

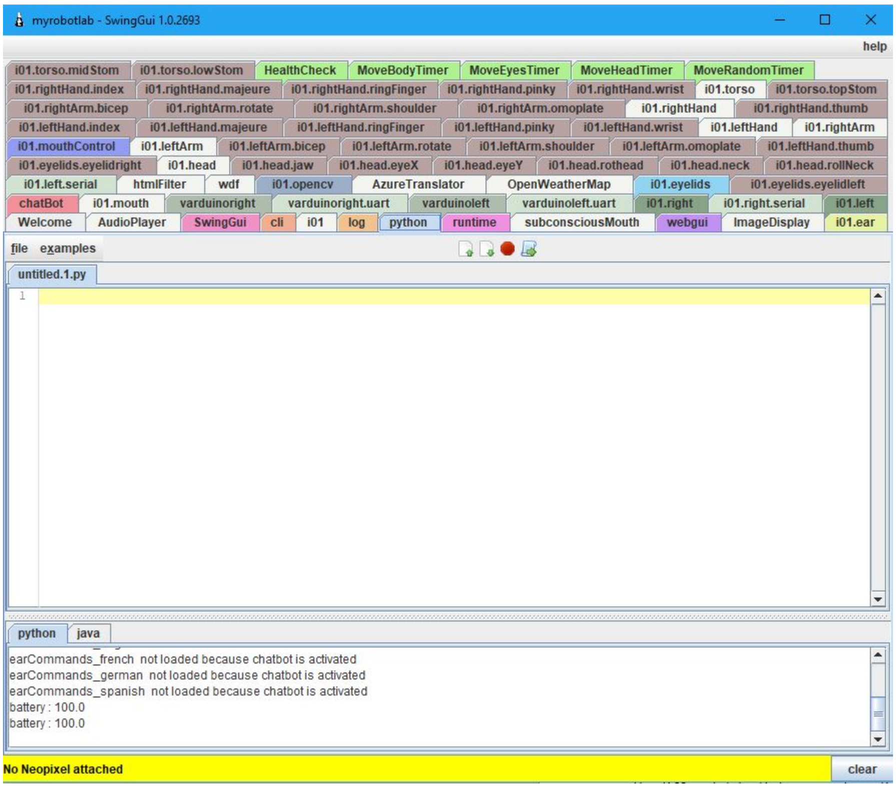


Figure 3.12: Python screen Myrobotlab

Beside the python surface there are several windows which are also handy to use. For example, there is the possibility to test every servo manually by controlling its angle with a slider. The velocity can be set to any desired level. On default this is always set to -1 this means the servos run on full speed. In the picture below this tab is shown:

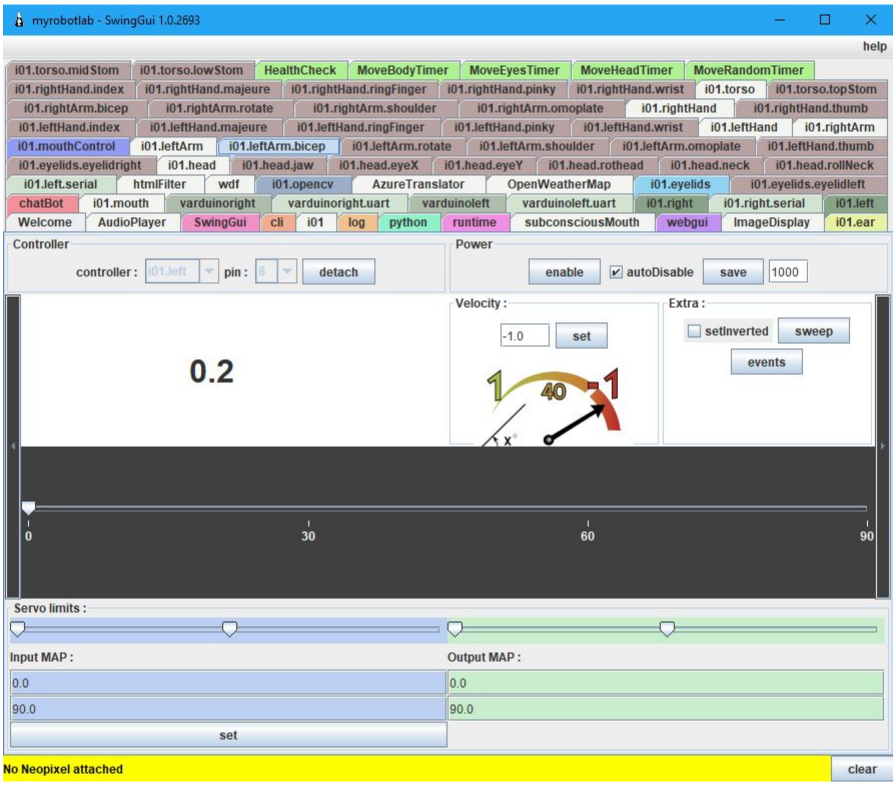


Figure 3.13: Manual movement screen Myrobotlab

In the Arduino tab, the Arduino COM (USB) port can be set (can also be configured so the software fills in automatically). It also enables the user to see and edit the Arduino code that stablish the communication with the Arduino. A screenshot is given below. The most important part is to check or set the COM port that is used for the Arduino.

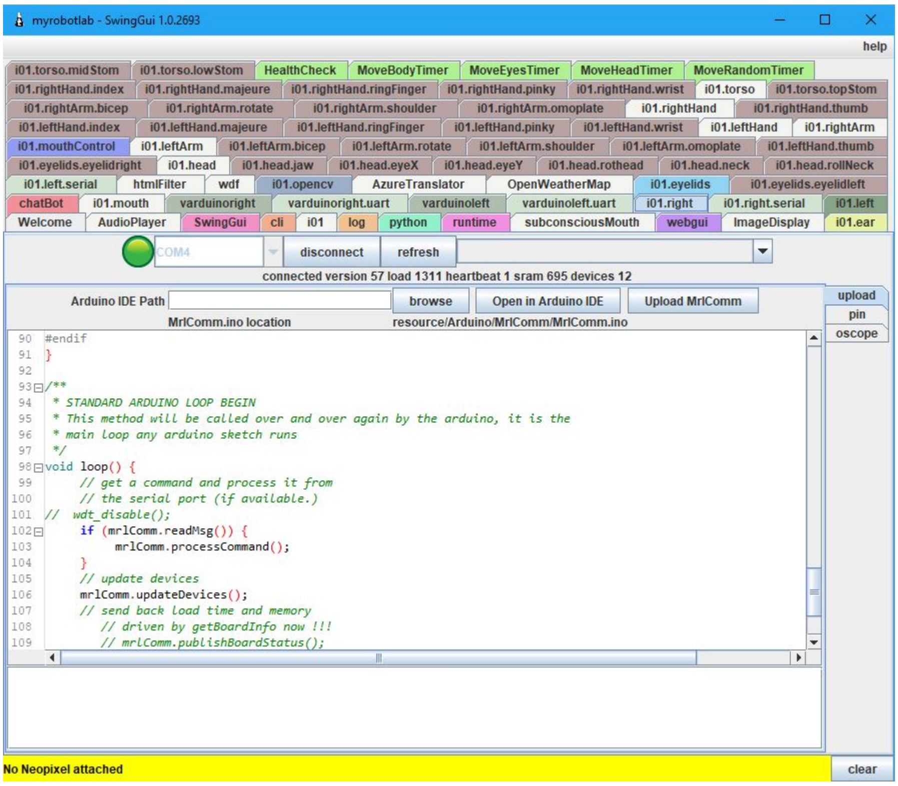


Figure 3.14: Arduino screen Myrobotlab

And the last tab to be mentioned inside of the Myrobotlab environment is the runtime tab. Here all the different plugins for Myrobotlab can be installed, uninstalled and manually started:

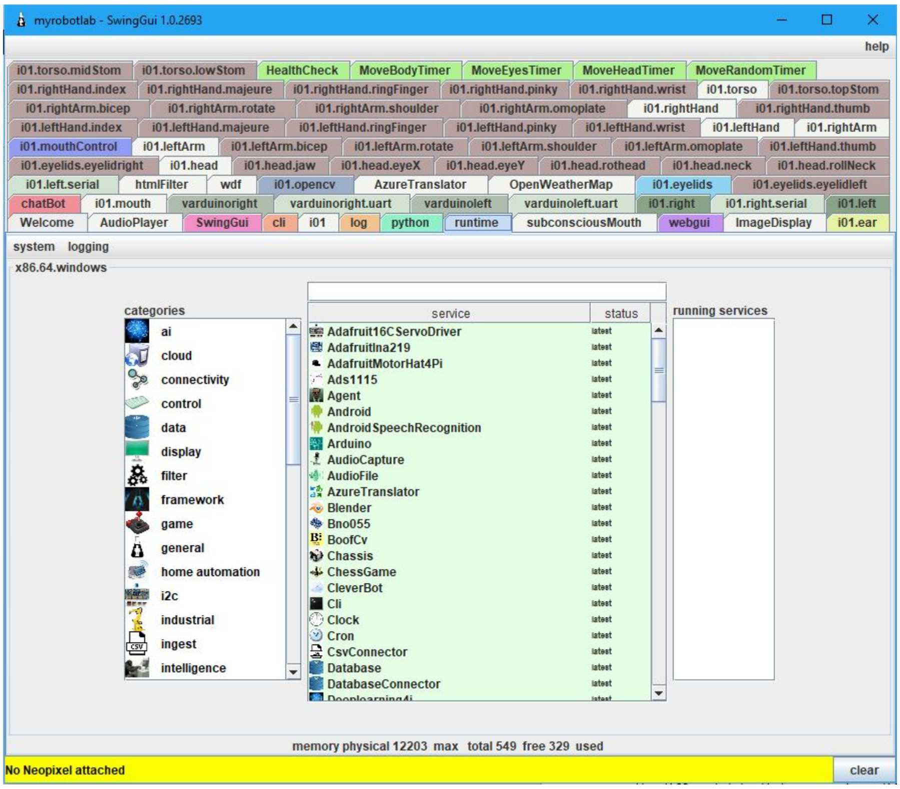


Figure 3.15: Runtime screen Myrobotlab

As last the webui is loaded. This webui is the google speech, which enables the user to communicate with InMoov using voice commands. This is opened in google chrome and handles all the speech functionality:

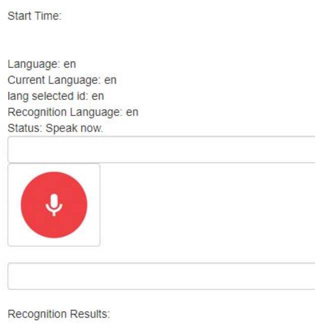


Figure 3.16: Speech webpage

3.2.2. Myrobotlab modes

Now the different modes of Myrobotlab will be discussed. They have been mentioned in the beginning of this chapter.

Virtual mode:
One of the great and just added functionalities to the Myrobotlab software is the option to control InMoov using a virtual environment. Everything in this mode works the same as in the real world, and it can be programmed and tested without connecting the real robot. The movements can be checked using a virtual window (see figure below).

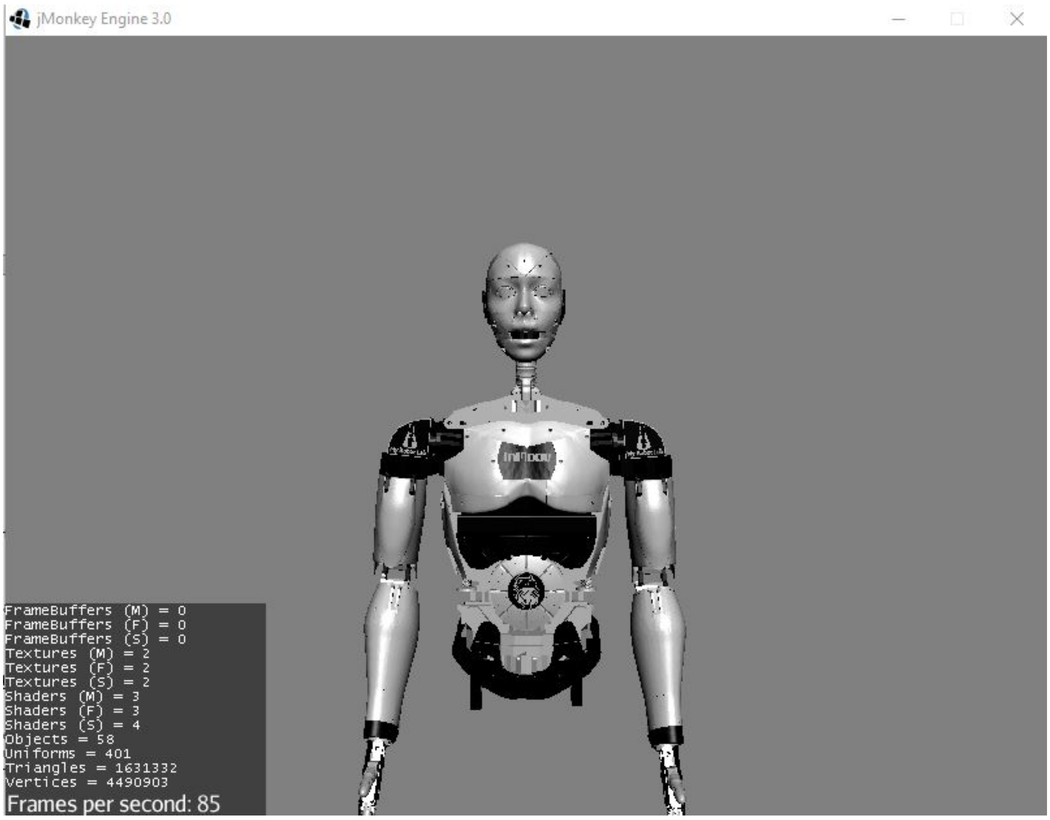


Figure 3.17: Virtual InMoov

This option offers great testing capabilities, because the software can be tested without the need of the full robot. The virtual mode is not perfect yet, because for example the legs still need to be added and the fingers and eyes cannot be moved.

Rightside:
This mode does what it says and only enables to control the right side of InMoov. This is handy to get familiar with InMoov and not have the complications of the complete robot. In this way, the right arm of the robot can be validated without being worried about the left arm of the robot. This functionality is only available for the right side of the robot and not for the left side.

Full:
The last and most important mode of Myrobotlab is the full mode. This mode enables the full control of InMoov. This means that both, the left and right side of the robot, are available to use. This is the environment which is used to get the final results and the mode that is probably used when the robot is doing his job as a final product.

4. Project management

4.1. Mission and Vision

4.1.1. Mission

The importance and the progress of humanoid robots is massive these days, because the need of robotic help is exponentially increasing nowadays. Therefore, the mission for this project is:

To enlarge the knowledge of humanoid robots and to use the skills of robotics to our advantage

The project will be done in contract with Technobothnia (the technical laboratory of the three universities of Vaasa). The final product shall be used to represent Technobothnia as a high-tech laboratory.

4.1.2. Vision

The vision for the completion of InMoov is:

A robot in the shape of a human being, that can assist in learning, research and education to achieve the future scientific goal (for example AI or deep learning).

4.2. Work breakdown structure (WBS)

In the WBS, the entire project is split down into four different sections and after that in further subsections, to reach the construction of a detailed WBS that will give insight into the needed time and resources. This WBS will be used to make the main task manageable. On the next page the complete WBS can be found with the different layers.

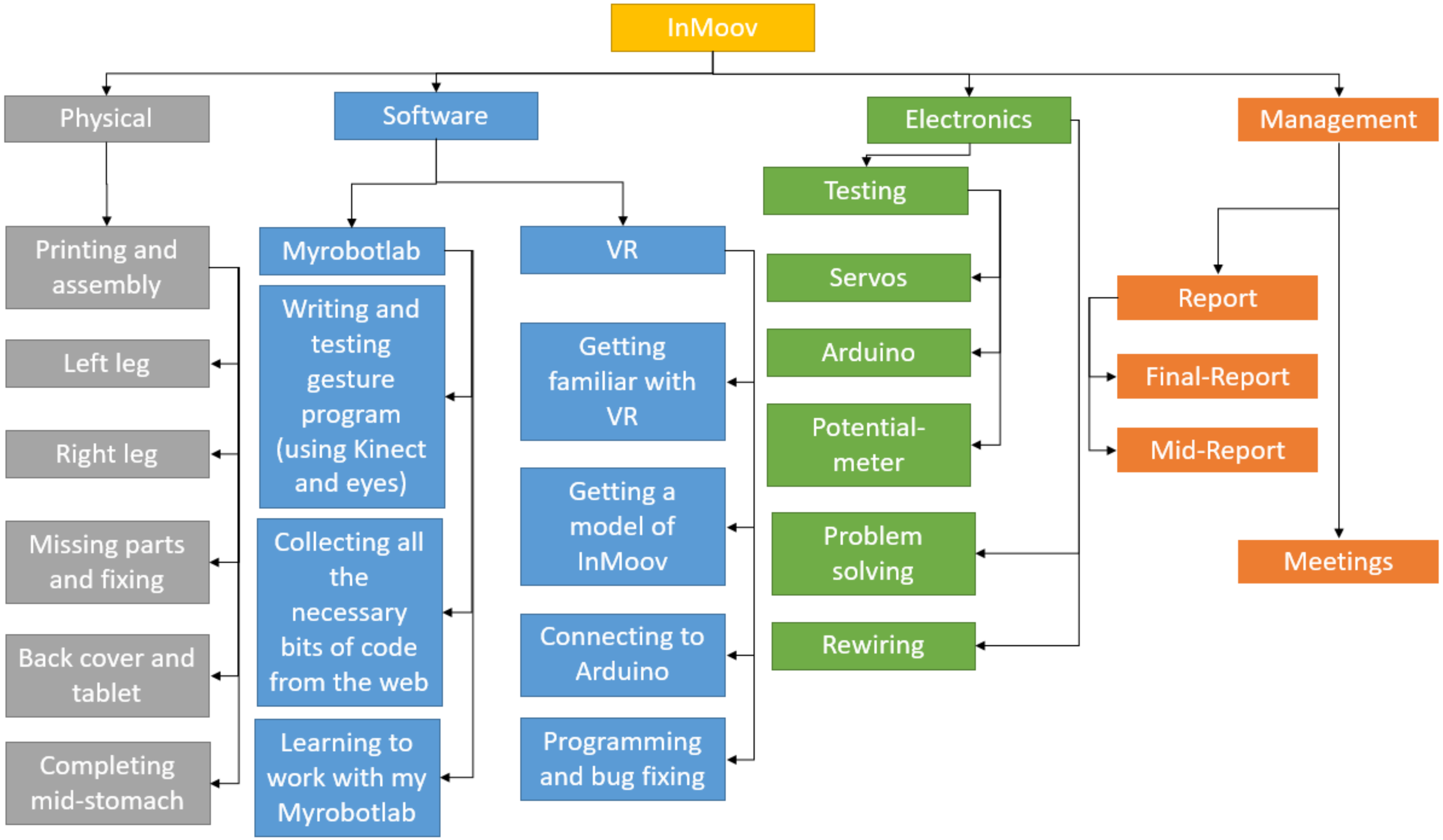


Figure 4.1: WBS

First of all the four main sections will be mentioned: the Physical Aspect of InMoov, the Software development and implementation, the Electronic Aspect and the Management work that this type of project requires. By the physical point of view, completing the manufacturing and assembly phase of the legs are the main goals followed by a continuous fixing period of other parts. The software section concerns the study and the implementation of a wide arrange of commands for InMoov, supported by the huge InMoov community and also study the VR (optional task) and the possibility to make it work together with InMoov. The electronic aspect of InMoov provides: the testing of all the components (for example the servos and Arduinos), solving problems for different components and rewiring InMoov. The final main section is the management, this section describes all the tasks that the different project members have to full fill. During the entire course of the project in order to organize their work and tasks as best as possible. The completion of this part is based on many meticulous meetings and has its main goal in publishing the final report.

4.3. Schedule

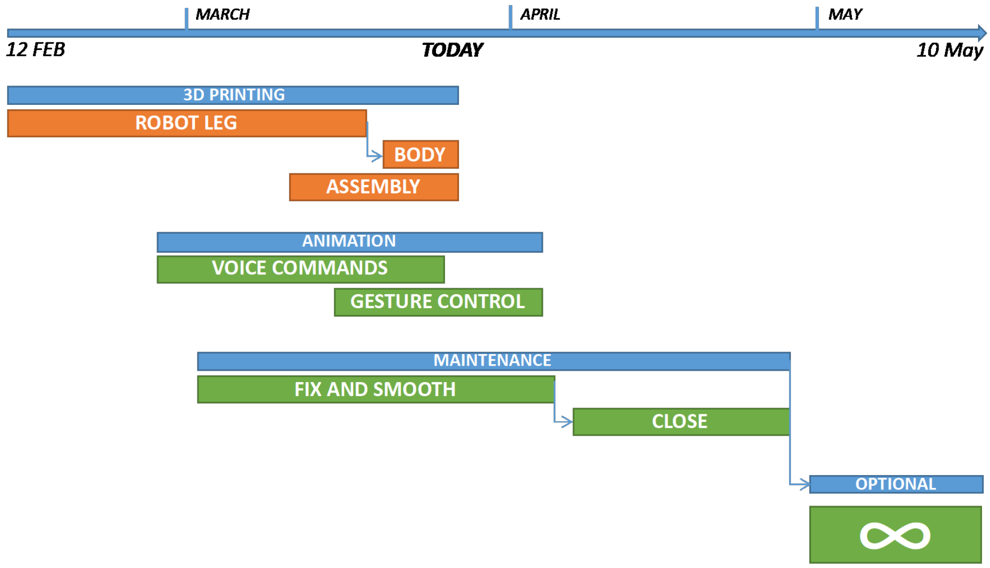


Figure 4.2: Schedule to finish the project

A precise constructed schedule is the base of a smooth running project based on Microsoft Project Software. R3-DFin developed a timetable to manage the time of execution of every main activity, to then itemize what each is composed of. Below, there is a vision of the entire schedule of the InMoov project represented in a detailed Gantt Chart.

