**CHALMERS**

# Humanoid Robotics

3D printed humanoid robotic hand controlled via Leap motion

Project report

**Borna Vukadinović, Marsela Polić, Mickaël Fourgeaud**

01/11/2015

## Abstract

In this document, the theory, method and results of Humanoid robotics class project are reported. The class project was focused on the realization of a 3D printed robotic hand controlled via Leap Motion sensor. This system should allow a user to control a bionic limb in real time with no prior training by using movement of his natural hand as input. An experimentally tested solution based on consumer grade technology and prototyping tools is proposed. The software layer, written in C++, uses Leap Motion to acquire natural hand movements, and converts them into usable servo values. Most of the mechanical structure has been 3D printed using PLA material on a ZYXX 3D printer. The system is robust and sensitive enough to allow the operator to grab an everyday life item such as empty aluminium soda can.

# Contents

# 1 Introduction

## 1.1 Background

Biologically inspired technology is currently a major field of interest in engineering [1]. A point of focus in humanoid robotics is the development of systems presenting humanlike features destined to interact with a human-centered environment.

One especially interesting biological system is the human hand. The limb is strong enough to carry heavy weights, and at the same time precise and flexible enough to carry out the most gentle operations [2]. The integration of a robotic human-like hand to a teleoperated humanoid robot opens a wide range of applications but increases the complexity of its controller.

The recent developments in the accuracy of optical sensing technologies allow precise gesture recognitions and position tracking through devices with sub-millimeter precision reported [3]. Such devices can create a simple interface for the operator by allowing the use of a natural limb as input for a remote agent's limb.

In the recent years, the development of 3D printing gave engineers the opportunity to closely replicate the anatomy of a complex human hand. Achieving similar behaviour to a biological hand of 30 DOF is still a challenge for engineers. Mimicking is therefore often facilitated through mapping of different motions of human arm to motions of an artificial arm with 5-7 DOF [2].

## 1.2 Aim and Scope

In this project the aim was to build a remotely operated robotic hand. The hand should have five controllable fingers, each having a joint powered by a single servo motor. The hand should be mounted on forearm, where the hand and the arm should be connected by a wrist joint with 2 degrees of freedom. The hand and the arm should be 3D printed using existing STL schemes from InMoov project [4] with slight modifications. Servo motors for the joints should be placed inside the forearm, connected to the joint using fishing line with low elasticity. Different parts of arm should be connected with acetone, glue, and/or bolts and screws, depending on the parts and the nature of connection. The servo motors should be controlled via an on-board Arduino.

Teleoperation should be realized through Leap motion sensor which enables detection of motion of operator's hand with a reported 12mm accuracy [3]. Software built in addition to existing Leap motion SDK should enable mapping of hand movement to an analogous motor rotation.

The hand could be used in various operations where presence of human beings is not possible, and precision of human hand movement is needed. Final product could be mounted on various platforms depending on application. Future plans include additional features such as stereo vision feedback.

## 1.3 Goal Statement

During the project planning, a goal statement was formulated.

> *"For educational purposes, we want to build a 3D printed robotic hand remotely operated with the operator's natural hand movements as inputs."*

# 2 Theory

## 2.1 Project Inmoov

The project InMoov has been started and mainly developed as a personal project of the French sculptor and designer Gael Langevin. The original goal was to develop 3D printed prosthesis. InMoov is now the first Open Source 3D printed robot.

All the parts have been designed to be compatible with most printers. The complete set of the parts is available online for free on the project's webpage [4].


Figure 1: Artificial upper body by InMoov. Picture taken from [4]

## 2.2 3D printing

Additive manufacturing, also known as 3D printing, refers to the creation of 3D objects by gradual addition of layers of material. While multiple different processes exists, more information on fused deposition modeling will be given here, because this is the principle on which the printer used, ZYXX, is based on.

### 2.2.1 Fused deposition modeling

Originally patented by S. Scott Crump, but now expired, fused deposition modeling is the main technology used by commercial products.

The digital information of the object to print, contained as an STL file, is mathematically **sliced** to multiple layers of strings. Support is added if necessary to respect the structural integrity of the object. This information is then translated into machine code and transmitted to the printer which replicates the different layers, from bottom to top, by extrusion of molten material, as shown in Figure 2.

### 2.2.2 Material - Polylactic acid (PLA)

The ZYXX printer is using rolls of 1.75mm filament of PLA, a strong biodegradable plastic coming from starchy grain such as corn or sugarcane. Some properties of the PLA are given in a table 1. One of the important properties of the PLA is that it does not need to be printed on heated bed. On contrary, other commonly used material, ABS, requires heated bed because it melts in higher temperatures. Another

Theory continued on next page...

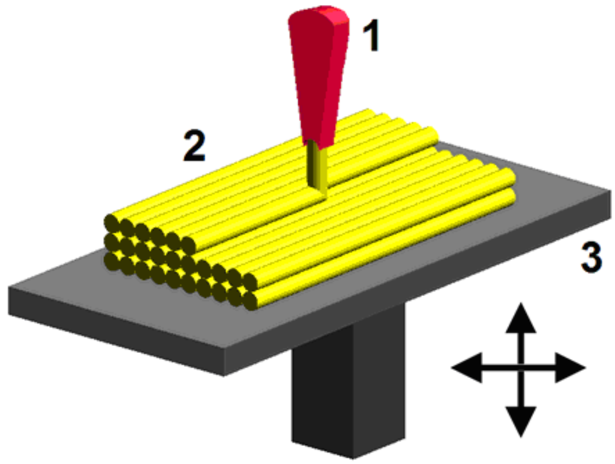



Figure 2: Fused deposition modeling working principle

important property is the need for cooling while printing, i.e. the printer should have a ventilator. Both PLA and ABS have their advantages, as well as drawbacks. The decision of using one instead of the other depends on the application of the final product. In this case, the decision was made inherently by properties of the available printer.

Table 1: Engineering Properties of PLA

| **Property** | **Value** |
|---|---:|
| Vitrification Temperature | 56.7∼57.9°C |
| Fusion Temperature | 140∼152°C |
| Drying Temperature | 40∼90°C |
| Drying Time | 6 ∼ 8 h |
| Drying Air Flow | 0.96 $m^3$/kg h |
| Density | 1.24 kg/L |
| Bulk Density of Pellets | 0.705 kg/L |
| Bulk Density of Sheets | 0.593 kg/L |

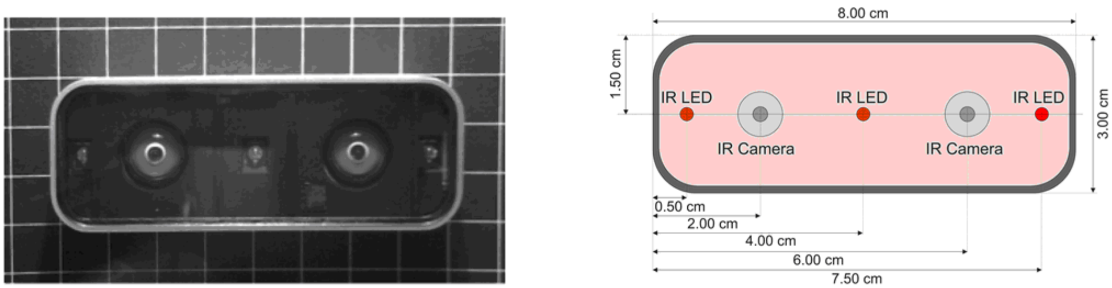
## 2.3 Leap motion

The Leap Motion controller is a system first commercialized in 2013 by Leap Motion Inc. The system consists of hand gesture, position and motion recognition sensor, figure 3, and associated algorithms. It is able to track hands, fingers and tools.

