Bachelor Degree Project

UNIVERSITY
OF SKÖVDE

VIRTUAL COMMISSIONING WITH VIRTUAL REALITY

Bachelor Degree Project in Automation Engineering

Bachelor Level 30 ECTS

Spring term 2020

Authors:

Jesús Tomás Almansa Fernández

Juan Pablo Vargas Maqueda

Company Supervisor: Mikel Ayani

University supervisors: Martin Birtic and Aitor Iriondo

Examiner: Kaveh Amouzgar

ABSTRACT

The industry is nowadays going through a transformation in networking technologies which leads to what it is called Industry 4.0. This stage of the industry is arriving from the hand of the internet of things. Having connected all the elements within a factory allows to control and keep track of them telematically. Virtual commissioning is in charge of designing, testing and debugging the system before it is even built. This project tries to approach virtual reality to virtual commissioning and create the virtual model of a human in these simulated environments. Technology allows introducing human interaction for many purposes such as operator training. To do this project properly a methodology will be followed for the design and creation. Once the background, frame of reference and literature review are established, the development can start. The development of the project has taken place alongside Simumatik Open Emulation Platform, consisting of creating the body of a person, as simple as possible, following ergonomics, into this platform for commissioning purposes. The model will be able to interact with the virtual environment like robots, boxes, and sensors. To sum up, the complexity of the model will be limited to the inputs coming from the head and the hands. There exist infinite solutions for which position should be the rest of the body, therefore, this project aims to fix some variables to find valid solutions. Finally, the project achieved building a digital human model in which the main goal was building the arms that are estimated. The model is capable of interact with Simumatik´s environment that has been created specifically to show the functionalities of this project, being detected for sensors and robots of the system.

ACKNOWLEDGEMENTS

In these brief lines, we would like to thank to all the people that have helped us during the development of this project.

First and foremost, we would like to thank to our supervisor Martin Birtic for all the help that he has provided us during the development of this project.

In addition, we would like to thank to Mikel Ayani and Simumatik for the support that they have given to us and for giving us the opportunity of doing this project alongside them.

Furthermore, we wish to express our sincere thanks to the University of Skövde, for providing us with all the necessary facilities for the research.

Finally, we would like to give special thanks to all the people that has support us during the entire year.

Certify of Authenticity

This thesis has been submitted by Jesús Tomás Almansa Fernández and Juan Pablo Vargas Maqueda to the University of Skövde as a requirement for the degree of Bachelor of Science in Production Engineering.

The undersigned certifies that all the material in this thesis that is not my own has been properly acknowledged using accepted referencing practices and, further, that the thesis includes no material for which I have previously received academic credit.

JT Almansa Vargas

Jesús Tomás Almansa Fernández *Juan Pablo Vargas Maqueda*

Skövde, 18th May 2020
School of Engineering Science

Abbreviations

VR	Virtual Reality
VC	Virtual Commissioning
OT	Operator Training
DoF	Degree of Freedom
DHM	Digital Human Modelling
SWOT	Strengths, Weaknesses, Opportunities and Threats
PLC	Programmable Logic Controller

Table of Contents

List of figures

List of tables

1. INTRODUCTION

Virtual Commissioning (VC) is the early development and validation of the code that will be conducted by the programmable logic controller (PLC). A simulated model will be used to test the logic in a context of simulation of the process and automation and validation of the behaviour of the system. Most of the commissioning is nowadays, focused on testing and fixing control errors in the software once the entire system is already built, by relying upon a digital software platform you can test, debug, and validate the software on your computer. Research has shown that VC improves the quality of control software by more than 100% and reduces the commissioning by 75%. (Reinhart and Wünsch, 2007).

Virtual Reality (VR) is a term used to describe a three-dimensional space, an environment generated by a computer that can be explored and interacted with. The person who is experiencing this virtual approach to reality becomes an element of this world and can interact with it performing several actions such as manipulate or interact with objects. This new reality allows people to experience as if they were inside this virtual environment. Several complex technologies are being developed right now to achieve this goal but lately, it is becoming cheaper. It is very likely to see more of these technologies soon (Virtual Reality Society, 2017).

Operator Training (OT) is a specialized education of employees for jobs in which specific knowledge and skills are needed. The main objective is to train the operators for the work that they are going to perform so they can achieve the goals satisfactorily. It should contribute to increase the quality, productivity, and safety of the operator and decrease errors, wastes, accidents, and danger by building human virtual models that can interact with virtual environments. This kind of relationship can be helpful on educating operators showing them situations that will need to cope with in their actual work avoiding wastes such as raw material and energy, for the development of a solution that allows creating a safe working environment for operators to train them before they get into the actual working cell.

1.1. Background

Manufacturing is an industrial process that goes through different stages such as designing, assembly, programming, and commissioning, in which the previous stages are tested looking for errors and the optimization of the system in terms of programming. In the past decades, every stage was built independent of the other and later being put together trying to make them work as a system. Nowadays, the way to create a new manufacturing system could be done through VC. Virtual models simulate the entire system that is planned to be built. Besides, this eliminates future drawbacks such as possible errors that could have appeared in the assembly and/or programming stages. Being able to test every step of the manufacturing system without physically building it and, therefore, saving money, time and increasing safety in worker's environment (Hoffmann, Maksoud, Schumann and Premier, 2010).

Planification of automated systems and their methods are reaching new stages along with Industry 4.0. To provide a better understanding of a manufacturing plant, the concept of VC has been adapted to the industry. This has led to a new possibility for the industry regarding the commissioning of projects and the apparition of new techniques that allow testing the different parts of the process. New technologies such as VR sets like HTC Vive, Oculus Rift, among others, have come to stay in this new stage of the industry allowing to simulate the interaction between the operators and the systems, to train them (Tudero and Azkue, 2017) and avoid possible risks that they would be exposed in a real system (Rouse, 2018).

Nowadays companies are constantly developing software for VC using the concepts mentioned before to keep up with competence. The company behind this project already has VC software, but it is looking forward to implement VR for improvement.