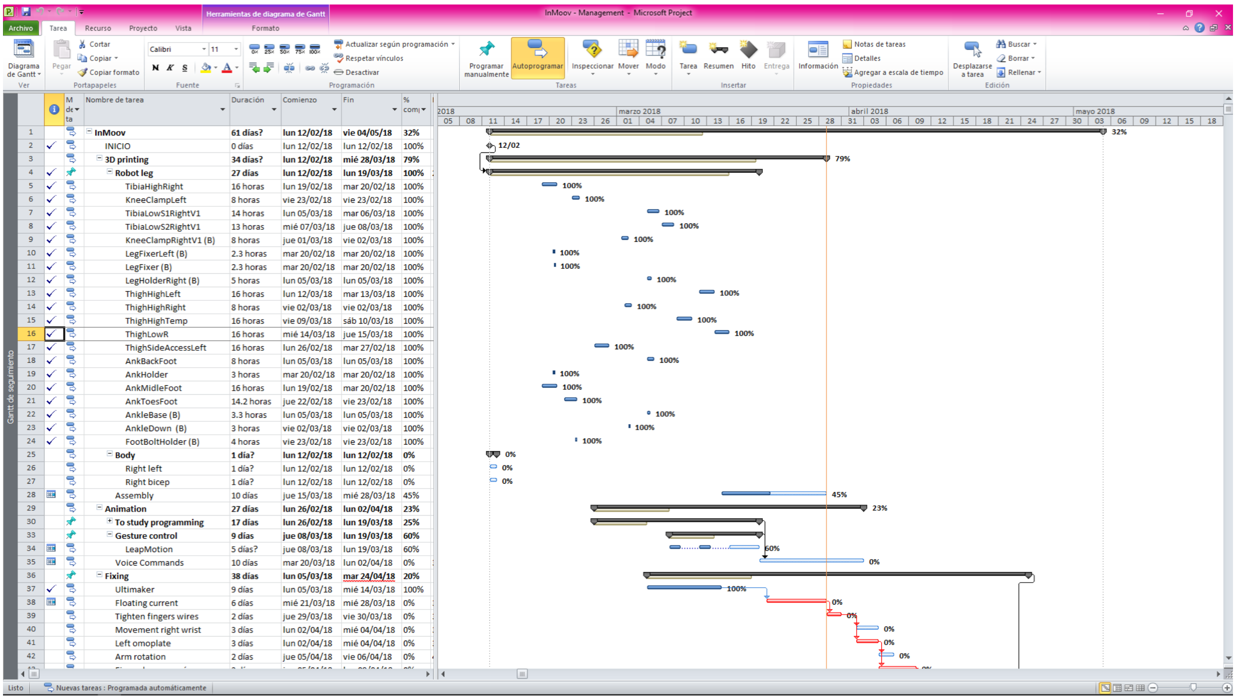


Figure 4.3: Gant chart for the InMoov project(1)

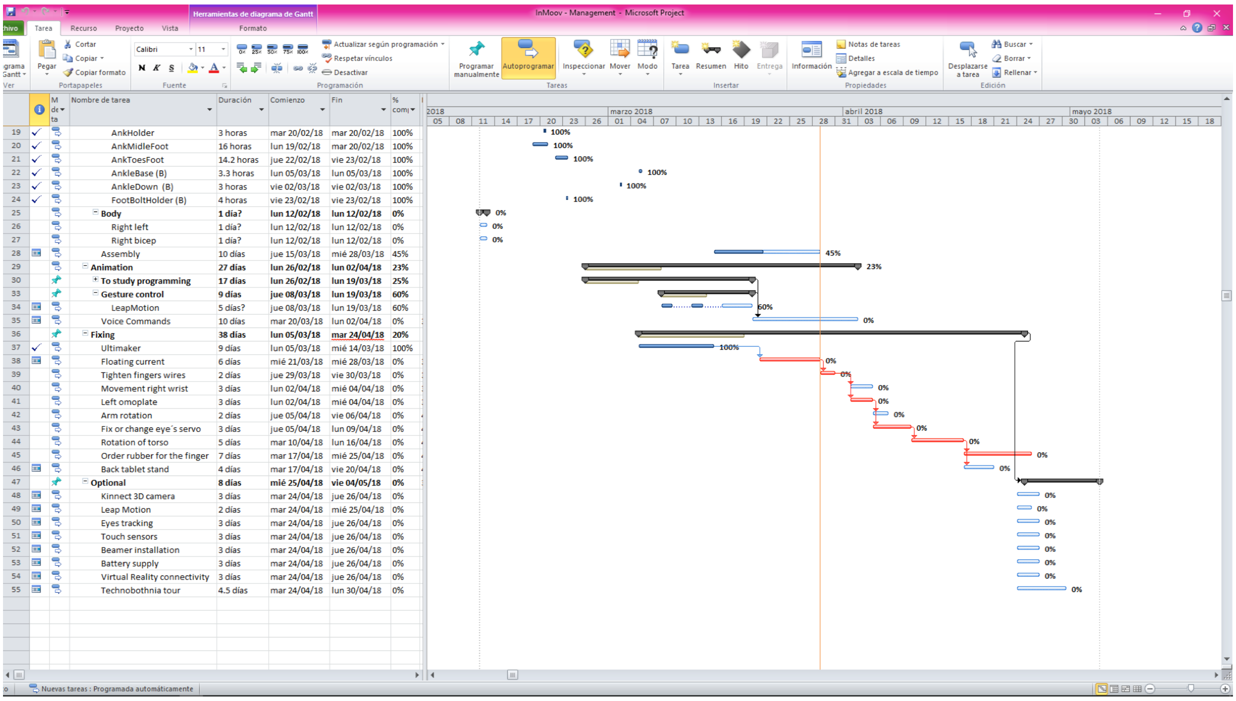


Figure 4.4: Gant chart for the InMoov project(2)

4.4. Milestones

R3-DFin paid attention to define correct milestones to have a clear idea about which are the main steps to pursue the main goal. The different milestones were put on a time line. Primarily, printing the leg is the main action to accomplish before starting to write the midterm report. Robot movements need attention later. This will be done when the main step of completing the robot is accomplished. When the robot is able to move perfectly without any type of mechanical or electronic problems. Software completion is going to take most of the remaining time of the project. There are also a couple of "optional milestone": implementation of VR support for InMoov, the combination between the Technobothnia Omron Robot[2] and InMoov. In addition, it is also considered the idea of implementing a video-projector on the robot. The team is ready to do their strongest effort in order to have enough time to complete all these tasks.

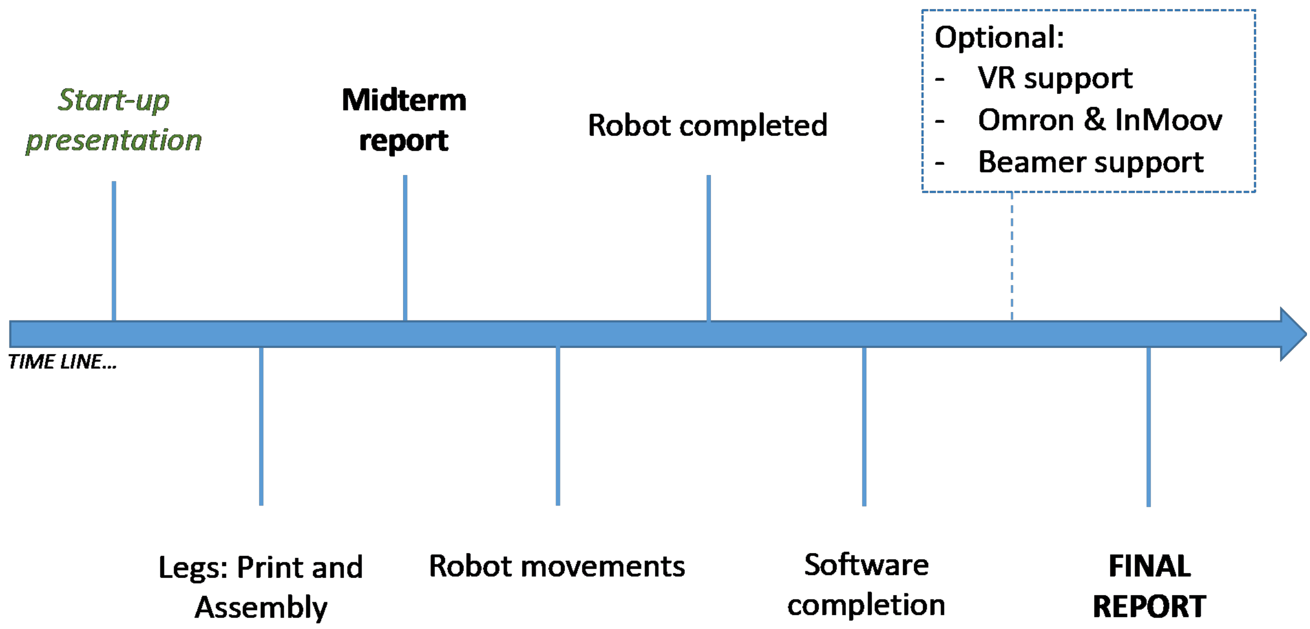


Figure 4.5: Milestones

[2]The Omron Robot of the LD-Series is an autonomous vehicle with the ability of self-mapping the area around him. It is used for Research and educational programs.

4.5. Human resources plan

A RACI chart has been structured to ensure clear distribution of the different roles and responsibilities of team members and collaborators. All members are responsible of the writing of the reports and presentations during the semester. Everyone has been linked to different tasks on different levels of responsibility: Responsible (R), Accountable (A), Consult (C) and Inform (I). As the Coordinator, Federico performs mainly management roles, organizing and monitoring the progress of the project. He is responsible for the Project Statements and Risk & Cost Management. He also is responsible about the work of the mechanical system of InMoov. Patricia, as the secretary, has the responsibility of the schedule in collaboration with Carlos and she also supervises the 3D printing phase. Next to this she makes an inventory of all InMoov printed and to-print parts. Carlos is the head responsible person for assembling, fixing broken parts and ordering, and he carries out his job mostly in laboratory. Matthijs, mechatronic engineer, is responsible for the electronic system of the robot and also about the writing & testing programs. So he manages mainly the software perspective. In addition to the project members there where various other people that where taken into the chart: the project supervisor Rayko Toshev, EPS Coordinator Roger Nylund and three external people. Rayko Toshev should be contacted and consulted for any subject that concerns the project. Roger Nylund is the reference person in case of doubts about the PM. Sulaymon Abiodun Tajudeen, master student, helps with his knowledge on the electronic components and the same is true for Thomas, who influences this project with his knowledge of mechanic and electronic challenges. Finally, Hanna Latva, the EPS English teacher, is consulted for a final check of any report or presentation.

Symbol	Meaning
Fe	Federico
Pa	Patricia
Ca	Carlos
Ma	Matthijs
Ra	Rayko
Ro	Roger
Su	Sulaymon
Th	Thomas
Ha	Hanna

Figure 4.6: Legend to the RACI-chart (fig. 4.7)

RACI Matrix		Project team				SV	Indirect people			
Project area	Activity	Fe	Pa	Ca	Ma	Ra	Ro	Su	Th	Ha
Project management	**Project statement**	**R**	**I**	**A**	**I**	**C**	**C**			
	Schedule	A	R	R	A	C	I			
	Risk management	**R**	**C**	**I**	**A**	**C**	**C**			
	Cost management	R	A	I	I	C	C			
	Midterm report	**R**	**R**	**R**	**R**	**C**	**I**			**C**
	Final report	R	R	R	R	C	I			C
	Presentations	**R**	**R**	**R**	**R**	**C**	**I**			**C**
Realize InMoov	3D printing	C	R	A	A	C		I		
	Assembly	**I**	**A**	**R**	**I**	**C**		**I**		
	Fixing of printed parts	A	I	R	I	C				
	Ordering parts	**I**	**I**	**R**	**I**	**A**				
	Making inventory of the parts	I	R	A	I	C				
	Writing & testing program	**I**	**I**	**A**	**R**	**C**		**C**		
Problem solving	electronic systems of InMoov	A	I	I	R	C		C	C	
	mechanical systems of InMoov	**R**	**I**	**I**	**A**	**C**			**C**	

Figure 4.7: RACI-chart for the InMoov project since February 2018

4.6. Project budget

Concerning the budget managing R3-DFin generated a list of every cost it will have to full fill. Defining all the fixed costs and variable cost possible. Once the estimated costs are grouped, the authorized costs baseline is obtained. The fixed costs correspond to the depreciation of the two 3D-Printers (Ultimaker 3 Extended and Makerbot Replicator Dual) and is calculated based on a standard 1000€annual depreciation divided by 252 working days.

$$\frac{1000€}{252\ days} = 3.96\frac{€}{day}$$

Moreover, multiplying the 3.96€by the 5 days of a working week, the total is 19.8€per printer, 39.6€per week (14 weeks is the duration of the complete project). The estimation of the fixed cost for the project are completed by engineers' salary. The average of gross income of engineers in Finland is 3,706€ [Bernard Parent,2008], that entail an expense of about 5000€for the enterprise (and the 14 weeks of work are rounded off to 4 months). Then:

$$5000\frac{€}{month} * 4\ engineers = 20,000\frac{€}{month} \quad (1)$$

Calculated on the whole duration of the project:

$$20,000\frac{€}{month} * 4\ months = 80,000€ \quad (2)$$

The electricity needs to be taken into account. Finnish electricity costs are about 0,065 €/KWh [Il Sole 24 Ore,2013] and 200W is the standard usage of a 3D-printer. With this information it is possible to calculate what the cost of electricity are (staying ON all day long every day during printing period, 7 weeks)

$$0.65\frac{€}{kWh} * 268.8\ kWh \simeq 20€.$$

this by 7 concludes that the electricity costs are 2.9€per week. Next to this it is necessary to take the price of ABS in consideration. This material is used to print all the different parts of InMoov. The regular cost of 1 Kg of ABS is about 30 €. And the 2 legs completely assembled are about 5.7 kg, and the material wasted because of defects during printing is estimated to be around 1.5 kg.

$$(5.7 + 1.5)\ kg * 30€ = 216€$$

Dividing this result by the 7 weeks of printing and reprinting, the result is 30,8€per week. During Week-4 a set of screws and bolts has been bought and the price was about 38.7€. In the middle of the project, circa Week-8, it will be necessary to purchase a Windows 10 Tablet. Considering a standard price of 300€. During the week 8-9-10 R3-DFin considers the possibility of replacing some malfunctioning servos, with an average price of 23.1€(30.8€/week).

	Week1	Week2	Week3	Week4	Week5	Week6	Week7	Week8	Week9	Week10	Week11	Week12	Week13	week14
	30.8 €/week							← Printing and Reprinting						
	2 €/week							← Electricity						
				38.7€	← Screws and Bolts									
	Extra Servos →							30.8 €/week						
	Tablet →							300€						
	← Repreciation Machine 39.6 €/week →													
	← Hypotetichal Engineers salaries 5714.28 €/week →													
Weekly Spending	5716,29	5716,29	5716,29	5716,29	5716,29	5716,29	5716,29	6014,29	5714,29	5714,29	5714,29	5714,29	5714,29	5714,29
Cumulative Spending	5716,29	11432,57	17148,86	22865,14	28581,43	34297,71	40014,00	46028,29	51742,57	57456,86	63171,14	68885,43	74599,71	80314,00

Figure 4.8: Calculation of expenses during the project

In the picture below all the expenses are shown in dependency of the weeks. The variation between the ideal (blue curve) and the unideal (orange curve) condition is too small to be observable, because the fix costs of engineers are prevalent on all the other costs. For the same reason, the graphics that resume the total cost of a real project light this is almost a straight line. The light increase of inclination in the weeks 8-9-10 caused by the purchase of tablet and extra servos is nearly canceled compared with the engineers salary. Range: week1-14, start 19/2, end 21/5 2018.

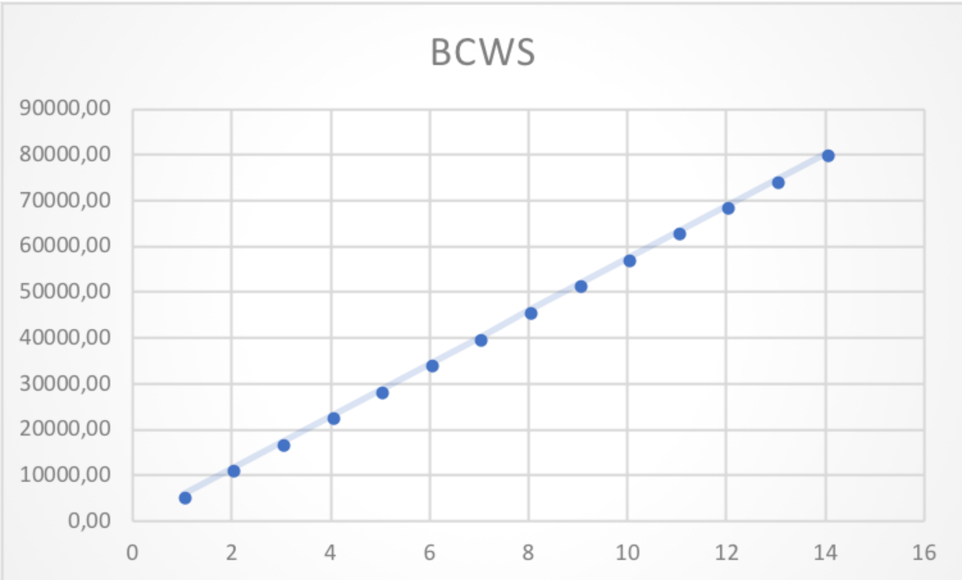


Figure 4.9: Expenses in dependency on the project weeks

4.7. Project risks

Risk Management is the term which is used to define the complex of activities that an enterprise acts to identify, analyse, assess, eliminate and control every type of hitch. The goal is to maximize the efficiency of a process and minimize declines. It is perceived as an accurate analysis of a big amount of statistic and probabilistic facts, but it is also true that Risk Management can be decomposed in main steps of action, also corresponding to some main question asked themselves to evaluate the situation. Once understood what the group is trying to do, studied the environment and the context, it is fundamental to understand what might affect the stability of the project, identifying unsafe matter and possible risks. Now the scope is to analyse which where the most important risks of InMoov R3D-Fin project and how the members manage them. Firstly, the main risk was that something during the fixing phase could have gone wrong. For example, it was possible that some other pieces would break, in addition to the ones already calculated previously. It could look like an unmanageable aspect, but it could be limited with precise interventions of maintenance and periodic checks of all the parts of the robot. It also guarantees a fast emergency service in case of necessity. The time plays a fundamental role in every project, and also in InMoov project attention was paid to respect time lines and deadlines. Another possible risk was that some deadline would not have been respected by third parts, for example the couriers for the ordered pieces. By this point of view, the most natural way of action has been to pay extra money to purchase the fast delivery service instead of the normal one, in order to have more certainties. Another unlikely but possible event is that a member would leave the group giving in fact more responsibilities to the remaining people. This is what happened in the R3D-Fin group, where a member left his position after just one week of work. The team reorganised roles in order to cover every lacking task. Fortunately, it happened at the beginning of EPS project, so the incisiveness of this event on the project has been lower than in worst case of happening in an advanced phase.

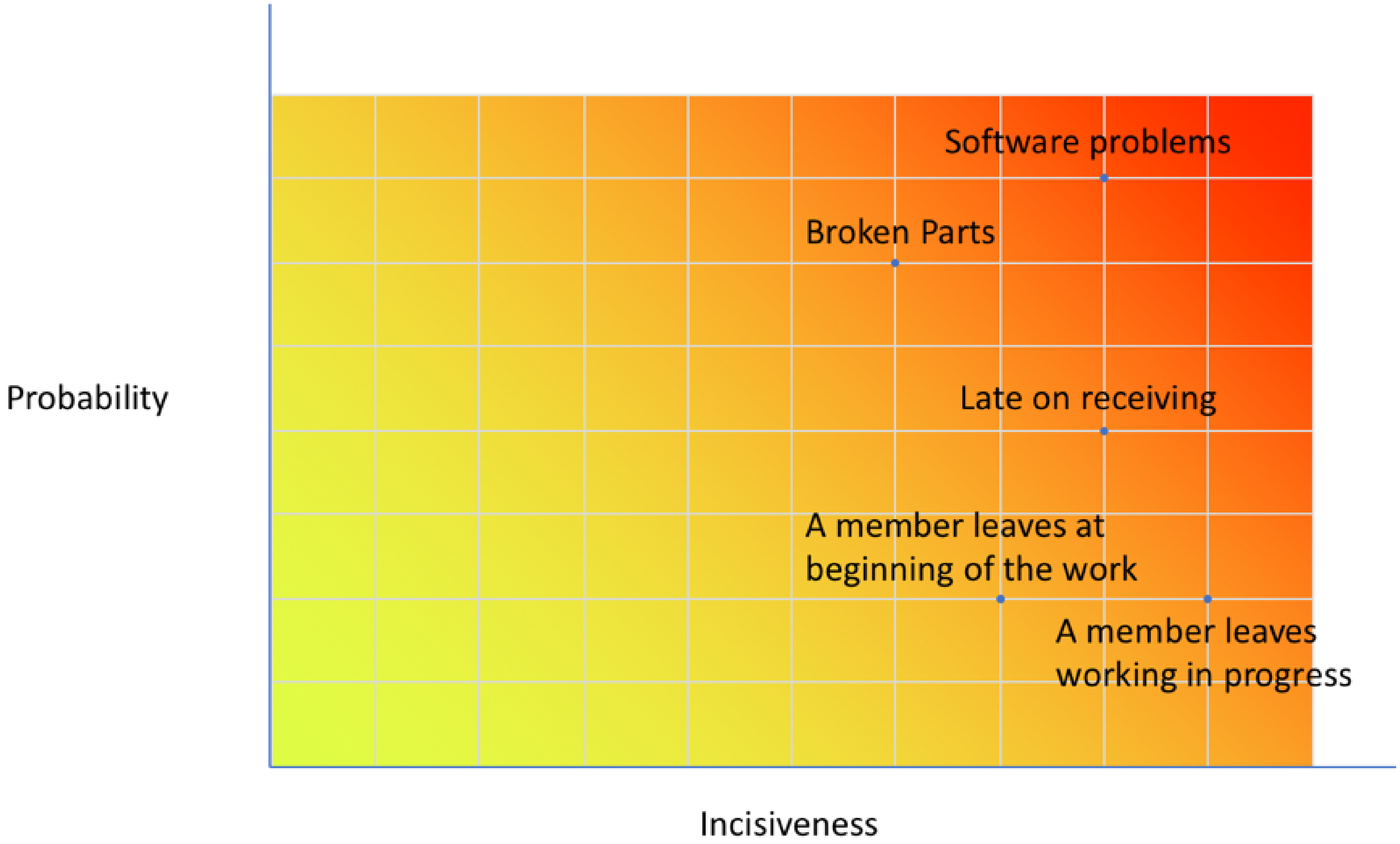


Figure 4.10: Inmoov's Risk Assessment

5. Additive manufacturing and assembly

5.1. Printer overview

The first milestone of the Project is to finish the 3D printing of InMoov. The parts that still need to be manufactured are the legs, lower stomach and components from the rest of the top body which need to be repaired. There are two 3D printers available for the InMoov project at the Technobothnia laboratory.

One of them is the Ultimaker 3 Extended. This printer is one of the latest models from Ultimaker. This product is offered for sale since October 2016. It is a popular 3D printing company located in the Netherlands(established in 2011). This printer has a big print volume (215x215x300 mm) and therefore it will be used for printing the bigger parts. The model can be loaded on to the printer via Wi-Fi or USB-stick. In the upper-front corner there is a camera, which allows to follow the printing process via the internet.