Hardware of the device includes 3 IR (Infrared) LEDs and 2 cameras with wide angle lenses, which enable detection of objects inside the interaction area depicted in figure 4. The system is tuned to a wavelength of 850nm. Interface with the host system is realized via USB. The Leap Motion SDK offers different types of outputs. In this project, the cartesian 3D coordinates of detected hands in the frame of reference tied to the controller position are used. A hand, seen by the SDK, is a set of fingers, themselves composed of bones as depicted in figure 5.

Due to the proprietary state of the Leap Motion SDK, no real information is available on the internal algorithms.

Theory continued on next page...




**(a)** **(b)**

Figure 3: (a) Picture of leap motion controller with transparent top cover (b) Schematics view of the internal hardware of the Leap motion controller

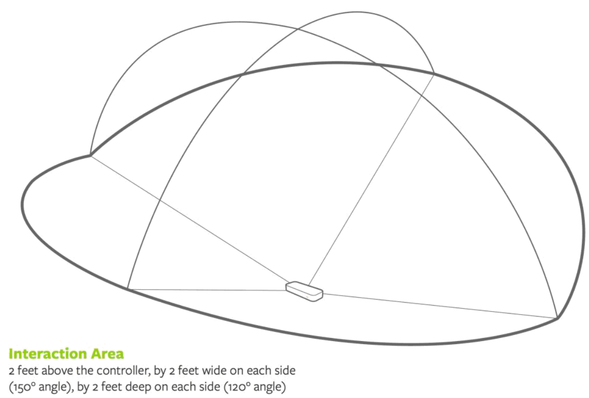



Figure 4: Interaction area of Leap Motion Controller

## 2.4 Arduino & Arduino IDE

Arduino is a widely used open-source microcontroller with multiple input and output ports enabling communication with its environment. The communication can be sensing through various sensors, affecting the environment through actuators, or most commonly both. Arduino can be programmed for a variety of applications using Arduino IDE.

In this project, Arduino is used for communication. If there is a sensor or other program running on a computer which controls the movement of the hand, the output values for the servo angles are sent to the Arduino. Arduino then controls the servos using PWM (see further).

Other microcontrollers could also be used in this project, but Arduino was chosen for several reasons. The main reason was that it was available for the project, and had sufficient functionality for the problem. It is also very easy to use, and the community is quite big, so there are a lot of open source libraries available online.

## 2.5 Servos

Servo motor is a commonly used type of motor in robotics. The main virtues of servo motors are simple control and high precision concerning position, velocity and acceleration. The precision is obtained through closed loop control using position feedback. In this project, high voltage coreless digital servos were used.

Servos for motion of the fingers have to have high torque. The high torque requirement exists mainly because of short lever-arm length inside the fingers, but also because of friction between the rope and many

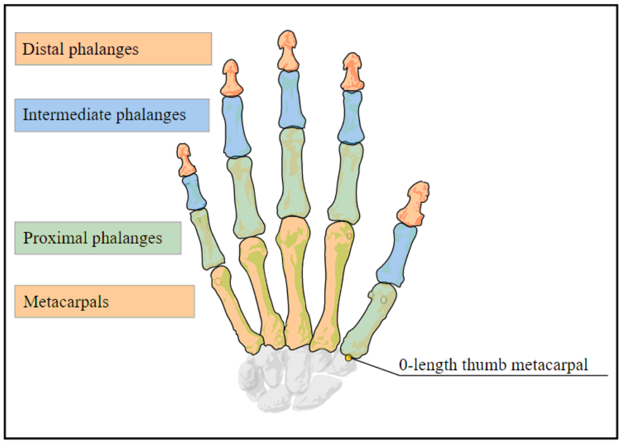



Figure 5: Anatomy of human hand

parts of the hand and arm. Movement of a finger is explained in Finger movement schema (see below).

Servo for the wrist rotation should also have high torque because of friction between plastic components during rotation.

The servos used in this project are coreless. The advantage of coreless motors compared to the ones with the core is in their faster response. Having no heavy steel core, the motor does not need to overcome its' inertia, so it can react faster, i.e. there is no time lag due to inertia of the core.

### 2.5.1 Finger movement schema

Movement of fingers is achieved using artificial tendons (e.g. rope). This type of movement was chosen in order to achieve greater visual similarity of artificial hand with the real human hand. Since the servos are too big to fit inside a finger, or even the palm, they are situated inside the forearm. From the forearm, rotational movement of the servos is transmitted through the tendon to the fingers, as depicted in figure 6. The picture shows cross-section of one finger (together with the palm and forearm), actuated with a servo motor.

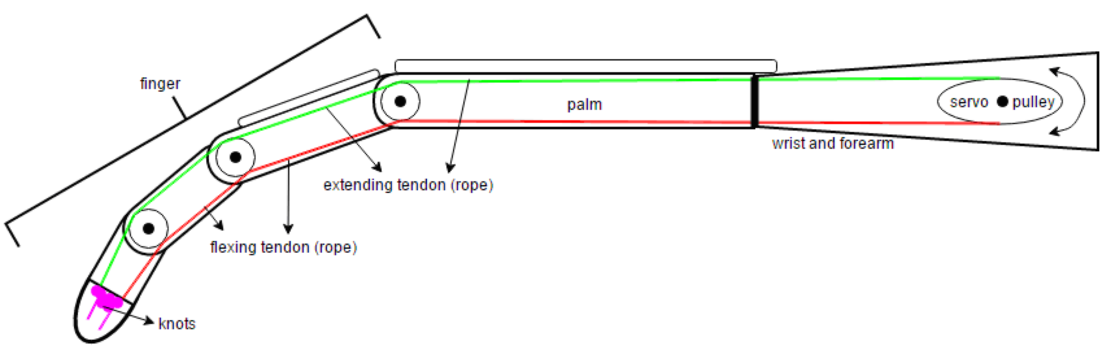



Figure 6: Cross section of the hand

When the servo rotates clockwise, tension in the extensor tendon (green) increases, and in the flexor tendon (red) decreases. That results in a torque around each axis of rotation of a finger segment, which in turn extends the finger. When the finger $i, i \in \{thumb, ..., pinky\}$ extends, length of the extensor tendon (green) is reduced, and the flexor tendon (red) increases its length. The change of length of tendons is only