1.2. Problem description

The main problem is that the company Simumatik has VC but right now it is not able to use VR in it. Nowadays, the implementation of VR in a simulation system regarding the manufacturing industry is not very common. There exist few companies that specialize in the development and implementation of this feature for any other company that needs it, but what Simumatik is looking for is to implement VR by themselves. This thesis focus on the development and implementation of this feature for Simumatik.

1.3. Aim and Objectives

The project aims to define and implement a 3D virtual human body model, utilizing parameters provided by a 3-point VR system (head and arms). The model is implemented within the Simumatik3D software using Python and is visualized in the software as a simplified human body representation using simple geometric forms. The objectives of the project would be:

- Define a model: Firstly, approach VR to get a better understanding of these new technologies. Then define what and how it is going to be done.
- Implement a model: Start working and implement the ideas that have been defined, fixing errors, and advancing step by step until reaching the final stage.
- Test the results: Check that the results obtained represent faithfully what was initially thought. To test the accuracy of the system, the user will need to feel the feedback of the system and test that the movements in the virtual world are similar to the movements in the real world.

1.4. Delimitations

Programming focused on VR is difficult from a first approach, therefore, the project is going to be limited. Regarding which parts are going to be simulated in VR environment, this project focus on emulating a simplified version of the human body. As there are only three points of reference, head and hands, as input information for position and orientation, the emulation of the lower part of the body is unlikely to be precise.

Focusing mainly on modelling the torso, arms, and head, to be able to simulate possible collisions within the workspace. Accuracy will be limited to simulate the position of the body without considering the position of the fingers.

For the hand, there will be only 3 actions available:

- First action with the hand in rest but being able to push objects and interact with the system.
- Second action with a laser beam pointing forward to interact with the buttons that can be pressed in the simulation.
- The third action will be in charge of grabbing objects of the cell.

To calculate the position of the arms in 3D, assumptions must be made. Two Degrees of Freedom (DoF) of the wrists are going to be fixed to obtain one solution of the position and orientation of the arms. This assumption is necessary to avoid multiple solutions of the arms.

Regarding the shoulders, they are going to be fixed following the same orientation as the head, despite that, the neck is fixed in 4 DoF.

2. SUSTAINABLE DEVELOPMENT

Sustainable development is defined as mentioned in Our Common Future (1987):
Sustainable development is the relationship that meets the needs of the present without compromising the ability of future generations to meet their own needs. It contains within it two key concepts:

- *The concept of 'needs', in particular the essential needs of the world's poor, to which overriding priority should be given; and*
- *The idea of limitations imposed by the state of technology and social organization on the environment's ability to meet present and future needs.*

It can be mainly divided into 3 different parts, environmental, economical, and social. The relation between each one of them can be represented in a Venn Diagram (Figure 1).

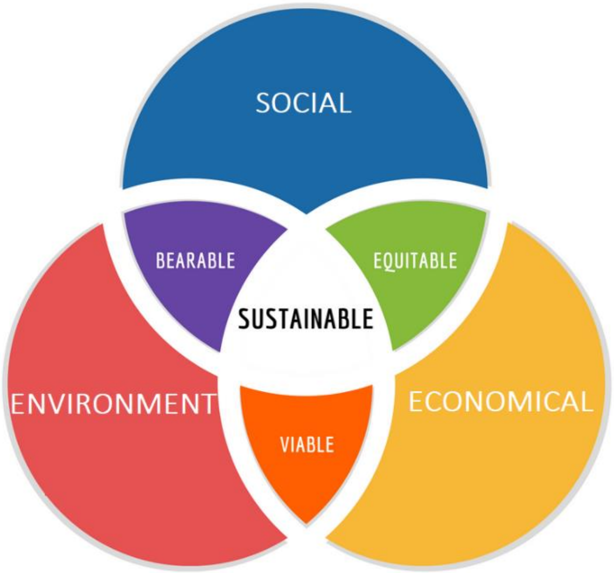


FIGURE 1. VENN DIAGRAM FOR SUSTAINABLE DEVELOPMENT

2.1. Environmental Sustainability

The objective of environmental sustainability is about taking care of the resources that the world provides, by not wasting them unnecessarily. The minimum natural resources (Figure 2) that should be left for the future generations should be the same quantity and type that the previous generation left. Also, it should be enough so they can have the same quality of life that we have, including meeting their basic needs (Wangel and Leap, 2015). Our project is favoring environmental sustainability because it is saving energy, materials and time by not depending on the physical resources to simulate any kind of situation in the workplace.

FIGURE 2. WINDMILL GENERATING CLEAN ENERGY

2.2. Economic Sustainability

For economic sustainability, the goal is to achieve economic growth using the resources optimally but without harming any other, including harming the environment. In the case of using VR, by emulating real-life situations in an industrial environment, it eliminates wastes such as leftovers, the energy necessary to power the plant and the commissioning of the process.

2.3. Social Sustainability

It could be said that social sustainability is in relation with economical sustainability, in the way that if a society has economic growth, it would mean that it could spend more money in education and health, so it would raise human productivity (Our common future, 1987). Also, in the case of this project, implementing VR means that the people would not need to be exposed to danger by testing the possible hazards of the workplace. VR could allow the operator to train without the fear of physical harassment and feel confident in every decision it makes during the training knowing that if he/she makes a mistake, it will not impact the company or his/her surroundings.