Figure 5.1: Ultimaker 3 Extended

On the other hand, the MakerBot Replicator Dual shall print the smallest parts. This printer has a printing volume of 220x150x150 mm [Wikipedia, 2018]. Older than the Ultimaker, this one does not dispose of a USB connection nor camera to follow the printing.

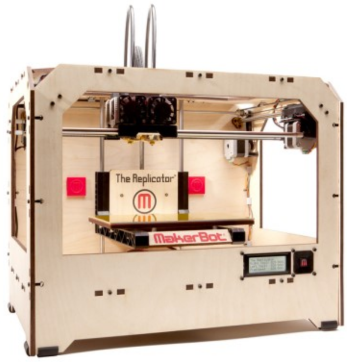


Figure 5.2: Makerbot Replicator Dual

The left one is the main nozzle which expends the principal material; the second one, if it is required, will build a base in order to help the printing, preventing falls of the melted plastic. They provide an intuitive display to change configurations, through the wheel/arrows.

The printers follow the next printing pathway:

1. A careful cleaning of the base with a brush is needed to remove every remaindering material from the previous print.

2. The base has to be glue-brushed with a thin layer to keep the first layers of plastic bonded. (only needed by the Ultimaker)

3. The desired model is uploaded in the printer, either through a Wi-Fi connection (or directly via USB-stick on the Ultimaker).

4. The printer starts the heating of the base, usually until 200-250 °C, which can be set in the printer software.

5. Once the base has reached the desired temperature, the ultimaker will auto-level his base (with the Makerbot this has to been done manually).

6. If everything is right, the printing starts.

7. When finished, a sound will mean that the finished part can be removed from the base, with the aid of a spatula.

8. Finally, the base and printer-head starts cooling and the printer is ready for the next print.

5.2. Printing software

Each of the printers has its own software package. This type of software is not a CAD tool. It is only used to adjust the model, and change the configuration of the printer like temperature, speed, density, etc. The model has to be pre-designed and formatted as a ".stl" file and then converted by the printing software to a ".gcode" file, the extension required by the software.

5.2.1. Ultimaker Cura 3.2.1

The Ultimaker software (Cura 3.2.1) allows configuring of the printing of the Ultimaker. It happens to be an intuitive and modern program with a easy to use interface. Its appearance is shown in the next figure.

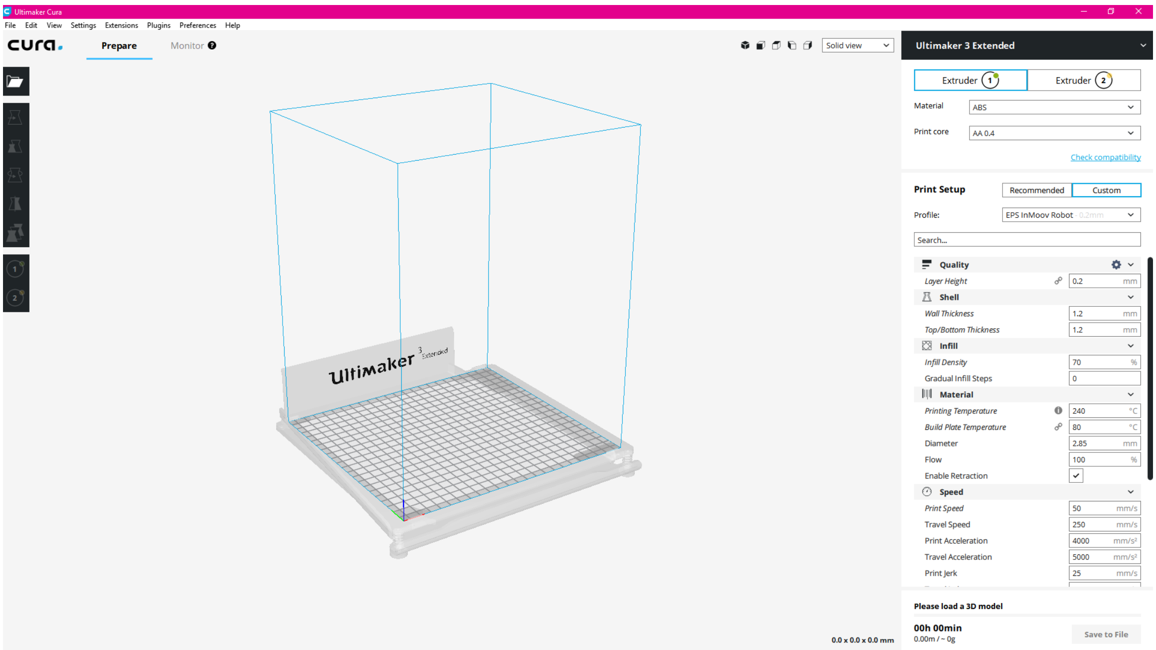


Figure 5.3: Ultimaker Cura software

With the tools in the left (figure 5.4) on the main screen the user can move, scale, rotate or change which extruder will print the model.

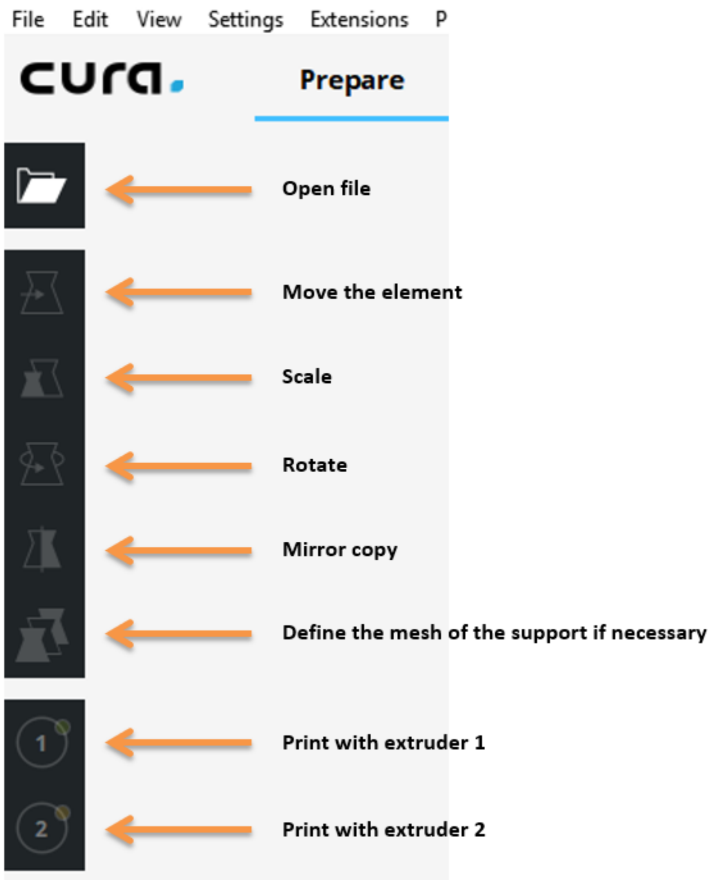


Figure 5.4: Setting tools in the Cura software

The advanced tools (figure 5.5) allow multiple settings for example: choosing the printing material, pre-set a configuration, the density of the inter mesh, temperature, speed and when a model is loaded in the program. After all the settings are set up the software calculates an approximated time of printing for the desired object.

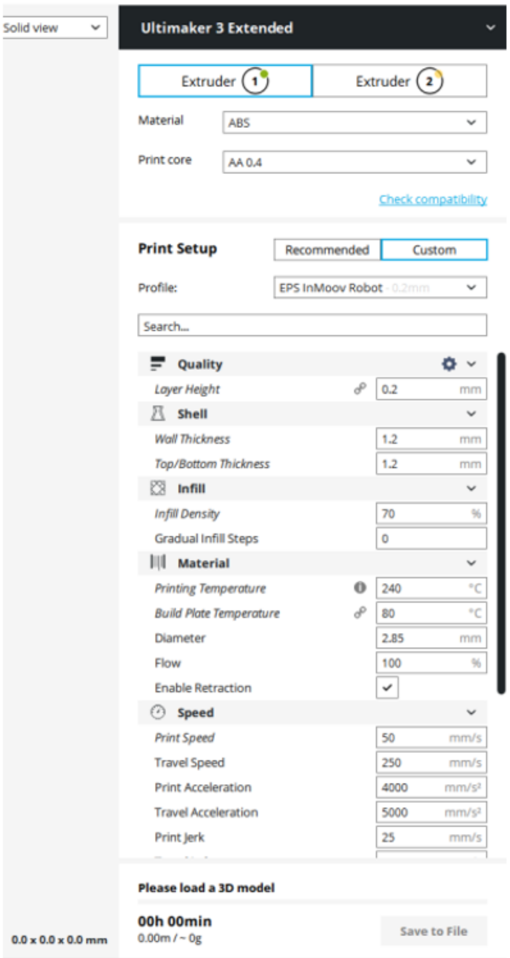


Figure 5.5: Advanced Setting tools in the Cura software

The setting, that is chosen for a print depends of the purpose of the object. The main setting, which was adjusted, is the percentage of infill. Default setting for the infill was 70%. For all the printed bolts throughout the project, the infill was set to 80%. Once the configuration is set as wanted, Cura will ask for a pen drive to save the model to.

5.2.2. Makerbot Print Software

Likewise, the MakerBot Print software works similarly with a tool for rotating, moving and scaling the object as shown below (fig. 5.6).

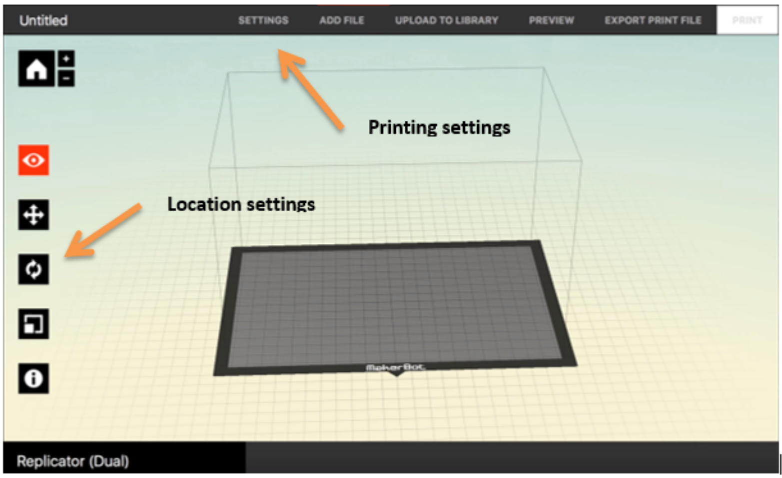


Figure 5.6: MakerBot Printing Software

In the printing settings the temperature, density of internal mesh, material and more settings can be changed by the user (fig. 5.7).

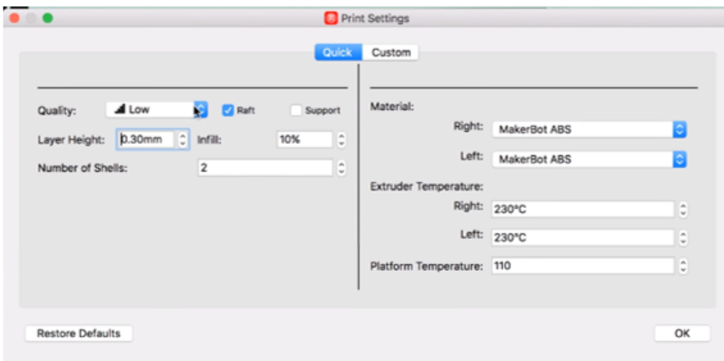


Figure 5.7: MakerBot Print settings

5.3. Materials

The market of the thermoplastic filament is under permanent development, but at the present time the best plastic for printing are **ABS** and **PLA**.

- **ABS**: thermoplastic composed of three major components: acrylonitrile, butadiene and styrene. Acrylonityl provides stiffness, speed in the face of high temperatures and resistance to chemical attacks; while butadiene gives tenacity and protection against impacts. Regarding styrene mechanical strength and brightness. This plastic can be polished, sanded, filled, painted, glued and so on. The impressions with this material work with temperatures of between 230 and 245 °C. It is one of the most used, so much, that even LEGO pieces are made of ABS.
- **PLA**: Its main virtue is that it is a biodegradable, non-polluting compound, usually obtained from corn starch. The texture of the pieces is not as smooth as with ABS but it is brighter and the corners results better. The temperature needed for printing is about 210 °C.
- **PVA**: is also a biodegradable plastic that is mainly used on multi-head printers as a structure for those areas that are more fragile or hang in the air (support material). The temperature needed is around 180 °C and is perfect for complex objects. When the printing is over this plastic can be dissolved in water and easily removed.
- And another plastics less utilized like HDPE (High-density polyethylene), NYLON, PET (Polyethylene terephthalate), HIPS, NINJAFLEX, etc.

The entire robot has been printed with ABS - black colour in the MakerBot and white in the Ultimaker.

5.4. Maintenance

There are many problems that arise when printing with 3D printing. These complex machines with many gears and complex technology requires human attention when things do not go as planned. In this chapter the most common issues with 3D-printers are going to be explained.

1. *The nozzle is blocked:* this will prevent material for being extrude. It usually happens when dirt blocks the nozzle or a too high temperature melts and carbonizes the plastic. There are three procedures to solve this issue like using a needle to poke through the nozzle, applying a cold pull with a PLA or cleaning filament, applying a hot pull with a PLA or cleaning filament and as last resort replacing the nozzle. Check out the full guide to clean the head with any of these pathways. In the figure below a hot or cold pull can be seen (fig. 5.8).

Figure 5.8: Removing the bowden tube

2. *Problems with the engine:* if the nozzle does not print, it could be caused by the engine which pushes the material. It can be cleaned by disassembling and cleaning the motor in order to avoid any obstruction (fig.5.9).

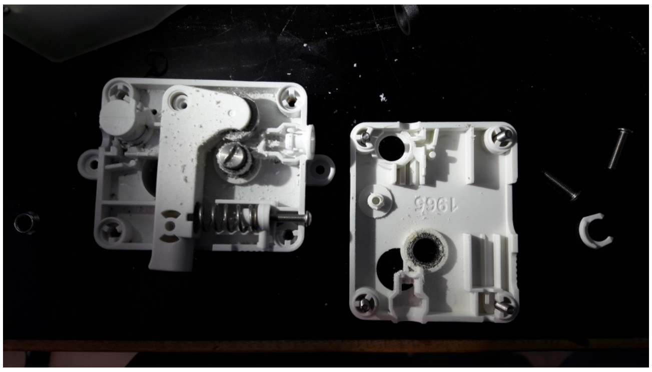


Figure 5.9: Disassembling the engine

3. *Not enough material:* this is a situation that needs to be considered. If there is not enough material to complete the print, the print will fail. As well as lost material a some time will be lost for reprinting the part. Both Cura and MakerBot Printers have an option to show what amount of material is needed to complete the print and give an idea if a change of spool is needed. The roll is at the back of the printer (fig. 5.10).

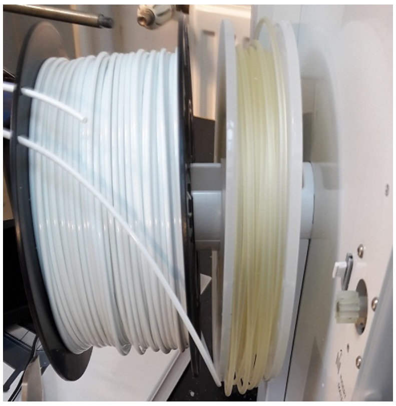


Figure 5.10: Situation of the rolls; left: ABS, right: PVA.

4. *Incorrect cleaning of the base:* usually, when the printing starts, the most important point is that the first layer remains bonded to the plate. Therefore the base must be carefully cleaned without residues of previous prints, and correctly glued with the glue stick (only Ultimaker). When this is not done properly there is a chance the first layer will not stick to the base and the print will fail.

5.5. Printing InMoov

As stated above, the remaining parts are the legs, and some parts of the torso. The legs are completed on this point. Some parts from the right leg were printed already when the project was started. So these ones where checked to make sure they were right and ready to be assembled.

All the designs were downloaded from the website of InMoov, section "Gallery" and "InMoov STL parts library". Selecting from the drop down list the parts of the leg, all the models can be downloaded and are ready to print (see fig. 5.11).

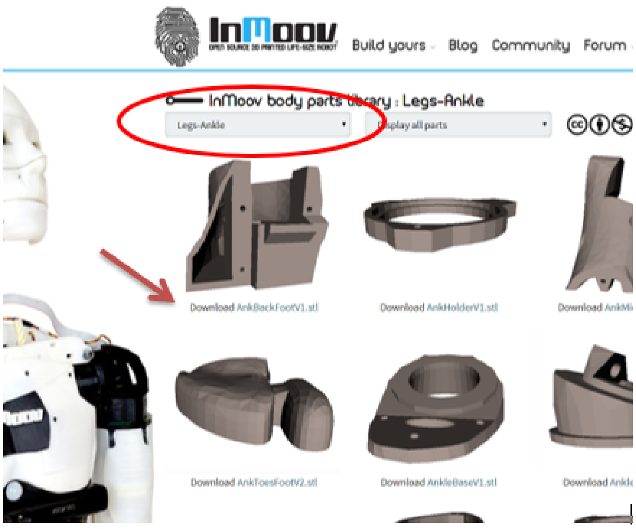


Figure 5.11: InMoov Library - Ankle

Some of them have some instructions to be taken into consideration for example if support material is needed. The support material (PVA) can be needed due to the complex shape (indicated in the section of "Legs Non-Motorized") or the density of the internal mesh of the piece. Which depends on the strength the part needs to have. The same website advices that some of the parts are really big, so it is recommended to use a printer with a higher printing volume (These parts will be printed with the Ultimaker).

Once the ".stl" files have been downloaded and the settings have been changed to the correct values, the software can convert the ".stl" file into a ".gcode" file. After this they have to be loaded in each printer, observing which colours belong to each piece.

5.6. Assembling InMoov

Once the 3D printing has been finished, it is time to assembly the different parts, a task that could take a great amount of time, depending on the quality of the printer and other external conditions.

Most of the parts cannot be assembled to each other directly, because all the pieces have a design tolerance where the parts fit with each other and, if the printer is not accurate enough, this tolerance could be increased. So, this tolerance has to be removed until the pieces match correctly.

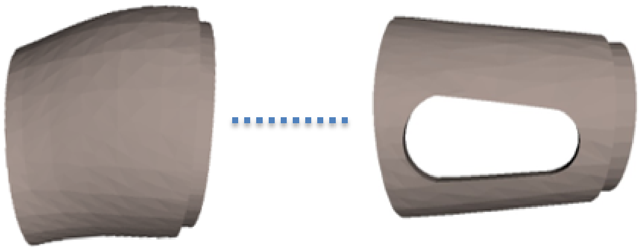


Figure 5.12: Assembling the leg

Besides this sometimes the pieces do not come out properly, with some defects after the printing. This happens when the layers or the wire that the nozzle extrudes do not glue to each other rightly, appearing howls and cracks that can lead to the rupture of the piece.

In the Technobothnia Lab, there are many tools available to deal with the fitting problems. These tools are:

1. Sanding machine (Dremel) (see fig 5.13): to sand the pieces where it is required to fit with each other.

2. Drill and drill bits: every needed holes are pre-designed in the models, but if it is necessary, they can be enlarged to facilitate the join.

3. Sandpaper: to sand manually and accurately or better the shape.

4. Soldering iron (fig 5.14): some defects and fissures can be fixed using a soldering iron. Setting the optimal temperature of the ABS (240°C) the fractures can be melted to improve the bound.

5. Acetone: the ABS plastic is acetone-soluble, so small imperfections or crack can be fixed with this solvent.

6. Glasses (protection)

7. Leather gloves (protection)

Figure 5.13: Sanding machine

Figure 5.14: Electric soldering iron

After the fixing of the different parts, the InMoov website can be visited to proceed with the assembly instructions. Next to these instructions there are detailed pictures of the connections and how the bolts, screws and nuts are disposed. The following picture (fig. 5.15) shows the assembled leg.

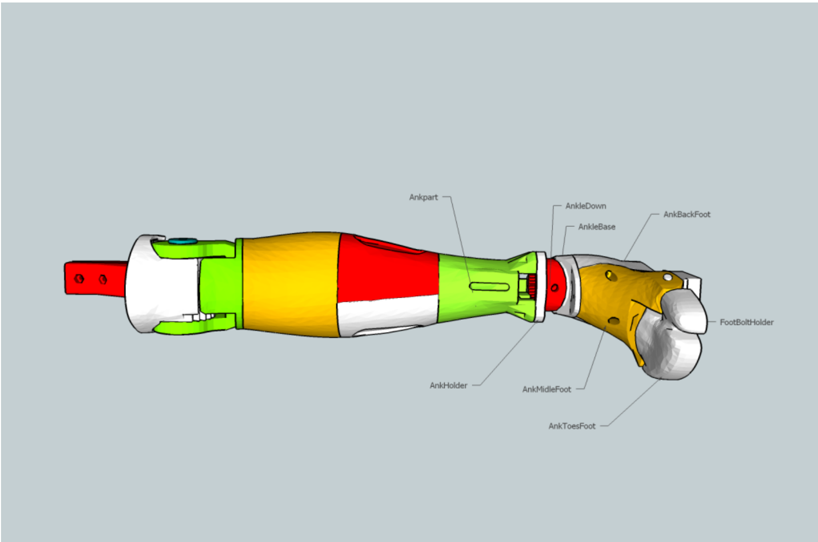


Figure 5.15: Pathway to assemble the leg

6. Building guide

6.1. Testing of servos

Before any new servo can be inserted into the robot, it needs to be tested on its functionality and set to a certain position (usually to the rest position). The same applies to the old servo. In order to do that, the following items are required:

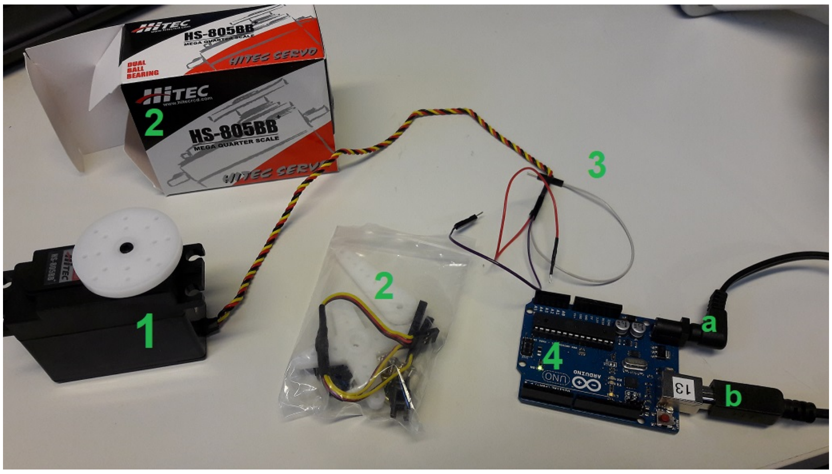


Figure 6.1: Set up for a manual test of a servo including: 1: Servo to test; 2: Items coming with the servo; 3: wires; 4: Microcontroller (here Arduino Uno) with a: charger & b: connection to PC

The wires on the servo should be connected as followed:

- Black –> GND
- Red –> 5V
- Yellow –> Pin (default 3)

The myrobotlab.jar file needs to be opened, Arduino and Servo started by clicking the right mouse button in the runtime and the connection to the COM port on the PC needs to be set differently according to every computer. In the servo tab, choose the signal pin (yellow wire), attach and move the servo.