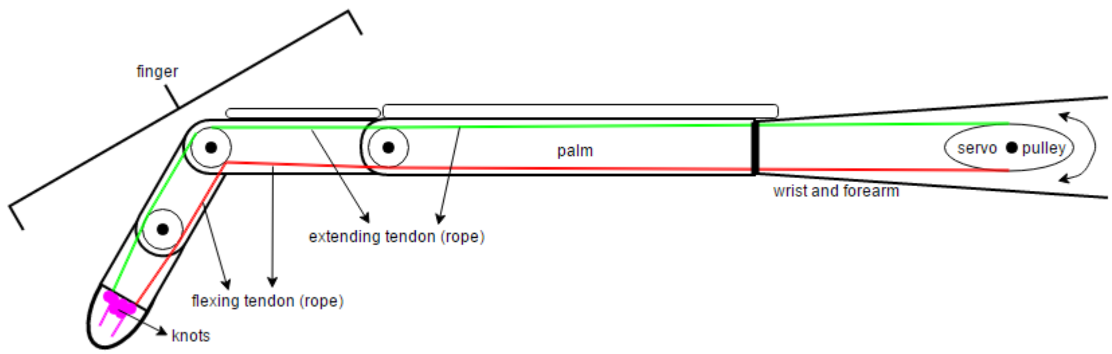



Figure 7: Equivalent finger position

illusory, because tendon is only winding or unwinding on the servo pulley. With the change of length, tension is also changed, and the movement of the finger stops when there is no net torque around the axis of a finger segment. The difference in tendons' lengths should then be approximately equal to the product of the servo pulley radius and the angle the servo has moved, as in equation 1. The expression 1 is not exactly equal because of elasticity effects on the rope length.

$$|l_{extensor,i} - l_{flexor,i}| \simeq r_{pulley} \cdot \alpha_{servo,i} \tag{1}$$

It should be noted that for a given servo angle, the position of the finger is not unique, due to the inability of controlling each finger segment independently. The variable that is controlled via servo angle is the sum of angles between subsequent finger segments, so the finger position on figure 6 is equivalent to the position shown in figure 7. One can move a finger between equivalent positions without any effect on the servo.

Mechanism for flexing the finger is analogous. Small radius of the pulleys in the finger joints requires a lot of tension in the tendon, i.e. high torque in the servo motor.

Other than short lever-arm lenght, the presence of friction also requires high torque motors. The friction is present between the tendon and the plastic parts that guide the tendon through the forearm, the wrist and the fingers.

### 2.5.2 Pulse Width Modulation (PWM)

The servos are controlled digitally with Pulse Width Modulation (PWM) via Arduino. PWM control is a simple digital control commonly used with servo motors. As the name suggests, pulse width, i.e. duration of high voltage pulse indicates the angle which the servo should reach. Depending on the ratio of duration of high and low voltage during a given time interval (duty-cycle), the servo rotates to the corresponding angle (figure 8). In this way, analogous information is transferred using digital signal.

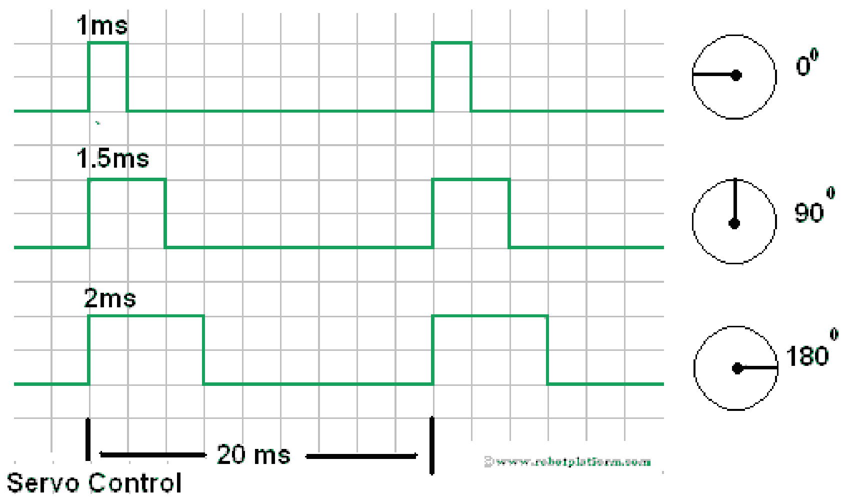



Figure 8: Pulse width modulation control of servo motor

# 3 Method

In order to reach the goal, the project was divided into smaller problems, which were then solved step by step. The main parts were planning, hardware and software.

## 3.1 Planning

The project was planned using the Value Model method [5]. Important parts were defined for the project: players, goal, segments, boundaries, deliverables, milestones, budget, and risk management. The course of work on the project was organized through a Gantt chart. [6]

The project players include the sponsor, Krister Wolff, the steering committee, Ola Benderius, and the team members which are also the customers: Borna Vukadinović, Marsela Polić and Mickaël Fourgeaud. The goal was formulated in a form of goal statement (see above). Boundaries and deliverables were respected, roughly following the milestones as defined in the Gantt Chart in [6]. Risk management predicted problems, of which some had occurred, but due to other circumstances (time frame), not all of the problems were solved as planned, see Results.

## 3.2 Hardware

Hardware used in this project consists of several important parts: plastic arm parts, motors, auxiliary connecting parts (bolts, rope, glue, etc.), and Arduino.

### 3.2.1 Artificial arm parts

Humanoid arm resembles natural human arm to a great extent through 3D printing. The arm was printed in 60 parts. 3D models were downloaded from InMoov [4] in STL format. The used STL files from the website are listed in table.

Table 2: Used STL files

| | | | |
|---|---|---|---|
| WristlargeV4.stl | WristsmallV3.stl | Bolt_entretoise7.stl | RobServoBedV5.stl |
| thumb5.stl | index3.stl | arduinosupport.stl | servo-pulleyX5.stl |
| majeure3.stl | ringfinger3.stl | robcap3V1.stl | RobCableFrontV3.stl |
| auriculaire3.stl | robpart2V3.stl | topsurface4.stl | RobCableBackV3.stl |
| robpart3V3.stl | robpart4V3.stl | WristGearsV4.stl | rotawrist3V2.stl |
| robpart5V3.stl | coverfinger1.stl | rotawrist1V3.stl | rotawrist2.stl |

Some preprocessing of the files was required. The preprocessing was done using Simplify 3D software. The infill of all the parts was set to 30%. Infill is a percentage of material consisting a part, the rest being air. Different infill percentages are shown in 9. Number of shells (outside layers) was set to 3 for all parts except for fingers, which had 1 shell. Some parts had to be printed with raft (several horizontal support layers of filament for stabilization). Likewise, some parts were printed with support (additional parts for stability during printing, which are removed after printing). Supporting structures (raft and support) were used mainly for the bigger parts of the arm, i.e. for the four big parts of forearm. Support was also used for the finger covers.