3. FRAME OF REFERENCE

3.1. Industry 4.0

The industry is constantly evolving and nowadays we are facing a new stage of this economic sector, Industry 4.0. The automation of industrial processes, due to the digitalization of the tools and manufacturing processes such as simulation software, allows creating accurate and reliable virtual systems. The industry is, currently, going through significant transformations such as the optimization of the connections between computers and machines and the control of the flow of data regarding those systems (Marr, 2018). Computers and electronic systems are connected and are autonomous in terms of making decisions without human interaction involved with it. The internet of things and systems will make factories more efficient regarding productivity and reduction of wastes. The connection between the different systems within the factory and the proper transmission of information is the key to industry 4.0. This new stage of the industry could impact businesses that do not adapt to this trend. Preparing for a future in which smart machines improve and add quality is one of the goals that a company should aim for, having industry 4.0 in their scope. Possible applications are (Marr, 2018):

- Identify opportunities: connected machines collect huge volumes of data that can inform about maintenance, performance, and possible issues, as well as analyze that data to identify patterns and insights that would be impossible for a human to do in a short and reasonable time.
- Optimize logistics and supply chains: a connected supply chain can adjust and accommodate when new information is presented.
- Autonomous equipment and vehicles: there are shipping yards that are leveraging autonomous cranes and trucks to streamline operations as they accept shipping containers from the ships.
- Robots: in the past decades it was only possible for large enterprises with equally large budgets. Robotics are now more affordable and available to organizations of every size.
- Internet of things and the cloud: a key component of this industry is the internet of things that is characterized by connected devices. By combining these connected devices with automated systems, it is possible to gather information, analyze it and create an action plan.

3.2. Virtual Commissioning

VC is the creation of a 3D simulation model of a manufacturing system to test the system or program before building and running it in the real system (Turnbull, 2019). Commissioning is one of the most important parts of the process and usually happens at the end of the process. This is when most of the delays occur even if it is only 25% of the entire development time (Hoffmann, Maksoud, Schumann and Premier, 2010). Errors in the control software are predicted to cost up to 70% of the delays (Reinhart and Wünsch, 2007). Thus, the goal of VC is an early validation of the logic of the device, the code and the reduction of the risk of launching a program with errors. The most time consuming is spent in building/implementing the software which will be in charge of the VC. Implementing VC increases quality software and reduces commissioning time. Virtual prototypes allow manufacturers to test control software along with other engineering phases (Reinhart and Wünsch, 2007) which reduces the risk of errors in upcoming processes. VC allows to test in virtual environments, improving software quality and increasing safety for employees, furthermore, reduces the risk of damaging machinery and waste energy, raw material, and time.

The benefits of VC are the following (Reinhart and Wünsch, 2007):

- Faster, in terms of testing and debugging.

- Cheaper, VC tools are way less expensive than building entire systems.
- Safety, due to trying all kinds of changes and upgrades in a virtual environment.
- Target scenarios of failure situations.
- Independency from the physical system’s location and installation time.

In this way, errors and wastes are detected earlier and reduced or eliminated. It is beneficial for manufacturing in which automation of a high level of complexity is involved.

To make the VC more reliable, it is necessary to collect detailed information about the exact placement of all the resources as well as having the real PLC programs and Input/Output signals.

3.3. Virtual Reality

VR is the use of computer technology to experience a world that does not exist. Nowadays, new computing technology allows creating an immersive experience for the user in a 3D simulated environment in which the user can navigate and interact with virtual features and items. VR is "a multi-dimensional human experience which is totally or partially computer generated and can be accepted by those experiencing the environment as consistent" (Seidel and Chatelier, 1997). The most immediately recognizable component is the head device.

People are visual creatures. The display will typically be shown between your eyes, splitting the 3D effect and stereo sound. Commonly, VR incorporates visual and audio feedback, but it can also allow the user to receive other types of feedback, like forces or vibrations, depending on the technology that it is used. Immersion and believable experience will be held by the combination of the input tracking systems and output signals such as images and sounds displayed by the VR peripheral (Shiratuddin and Zulkifli, 2001).

There is a concern about if the VR can damage your eyes. The paper "Some effects of using VR technology" (Seidel and Chatelier, 1997) discusses the effects that VR has on the user. Conflicts of the senses lead to uncomfortable and confusable situations, some symptoms depend on technological factors such as screen resolution, speed of the simulation, etc. The answer to possible damages is that it is like other screen technology, people tend to blink less so the eyes become dry faster, otherwise, there is no special risk about getting the eyes damaged.

There exists what is called Cybersickness, it occurs when people experience nausea and discomfort when using VR systems. This technology is a double-edged sword as it has positive and negative effects regarding cognition and emotions, the boundary between actual reality and VR is narrow. Some people experience headaches or nausea because their brain thinks that they are standing while the VR shows images in movement. Also, our eyes use to focus and converge depending on the distance of the object that they are looking to, but in VR, the distance is always the same. By using the VR system for so long, you can experience Cybersickness, so it is recommended to have a break and not use it without resting.

3.4. Available technology for tracking movements for VR

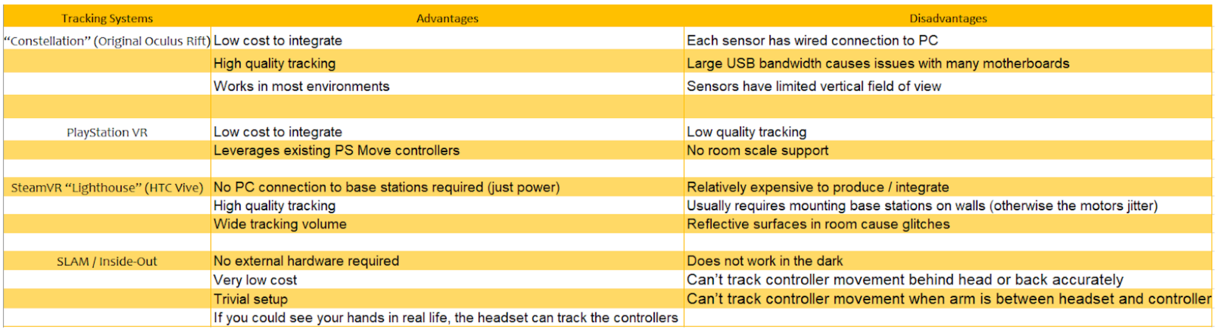
Nowadays, VR is a frontier, breakthrough technology that has been introduced in the past years. Different approaches to this incoming technology are appearing in the hands of several companies. Hence, as every brand is developing its unique VR System, there is no standardized methodology in which VR works. Currently, it is possible to find the same product but with different names and their working method. Six DoF allow you to move in all three axes, and rotate around them also.