6.2. Head

In order to fix the problems that have been stated in chapter 1.3, the entire head needed to be taken apart. The disassembly and assembly was done by following the video on the InMoov website (https://youtu.be/fMeiSKzO2p8).

6.2.1. Eyes

Afterwards, the missing eye servo (DS929HV ordered on HobbyKing[3]) for the y-movement of the eyes, was first tested manually with the Arduino to find out the limits and to check whether it moved smoothly. The servo was set to 90 before disconnecting from the microcontroller. Then, the servo was implemented below the eyes with two screws, that were delivered with the servo, and attached to the eyes as shown in picture 6.2. The connection between the eyes and the servo was printed in the Makerbot (see chapter 5.1).

Figure 6.2: Implementation of the servo DS929HV for the y-movement of the eyes

To finalize this part, the limits for each direction of the movement were determined by using the Arduino board.

[3]Online provider for all sorts of electronic devices for hobby users

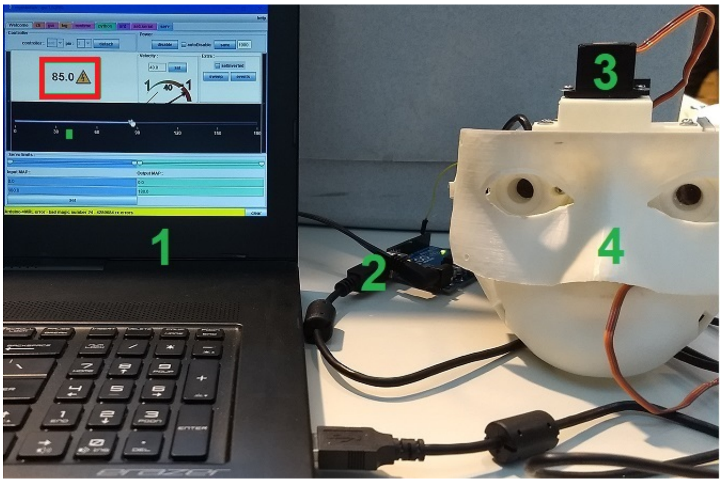


Figure 6.3: Set-up to determine the limits: 1: Laptop with myrobotlab.jar-File (minimum limit in red square); 2: Arduino Mega 2560; 3: servo DS929HV for x-movement; 4: eyes of InMoov looking far to the right

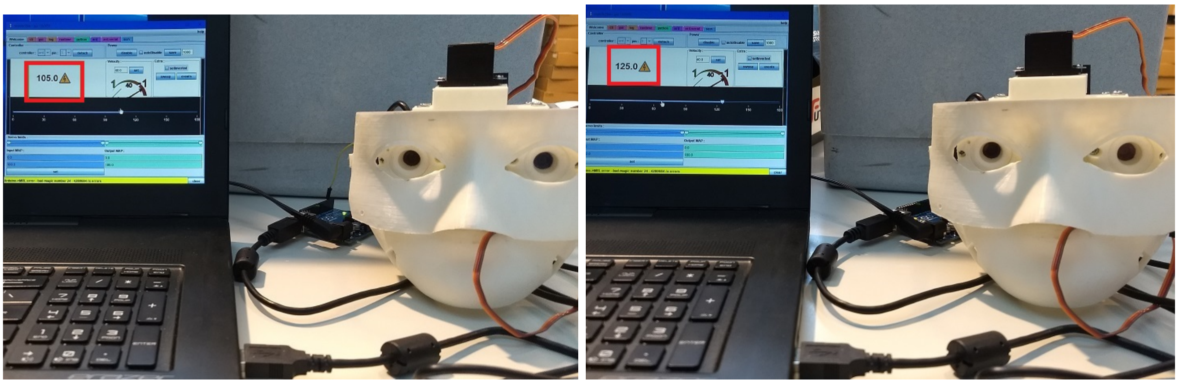


Figure 6.4: Rest position (left) and furthest point left (right)

The same procedure was repeated for the newly-installed servo to look up and down. The final limits can be seen in the table in the appendix E

6.2.2. Jaw

The movement of the jaw servo was not possible due to jittering. Therefore, the servo for the jaw was replaced by a new one, that was tested in the same way as the servo for the eyes and set to its rest position. The used servo was called MG996R (see figure 1.6 no. 4). The new servo can be seen in the picture below.

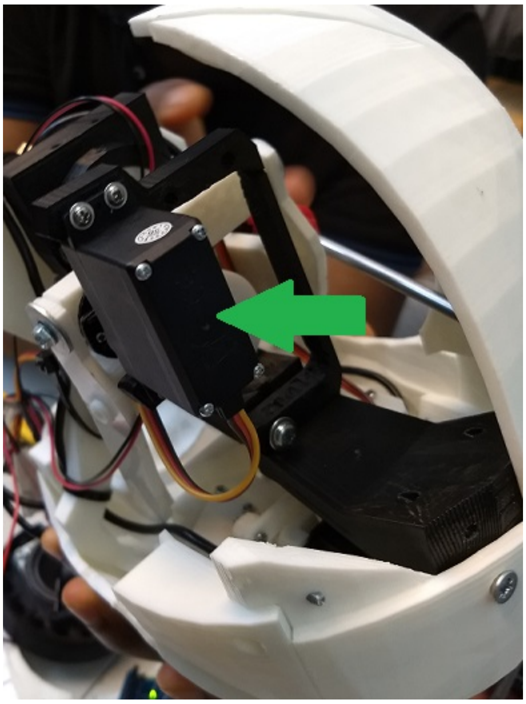


Figure 6.5: Newly installed servo for the jaw movement

The connection between the eyes and jaw had no need to be disassembled as it was always connected due to the wires of the cameras. Due to that, the new jaw movement could be tested directly to determine the limits of the servo for an opened and closed mouth. The limits can be found in the Appendix E.

6.2.3. Head rotation

As stated already, a new servo needed to be installed for the head. After testing the servo (see chapter 6.1), it was set to its middle position (=90°).

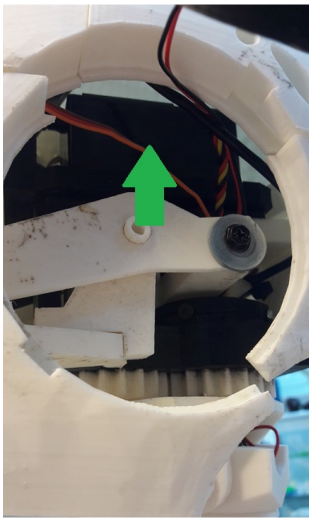


Figure 6.6: Newly installed servo for the head rotation(picture taken after the re-assembly)

The servo moves a system, that consists of two intermeshing gears. The teeth needed some adjustments to make them move smoothly together.

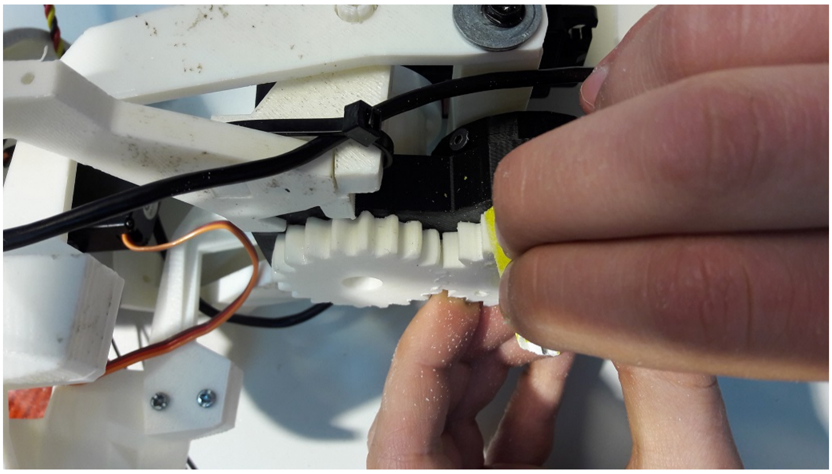


Figure 6.7: Sanding gear system

6.2.4. Loudspeaker

The robot is designed to be able to speak using his ears as loudspeakers. The installation requires a resoldering of an AUX cabel, connecting them to the plus and ground wire of the loudspeakers.

Figure 6.8: Loudspeaker with ground wire (black) and signal wire (red)

The AUX wire consists of three parts, a ground, a plus and a minus. The three parts are isolated through the black isolator. The set-up is shown in the picture below.

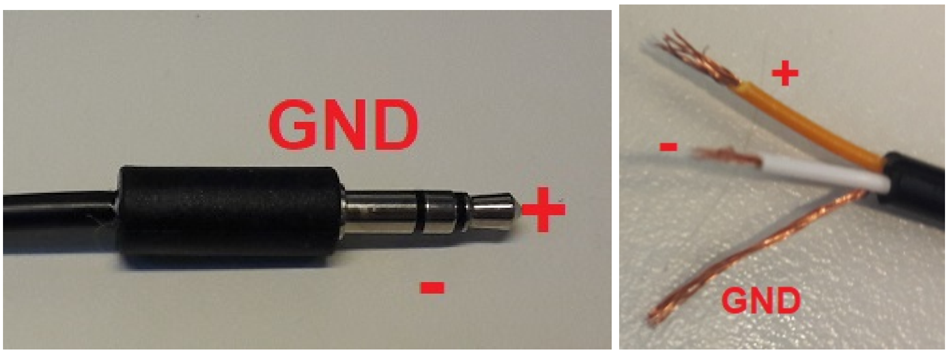


Figure 6.9: Sections of an AUX wire

In this case, minus and plus functions as the direction, where the sound comes from, the left or right loudspeaker. The soldering should be done by soldering the black wires of both loudspeaker to the ground part. One red wire of the left loudspeaker is then soldered to the + pole and the right one to the - pole. By using a video (e.g. https://www.youtube.com/watch?v=hTvJoYnpeRQ), it can make sure, what speaker is the right and which is the left.

6.2.5. Reassembly

After the change of all the servos, the wires of all the servos as well as the wire of the cameras needed to fit through the hole in the neck and the spine. Finally, the loudspeaker wire should be put through the back hole of the head.

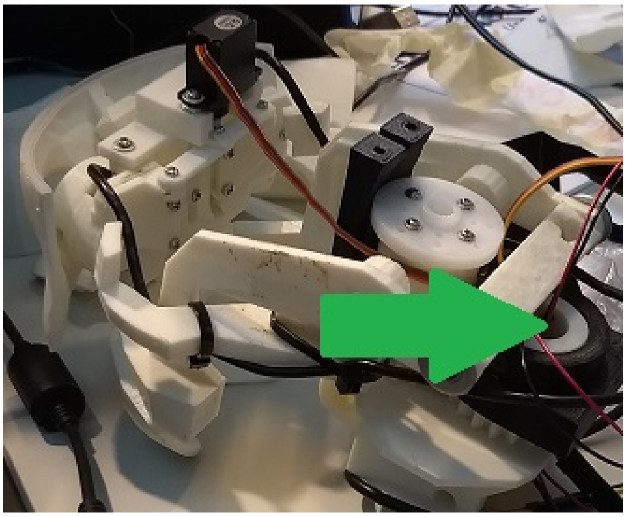


Figure 6.10: Set-up before reassembly to put the wires through

Any tension on any wire should be avoided at all times. The head was at this point reassembled, following the build guide mentioned in the beginning of this chapter.

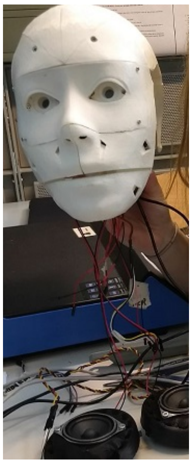


Figure 6.11: Reassembled disconnected head

Finally, all wires should go through the neck to be able to connect them to the Arduino. The head is connected to the rest of the body with a small and a large bolt.

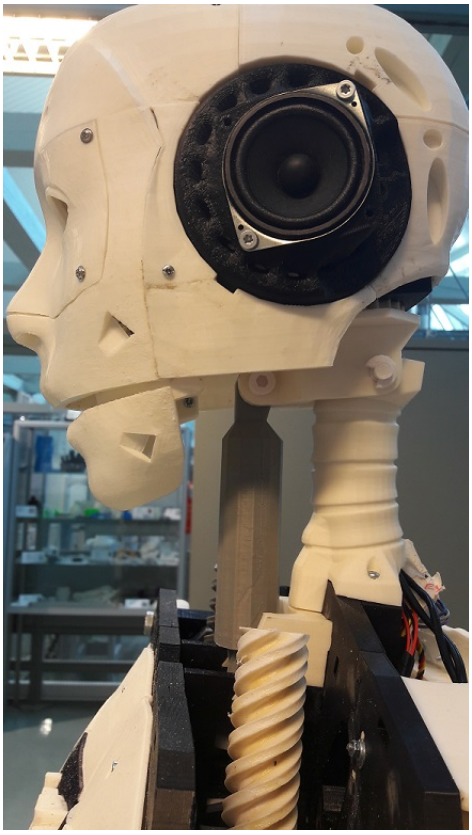


Figure 6.12: Fully Reassembled and connected head connected to the body with two bolts

6.3. Arms

The main tasks for the rebuild of the arms are replacing the servos in the wrist and improving the tension of the cables. These two tasks will be explained in this chapter.

6.3.1. Replacing the servos of the wrists

To replace the servos inside the wrist the first step is to disassemble the arm. The arm is assembled to the torso with three screws these can be disconnect and the arm can be pushed up including the bicep.

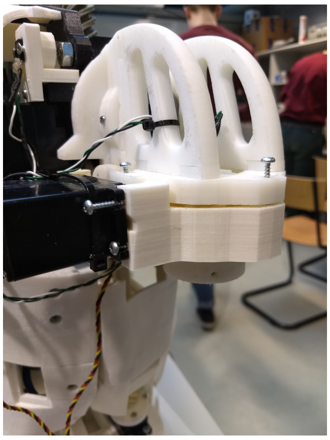


Figure 6.13: Arm removed from bicep

After the arm is taken separately the wrist assembly can be opened.

Figure 6.14: Closed wrist assembly

The assembly is open the gearing of the wrist is accessible and also the servo for the wrist is accessible on this point the servo can be replaced for a new model.

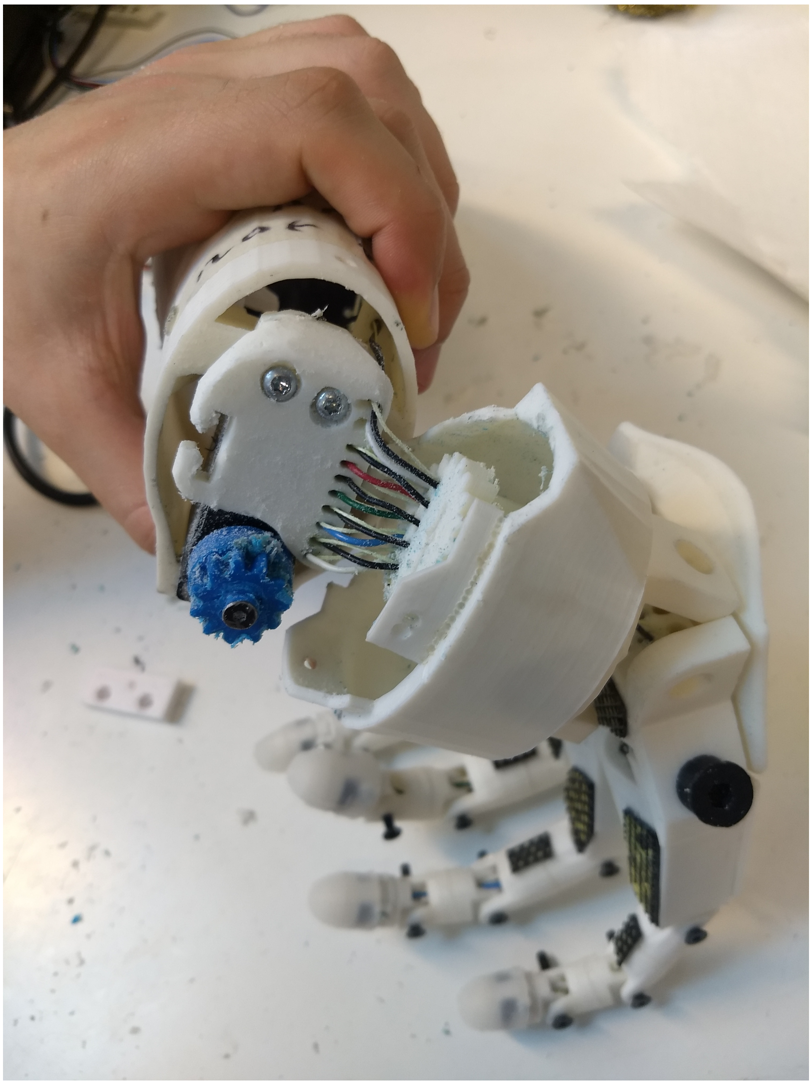


Figure 6.15: Open wrist assembly

After that the new servo is installed correctly and tested the assembly can be closed again. Then the arm can be installed on the torso again. For a detailed guide on the assembly of the arms see [Gaël Langevin, 2012].

6.3.2. Improving tension on cables

To improve the operation of the fingers, the tension on the cables can be increased. The tension on the cables is increased by putting a finger in the complete open or complete closed position and tightening one of the cables. The other cable has to have some slack on this point.

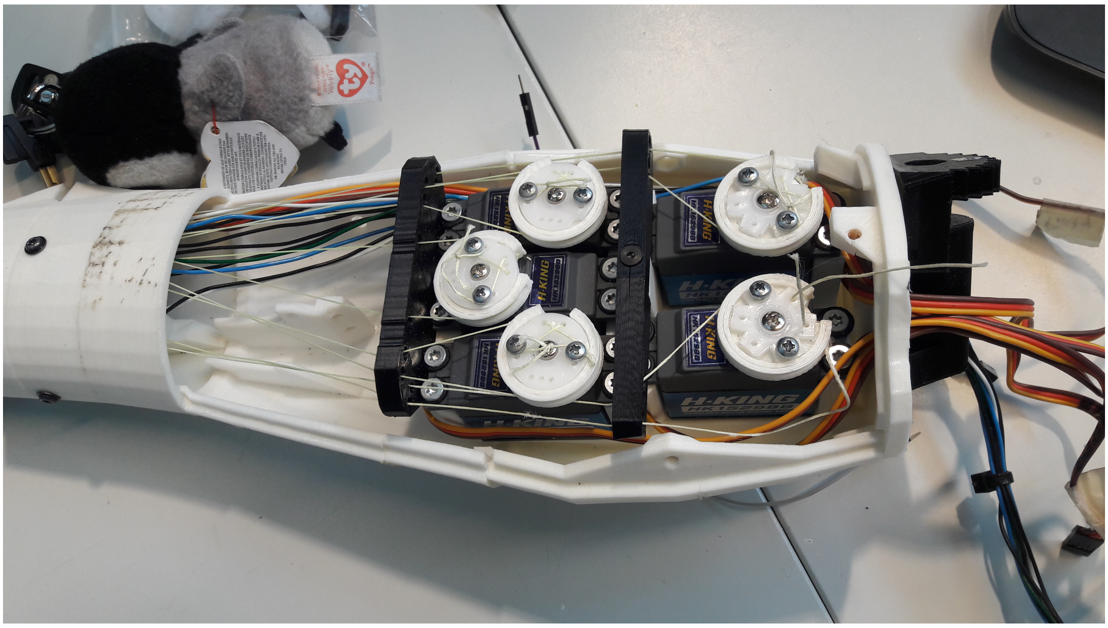


Figure 6.16: Cables fingers

For a detailed guide on the complete assembly of the cables and hands see [Gaël Langevin, 2012].

6.4. Back

The final version of the robot is controlled by a tablet, that needed to be installed in the back of the robot running MyRobotLab. Chapter 6.9 explains the installation of the software on the tablet. The starting point (without the microcontrollers) can be seen in the picture below.

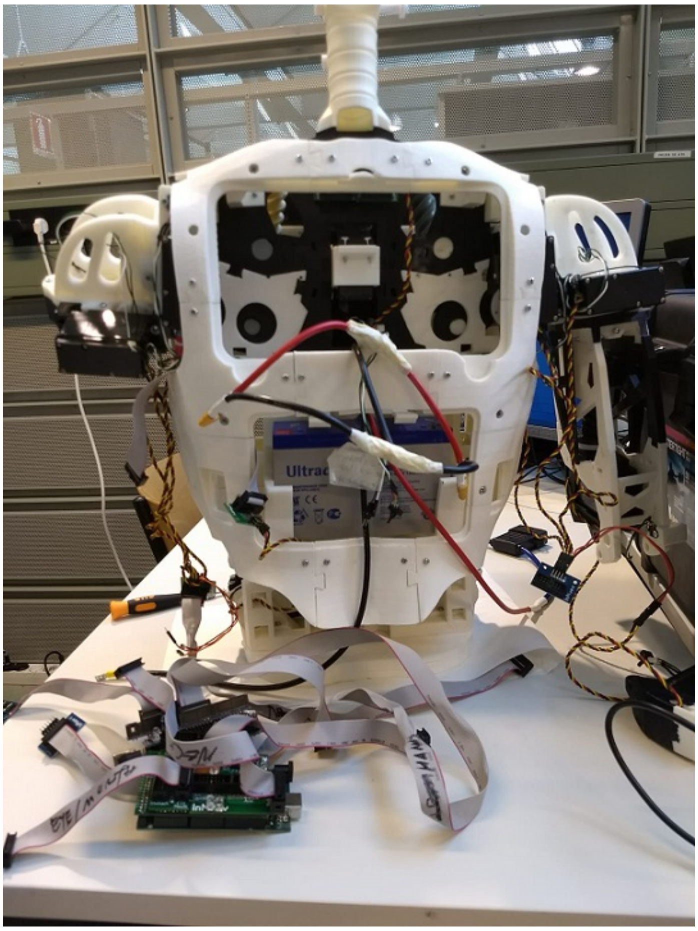


Figure 6.17: View on the back before the installation with the Arduinos taken out

After the installation of the Arduinos to the back (connected with two screws each on top), the nervoboard was installed on top of the arduinos and the connections to all servos as well as the battery made.