After setting all the parameters listed, model was translated into machine code and sent to print. Printer that was used was ZYXX. Material used was 1.75mm PLA. In total, around 200m of PLA was used. The printing process lasted around 50 hours. At the end of printing, some post-processing was necessary. Raft and support parts had to be removed. Holes had to be redrilled, and contact surfaces had to be smoothed.

Method continued on next page... 

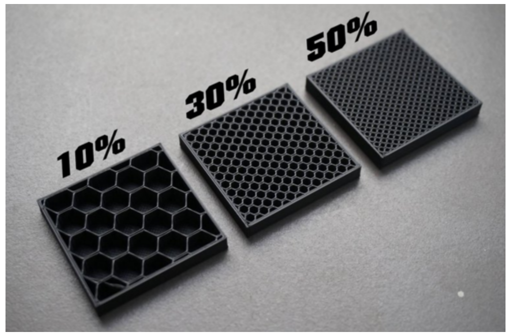



Figure 9: Infill. Picture taken from http://triplaxis.deviantart.com/art/3D-Printing-Infill-433183209

Arm was assembled with the help of tutorial on InMoov website [4]. Some auxiliary parts and materials had to be used for connecting parts of the arm. Different parts were connected using several methods: glue, bolts, rope, filament. For instance, fingers, printed in 6 segments each, were connected using glue and filament between the segments. The whole finger was then connected to the servo with a rope representing tendon. Details of assembly can be found online [4].

### 3.2.2 Motors

In the artificial arm, movement of fingers was realized using servo motors. Servos had the role of muscles. Rotation of the servo was translated to movement of fingers using rope, which had the role of tendons. Each finger was motioned using one servo. One direction of rotation corresponds to extension, and the other direction to flexion of the finger. Another servo was used for rotation of the wrist.

Two different models of servos were used for motion of fingers and of the wrist. For the fingers, HK15298 motors were used, with specifications given in table 3. The main reason for choosing this servo is the need for high torque, as explained in theory part.

Table 3: HK15298 specifications

| **Property** | **Value** |
|---|---|
| Torque | 14kg @ 6V, 15kg @ 7.4V |
| Weight | 66g |
| Speed | 0.13 s / 60 deg @ 6V, 0.11 s / 60 deg @ 7.4V |
| Voltage | 4.8V ∼ 7.4V |
| Motor type | Coreless |

Other servo used, for the wrist rotation, is MG996r. Although high torque is also needed for this joint, different servo was chosen in order to reduce the price of the whole arm, since the total price was an important constraint. The chosen servo has rotation range of 180 degrees, which is more than needed to mimic a human hand, therefore a reductor was placed inside the wrist to increase the torque (but consequently decrease the range of rotation). Reductor placed inside the wrist has a 20:11 ratio. Specifications of MG996r are given in table 4.

Five servos for the fingers were mounted on the servo bed inside the forearm, as shown in figure 10. 3D printed pulleys were mounted on each servo, and tendons were attached on the pulleys. Servo for the wrist rotation was mounted inside the wrist, as shown in figure 11. This servo rotates the joint through a reductor, with ratio 20:11. Inside the wrist, there is some space available for an extra servo. This one would enable the thumb or the wrist to have one more degree of freedom.

Method continued on next page. . . 

Table 4: MG996r specifications

| **Property** | **Value** |
|---|---|
| Torque | 10kg @ 7.2V |
| Weight | 55g |
| Speed | 0.2 s / 60 deg @ 7.2V |
| Voltage | 4.8V $\sim$ 7.2V |
| Motor type | Coreless |


Figure 10: Servo motors for fingers mounted on the servo bed inside forearm

Motors need more power than Arduino can provide. Therefore, an external power source should be used. Servos should be powered from a power source that can provide voltage of 7.4V, and current of at least 6 - 7A. It is advisable to use more than one power source, e.g. one power source per 3 servos. More detail on powering the servos can be found in Manual for future use.

### 3.2.3 Other parts

As mentioned before, tendons are used to transmit motors' rotation into finger flexion or extension. Instead of fishing line as suggested by InMoov, a rope of low elasticity was used for tendons. Tendons should be very tight for precise movement, otherwise deadzone type nonlinearity occurs. Tendons should however have a low elasticity (non-zero) because they are fixed to the servos using knots. Fishing line did not meet this requirement, and because of too low elasticity, it broke under tension. The rope was chosen as a substitute after testing it with 180 N tension. This test was sufficient because the servos with the pulleys currently mounted cannot produce more than 150 N force.

From other materials used in assembly, wires should be mentioned. There is a set of wires connected to the servos. Each servo has to have 3 wires: ground (brown), power (red), control (yellow). Endings of these wires are connected in a common jack with three female plugs. These plugs are then used for longer wires which can reach to Arduino and power supply. Currently, the ground wires (from the jack towards power supply) are black, the power wires are in warm color (yellow), and the control wires are in cold colors (green, blue, purple), except for wrist control, which is orange (due to a lack of wires in the same color). These wires can of course be switched. They all have a male plug ending, convenient for breadboard (protoboard) pins. To enable the hand to interact with surroundings, a gripping material was added on the palm. Mouse pad was used as a gripping material for several reasons: reasonable price, easy manipulation (cutting and

Method continued on next page. . . 

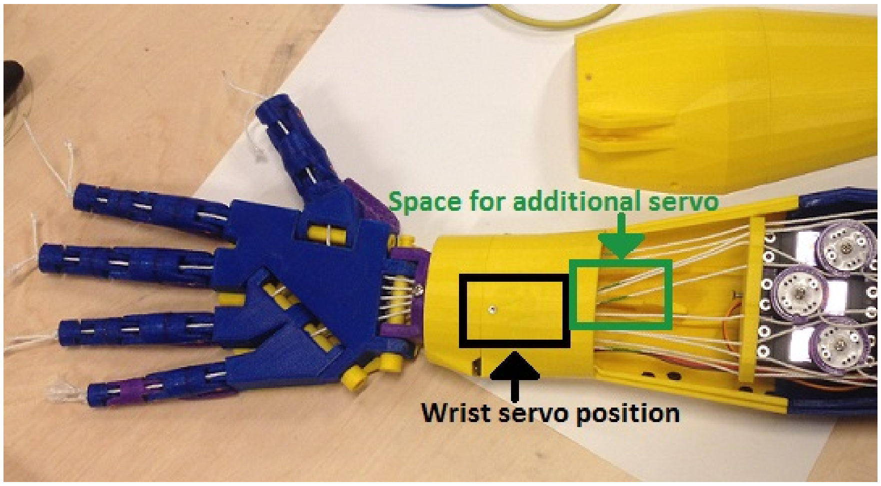



Figure 11: Position of wrist servo

glueing), relatively high friction, softness. Mouse pad patches were glued on all the contact areas of the palm.