Nowadays, there exist in the market optical and non-optical tracking technologies. Optical uses image sensors, and non-optical uses other kinds of sensors, even the more advanced options, use sound waves or magnetic fields. Most modern tracking systems use both combined. Here they are in more detail:

- The optical methods use cameras. The user should wear optical markers around the body or in strategical places. These markers may be passive or active. It is usually used the passive because you need to have the LEDs powered in an active marker.
- The non-optical methods typically use gyroscopes, accelerometers, and magnetometers. They measure the 360 degrees rotation, the XYZ axis movements, and the North direction respectively.

The combination of non-optical and optical can achieve incredibly precise motion tracking data. It is possible to find the following technologies for implementing VR in the market (Table 1):

TABLE 1. Differences between tracking systems



Tracking Systems	Advantages	Disadvantages
"Constellation" (Original Oculus Rift)	Low cost to integrate	Each sensor has wired connection to PC
	High quality tracking	Large USB bandwidth causes issues with many motherboards
	Works in most environments	Sensors have limited vertical field of view
PlayStation VR	Low cost to integrate	Low quality tracking
	Leverages existing PS Move controllers	No room scale support
SteamVR "Lighthouse" (HTC Vive)	No PC connection to base stations required (just power)	Relatively expensive to produce / integrate
	High quality tracking	Usually requires mounting base stations on walls (otherwise the motors jitter)
	Wide tracking volume	Reflective surfaces in room cause glitches
SLAM / Inside-Out	No external hardware required	Does not work in the dark
	Very low cost	Can't track controller movement behind head or back accurately
	Trivial setup	Can't track controller movement when arm is between headset and controller
	If you could see your hands in real life, the headset can track the controllers	

There exist two ways to track the whole body, HTC VIVE Tracker Pucks (VIVE only) or Kinect from Microsoft. Xbox Kinect depends on the headset in use, it is limited as its quality depends on how well it can detect you from certain angles. The more expensive way is to use the HTC VIVE Trackers, they are designed to be used as accessories.

Usually, a VR system works with 3 tracking devices that are enough to track the position of the head and the hands. However, it is possible to emulate the full body in different ways, for example, by wearing a suit with many sensors to track every movement of the entire body, but it is also possible to estimate the position by adding three more tracking devices, two of them should go to the feet and the other one to the waist (Figure 3). Making sure that the trackers are always visible to the cameras. The tracker on the hip can be placed anywhere around the middle of the body, being careful not to cover it with your clothes. With the information provided by six devices, it is possible to emulate a lot of complex movements such as walking, gestures and turning around respect to the waist.

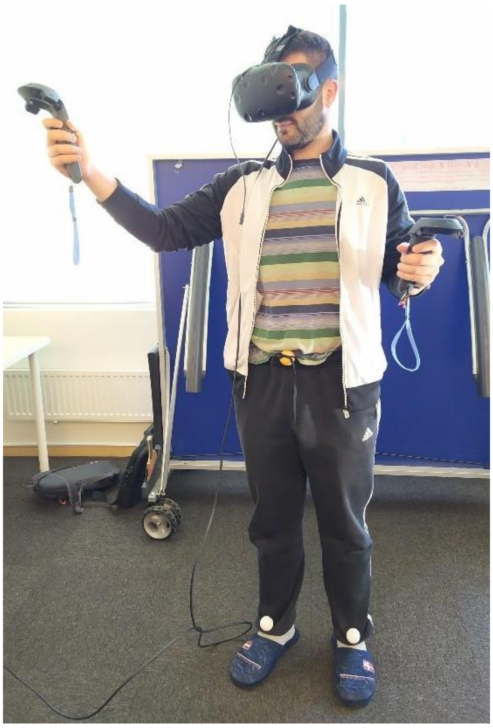


FIGURE 3. POSITION OF SENSORS FOR FULL-BODY TRACKING

3.5. Euler angles and Quaternions

Regarding computer graphics, robotics, and aviation, transformation matrices are used to express the position of a point in the space (translation) and its orientation (rotation). The representation of orientation in space is a complex issue. Euler's rotation theorem states that, in 3D space, any displacement of a rigid body in such a way that a point on the rigid body remains fixed is equivalent to a single rotation about an axis that passes through the fixed point. Accordingly, such rotation can be described by three independent parameters: two for describing the axis and one for the rotation angle. Orientation in space, however, can be represented in several other ways, each with its advantages and disadvantages. Some of these representations use more than the necessary minimum of three parameters. Euler angles are a set (or rather a sequence) of three angles, which can be denoted for example by ϕ, θ, and ψ. (Often, Euler angles are denoted by roll, pitch, and yaw) as seen in figure 4:

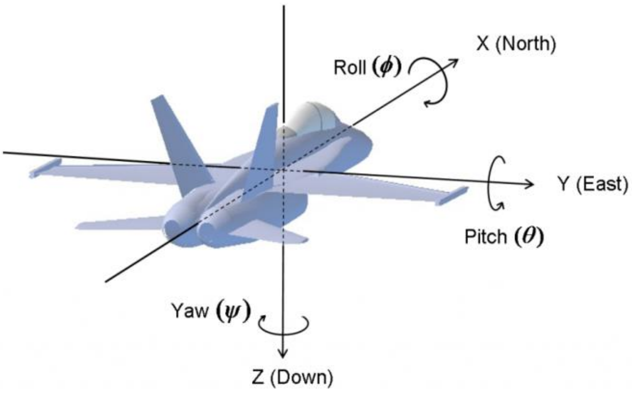


FIGURE 4. THE INERTIAL FRAME

A quaternion is a four-element vector that can be used to encode any rotation in a 3D coordinate system. Technically, a quaternion is composed of one real element and three complex elements, and it can be used for much more than rotations.