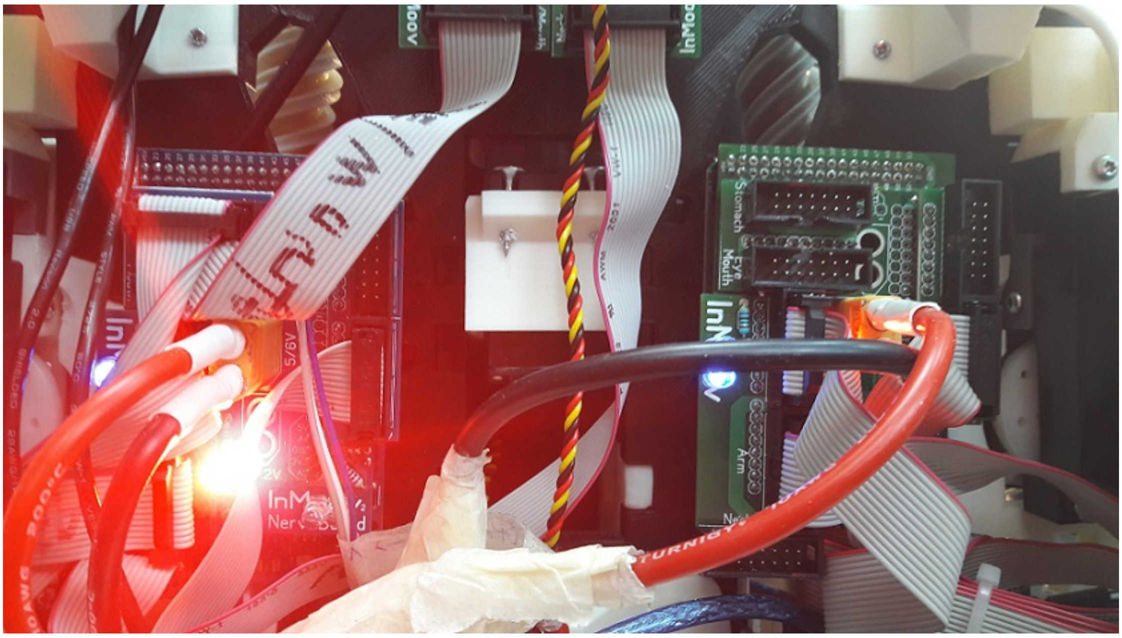


Figure 6.18: Back after all connections are made

Due to the soldering of the wires that lead to the battery, the tablet cannot be mounted directly on the side. Rather, a spacer block is needed to be designed (see fig 6.19). The new tablet is a NuVision with a screen size of 8-inch [Amazon, 2018]. The diameter is too small to fit into the back of the robot. Therefore, two supplementary holders are needed to be designed. The holders are 3D-printed parts, created using the free design web page called Tinkercad®. They are designed to guarantee stability for the device through simple application of two screws. Once fixed, the tablet will fit perfectly.

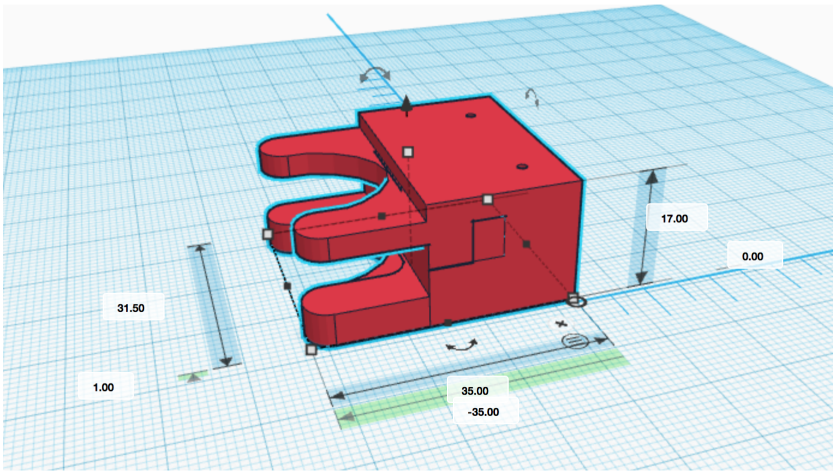


Figure 6.19: Design of the Left Tablet Holder

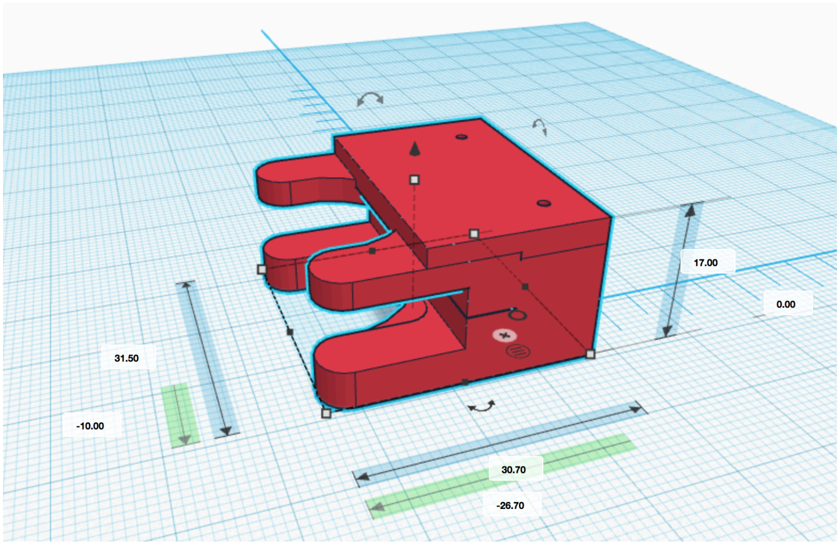


Figure 6.20: Design of the Right Tablet Holder

The design is obtained by the virtual assembling of basic geometrical shapes: cylinders and parallelepiped. In the ".stl" file the space for the screws is already done.

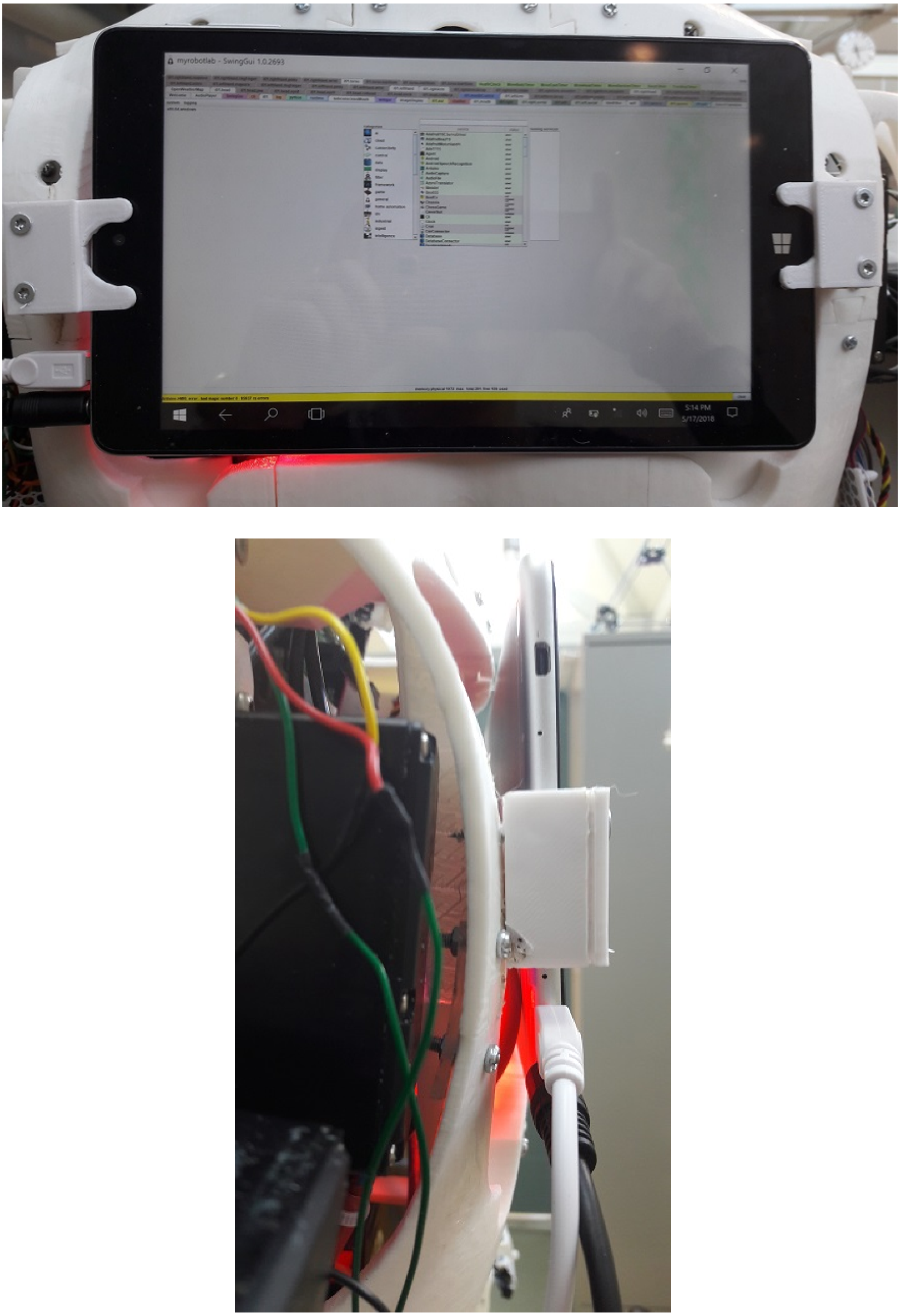


Figure 6.21: Installed tablet in the back (top) and side view(bottom)

6.5. Torso

The building of the torso consists of work on the lower stomach. This is the location of the servos, that rotates of the upper body. This rotation is made by two servos, which spin in reverse directions. This means, that one of the servos need to be hacked to change the rotation. This process consist of switching the + and - on the DC motor. Also, both DC motors of the servos run on the same motion controller board (for more information of servos inform 3.1.3). For the detailed guide see [Gaël Langevin, 2012].

6.5.1. Hacking servos

The first step is to open the servo and remove the motion-controller board (see figure: 6.22).In the end, only one servo shall still remain with his motion-controller board. This is because both the DC motors will be run by the same motion-controller board.

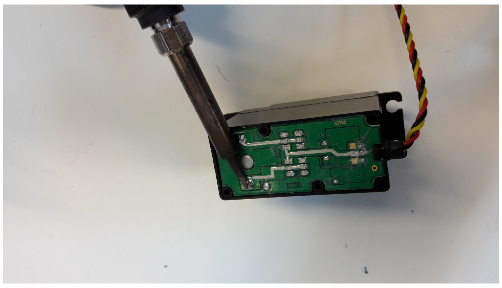


Figure 6.22: Removing motion-controller board

When the board is removed, the potentiometer can be taken out. Only the servo including the motion-controller board will be used.

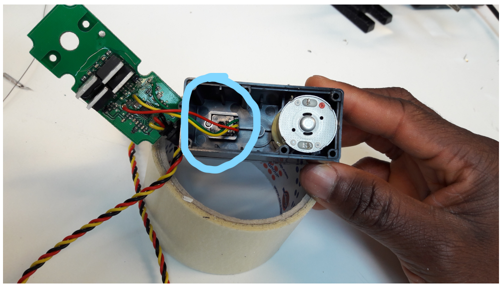


Figure 6.23: Potentiometer

For the servo without the motion-controller board and potentiometer, the poles of the DC motor can be switched (see figure: 6.24). After this process the motor will rotate in the inverted direction.

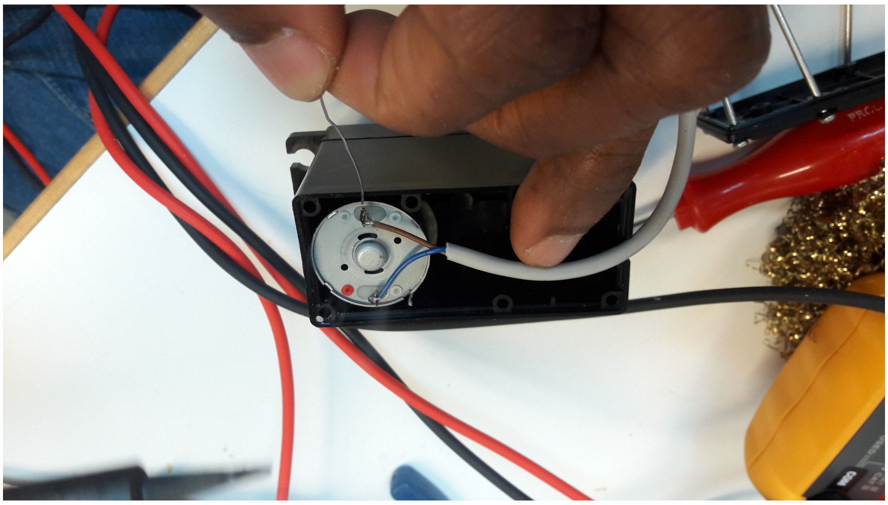


Figure 6.24: Switching poles DC motor

The last step is to make the rotation of 360 degrees possible by manipulating the gears of both servos. This process can be seen in figure 6.25. The 360 degree turn is realised by cutting of the mechanical stopper on the gear.

Figure 6.25: Cutting the gear

When all the parts are correctly assembled, the end result will look like the following figure. Afterwards, the covers can be put on again and the servos can be used for the desired movement.

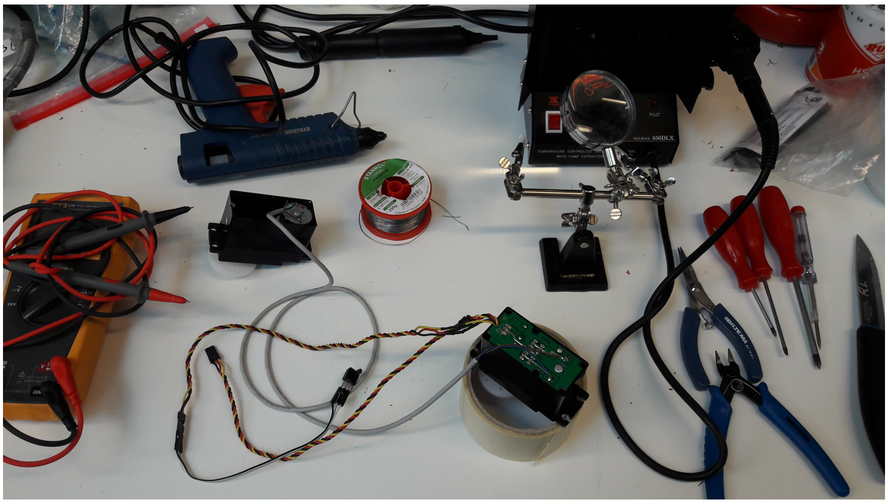


Figure 6.26: Hacking completed

6.5.2. Installation servos

The hacked servos can now be installed on the lower stomach assembly (see figure: 6.27).

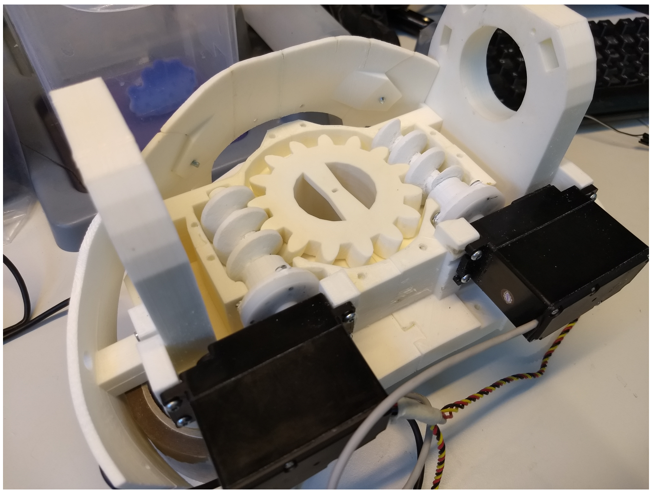


Figure 6.27: Servos installed on the assembly

This assembly can be installed on the rest of torso and is ready for use. Also, just as all the other completed assemblies this complete assembly is tested.

6.6. Legs

6.6.1. Checking for breaks

Once all the parts are printed, the defects due to the printing must be fixed. It is necessary to study carefully every piece checking for any holes, cracks and rifts. Any small fissure can trigger to the rupture of the piece, due to any minimum shear stress that the piece must withstand, which implies to fix it again probably when the piece is assembled to the robot, and if the break is critical, another new piece would be needed to print.

The following picture shows a crack in the connection between the torso and the legs.

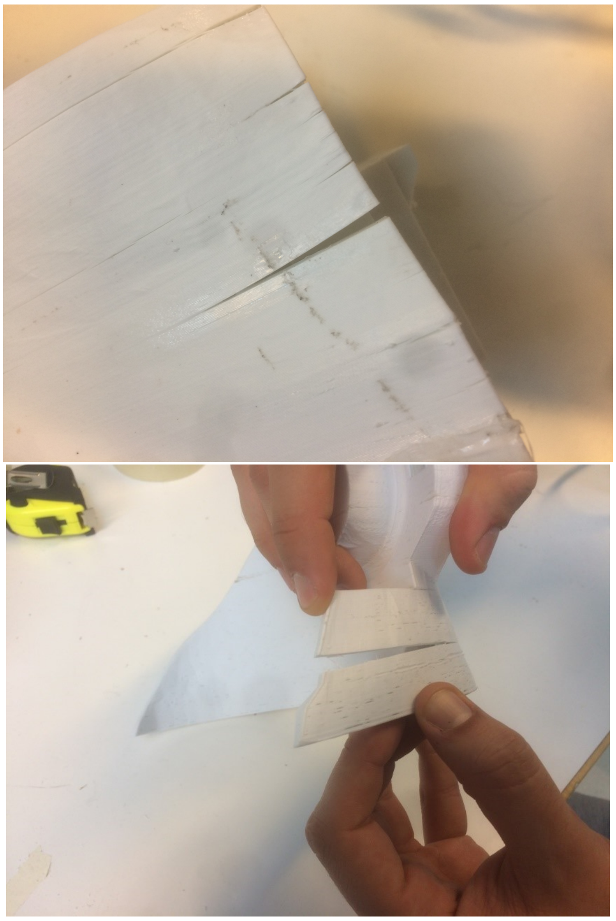


Figure 6.28: Cracks in the printed part (here the back of the foot)

6.6.2. Workover

Using the laboratory tools (see chapter 5.6 (sander and soldering iron)), it will first be necessary to weld all the cracks, at the lowest temperature that the machine allows, so that, melting the joint, the material will seal the break.

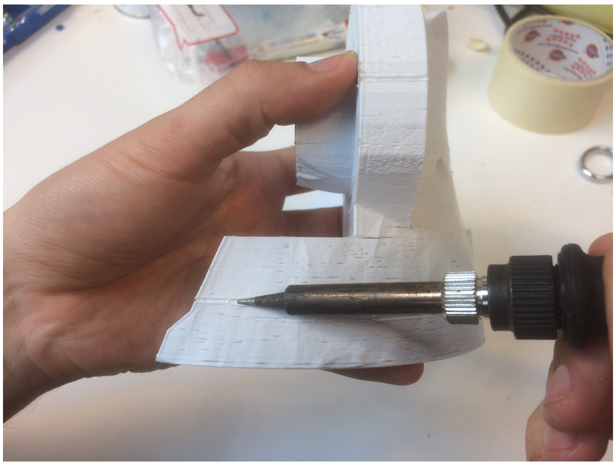


Figure 6.29: Repairing a small crack with a soldering machine

If the crack is too large, as shown in figure 6.29, and both walls are far apart, a piece of ABS filament will be added as the necessary material with the welding machine.

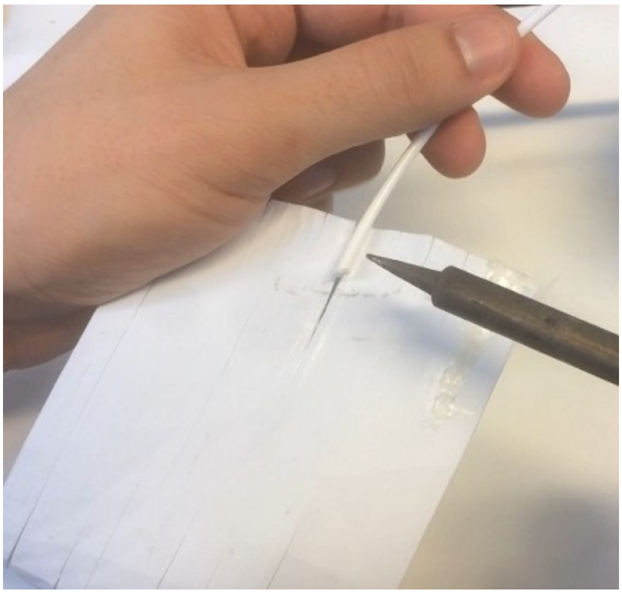


Figure 6.30: Repairing a large crack with a soldering machine

It is even recommended to add excessively, until it is completely covered, taking care not to keep the material in contact with the punch for a long time, otherwise the plastic will start to burn and will acquire an unwanted brown color. This is a process that requires practice, so each step will be shown in details in the following paragraphs.