### 3.2.4 Arduino

Important part of hardware is a microcontroller board. In this project, Arduino Uno was used. The main function of Arduino PWM is control of the servos based on control instructions from the computer, i.e. communication. As mentioned earlier, the Arduino was not used as a power source for the motors. The servos' ground was connected to the ground of Arduino (GND pin). Control wires of servos for fingers (thumb, index, . . . , pinky) were connected to pins 2, 3, . . . , 6 respectively (wires' colors: purple, green, blue, green, purple). The wrist servo was connected to pin 7 (orange wire). Connection diagram is shown on the figure 12. Arduino was powered and received control instructions from the computer via USB.

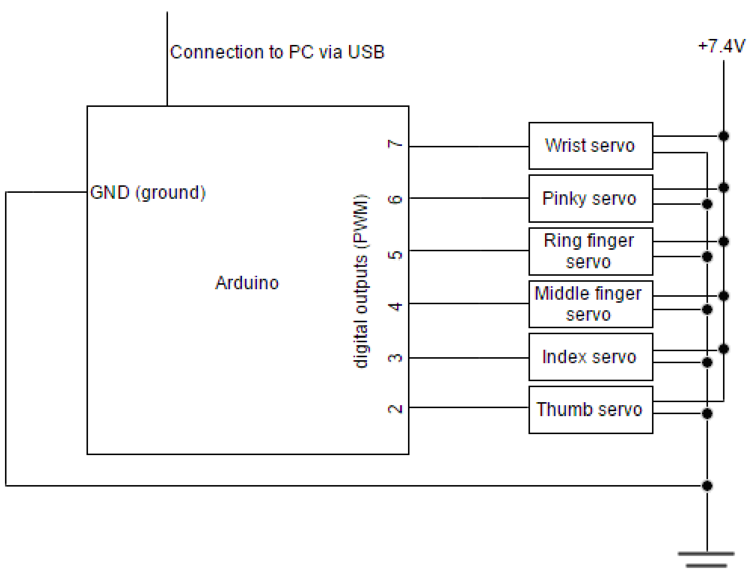



Figure 12: Connection diagram

## 3.3 Software

In this section, different parts of the software layer are presented. Arduino related software controls the servos. Host software is responsible for data acquisition and mapping. Communication protocol is was defined from communication between Arduino and the Host. Software layer can be depicted as in figure 13

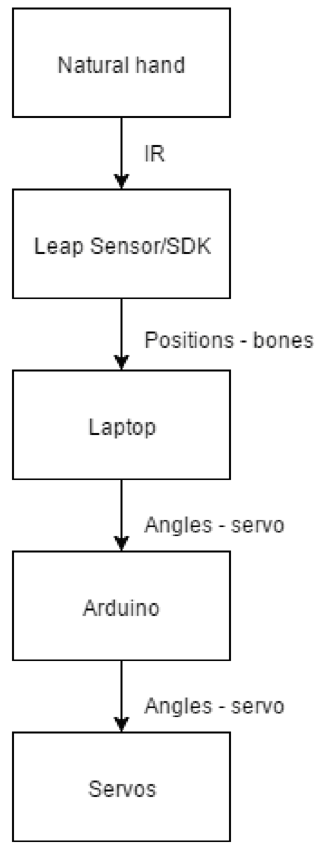



Figure 13: Software

### 3.3.1 Communication protocol

The communication protocol is used between the laptop (Host) and the Arduino board to transmit the control instructions, i.e. angular values of the servos.

To be able to control a servo through the Arduino board, the angle $\alpha$ of the servo and servo's pin index $N$ have to be communicated. This communication protocol sends messages containing instruction for one servo at a time by coding a pair ($\alpha$,$N$).

A message is coded with the following sequence of characters: $\#M$;. The parts of the message have the following meaning: # is the beginning delimiter of a message, ; is the ending delimiter of a message, M is the coded information. Beginning and ending delimiters were added to the message as flags denoting beginning and end of the message, because otherwise messages tended to get mixed during serial communication. The message information M can have two different forms:

1. M = XYYY
   In this case, X and YYY are integers with the following values: X = N (Arduino pin index of a servo), and YYY = $\alpha[deg]$.

2. M = none
   This message is sent if there is no valid output from the leap motion sensor.

The message M is encoded before sending to Arduino, and decoded by Arduino during servo control. Message encoding can be represented with equation 2.

$$XYYY = 1000 * N + \alpha \tag{2}$$

Message decoding is an inverse operation represented with equations 3 and 4. Note that decoding results in two values, whereas encoding resulted in one value, as a value of function of two variables. In equation 3 *trunc* is the truncature function. In equation 4 $mod(a, b)$ denotes remainder after devision of $a$ by $b$.

$$N = trunc(XYYY/1000, 0) \tag{3}$$

$$ALPHA = mod(XYYY, 1000). \tag{4}$$

#### 3.3.2 Arduino

The Arduino unit is responsible of controlling the servo motors in real time. Servo control is based on the information received from serial communication input as explained in the Communication Protocol section (see above).

The servo library included with the Arduino board was used. It offers several methods of control of servos connected to input pins of the board. The required position of digital servo is transmitted by simulation of Pulse-width modulation (PWM).

The Arduino algorithm shown on figure 14 as a flow chart.

#### 3.3.3 Host - Laptop

The host is responsible for linking of the data acquired by the Leap Motion sensor to the position of the servo in real time. C++ program is implemented under Ubuntu 12.04. The program runs the following set of operations for each frame recieved from the Leap Motion sensor.

1. Hand detection
   If no hand has been detected by the Leap Motion sensors, the Host discards the frame and communicates the lack of a valid input to the Arduino unit as the message "none" using the communcation protocol (see above).

2. Raw angles
   If a hand has been detected, *raw angles* are computed. Each finger of a human hand consists of several bones, which are connected with joints, as shown in the figure 5. Each of these links (bones) can be represented with its direction vector. Leap Motion gives information about the direction vectors of all the links. Angle of a certain joint can then be calculated as an angle between two consecutive direction vectors, representing the links before and after the joint. The *raw angle* for each finger is calculated as the sum of the angles of all the joints of the finger. This step represents **mapping** the positions of human fingers to artificial ones.