The general form to express quaternions is

$$q = s + xi + yj + zk \quad s, x, y, z \in \mathrm{R}$$

Where, according to Hamilton's famous expression

$$i^2 = j^2 = k^2 = -1$$

And

$$ij = k \mid jk = i \mid ki = j$$

$$ji = \text{-}k \,/\, kj = \text{-}i \,/\, ik = \text{-}j$$

The relationship between i, j, k is very similar to the cross-product rules for the unit cartesian vectors:

$$x \times y = z \,/\, y \times z = x \,/\, z \times x = y$$

$$y \times x = -z \mid z \times y = -x \mid x \times z = -y$$

Hamilton also recognized that the i, j, k imaginary numbers could be used to represent three cartesian unit vectors i, j, k with the same properties of imaginary numbers, such that

$$i^2 = j^2 = k^2 = -1$$

Euler angles are more human-understandable and good for decomposing rotations into individual degrees of freedom (for kinematic joints) but have disadvantages like ambiguity and gimbal lock. In practice, quaternions are used more often as it is easier to calculate with computers and more efficient. To sum up, three rotations are multiplied together in a specific order to avoid Gimbal lock when rotating by Euler angles, however, a Quaternion is only one rotation, conversion from quaternion to a matrix is quite efficient.

3.6. Kinematics

The behaviour of physical systems in many situations may be better expressed with an analytical model. Kinematics in robotics is a statement form about geometrical description of a robot structure. Geometrical equations show the bound between joints in spatial geometry on a robot with ordinary coordinates to determine the position of the end effector or the final position of an arm. Kinematics is the relationship between position, velocity, and acceleration of the different links of the robotic arm. The objective of kinematics is to define a new position from its original frame of reference or coordinates (Rehiara, 2011).

3.6.1. Forward Kinematics

The problem that forward kinematics has to cope with is finding the position and orientation of the end-effector as a function of its joint angles. There exist two methods for building forward kinematics:

- Graphical Method (Figure 5)

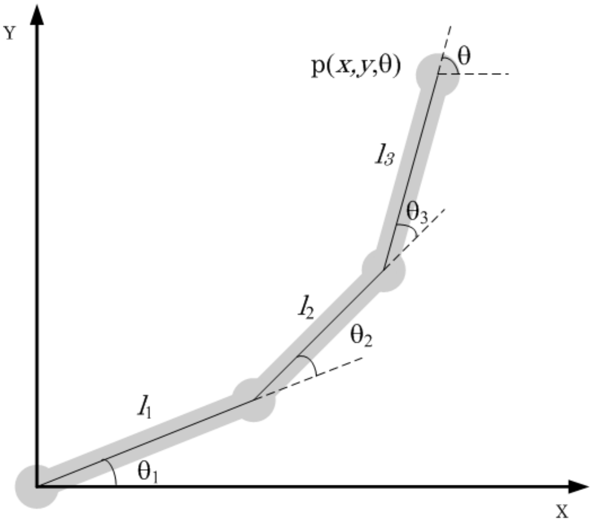


FIGURE 5. GRAPHICAL METHOD, FORWARD KINEMATICS

- D-H convention

A method of assigning coordinate frames to the different joints of a robotic manipulator. The method involves determining four parameters to build a complete homogeneous transformation matrix.

The rotational matrix is a homogeneous 4x4 transformation matrix which describes the position of a point or an object and the orientation of the object in 3D space. The homogeneous rotation matrix (Figure 6) (*alpha* and *theta*, are angles of rotation around the axes, respectively):

$$Rot = \left[\begin{array}{ccc|c} \cos\theta & -\cos\alpha\sin\theta & \sin\alpha\sin\theta & 0 \\ \sin\theta & \cos\alpha\cos\theta & -\sin\alpha\sin\theta & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ \hline & 0\ \ 0\ \ 0 & & 1 \end{array}\right]$$

FIGURE 6. PURE ROTATION MATRIX

And the homogeneous translation matrix coordinates the transformation from the previous frame to the next one, it is given by (Figure 7) (*a* and *d*, are distances in the X and Z axes, respectively):

$$\boldsymbol{Trans} = \left[\begin{array}{ccc|c} 1 & 0 & 0 & \boldsymbol{a} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \boldsymbol{d} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

FIGURE 7. PURE TRANSLATION MATRIX

3.6.2. Inverse Kinematics

The problem that inverse kinematics has to cope with is finding the appropriated joint angles of every link to get a certain position and orientation of the end-effector. Sometimes finding the inverse kinematics solution can be a very tricky task, and sometimes there is no solution to the problem. There are two methods for solving inverse kinematics:

- Geometric Method (Figure 8):

Cosines law can be used with this method. It is recommended to use this method when the problem has three DoF as maximum because if it has more, the mathematical solution becomes very complex.

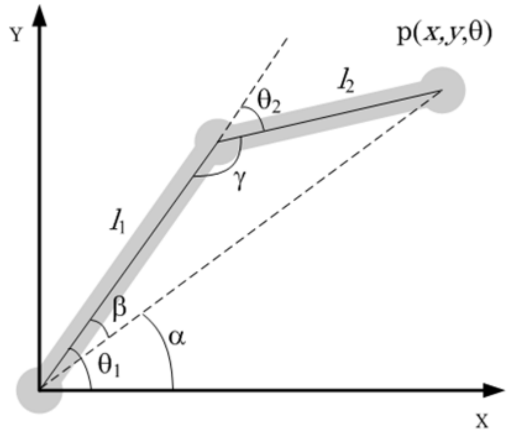


FIGURE 8. GEOMETRIC METHOD FOR INVERSE KINEMATICS

- Algebraic Method

With this method, sometimes the solution is extremely complicated because some equations might be nonlinear, but the advantages are that this method is reliable, it requests less computer cost, so the results are obtained faster. It also allows finding all the possible solutions.