- First, holding the piece firmly on a comfortable surface to work, begin by melting the tip of the filament keeping it almost in contact with the piece:

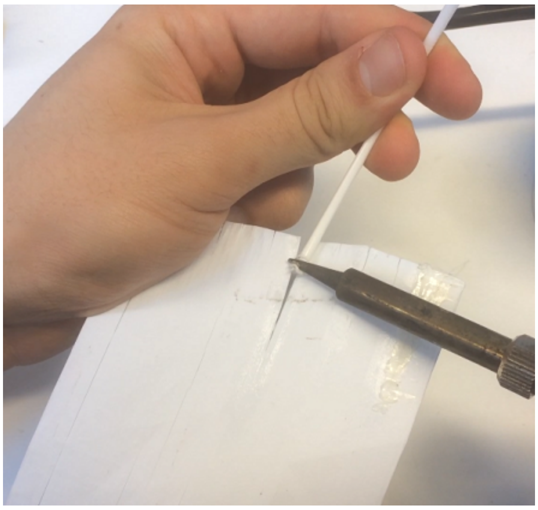


Figure 6.31: Repairing a large crack with a soldering machine(1)

- Once the material is melting, try to push it towards the crack, with an inclination of the punch of 90 degrees with respect to the filament. In this way, the crack will seal more evenly:

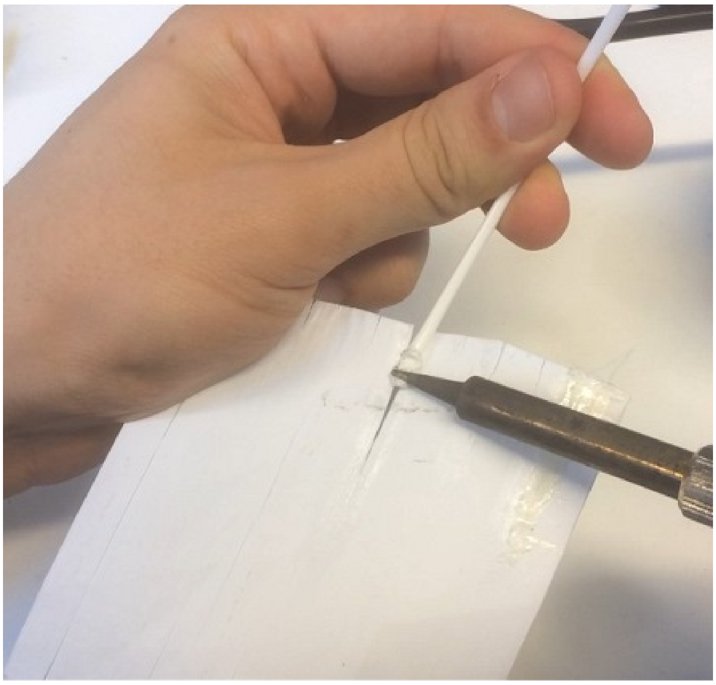


Figure 6.32: Repairing a large crack with a soldering machine(2)

- Furthermore, try that the punch is in contact with both edges of the fissure, because it is necessary that these are mixed with the new material, and thus ensure the sealing, otherwise the material will not adhere properly:

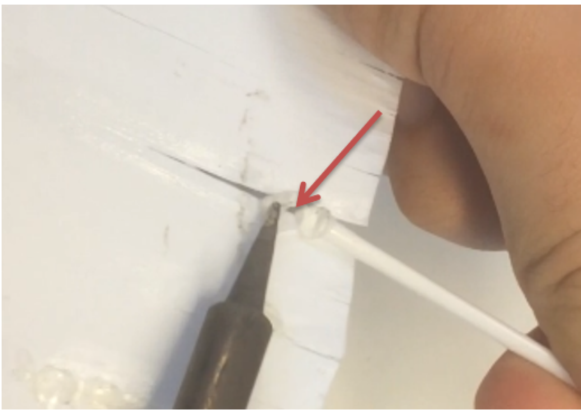


Figure 6.33: Repairing a large crack with a soldering machine(3)

- Continue to melt more filament material trying to add excess plastic, which will be easy to remove later, until the crack has been completely sealed:

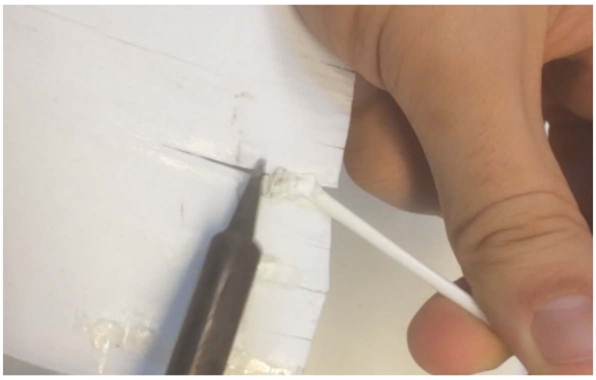


Figure 6.34: Repairing a large crack with a soldering machine(4)

Once the crack has been sealed, with the sander and choosing a tool to smooth surfaces, the excess plastic will be removed until there are hardly any marks of previous ruptures, and giving the piece a better appearance.

Following the procedure described and being careful and patient, the result can be quite acceptable. Below are more examples of other parts that have been repaired, and the final result.

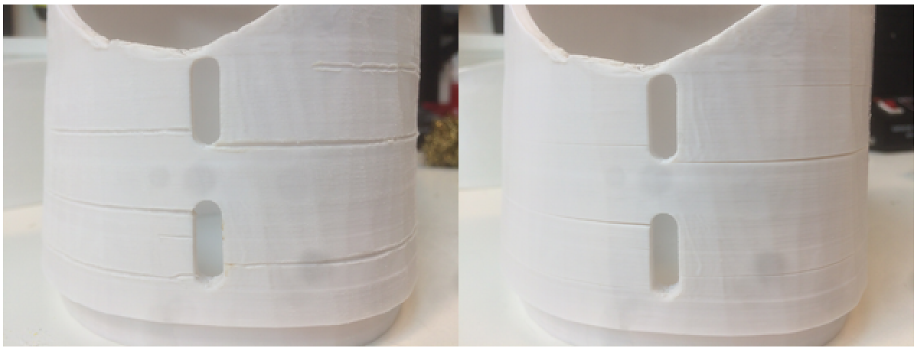


Figure 6.35: Before (right) and after the soldering (left)

One of the heels has been one of the most critical pieces for the printer as it showed various cracks with after every try to print.

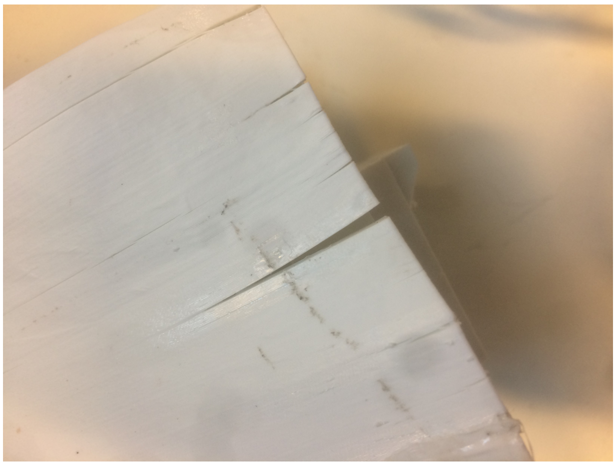


Figure 6.36: Difficult part of the foot to print

After soldering, the best result is shown below.

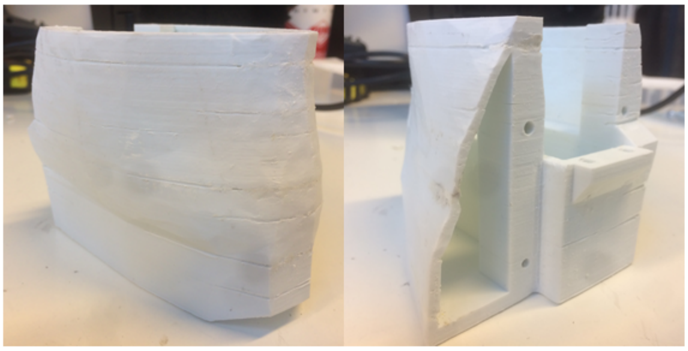


Figure 6.37: Result of the reparation

As it can be seen, the repair is quite acceptable and the appearance is much better without visible cracks, in addition to reinforcing the piece and avoiding subsequent breaks.

In case, that any part breaks completely and the rupture leads to the total division of two or more parts, there is another possibility to fix the damage. By using a plastic solvent such as acetone and wetting both sides separately with a brush, the union is simple and fast. Subsequently, if desired, with the soldering iron the connection can be melted carefully to reinforce and secure the joint.

Figure 6.38: Acetone used to repair damaged parts

6.6.3. Collection

With all the parts repaired, it is advisable to order them accordingly to their parts of the body. These include: foot, ankle, tibia, knee and thigh.
Each section will need the required screws with their diameters, lengths, or thread type, that is, if it is necessary to use a nut and washer or not. Creating a simple table in Excel of each part, with the screws and their characteristics simplify the assembly. Additionally, interruptions will be avoided due to missing tools, which will lead to unnecessary consumption of time to acquire new items.

Figure 6.39: Overview of every part needed to build the legs

Make sure everything needed to continue with the assembly is readily available. This includes parts, screws, pins, nuts and washers. If desired, with paper tape and a marker a number and name the pieces can be applied, to facilitate the location and disposal of them.

6.7. Assembling the legs

When every part is printed and adjusted (cracks repaired, edges sanded) the final assembly can begin. Start the guide from the feet, it will be completed with the thighs, which connect the legs to the rest of the body. References to the necessary pieces will be carried out using the names that the InMoov website uses for each one in the body parts library (check http://inmoov.fr/inmoov-stl-parts-viewer/). Thus, the reproduction will be easier. The following part describes the assembly of the right leg. It can be done in the same way for the left.

6.7.1. Foot and Ankle

In order to assemble this part, all the items mentioned as "Legs-Ankle" in the drop-down list of the library are required.

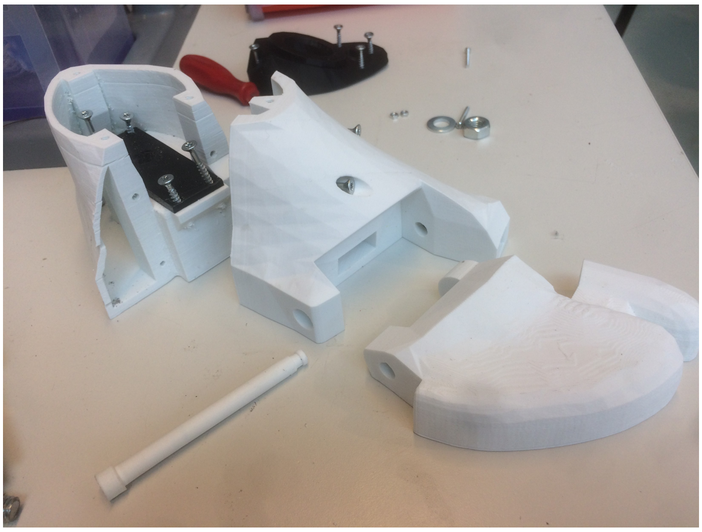


Figure 6.40: Layout of the parts and screws

The first assemble will be the pieces of the heel, that will hold the rod, which goes through the rest of the leg, until the union with the torso. Therefore, this union is the most important, since it will bear a considerable weight. Correcting the holes with the drill if necessary, *"AnkBackFootV1"* will be joined with *"FootBoltHolder"*, with wooden screws and insure a firm connection.

Next, the "AnkMidleFootV1" (see fig. 6.40: center part) will be added to the complex, taking into account that several joints are necessary from inside the piece:

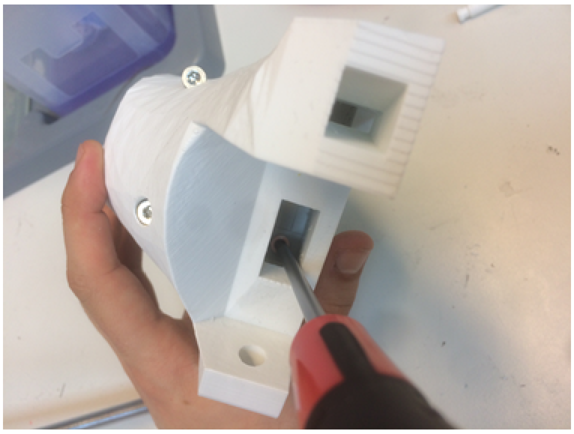


Figure 6.41: Screwing *"AnkMidleFootV1"* with the last parts

After that, with the *"AnkToesFootV2"* and through the pin *"AnkToesBoltV1"* the toes will be joined to the previous assembly:

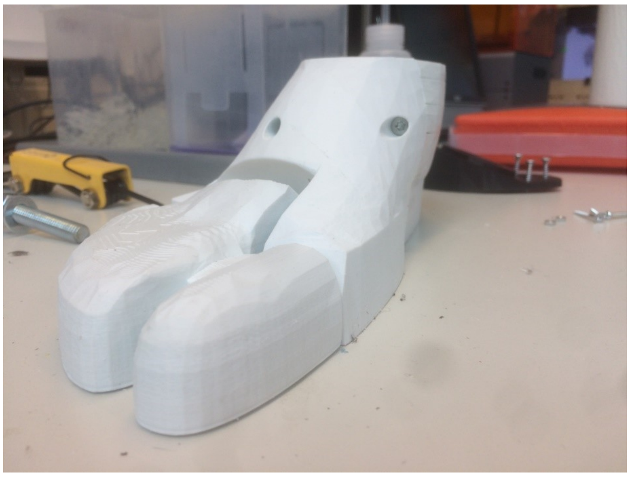


Figure 6.42: Adding the toes to the foot

A M8-rod of 1 meter length, is neccessary to hold the structure from the inside of the legs. It should be implemented to the piece called *"FootBoltHolder"* by using two nuts and washers from the top and the bottom, as shown in the following image:

Figure 6.43: Implementing the rod to *"FootBoltHolder"*

The black part of the following image, *"AnkleBaseV1"*, is screwed to the top of the foot, with four wooden screws:

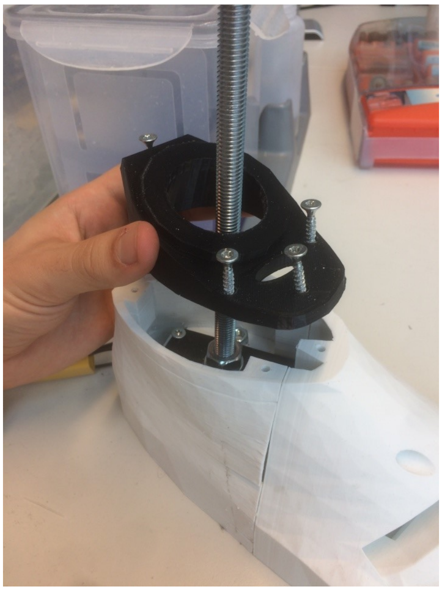


Figure 6.44: Screwing *"AnkleBaseV1"* to the foot

In the next step, clip the rotating piece *"LeftAnkHolderV1"* the foot to finish:

Figure 6.45: Assembled foot and ankle

6.7.2. Tibia

The necessary elements for the tibia are those shown in the following figure:

Figure 6.46: Required parts for the tibia

The *"LeftAnkHolderV1"* (upper part of the ankle) will be connected to the *"AnkpartV1"* by using wooden screws:

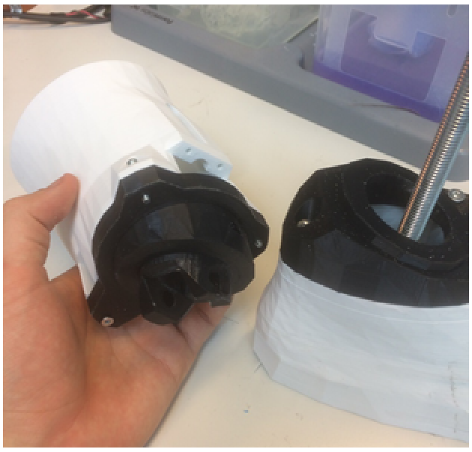


Figure 6.47: screwing the *"AnkPartV1"* to *"AnkleBaseV1"*

The complex needs to be clicked onto the finished foot:

Figure 6.48: Union between the foot and *"AnkleBaseV1"*

The pieces, that are assembled so far, are bolted. The following ones up to the knee are going be glued with acetone and the soldering iron at the end of the assembly. Until then, paper tape is sufficient.

Now, the parts *"TibiaLowS2RightV1"* and *"TibiaLowS1RightV1"*,

Figure 6.49: *"TibiaLowS2RightV1"* to *"TibiaLowS1RightV1"*

are added to the assembly:

Figure 6.50: Tibia mounted on the ankle

To have a secure connection between the upper and lower part of the tibia, some points (2 or 3) inside can be melted using the soldering machine.

The tibia completes by adding the upper part of it called *"TibiaHighRightV1"* This simply fits into the union of the previous two parts without any screws.Paper tape ensures the further assembly:

Figure 6.51: Assembly of the foot and tibia

6.7.3. Knee

The printing models can be found in the drop-down list of the library in the *"Legs-Knee"*-section.

The *"KneeLowRightV3"* will screwed to *"TibiaHighRightV1"* using four metal screws, nuts, and washers.

Figure 6.52: Start of the knee

The holes, made in the internal supports of the piece *"TibiaHighRightV1"*, have been made with a 10mm bit. Thus ensures, that the nut and washer are arranged properly.

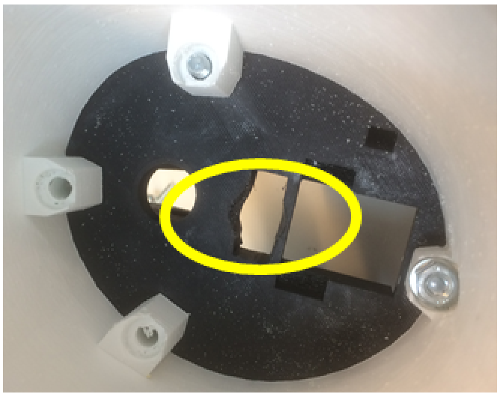


Figure 6.53: Sanding needed inside the knee for the rot

As shown in the picture 6.53, it is necessary to eliminate material to make the M8-rot fit through. The part is ready, to put on top of the already assembled part. Furthermore, the right part of the model *"KneeClampRightV1"* will be added as well as the bolts(*"KneeSmallBoltsV1"*):

Figure 6.54: Knee connection with the *"KneeClampRightV1"* and the *"KneeSmallBoltsV1"*

Afterwards, the pieces *"KneeHighRightV3"* and *"ThighLowRightV3"* from the drop-down list *"Legs-Thigh"* will be screwed in. Then, the left part of the model *"KneeClampRightV1"* goes through the slot, and with the bolts named *"KneeBoltsV1"* the previous set will be assembled:

Figure 6.55: Assembling the knee to the thigh

The zoomed-in hole needs to be enlarged with a 10mm in diameter bit to make it fit.

The result so far is as follows:

Figure 6.56: Finished complex of foot, ankle, tibia and knee

6.7.4. Thigh

The legs are almost done. Now, the remaining elements are in the *"Legs-Thigh"* drop-down list.

The assembly continues mounting the part *"ThighMidRightV1"* on top of the current build:

Figure 6.57: Assembly of *"ThighMidRightV1"*

Additionally, the part *"ThighHighRightV1"* is placed on top:

Figure 6.58: Assembly of *"ThighHighRightV1"*

The piece *"ThighHighTempV1"* works as a connection between the legs and the torso:

Figure 6.59: Assembly of *"ThighHighTempV1"*

As previously stated, it is helpful to add 2-3 soldering points to all these parts, to ensure the joint before sticking them completely.

Next, the piece *"LegHolderRight"* (black in figure 6.60) serves to ensure that the rod is in its correct position. It will be placed simply supported on *"ThighHighRightV1"* and fixed to the rod by using of two nuts and washers, one above and one below:

Figure 6.60: Assembly of *"LegHolderRight"*

Finally, the piece *"ThighSideAccessRightV1"* serves as reinforcement of *"ThighHigh-TempV1"* due to its fragility. In order to realize the support, two holes will be made in *"ThighHighTempV1"*, so that the reinforcement rest on *"ThighHighRightV1"*:

Figure 6.61: Installation the supporting parts

Having done the support, the assembly of the right leg is finished. The following picture shows both assembled legs, tht are ready to be conneted to the rest of the body robot:

Figure 6.62: Fully assembled legs

6.8. Holder design

For secure transport and operation of InMoov, a good holder is necessary. The holder is designed to be used with a specific trolley. The holder will use the rots in the legs of InMoov and the horizontal tube on the trolley. The completed holder can be seen in figure: 6.63.

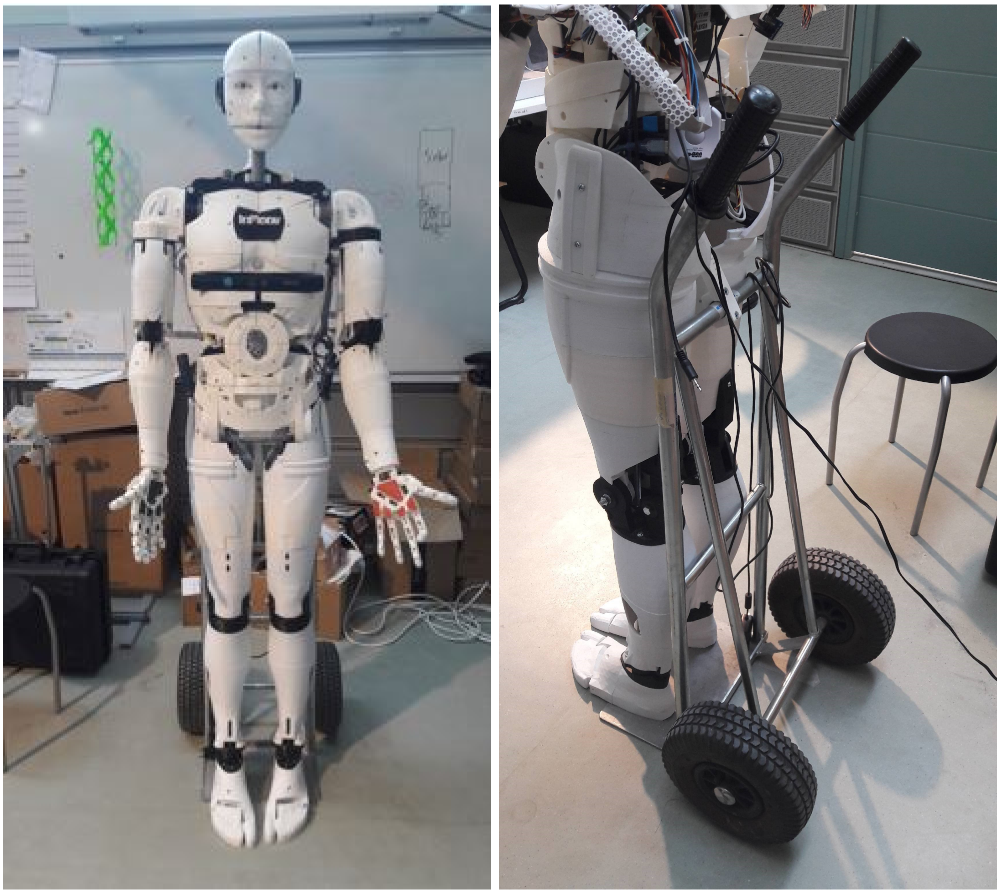


Figure 6.63: Completed holder design

The design of the holder can be found in the figure below (figure 6.64). The design was made in the CAD software Tinkercad. The design consists of three parts one part goes in between the legs and the other two parts are for the connection between the frame of the trolley and the first part.