   This way of mapping is chosen because of the nature of human and artificial hand. First of all, the artificial finger is not able to mimic the natural one fully, because it only has one degree of freedom, while the natural finger has four (thumb has five). Finger joints in an artificial arm all have a common power source, whereas in a human hand, they can be controlled independently to a certain degree (hence more DOF). Since the rotation of the servo contributes to the movement of each joint in the artificial finger in an approximately equal amount, the total angle of the servo is propotional to the

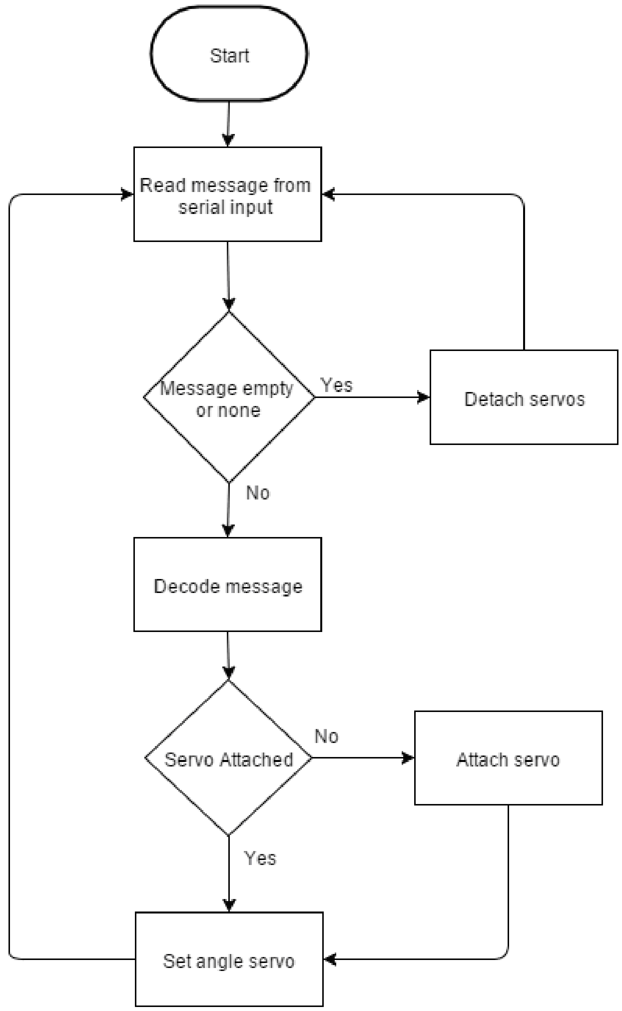



Figure 14: Arduino algorithm

sum of angles of joints in a finger. In this way, each joint is flexed for approximately the same angle. This angle is actually the average angle of joints in a human finger.

Using the average angle can be justified by the nature of human hand. Even though finger joints can be controlled somewhat independently, they are usually flexed in unison. This theoretical assumption resulting in described mapping was proven correct during testing, see Results.

The raw angle of the wrist servo was obtained as the roll value of the normal to the operator's palm. This value is provided from the Leap Motion SDK.

3. Final Angles
   Two different phenomena occur, which require conversion of raw angles before sending to the Arduino. Fistly, the sensor has an offset at a zero position (flat palm). This is corrected by applying an offset to the raw data, with the value of offset based on observations and measurements during testing. Deviation of the average offset for different operators (users) was small enough to justify a constant offset correction. The second conversion is due to non-uniform angle ranges of the servos. The lengths of the tendons (i.e. finger links) are not equal for all the fingers. This results in different maximum angles of the servos. This is corrected by applying a linear transformation with parameters determined during testing. Parameters of these transformations are given in table 5. It should be noted that the

maximum raw angle of a human finger should be 270° (the maximum being reached when the fist is clenched). That would imply the need for scaling to a range $[0, 180]$, i.e. the servo angle range. However, during the testing of Leap Motion, it turned out that the maximum value was almost always around 180°. Therefore, the scaling was not included in computation. Another expected scaling should be from raw angles range $[0, 180]$ to a servo angle range $[0, 90]$. However, the Arduino library for PWM control has the input range $[0, 180]$, with maximum value of 180 corresponding to maximum value of 90 for the servo.

The final angles' values obtained after described transformations are then transmitted to the Arduino unit using the communication protocol (see above).

Table 5: Parameters for conversion from raw to final angles

| **Servo** | **Offset** | **Maximum range** |
|---|---|---|
| Thumb | 20 | 300 |
| Index | -20 | 180 |
| Middle | -20 | 180 |
| Ring | -20 | 140 |
| Pinky | -20 | 160 |
| Wrist | 0 | 180 |

# 4 Results

The goal was to make an artificial arm which could mimic the movements of a human arm. The arm was made based on a design from InMoov. Leap Motion sensor was used for generating control instructions which were then communicated to the arm over Arduino.

Artificial hand was 3D printed with 5 controllable fingers, each finger having one degree of freedom (DOF). The hand was mounted on a forearm through a rotational wrist joint with one degree of freedom. The six servo motors were placed inside the forearm. Control of the servo was realized through Arduino board. Teleoperation was realized through Leap Motion sensor. A software was built in addition to existing Leap Motion SDK which enabled mapping of hand movement recorded with Leap Motion to analogous motor (joint) rotation. Mapping had to be done because of a significant difference between the DOFs of a human and artificial hand.[**?**] Functionality of the hand was tested through observation, and the results were satisfactory. The hand mimicked human hand movements to a great extent. The hand was able to grip common objects and tools designed for human hand, such as screwdrivers and cans. Numerical measurement of the hand was not obtained because of inability of Leap motion to recognize an artificial hand.

The project was supposed to implement slight modification to the InMoov design, in order to achieve an additional degree of freedom to the wrist joint. The place for an additional servo motor is already available, and the mechanical part of the joint was smoothed and rounded to enable the flexion and extension of the wrist. Places to tie additional tendons were determined on the hand. However, the new functionality is not included in the final product due to a burned servo only 3 days prior to project deadline. It should be noted that the servo was burned during testing, and not during normal operation. Therefore, there should be no danger of destroying other parts during normal operation. This risk was included in risk management analysis, but due to an event occurring a short time before the deadline, new servo could not be obtained on time.

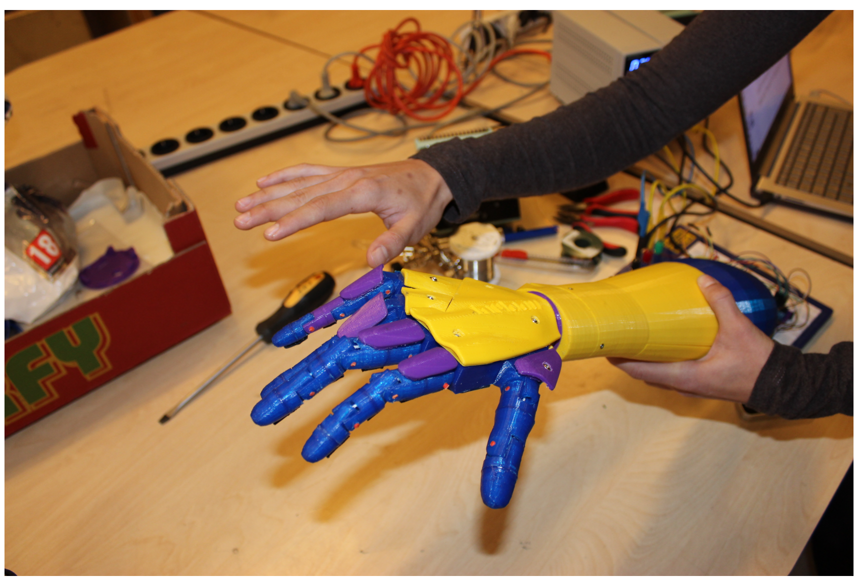


Figure 15: Artificial and human hand

## 4.1 Discussion

The artificial hand mimicked the human hand in a satisfactory way, as explained in the previous chapter. However, there were some problems present. The detection of the human hand was sometimes flawed, resulting in decreased mimicking ability of the artificial hand. In most of the cases, the thumb

Results continued on next page... 

detection was the source of flaw. Different distances from the Leap Motion sensor were tried, but it was concluded that the most relevant factor in detection accuracy was the person whose hand was being detected. Another problem with hand position detection occurrs when the palm is facing up. In that case, the Leap Motion cameras cannot see the positions of the fingers, and therefore the finger angles cannot be calculated. This problem is easily solved with restriction on turning of the palm of the person controlling the artificial arm, or with additional Leap Motion sensor above the manipulating area. Except for the detection problems, the accuracy of finger movement can be decreased due to tendons loosening. In that case, tendons should be tightened, in order to have finger movements that correspond to the servo angles.