4. LITERATURE REVIEW

A part of the research that has been carried out is about looking for articles, documents or files that could help to build the prediction of the human movements by using Forward and Inverse Kinematics, to model the entire body. Many articles talk about robot arms with many DoF, or human arms but using complex calculations, and did not contain enough data to replicate the system (Parger et al, 2018; Al-Mashhadany, 2011). Besides, some scripts for Human Arm Inverse Kinematics have been found, many of them were not written in Python language, they were precise, but also, complex, and hard to translate into python language. Because all of that, the limitations of the project have been set to estimate only the arms. Their positions will be calculated with the simplest code as possible, to fulfill the requirements. A file in Python language has been found, which contains enough code to create a scene (that could not be changed, or interacted with), and can connect the VR system and see the controllers in the virtual world. This code has been used for this project to get an early approach to python programming in VR.

4.1. Digital Human Modelling

Digital Human Modelling (DHM), is a technique of simulating human interaction with the workplace or product in a virtual environment. This virtual evaluation process is useful in developing user-centered products by incorporating human factor principles at an early design phase, which improves quality and reduces the design time (Maurya, Karmakar & Kumar Das, 2019). Application of DHM has gained attention in the design process of the manufacturing industry, agriculture, healthcare sectors, transportation, aviation sectors, etc. Nowadays, DHM software is being developed, a computer-aided design tool for the construction of 2D and 3D human models from anthropometric data of targeted users/population for ergonomic analysis.

The study of digital prototypes in the virtual environment reduces the developmental cost and design time. In an industrial workplace, DHM has been applied for improving the designs of work cells in car manufacturing plants, designing of small fishing vessels to reduce work-related musculoskeletal disorders of fisherman, redesigning of work accessories for minimizing awkward/uncomfortable postures in Indian shop floor workstation, workplace evaluation of coir industry, etc. (Maurya, Karmakar & Kumar Das, 2019). The previous article showed how valuable is the integration of human interaction into virtual environments to increase productivity and reduce costs.

In the article written by Tolani and Badler (1996), they propose an inverse kinematics procedure for a seven DoF model of a human arm. They show two schemes, in the first one, they specify the elbow position and solve the wrist and shoulder angles, but singularities might appear and also, multiple solutions due to the position and orientation of the arms. In the second scheme, they fix a shoulder or wrist joint, but they still get multiples solutions. This article has been useful to gain the knowledge of fixing joints to get fewer solutions and learning some calculations that might be used.

4.2. Simulation of Human/Robot Collaboration

Human/Robot collaboration is a field that requires safety regulations and configuration to avoid people getting injured or even killed by a robot. The principal aim is to have a human and a robot working close to each other and evade possible collisions that may occur. Due to the possible hazard of this collaboration, it is necessary to have simulation to test the behaviour of the robot when a human is close to it.

A study about safety human/robot collaboration has been conducted by Mainprice and Berenson (2013) in which manipulation tasks are performed simultaneously and close together. The prediction of human workspace computes the learned volume and human motion trajectories throughout the gesture recognition system. Robot trajectories are intended to minimize the penetration cost in the human workspace occupancy meanwhile the robot executes several tasks that occurs in parallel to the human operator. Considering the predicted human workspace occupancy in the robot's motion planner, this leads to safer and more efficient interactions between the user and the robot than only considering the human´s current state.

In addition to this, a robot can be manipulated using a force feedback device. In a continuously moving line in a vehicle factory, the robot will wait in a safe position, until it is ready to be used, then it will move to a comfortable position for the operator so he can pick it and use the screwing tool, while the robot absorbs the forces and torque (Dombrowski, Stefanak and Perret, 2017).

Moreover, regarding human/robot collaboration, there is a project that aims at inherent design and control throughout the simulation by means of power and force limiting, it determines how to demonstrate the force and pressure in case of a direct collision between humans and robots in a safe industrial environment (Dombrowski, Stefanak and Reimer, 2018).

The previous articles provide insight for this project, in which it is expected to have a robot that can detect the presence of a human by using distance sensors and stop if necessary. For this, it is necessary to have the body built and be able to move inside the simulation with the VR system, to make the sensors work and test dangerous situations in which the human would be exposed to.

4.3. Simulation of the assembly process and operator training

The article written by Shiratuddin and Zulkifli (2001), talks about the definition of VR and Virtual Manufacturing, and the applications of VR in manufacturing, such as training, assembly, and design. It talks about the benefits that exist and the issues that might solve. In design, when building a factory, it is important to choose the placement of the machinery wisely, because an incorrect decision can make a plant inefficient. The simulation of the factory allows you to try different configurations until finding the most efficient one. In a simulation, it is possible to test the failures, the cycle time, the work in progress and many useful statistics that do not necessarily require the physical factory. This article has useful information to get an early approach to all these concepts.

A study has been carried out by Al-Ahmari, Abidi, Ahmad and Darmoul (2016), about creating a fully functional virtual manufacturing assembly simulation system that solves most of the issues related to VR environments, such as data-translation, integration of different hardware, software systems and the collision detection in real-time. They create a virtual system that provides visual, auditory, tactile and forces feedback to use it on training operators for the assembly process and evaluating assembly operations alternatives. This study has shown the importance of the implementation of VR for assembly processes due to the benefits that it brings.

As it is mentioned by Tao, Leu and Lai (2019), the 3 main parts of a Manufacturing Assembly Simulation (MAS) are:

- Modelling: To develop a MAS, there is a need to create a 3D model of the objects, that needs to be accurate in dimensions, realistic in physics and functional as real parts in the virtual environment.

- Pose Estimation: In VR, it is needed to track the position of the body, to be able to change the view and make the body act as it moves in the real world.
- Interface and Interaction: Using all kinds of sensors can improve the experience to be fully immersive. Besides, the interaction with the virtual objects should be like manipulating the objects in the real world.

This project is focused on the pose estimation, an important part in a MAS, and the more accurate it is done, the more immersive experience it is achieved.