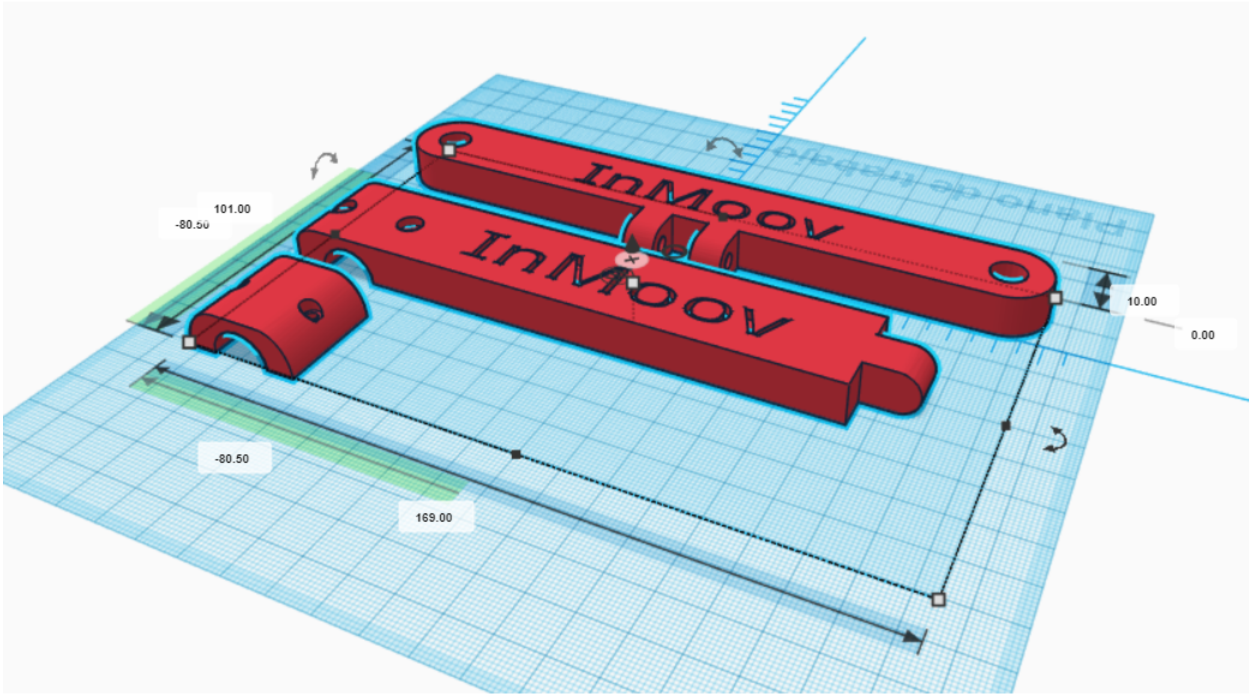


Figure 6.64: Design of the holder (Tinkercad)

In the close ups of the design the exact location of all the parts can be seen and how they ensure a sufficient connection between the robot and the trolley. In this close up the three different parts can also be easily distinguished (see figure 6.65).

Figure 6.65: Close up holder

6.9. Software

To setup the software some steps need to be taken. These consist of installing, configuring the program with the help of the config files, inserting your own personal programs and debugging. This chapter will go over the complete setup of MyRobotLab.

6.9.1. Installation

First the installation of the software has a good explanation on the website [Gaël Langevin, 2012] this guide is added in appendix C . This guide helps the user through the regular setup and gives the user to basic tools to start with Myrobotlab. If further functionality is needed extra steps need to be taken.

6.9.2. Drivers

For the drivers there is one special note that needs to be made. This is that both the online and offline drivers need to be installed to get the right functionality out of Myrobotlab. These consist of 32bit versions and 64bit versions for the offline version and one version for the online version.

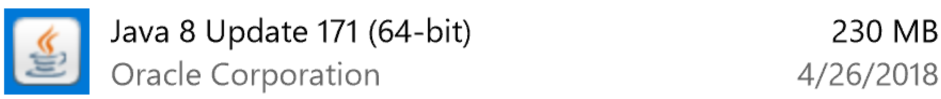


Figure 6.66: Java 8 in the programs list.

6.9.3. Kinect installation

The normal setup guide does not give a clear driver installation guide for the kinect. For the kinect it is necessary to install the drivers to run the Kinect attached on InMoov. Both Kinect and Windows are Microsoft products. In order to use Kinect, there are some installation needed: the Runtime file of Kinect for Windows updated to the latest version, the Software Development Kit of Kinect (SDK), the updated Developer Toolkit, and finally the pack for the Speech Recognition.

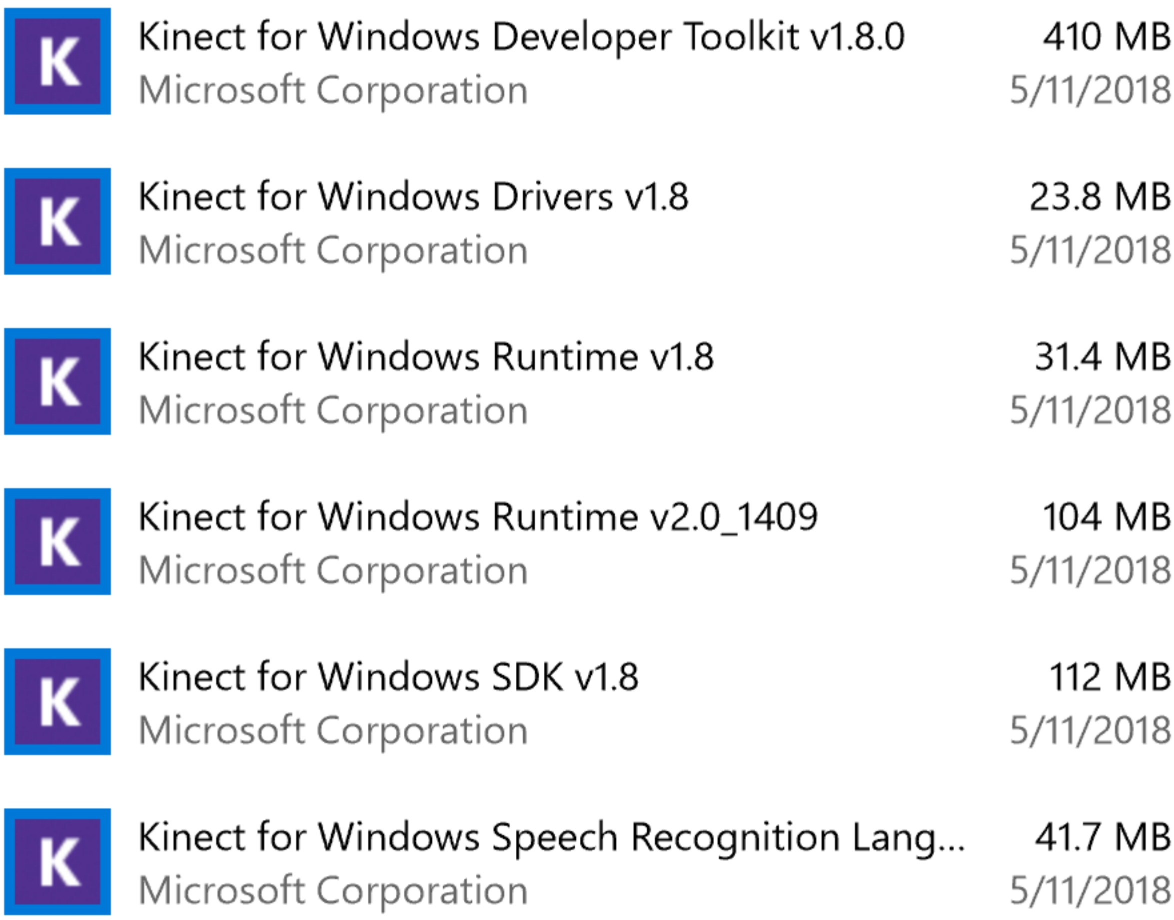


Figure 6.67: Lists of all the Kinect items installed on the tablet.

After the installation of these drivers the kinect is functional and ready for use in MyRobotLab.

6.9.4. Configuration files

After completing the installation of Myrobotlab and the different drivers the configuration files can be configured correctly. These configuration files can be found in: C:\mrl\myrobotlab.1.0.2693\InMoov\config
These files that are in this folder are:

_InMoov	16-5-2018 14:37	CONFIG-bestand	2 kB
InMoovLife	11-5-2018 16:16	CONFIG-bestand	1 kB
service_0_WebGui	11-5-2018 16:16	CONFIG-bestand	1 kB
service_4_Ear	11-5-2018 16:16	CONFIG-bestand	1 kB
service_5_Mouth	11-5-2018 16:16	CONFIG-bestand	1 kB
service_6_Arduino	14-5-2018 11:21	CONFIG-bestand	1 kB
service_8_NervoBoardRelay	11-5-2018 16:16	CONFIG-bestand	1 kB
service_9_neoPixel	11-5-2018 16:16	CONFIG-bestand	1 kB
service_A_Chatbot	11-5-2018 16:16	CONFIG-bestand	1 kB
service_C_Pir	14-5-2018 11:21	CONFIG-bestand	1 kB
service_D_OpenCv	14-5-2018 11:22	CONFIG-bestand	1 kB
service_E_OpenNI	14-5-2018 11:22	CONFIG-bestand	1 kB
service_F_VirtualInmoov	11-5-2018 16:16	CONFIG-bestand	1 kB
service_G_Translator	11-5-2018 16:16	CONFIG-bestand	1 kB
service_H_OpenWeatherMap	11-5-2018 16:16	CONFIG-bestand	1 kB
service_I_UltraSonicSensor	11-5-2018 16:16	CONFIG-bestand	1 kB
skeleton_eyeLids	11-5-2018 16:16	CONFIG-bestand	2 kB
skeleton_head	16-5-2018 14:42	CONFIG-bestand	2 kB
skeleton_leftArm	14-5-2018 11:31	CONFIG-bestand	2 kB
skeleton_leftHand	14-5-2018 11:31	CONFIG-bestand	2 kB
skeleton_rightArm	14-5-2018 11:35	CONFIG-bestand	2 kB
skeleton_rightHand	14-5-2018 11:38	CONFIG-bestand	2 kB
skeleton_torso	14-5-2018 11:38	CONFIG-bestand	1 kB

Figure 6.68: Configuration files

The different configuration files are separated in two different categories: service files and skeleton files. The service files give access to different components and functionalists from InMoov and the skeleton files give access to all the servo related settings. For every configuration file a small description is given what can be changed in this specific file. These descriptions can be found in appendix D. For the limits of this specific InMoov a table is present to use these limits are tested and ready for use (see appendix E).

6.9.5. Standard voice commands and functions

When Myrobotlab is started various voices commands and functions are directly usable (if they are activated in the configuration files). A large array of available voice commands commands is given in appendix F.

One voice command needs to special attention the track command this one starts the eye tracking. To use this command correctly, also open the tab of OpenCV in Myrobotlab to set the point to follow.

6.9.6. Custom code

To write an own custom code go to:

C:\mrl\myrobotlab.1.0.2693**\InMoov\custom**

In this directory the custom.py can be found. This custom file can hold all the custom code desired by the user. Also multiple files can be added here to be run. In the directory:

C:\mrl\myrobotlab.1.0.2693**\InMoov**

The main InMoov code can be found inside of the InMoov.py file. This python file is the complete Python file of the whole of the myrobotlab software. The modular setup of InMoov can also be seen here. Most importantly for writing custom code is the last line:

execfile(RuningFolder+'custom/InMoov_custom.py')

This gives the executing command for the costume file. If there is a need to run multiple custom files they can be added here or a main custom file with different executions can be made. To write your own custom code a couple of different commands can be used to ensure smooth coding and operating. Various different commands are given in appendix G. Most of the commands are pretty self explanatory.

7. Discussion and conclusion

7.1. Discussion

The main goals, that were set in the beginning, could be achieved to a functional unoptimised extend. A significant amount of time was spent on ordering new servos and delivery times made a big impact on the planning of the project and led to the postponement. The last major factor of time loss was damage that arose during the work on critical parts (e.g. the back of the foot). The programming of InMoov went according to the planning, but the start of that task needed to be postponed due to other programming, that still needed to be completed. The fingers and especially the fishing braids, that make the movement of the fingers possible, is not optimal. This tasks is not fully completed, because of the decision to invest time on different tasks. The secondary objectives where not yet achieved, due to a lack of time (VR, Omron, Leapmotion and Beamer). The secondary objectives will be kept in place to ensure later improvements can be made by future work groups. The reason for the time issue on the objectives is caused by the other tasks taking more time then planned (fixing, maintenance and disassembling). To conclude, starting to implement innovative and complex devices such as VR, Omron and Leapmotion on InMoov just a few weeks before the deadline of the project would have meant abandon some other more important tasks.

R3-DFin prioritised, to achieve the main goal and obtain the best result from the basic targets, sure that in the close future, there will be enough time to apply the devices to InMoov.

7.2. Conclusion

To sum up, all the main goals of the project have been achieved. Having a look back to the milestones, it is possible to claim, that all the robot movements (first milestone after midterm report) are installed and ready to be used (limits approved). The robots in its entirety is completed, with all the pieces and components needed and tested (second milestone) and also the software completion (third one) has been done by writing a script for the presentation of Technobothnia.

To conclude, the main objective is reached successfully, but the secondary objectives still need work in the future. Also, future optimisation for the fingers and software are necessary to improve the functionality.

7.3. View into the future

For the InMoov development that goes on during the coming years. Various different goals can be set:

- To guarantee more interaction between people and InMoov, one option is to implement Virtual Reality device on the robot.
- Linking InMoov and Omron could be an opportunity for Technobothnia to be represented and present itself using a more complex combination of technologies.
- The beamer can be added to InMoov, to make the robot's explanations more clear with the help of images and videos.
- An improvement of the quality of the wire tension in the arm of the robot should be done to avoid constrained movement. The best option would be a total replacement of the current fishing lines.
- Conductive foam for the fingers is not working properly, so it has to be changed in the future.
- The Kinect is working in the right way, but it needs to be optimised, because more than once the robot has little problems representing the mirror-movements of the person standing in front of him. The movements are correct, but the angles and the limits are not always followed correctly especially the movement of the omoplate.
- Attach a 12 volt battery to make the robot completely independent from the power sockets on the wall.
- The basic programming code of the presentation of Technobothnia is completed. That is why R3-DFin recommends to improve the code and extend it to a full presentation.
- Implementing new type of sensor in addition to the already existing ones, for example ultrasound sensors in complementary action with the Kinect and the

eye-cameras. Using the PIR sensor can be an option too. Until now, it has a high sensitivity difficult to use as it recognises people

- Including switches to shut off power to the arduinos and other components from the batteries

8. Sources

8.1. Research sources

[Jorgen Hansen, 2017] http://europeanprojectsemester.eu/info/Introduction; called 22/03/2018

[Gaël Langevin, 2012] http://inmoov.fr/; called 22/03/2018

[Gaël Langevin, 2012] http://inmoov.fr/activity/; called 21/03/2018

[Myrobotlab, 2011] http://myrobotlab.org/; called 21/03/2018

[Google Group, 2013] https://groups.google.com/forum/#!forum/inmoov; called 21/03/2018

[Bernard Parent,2008] http://www.worldsalaries.org/engineer.shtml; called 01/05/2018

[Il Sole 24 Ore,2013] http://www.ilsole24ore.com/pdf2010/SoleOnLine5/_Oggetti_Correlati/Doc energia-elettrica-gas-naturale.pdf; called 05/05/2018

[Harriet Taylor, 16/03/2016] https://www.cnbc.com/2016/03/16/could-you-fall-in-love-with-this-robot.html; called 30/04/2018

[Oscar Raymundo, 2016] https://www.macworld.com/article/3045299/robots/meet-sophia-the-female-humanoid-robot-and-newest-sxsw-celebrity.html; called 01/04/2018

[Boston Dynamics, 2018] https://www.bostondynamics.com/atlas; called 10/05/2018

[Jason Ford, 22/11/2000] https://www.theengineer.co.uk/issues/november-2000-online/two-legs-good/#ixzz1TF8K9jyc; called 01/04/2018

[Alok Jha, 2004] https://www.theguardian.com/science/2004/feb/17/sciencenews.uk; called 01/04/2018

[Wikipedia, 2018] https://de.wikipedia.org/wiki/Arduino_(Plattform); called 27/03/2018

[Arduino Website, 2018] https://www.arduino.cc/en/Guide/Introduction; called 27/03/2018

[Arduino Website, 2018] https://store.arduino.cc/arduino-mega-2560-rev3; called 27/03/2018

[Howard Eglowstein, 2012] https://www.sciencebuddies.org/science-fair-projects/references/introduction-to-servo-motors; called 31/03/2018

[Gaël Langevin, 2012] http://inmoov.fr/default-hardware-map/; called 31/03/2018

[Wikipedia, 2018] https://en.wikipedia.org/wiki/Kinect; called 01/04/18

[Greg Borenstein, 2012] http://www.hmangas.com/Electronica/Datasheets/Arduino/LIBROS-%20Y%20MANUALES/[Making.Things.See(2012.01)].Greg.Borenstein.pdf; called 01/04/18

[Wikipedia, 2018] https://en.wikipedia.org/wiki/Ultimaker; called 15/02/2018

[Gaël Langevin, 2012] http://inmoov.fr/mid-stomach/;called 17/05/2018

[Gaël Langevin, 2012] http://inmoov.fr/hand-and-forarm/; called 17/05/2018

[Amazon, 2018] https://www.amazon.com/NuVision-Touchscreen-x5-Z8300-Quad-Core-Processor/dp/B01MYZEPGP; called 17/04/2018

[Gaël Langevin, 2012] http://inmoov.fr/how-to-start-myrobotlab/; called 16/5/2018

8.2. Sources pictures

[1] http://inmoov.fr/wp-content/uploads/2015/06/gael-langevin.jpg; called 22/03/2018

[2] http://inmoov.fr/wp-content/uploads/2015/01/Nervo-Board-300x300.jpg; called 23/03/2018

[3] https://avatars1.githubusercontent.com/u/6167429?s=280&v=4; called 23/03/2018

[4] https://www.fanuc.eu/ch/en/robots; called 30/04/2018

[5] https://assets.entrepreneur.com/content/3x2/1300/20171121194629-toyota-robot-video.jpeg?width=700&crop=2:1; called 30/04/2018

[6] http://robohub.org/three-concerns-about-granting-citizenship-to-robot-sophia/ ; called 30/04/2018

[7] https://media.wired.com/photos/5a0e13169639c5682ccdf3b2/master/w_799,c_limit/Atlas-FinalArt.jpg; called 30/04/2018

[8] https://en.wikipedia.org/wiki/ASIMO#/media/File:ASIMO_4.28.11.jpg ; called 14/5/2018

[9] https://reprapworld.nl/images/default/dynamic/products/originals/prod_7NDF9j.jpg; called 27/03/2018

[10] http://inmoov.fr/wp-content/uploads/2013/12/Map-ConnectionsV1.jpg; called 27/03/2018

[11] http://www.inmoov.fr/wp-content/uploads/2013/12/default-hardware-map4.jpg; called 27/03/2018

[12] http://2.bp.blogspot.com/-ty_tbM2DM-Q/VQ8ZBWReZfI/AAAAAAAANEU/-ui9tDAawLEc/s1600/servo-motor.jpg; called 31/03/2018

[13] http://kamami.com/6111-thickbox_default/phk-hk15298b-high-voltage-coreless-digital-mgbb-servo-66g–20kg–016s-16272p.jpg; called 31/03/2018

[14] https://upload.wikimedia.org/wikipedia/commons/thumb/f/f6/TiemposServo.svg/330p TiemposServo.svg.png; called 31/03/2018

[15] https://i0.wp.com/electricalstudy.sarutech.com/wp-content/uploads/2013/07/servo-motor-3.png?resize=618%2C464&ssl=1; called 31/03/2018

[16] https://cdn.makeuseof.com/wp-content/uploads/2013/07/leap-670x335.jpg; called 29/03/2018

[17] https://www.ncbi.nlm.nih.gov/core/lw/2.0/html/tileshop_pmc/tileshop_pmc_inline.html?title=Click%20on%20image%20to%20zoom&p=PMC3&id=3690061_sensors-13-06380f1.jpg; called 29/03/2018

[18] https://3bonlp1aiidtbao4s10xacvn-wpengine.netdna-ssl.com/wp-content/uploads/2017/03/leap-motion-3d-motion-gesture-controller-10-large.jpg; called 30/03/2018

[19] https://upload.wikimedia.org/wikipedia/commons/thumb/6/67/Xbox-360-Kinect-Standalone.png/1920px-Xbox-360-Kinect-Standalone.png; called 01/04/18

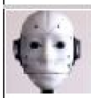
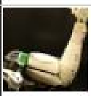
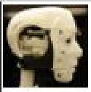
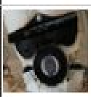
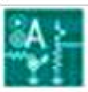
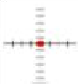
[20] https://nxeassets-ssl.xbox.com/shaxam/0201/69/73/6973bced-7796-4b85-bb06-f0279d9e1cf4.PNG?v=1#kinect-sensor-callouts-m-m.png;called 01/04/18

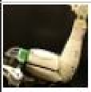
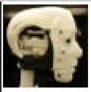
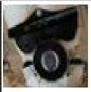
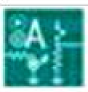
[21] https://www.mindtreatstudios.com/wp-content/uploads/2012/02/kinect_depth_image.jpg ; called 01/04/18

9. Appendix

A. MyRobotLab functions

Table A.1: Functions of Myrobotlab

Service	Name	Config files description : wiki	Description	Voice command sample
	InMoov	_InMoov.config	A composite service to control and allow easy operation of other all of InMoov's subsystems	REST METS TOI AU REPOS
	InMoovArm	skeleton_rightArm.config skeleton_leftArm.config		DA VINCI
	InMoovEyelids	skeleton_eyeLids.config		
	InMoovHand	skeleton_leftHand.config		OPEN YOUR RIGHT HAND OUVRE TA MAIN DROITE
	InMoovHead	skeleton_head.config		TILT YOUR HEAD ON THE RIGHT SIDE PENCHE LA TÊTE A DROITE
	InMoovTorso	skeleton_torso.config		
	Arduino	service_6_Arduino.config	Arduino control of the left, right sides of InMoov + 1 deported nano for Neopixel	
	Servo		A software servo service for motorized control and articulation of InMoov body parts	
	Tracking	service_D_OpenCv.config	A service which allows a pan / tilt camera system to track things	LOOK AT ME REGARDE MOI
	Speech	service_5_Mouth.config	A service to synthesise speech, basic multi-lingual capability MarySpeech Polly VoiceRss LocalSpeech IndianTts	SAY HELLO DIS BONJOUR