### 4.1.1 State of the art comparison

Many tendon driven robotic hands have been designed and implemented so far. A few examples are a tendon driven robotic hand developed in Movement Control Lab at University of Washington [8], another one developed in Humanoid robotics group at MIT [9], or the one developed at Saarland University [10]. When compared to the hand built in this project using, for example, numerical data provided in [11], the hand from Saarland University is clearly superior. It can manipulate fragile objects, like eggs, and it can lift a 5 kg load by 3 cm within a split second. The reason for that superiority is, of course, of the financial nature. The development of the Saarland University robotic hand was a part of European project, and therefore had much more financial support. Motors that are used for actuation are more precise than the ones used in this project. Furthermore, the hand possesses more motors, which results in more degrees of freedom and more maneuverability. Additionally, the tendons are made of strong polymer strings, which can withstand more tension. Another difference is the control of the hand, which is more precise in the case of the Saarland University robotic hand, due to the usage of better quality 3D computer vision system. However, the functionality of these two hands should not be a subject of comparison. Building of a hand in this project had an educational purpose, whereas the Saarland University hand, built by a team of experts over several years, has a research and industrial purpose. If conveniece of the hands is to be compared from an educational perspective, the one built in this project is surely more convenient for several reasons: price, availability, level of complexity, possibilities of further features implementation.

# 5 Prospects

The arm built in this project can be a platform for many other projects, tests and improvements. The process of assembling it was highly educational, and future work on it could be conducted in the same manner. Possible future work ideas are unlimited, and depend only on creativity of a future user. Here, only several ideas will be presented, based on the individual preferences and interests of the team members. The ideas include both hardware and software improvements, and several could also be implemented in parallel.

Due to the fact that Leap motion sensor might not be available in future work, other means of control should be considered. These could be purely program based, or include other sensor hardware which could enable teleoperation, such as camera, sonar, gyroscope, magnetometer, accelerometer, or any other sensory technology. Sensor fusion is also an interesting field with many possibilities for research and development.

The mechanical parts of the hand could be improved, for instance by implementing the additional wrist degree of freedom. For this, a new servo should be built inside the forearm, in the designated space in the wrist. The wrist was temporarily fixed in order to avoid unwanted flexion of the wrist with hot glue. When the glue is removed, the hand is able to rotate. Additionally, tendons should be tied to a servo and to the hand to enable control of the wrist. Some programming is also necessary.

The hand (i.e. the forearm) could be mounted on a movable platform, which would enable the hand a wider range of actions. Another approach would be adding an elbow joint which would add additional degree of freedom. Adding a shoulder joint would enable the hand to mimic most of the natural human movements. InMoov project already has schemes for the whole human upper body, but designing a different platform for the forearm would also be a very interesting approach.

The hand currently has no feedback. Different sensors could be implemented on the hand or in its' environment to enable feedback, and eventually closed loop control. The InMoov project has developed a design for touch sensors on the tips of the fingers. This improvement would require replacing fingertips with new, active ones. Development of new approaches using tactile sensors could also be interesting. Other feedback methods could include stereo vision feedback.

The precision of the hand could be improved through modification of control algorithm. Currently, the joints are controlled by position. Common approaches in robotics would also include control of the joints by position, velocity and torque. [7] This approach would enable both fast response in positioning, as well as precise handling of fragile objects. With a closed loop control, the code for hand movement control could be developed using machine learning methods such as evolutionary algorithms.

To conclude, the humanoid hand could be used for many applications, but those related to human actions, i.e. manipulation of environment designed for human usage are probably the most interesting ones. This hand should be a platform for exploration of a human body which could eventually lead to fully functioning humanoid robots capable of sharing natural human environment.

# 6 Manual for future use

This part of the report is written for the future user of the arm. Some set up needs to be done before using the arm. Without any accidents, the arm should be functional after following the following steps.

## 6.1 Check everything is in place

6.1 Check everything is in place

The arm should have all the important parts in place. That means that the plastic parts should be robustly connected. The fingers should be relatively easy to flex or extend, and a buzzing sound should be heard from the servos while moving the fingers.

### 6.1.1 Opening the forearm

If anything seems out of place, the forearm can be opened by unscrewing 2 bolts on the the forearm. The bolts are placed on the sides of the arm. After unscrewing the bolts, the yellow cover of the forearm should be carefully separated from the half yellow - half blue part. When the cover is removed, five servos should be visible. The servos should be fixed in the servo bed. If they are not, some bolts might have to be tightened.

### 6.1.2 Servos for fingers

If the motion of the fingers does not produce any sound from the servos, tendons might have to be tightened. This is described in a later subsection (Tightening the tendons).

### 6.1.3 Wrist servo

Similar to the fingers, rotation of the hand with respect to the forearm should produce a buzzing sound from the wrist servo. The wrist servo is located in the wrist. Upon opening the forearm (as in 5.1.1), one should unscrew the three bolts of the wrist part (figure?). The servo should be visible, with a reduction mechanism mounted on the servo axis. If the reduction mechanism is loose, the smaller gear should be screwed into the servo axis. This gear should rotate along with the servo axis. If there is any sliding, the wrist rotation will not work, and the gear should therefore be tightly fixed before usage. The teeth of the gears should be greasy to avoid tearing. If there is some backlash, new gears can be printed from the InMoov website (WristGearsV4.stl).

## 6.2 Arduino software

The communication with the servos is enabled through the Arduino board. The board used was Arduino Uno. There are a few steps if one has not used Arduino yet.

### 6.2.1 Download Arduino IDE

The Arduino is programmed from the Arduino Software which can be downloaded from the official webpage (arduino.cc). The programming language is very similar to C.

### 6.2.2 Obtain existing programming codes

There are some codes already implemented for the arm. These should be available with the teacher. The codes are used for setting the servos, and have hardcoded values of the servos. Note that the maximum values sent to different servos vary, i.e. in order to fully flex different fingers, different values should be communicated.