To achieve the optimal production efficiency, operators must receive a correct apprenticeship. High-quality and non-interrupted services are required for production lines; therefore, skilled operators are important. OT is, nowadays, necessary for industrial health. In the paper "Design and Operation of a Virtual Reality Operator-Training System" (Okapuu-von Veh et al., 1996) a study was conducted to test the benefits of a correct instruction of the staff. Power system operators have usually to deal with situations where their capacity of memory, their ability to put into practice their theoretical knowledge and their capacity to overcome continuous stress situations are continuously tested. The use of VR systems for training employees has been widely used and considered for companies due to the good results given. Some skills are learned better by experiencing it in a realistic environment, instead of reading it in a book, or a demonstration. Besides, there is a close relationship between learning capability and physical perception, VR can replicate the entire facility allowing each trainee to see how a product moves through the manufacturing system. Moreover, it allows the user to be able to experience extreme situations that would be dangerous or expensive in real-life systems.

5. METHODOLOGY

Following a methodology is important to be able to organize the information and conduct a research that adapts to the project. There are two main approaches to collect and analyze the data, the qualitative and quantitative (Oates, 2006; Streefkerk, 2019).

Quantitative research is about conducting surveys, experiments, and observations to collect the information, but that it is not the research that is going to be carried out. Therefore, qualitative research is going to be the method to work with. Collecting concepts, thoughts, and knowledge about the main objective, which is the implementation of VR to train operators in a safe workplace.

The research strategy called Design and Creation (Oates, 2006) is going to be followed for this purpose (Figure 9). This strategy uses an iterative process for a problem-solving approach. It is divided into 5 different steps that need to be followed in the right order to get an acceptable solution, but it is possible to jump to a previous step if needed in case you find out new information or new ideas.

The steps of the strategy Design and Creation (Oates, 2006) are the following:

- The awareness is the recognition and articulation of a problem, which can come from studying the literature where authors identify areas for further research or from new developments in technology.
- The suggestion involves a creative leap from curiosity about the problem.
- The development is where the tentative design idea is implemented. How this is done depends on the kind of IT artifact being proposed.
- The evaluation examines the developed artifact and looks for an assessment of its worth and deviations from expectations.
- The conclusion is where the results from the design process are consolidated and written up, and the knowledge gained is identified, together with any loose ends.

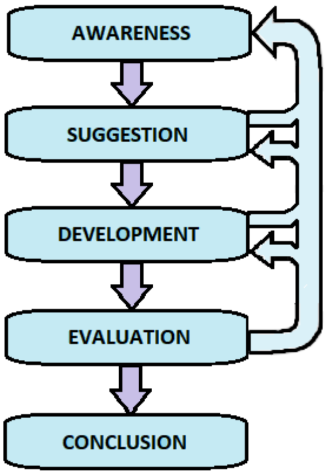


FIGURE 9. DESIGN AND CREATION METHODOLOGY

Awareness, in this case of study it comes from Simumatik Open Emulation Platform which is expressing a need to introduce interactive VR into their simulated environment. Afterwards, the suggestion for this project is to introduce a simplified version of the human body to be able to interact with the simulation. Following the methodology, development in this project will be conducted through Python programming to estimate the position of the upper parts of the body, having only three points as a framework. Once the upper part of the body has been modelled a comparison is needed to evaluate the model behaviour. It is possible to check that a distance and an angle in VR corresponds with the ones in reality. When creating a working cell in Simumatik, the floor is generated with one square meter, therefore, it was possible to measure the distance when walking between squares. If you walk from the edge of a square to the following one, the distance the user has walked in reality is exactly one meter. Finally, the conclusion of the project must be displayed to show that the actual results follow the requirements of the company. The process diagram of this project is shown in Figure 10.

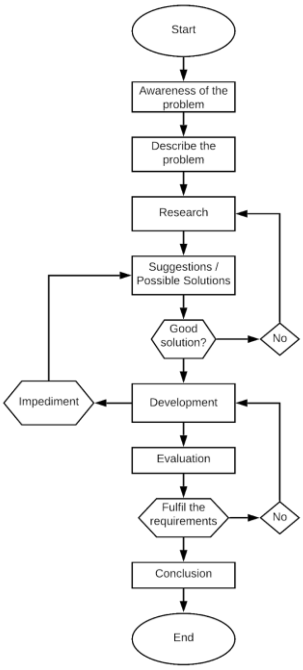


FIGURE 10. PROCESS DIAGRAM OF THE PROJECT

The suggestion is an important step, as it proposes the main idea about how to approach the problem and develop the project. To get the best idea, here are some tips that are recommended to follow:

On the one hand, it is important to follow certain steps to find quality and credible sources for the research (Wilczek, 2011). For example, being organized at the beginning prevents repeating searches in the same resources and will save a lot of time.

On the other hand, identifying the main concepts in the project (Ohio State University, 2018), avoiding research in general in the area of VR. Furthermore, looking for specific aspects that are directly related to the main purposes of this project and those that are a matter of interest. Body modelling in VR will be completed throughout products that are already in the market such as sensors or jackets. Another way to approach our goals is by looking for code that is used in VR programs. Besides, OT, but in an industrial environment to apply to our objectives.

Finally, when everything is clear and understandable, research is going to be conducted. Being careful, considering that not all the information available on the internet is reliable. Thus, when conducting the research, critical thinking must be our most important tool to identify which information can be useful and which not (McLean, 2012). Later, information obtained will be evaluated throughout asking several questions, to do it, it is going to be used journalism's Five W's and One H (Wilczek, 2011). Once the reliability of the information is confirmed, then focus on analysing if the information found was as expected, and if it is not the case, ask why and try to do it better for the next research.

6. DEVELOPMENT

6.1. Initial considerations

In this project, a study has been conducted about how to model a human being as accurate as possible by using the position and orientation of the head, and the hands. To make the code simple, it is needed to make some assumptions to calculate some parts of the body, because the human being is complex, and it is possible to have multiple positions of the body, by having the same position of the head and the hands.

The assumptions that are made are the following:

- The neck is unable to turn to the sides, it is fixed respect to the shoulders.
- The wrists are fixed, so it is possible to move the hands, but not to rotate them.
- The torso and the legs are fixed, following the position and orientation of the head.

The first assumption has been made because there is no way of knowing exactly the position of the shoulders without any extra sensor, so by fixing the neck, the shoulders will always point in the same direction of the head.