Service	Name	Config files description : wiki	Description	Voice command sample
	InMoov	_InMoov.config	A composite service to control and allow easy operation of other all of InMoov's subsystems	*REST* *METS TOI AU REPOS*
	InMoovArm	skeleton_rightArm.config skeleton_leftArm.config		*DA VINCI*
	InMoovEyelids	skeleton_eyeLids.config		
	InMoovHand	skeleton_leftHand.config		*OPEN YOUR RIGHT HAND* *OUVRE TA MAIN DROITE*
	InMoovHead	skeleton_head.config		*TILT YOUR HEAD ON THE RIGHT SIDE* *PENCHE LA TÊTE A DROITE*
	InMoovTorso	skeleton_torso.config		
	Arduino	service_6_Arduino.config	Arduino control of the left, right sides of InMoov + 1 deported nano for Neopixel	
	Servo		A software servo service for motorized control and articulation of InMoov body parts	
	Tracking	service_D_OpenCv.config	A service which allows a pan / tilt camera system to track things	*LOOK AT ME* *REGARDE MOI*
	Speech	service_5_Mouth.config	A service to synthesise speech, basic multi-lingual capability MarySpeech Polly VoiceRss LocalSpeech IndianTts	*SAY HELLO* *DIS BONJOUR*

	AzureTranslator	service_G_Translator.config	Use some Api to translate what the robot ear	*TRANSLATE hi IN japanese* *TRADUIT bonjour EN japonais*
	WebGui	service_0_WebGui.config	The WebGUI is a mini webserver and websocket server	
	ChatBot	service_A_Chatbot.config	Globalized AliceBot fork	*
	AudioFile		A service to play audio files voice command	*WHAT ABOUT STAR WARS* *A PROPOS DE STAR WARS*
	PID	service_D_OpenCv.config	An important service to implement a PID (proportional integral derivative) controller - e.g. allows smooth tracking of objects	
	PIR	service_C_Pir.config	Human basic detecor	*GO TO SLEEP* *FAIS DODO*
	UltrasonicSensor	service_I_UltraSonicSensor.config	Ultrasonic sensor for distance calculation	*MEASURING THE DISTANCE* *MESURE LA DISTANCE*
	OpenCV	service_D_OpenCv.config	The vision service which wraps opencv	
	OpenNi	service_E_OpenNI.config	Useful to use a Kinect	*BODY CAPTURE* *TRACKING SQUELETTE*
	Python		A python scripting IDE	
	SwingGui		A Swing graphical user interface	

	Voice recognition	service_4_Ear.config	speech recognition service WebkitSpeechRecognition AndroidSpeechRecognition Sphinx	
	Neopixel	service_9_neoPixel.config	Neopixel control service	
	OpenWeatherMap	service_H_OpenWeatherMap	Forecast	*WHAT IS THE FORECAST* *MÉTÉO*
	WikiDataFetcher		Knowledge bible	*QUELLE EST LA HAUTEUR DE LA TOUT EIFFEL*

B. Software mindmap

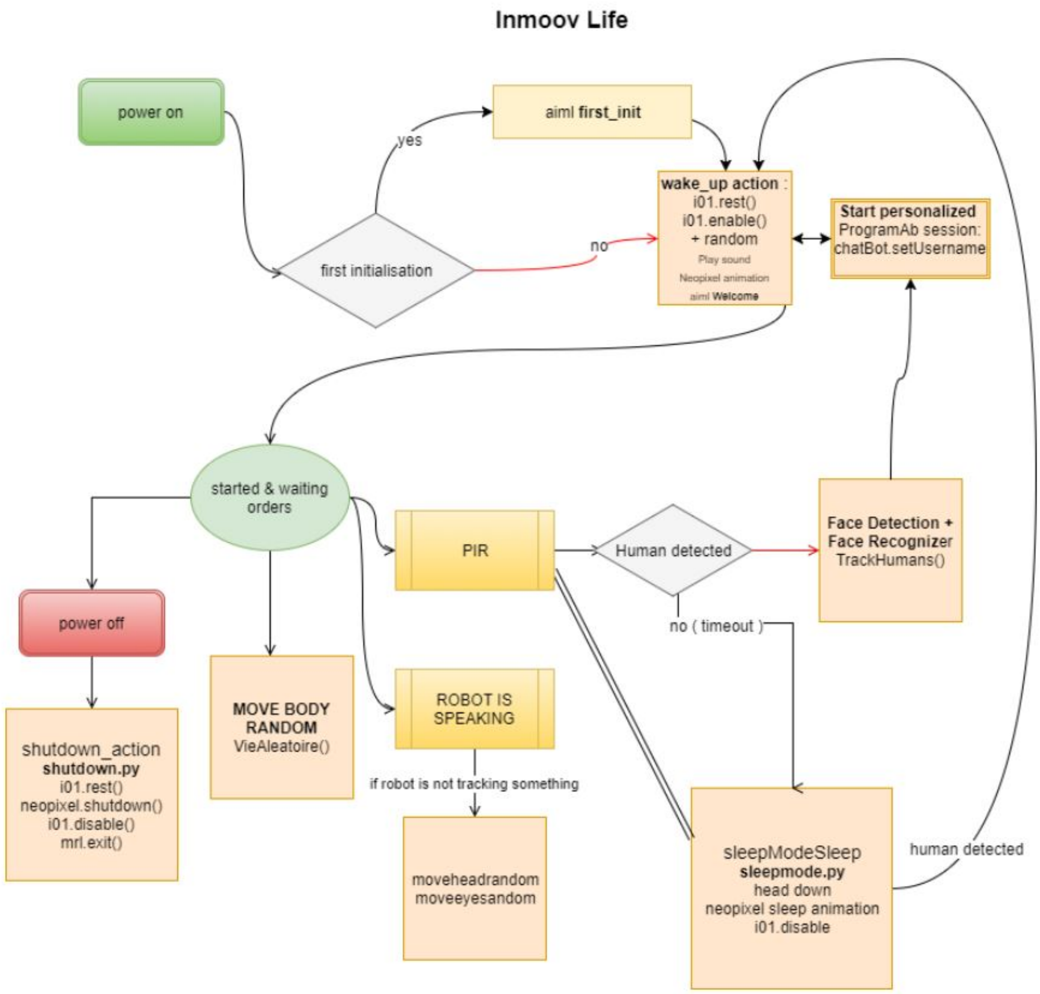


Figure B.1: Software flowchart

C. Software setup

SETUP & PREREQUISITES
STABLE RELEASE:

Manticore 1.0.2693

1. Download & update
   JAVA – https://www.java.com/fr/download/manual.jsp (if your computer is 64 bit, please take 64 bit version)
   CHROME – https://www.google.fr/chrome/browser/desktop/index.html (set it to default)
   ARDUINO – https://www.arduino.cc/en/Main/Software

2. Set the Port com of your Arduino(s) in device manager to 115200 BAUD.

3. Create a new directory [mrl] on root of your disk like this C:\mrl \

4. Download script : myrobotlab.1.0.2693.1.zip and extract like this in your C:\hmrl \

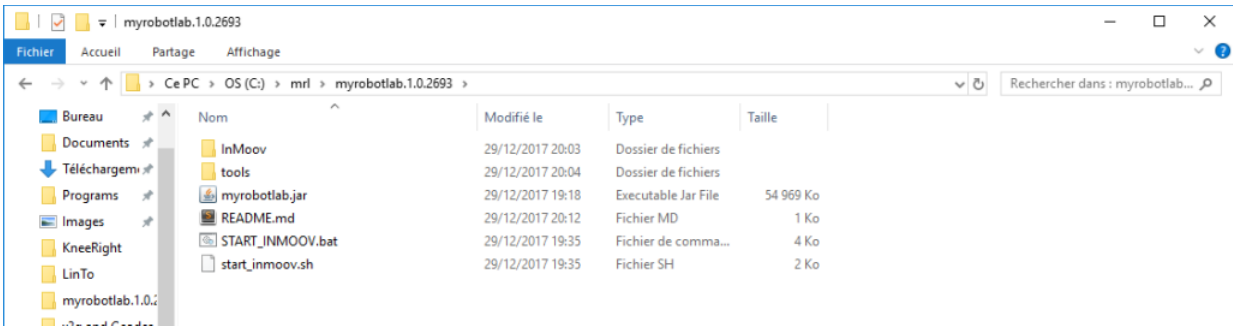


Figure C.1: C:\mrl \

5. Double click START_INMOOV.bat and wait a while for MRL to complete the install, click exit when the install is finished.

6. Double click START_INMOOV.bat to re-start. Once MyRobotLab has started, you can use InMoov in Virtual mode, to get in Full mode, follow the next steps.

7. Close MRL and upload the MRLcomm.ino code in your arduino from C:\mrl \myrobotlab.1.0.xxxx\resource \Arduino \MRLComm
8. You can setup your arduino portCOM , service_6_Arduino.config in config folder
9. Edit to activate the _InMoov.config to define which “real”parts you want to control.
10. In _InMoov.config you can also define language. To modify the voice type, edit service_5_Mouth.config and define the SpeechEngine and the voice you want to use. See below for more configuration info.
11. Edit to activate as True or False each of the skeleton_XXX.config files regarding each body part you have built.
12. In skeleton_XXX.config, set your mappings with min and max output for each servo and save. (A tool script is also available to test each servo one by one in C: \mrl \myrobotlab.1.0.xxxx \tool)
13. Double click START_INMOOV.bat
14. Enjoy!!

D. service & skeleton files

- _InMoov: gives the ability to set Full, Righside or virtual mode for Myrobotlab
- InMoovlife: gives the settings for the live of InMoov (for example sleeptimers)
- Service_0_WebGui: gives the setting to turn or off the WebGui
- Service_4_Ear: gives the setting for the ears
- Service_5_Mouth: gives the option to set the speech engine
- Service_6_Arduino: gives the option to set the communication port and type for the arduinos
- Service_8_NevrvoBoardRelay: Options and activation for the relays on the NervoBoards
- Service_9_neoPixel: Gives the options for the neopixel led ring (not available in the current build of InMoov)
- Service_A_Chatbot: Activation for the chatbot
- Service_C_PIR: Activation for the PIR sensor and settings for the arduino connection
- Service_D_OpenCv: Activation for OpenCv (webcam control) and settings regarding the OpenCv
- Service_E_OpenNI: Activation for OpenNI (Kinect control) and settings regarding the OpenNI
- Service_F_VirtualInmoov: settings regarding Virtual InMoov
- Service_G_Translator: setting regarding the translator (speech and voice of InMoov)
- Service_H_OpenWeatherMap: activation and setting weather map this so that InMoov can give you local weather updates
- Service_I_UltraSonicSensor: activation and setting ultra sonic sensor (not available in the current build of InMoov)

- Skeleton general: All the skeleton files consist of the option same options: disable en enable different parts of InMoov, set minimum limit, set maximum limit, set resting/reset position, set minimum mapping, set maximum position, set maximum velocity, set inversion, set auto disable and set pin numbers

E. limits

Table E.1: table InMoov limits

Group	Part name	Arduino atmega 2	arduino atmega 2	pin number	position min	position reset	position max		position min	position reset	position max		Part name
Hand	Thumb	Right	Left	2	37	37	120		35	35	135		Thumb
	Index	Right	Left	3	70	70	140		42	42	135		Index
	Middle	Right	Left	4	35	35	145		40	40	148		Middle
	Ring finger	Right	Left	5	30	30	145		35	35	145		Ring finger
	Pinky	Right	Left	6	35	35	138		60	60	125		Pinky
	Wrist	Right	Left	7	0	33	70		30	77	155		Wrist
Arm	Bicep	Right	Left	8	0	0	90		0	0	80		Bicep
	Rotate	Right	Left	9	50	90	145		10	25	55		Rotate
	Shoulder	Right	Left	10	0	13	180		0	13	180		Shoulder
	Omoplate	Right	Left	11	85	85	140		90	90	100		Omoplate
Head	Neck		X	12	0	35	100						Neck
	Rot. Head		X	13	5	90	175						Rot. Head
	Jaw		X	26	0	0	35						Jaw
	EyeX		X	22	85	105	125						EyeX
	EyeY		X	24	60	90	110						EyeY
Stomach	Topstomach		X	27	70	85	100						Topstomach
	Midstomach		X	28	60	90	120						Midstomach

(1) note: First Left of the blue vertical line is the rightside and right of the blue vertical line is the leftside.

F. standard voice commands

- rest: InMoov goes to resting position
- relax: InMoov goes to relaxing position
- look up: InMoov looks up
- look down: InMoov looks down
- how do you feel: InMoov sings New York
- track: starts eye tracking
- freez track: stops eye tracking
- tracking skeleton: InMoov uses Kinect to capture body movements
- freeze body capture: InMoov stops using the kinect and body capture
- capture gesture: captures the gesture (not working correctly right now)
- full speed: set all servos to full speed
- stop listening: InMoov stops listening
- quit your action: InMoov quits his current action
- surrender: InMoov makes surrender movement
- system check: InMoov checks all servos
- show your muscles: InMoov shows off his muscles
- rock paper scissors: start the game rock paper scissors

(1) note: This are only various available voice commands many more are available (look at full InMoov script for all voice commands)

G. custom code

- ear.addCommand"trigger command", "location", "def name": This line of code first of all gives the user the ability to add voice commands. The trigger command is the word(s) that the user uses to trigger the definition. The location is the location of the definition when in the same file use "python". The definition name is the name of the definition.

- def defname():: In this line the user starts to define the definition of the command. in the place of defname the name of the definition can be filled in.

- i01.moveHand"side of the hand", "angle thumb", "angle of the index", "angle of the middle", "angle of the ring", "angle of the pink", "angle of the wrist": in the side of the hand left or right can be filled in to define which hand is desired to be controlled. In the angle of every part a integer has to be filled in to define the angle. This command is available for every part of InMoov (moveArm, moveHead and moveTorso).

- sleep("length of sleep in seconds"): This commands lets InMoov sleep for a certain time the argument for this line is a integer.

- rest(): brings all the servos to the resting position.

- relax(): brings all the servos to the relax position.

- setHandSpeed("side of the hand", "speed thumb", "speed index", "speed middle", "speed ring", "speed pink", "speed wrist"): this command set the speed for different components. The scale for the speed is -1 = fast, 40 = medium, 1 = slow. This command is available for all the different parts (setArmSpeed, setTorsoSpeed, setHeadSpeed).

- openrighthand(): opens the right hand.

- openlefthand(): opens the left hand.

- i01.mouth.speakBlocking("funny texts"): this commands let InMoov say your own custom text and also blocks his auto reply function.

(1) note: This are only various available commands many more are available

H. Presentation of Technobothnia

The following code leads the robot to present the 3D-printing and -scanning lab in Technobothnia.

Presentation_TB_script.py

```python
i01.setArmSpeed("left",0.6,0.6,0.6,0.6) # set speed for all parts; the smaller the number, the slower the movements (0-1)
i01.setArmSpeed("right",0.6,0.6,0.6,0.6)
i01.setHandSpeed("left",1.0,1.0,1.0,1.0,1.0,1.0)
i01.setHandSpeed("right",1.0,1.0,1.0,1.0,1.0,1.0)
i01.setHeadSpeed(0.1,0.1)

ear.addCommand("Say Hello", "python", "introduction") # "Say Hello" = voice command InMoov recognizes, "python" = looking current file, "introduction" = name of the definition


def introduction():
    rest() # defined functions in the basic script
    openrighthand()
    openlefthand()
    i01.mouth.speakBlocking("Hello and Welcome to Vaasa in Finland. My name is Antonio and I was born in 2018.") # robot speaking
    i01.moveArm("right",80,95,70,85) # command to move the robot to the desired position of each servo
    sleep(2)
    i01.moveHand("right",37,70,35,30,35,5)
    sleep(1)
    i01.moveHand("right",37,70,35,30,35,70)
    sleep(1)
    i01.moveHand("right",37,70,35,30,35,5)
    sleep(1)
    i01.moveHand("right",37,70,35,30,35,70)
    sleep(1)
    i01.moveArm("right",0,90,13,85)
    sleep(2)
    rest()
    i01.mouth.speakBlocking("You are currently standing in Technobothnia. As far as I know, these halls")
    i01.moveHead(35,60)
    i01.moveHand("right",37,70,35,30,35,0)
    i01.moveHand("left",35,42,40,35,60,155)
    i01.moveArm("right",68,135,13,93)
    sleep(2)
    i01.moveHead(35,130)
    i01.moveArm("left",68,65,13,90)
    sleep(2)
    rest()
    i01.mouth.speakBlocking("were used as a cotton factory before 1996.")
    i01.mouth.speakBlocking("But I do not know for sure since I am programmed, I have to trust the information that people give me.")
    sleep(0.5)
    i01.mouth.speakBlocking("What I know for sure is, that nowadays, Technobothnia is shared by three different Universities for the education of their engineering students.")
    i01.moveArm("right",80,95,75,85)
    i01.moveHand("right",120,140,145,145,138,0)
```

```
    sleep(0.5)
    i01.moveHand("right",37,140,145,145,138,0)
    i01.mouth.speakBlocking("First, University of Vaasa")
    i01.moveHand("right",37,70,145,145,138,0)
    i01.mouth.speakBlocking("Second, Vaasa University of Applied Sciences, short VAMK")
    i01.moveHand("right",37,70,145,145,35,0)
    i01.mouth.speakBlocking("And third, Novia University of Applied Sciences, where some students came from, who worked on me for 4 months.")
    sleep(0.5)
    rest()
    i01.mouth.speakBlocking("I would like to tell you more about, is the 3D-Lab in Technobothnia, that I consider my home and the place where I was born")
    sleep(1)
    i01.mouth.speakBlocking("Uhmm,it is quite dark in here, isn't it?")
    sleep(0.5)
    rest()
    i01.mouth.speakBlocking("Let us turn the light on, please guys, pull that string!")
    i01.moveHead(35,50)
    sleep(0.5)
    i01.moveTorso(90,120,90)
    i01.moveArm("right",80,135,13,140)
    i01.moveHand("right",120,70,145,145,138,33)
    sleep(5)
    i01.moveHead(35,90)
    i01.moveTorso(90,90,90)
    rest()
    stopit()
    i01.mouth.speakBlocking("First of all, you have to know that every part of my body was printed either in Makerbot or in Ultimaker Printer. You can find them in the left corner!")
    i01.moveHead(20,105)
    i01.moveArm("left",0,39,65,93)
    i01.moveHand("left",35,42,40,35,60,100)
    i01.moveTorso(90,85,90)
    sleep(1)
    i01.mouth.speakBlocking("They use A B S plastic or P L A or you can even print multi-colored. That is pretty cool, right?")
    sleep(1)
    rest()
    i01.mouth.speakBlocking("Straight ahead of me, you can see the Object30 printer and next to it on the right side the dimension printer produced by Stratasys.")
    i01.mouth.speakBlocking("The Object30 can print objects by using 5 different kinds of material.")
    i01.mouth.speakBlocking("transparent and nontransparent, where as the dimension uses A B S with a very fine layer resolution")
    i01.head.eyeY.moveTo(100)
    i01.moveHead(0,90)
    sleep(0.5)
    i01.mouth.speakBlocking("I consist mostly of A B S plastic printed with lower resolution, I am a low-budget robot. Ha ha.")
    i01.mouth.speakBlocking("The time to print parts of me can vary between a couple of minutes for bolts, up to multiple hours for parts of my legs.")
    i01.moveHead(35,90)
```

```
    sleep(1)
    i01.mouth.speakBlocking("Furthermore on the left side,")
    i01.moveTorso(90,70,90)
    i01.moveArm("left",38,55,13,93)
    i01.mouth.speakBlocking("you can find a 3D printer, that uses liquid plastic to create the objects. That is amazing!")
    sleep(1)
    i01.mouth.speakBlocking("Every computer on that side were used to build and realize me by using various programs such as Cura, Makerbot Software and TinkerCad")
    sleep(1)
    i01.moveTorso(90,90,90)
    rest()
    i01.mouth.speakBlocking("Behind me,")
    i01.moveHand("right",37,140,145,145,138,33)
    i01.moveArm("right",90,90,130,85)
    sleep(0.5)
    i01.mouth.speakBlocking("you find the chart where it is possible to book printing time for each printer.")
    i01.mouth.speakBlocking("Remember always to make your reservation and clean the lab after the use!")
    rest()
    i01.mouth.speakBlocking("Here on my right")
    i01.moveTorso(90,120,90)
    i01.moveHead(35,13)
    i01.moveArm("right",3,126,4,139)
    i01.moveHand("right",120,70,145,145,138,70)
    sleep(1)
    i01.mouth.speakBlocking("we have a cupboard storing all the printing material and also the tool for the maintainance of the printers.")
    rest()
    sleep(0.8)
    i01.mouth.speakBlocking("I meet a lot of people here every day, I here a lot of speeches, and I can ensure you that some of them are not related to work!")
    i01.mouth.speakBlocking("ha ha ha ha ha")
    i01.mouth.speakBlocking ("I am kidding, I love to witness lessons of the universities working in Technobothnia")
    i01.moveHead(35,155)
    sleep(1)
    i01.moveArm("left",56,58,13,92)
    sleep(0.5)
    i01.mouth.speakBlocking("they work not just with computers, but also manually with all the different types of tools you can see there")
    i01.moveHead(0,155)
    sleep(1)
    i01.moveArm("left",38,65,13,92)
    sleep(1)
    rest()
    i01.mouth.speakBlocking("I have told you a lot about the printers now. But in this lab, there is also the opportunity of 3D scanning!")
    i01.mouth.speakBlocking("Please look to the right side!")
    i01.moveHead(35,53)
    sleep(1)
```

```
i01.moveArm("right",15,120,90,94)
sleep(1)
i01.mouth.speakBlocking("I am talking about that device close to the wall on the second table. The 3D scanner is from the company called nextengine!")
i01.mouth.speakBlocking("When the scan is done, you can even print the objects.")
rest()
i01.mouth.speakBlocking("Ladies and Gentlemen, my program comes to an end, there is nothing more programmed to say. It's been a while that i talk now. I started rusting already."
i01.moveTorso(70,90,90)
sleep(1)
i01.moveTorso(100,90,90)
sleep(1)
i01.moveTorso(90,70,90)
sleep(1)
i01.moveTorso(90,120,90)
sleep(1)
i01.moveTorso(90,90,90)
sleep(1)
i01.mouth.speakBlocking("You can have a look on various printed objects outside in the glassy box.")
i01.moveTorso(90,120,90)
sleep(0.5)
i01.moveHead(35,60)
sleep(1)
i01.moveArm("right",68,140,13,93)
sleep(2)
i01.moveArm("left",49,13,47,90)
sleep(1)
rest()
i01.mouth.speakBlocking("Goodbye and I hope to welcome you soon again in my house.")
i01.moveHead(0,90)
sleep(0.5)
i01.moveHead(35,90)
sleep(1)
rest()
stopit() # quits all actions and avoid the problem, that the code is executed twice
```

Figure H.1: Code presenting Technobothnia