## 6.3 Wiring

The servos have to be powered by an external power supply, and should also communicate with Arduino or other microcontroller. The wiring used in this project is presented here. The servos should be powered by a power supply that can provide 7.4V. It is also necessary to provide a current of at least 6A. It is advised to use several DC sources in parallel for this reason. This can be realized using a protoboard. Grounds of the sources should be connected, along with the grounds of servos, and also connected to the GND pin of the Arduino. The control wires should be plugged in to Arduino ports 2-7 as explained in Method - Arduino. The power wires of the servos should be connected to the + side of some of the power sources.

The Arduino should not be connected to an external power supply if it is powered from the computer via USB.

## 6.4 Tightening the tendons

If there is no noise from the servos while flexing/extending the fingers, the tendons might have to be tightened. However, small backlash will probably still be present (few millimeters). It should also be noted that a servo controls all three finger joints together (or two, for the thumb), meaning that the servo controls the sum of angles of all joints of one finger (see 2.5.1. Finger movement). That results in more than one position of a finger for a given servo angle (as shown in figures 6 and 7), so one must be careful to move the finger in such a way that it actually changes the angle of the servo to check whether the tendons need to be tightened. Upon opening the forearm cover, 10 tendons are visible. Pairs of two tendons tied to a common servo are used for controlling a certain finger. The tendon marked with green is the extending tendon, and the other of the pair is the flexing one. The servos corresponding to fingers are shown on the figure 16.

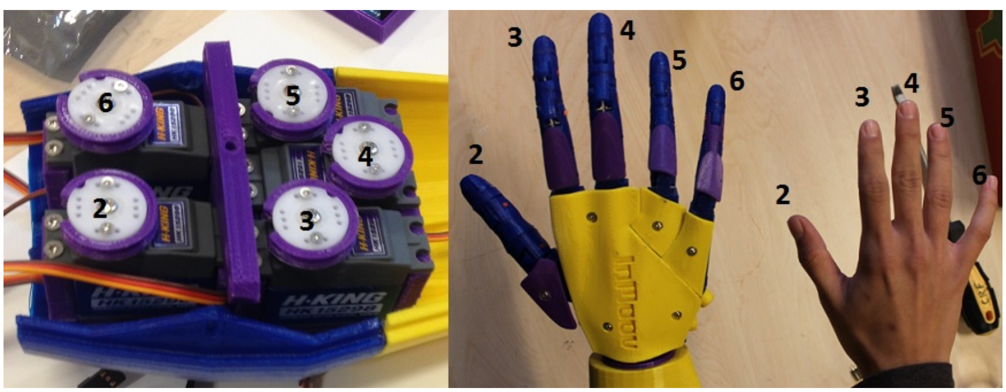



Figure 16: Servos corresponding to fingers. Numbers denote Arduino input pins

First step is putting the servos to zero position using Arduino. This can be done by downloading the SimpleServoControl.ino program with fingersToZero() function to Arduino. When the servos are connected to the power supply, they will rotate to zero position. Note that this position is the extended position of the fingers. Now, the extending tendons (ropes marked with green) should be tightened using pliers or other tools (fingers are usually not enough), and markings on the rope should be made. The marking should point to the part of the rope where a knot should be if the tendons are tightly knotted to the servo in zero position. The hand was tested with 180N force applied on the tendons, which is more than enough to tighten them. Then, servos should be put to maximum angle using fingersToMax() function from SimpleServoControl.ino program. Same tightening procedure should be repeated with other tendons (ones for flexing the fingers). Next, servos should be put to middle position using allToMiddle() function from SimpleServoControl.ino program. The servos should then be plugged out from the power source. Next step is unscrewing and unmounting servo pulleys from the servos. That will release the tension and enable untying the knots of the tendons on the servo pulleys. It is advised to keep the tendons pulled through the holes in the pulleys so that they do not get mixed up after untying. Then, the knots should be tied on all tendons in such a way that the knots are located on the marking made before. After the knots have been tied, pulleys should be mounted on the servos in the same position as before, and fixed with screws. During the whole process, one must be careful not to rotate the servos. If they are moved while pulleys are unmounted, they should be moved back to the middle position using function allToMiddle() again.

## 6.5 Control

The Leap motion sensor might not be available for future use. If it is available, the code developed in this project should be available with the teacher. Otherwise, another control mechanism should be developed, depending on preferences, interests and ideas of future users. Some ideas are listed in Prospects, but the options are almost infinite, and the decisions remains on the user.

# 7 Conclusion

In this project, a remotely operated robotic hand was successfully built. Teleoperation was realized using Leap Motion sensor data, and a mapping software that was developed for this cause. The hand of a human-like design was built using 3D printing technology. Hand was mounted on artificial forearm, which resulted in a robotic arm with 6 DOF. The design was explored and a way of improving was determined by adding an additional degree of freedom.

Building an artificial arm was a challenging and exciting process. Combining different technologies such as 3D printing, microcontrollers, servo motors and movement detection to reach a common goal gave an insight to the wide field of robotics. Successful assembly of a relatively complex robotic hand in a tight deadline, and with constrained resources was highly motivating for future work in humanoid robotics.

From this project, it can be concluded that humanoid robotics is an interesting field which is broadly available for research. Even though the hand built is not applicable in industry or other real life situations, it can serve as a platform for education. Possibilities for future development, as was suggested, are not lacking.

# 8 References

## References

[1] Chew, C., et al. *From science fiction to reality - humanoid robots*. HNICEM, March 27-30, 2003, Manila, Philippines

[2] Behnke, S. *Humanoid Robots – From Fiction to Reality?*. KI-Zeitschrift, 4/08, pp. 5-9, December 2008

[3] Weichert, F., et al. *Analysis of the accuracy and robustness of the leap motion controller*. Sensors 13, no. 5 (2013): 6380-6393

[4] Langevin, G. *InMoov — open-source 3D printed life-size robot*. http://www.inmoov.fr , October 2015.

[5] Lindstedt, P., and Burenius, J. *The value model: how to master product development and create unrivalled customer value*. Nimba, 2003.

[6] Vukadinović, B., Polić, M. and Fourgeaud, M. *3D printed humanoid robotic hand controlled via Leap motion - Project Plan*. October 2015.

[7] Kovačić, Z., Bogdan, S., and Krajči, V. *Osnove robotike*. Graphis, Zagreb (2002).

[8] Kumar, V. *Homes.cs.washington.edu*. http://homes.cs.washington.edu/~vikash/P_Hand.html , 2015.

[9] *People.csail.mit.edu*. http://people.csail.mit.edu/g_gomez/robots.php , 2015.

[10] Palli, G., et al. *The DEXMART hand: Mechatronic design and experimental evaluation of synergy-based control for human-like grasping*. The International Journal of Robotics Research, 2014, 33.5: 799-824.

[11] *Project DEXMART sensitive robotic hand — RobAid*. http://www.robaid.com/robotics/project-dexmart-sensitive-robotic-hand.htm 2012.