The second assumption exists because the human arm has many DoF, and by fixing some of them, it is possible to estimate the position of the arm more accurately. This assumption has been made also because it is supposed that a worker usually does not do complex wrist rotations.

The third assumption is due to the lack of information coming from these parts of the body. As there are no sensors located in the lower part of the body, their representation will be dependent on the input of the VR set.

The arms of the human that this project tries to digitally model are going to be treated as robotic arms, therefore, the approaches made in this project focus on the implementation of the mathematics related to kinematics used in robotics. The elbow is a complex point that must be found throughout forward and inverse kinematics, working together to find it. Forward kinematics from the base (head) to the shoulders and inverse kinematics from the end effector (hand) to the shoulders. The torso and legs have been built using forward kinematics.

6.2. Implementation of the DHM

At the very beginning, it was not possible to implement VR in Simumatik as this tool was not available at that moment, therefore a library called HARFANG VIRTUAL REALITY was used instead. This library allows creating a scenario with some geometries that are not interactive but mirrors the movement of the hand's devices. HARFANG includes several tools to introduce code, creating new stuff within it. These tools were useful since they allowed to paint geometries in places of interest to simulate the position of the body (Figure 11).

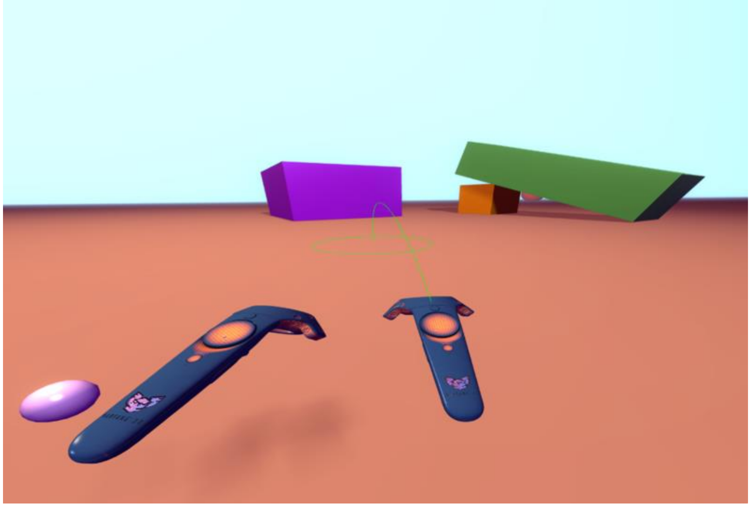


FIGURE 11. TESTING OF INITIAL CALCULATIONS

HARFANG ENVIRONMENT creates a scenario in which you can find some geometries. The user can point with a laser beam that pops out from the virtual controllers and by pressing the trigger button it is teleported to that location. The geometries do not have collisions so you can only trespass them.

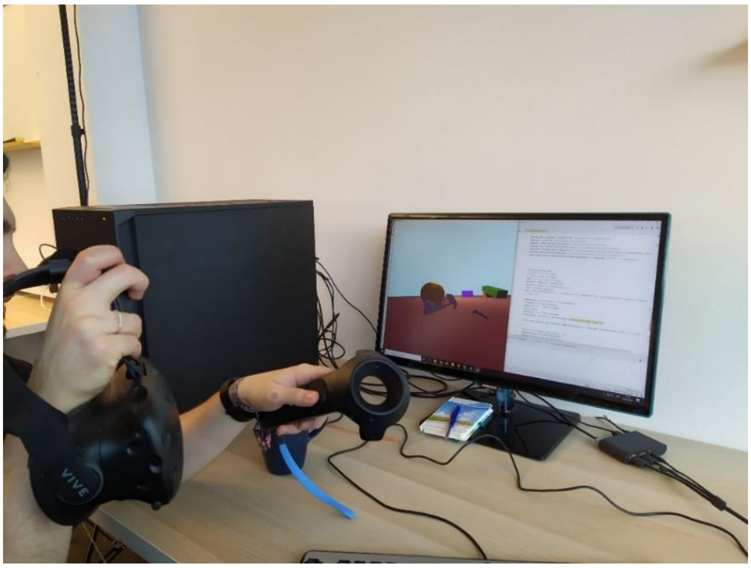


FIGURE 12. VR TESTING

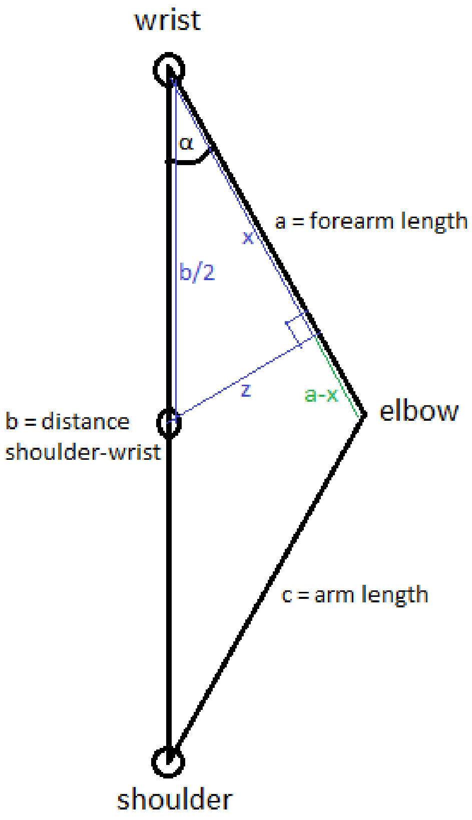
As a first approach with HARFANG environment, the objective was to understand the tools that this library provided such as teleporting through the scenario or creating new geometries. Once the library was mastered, spheres were created, throughout kinematics to represent points in the space in which the current parts of the body should be (Figure 12).

The main functionalities that have been used in HARFANG are:

- Creating a sphere with an offset in an axis.
- Creating a sphere following the position and orientation of the device.
- Creating a sphere in the middle of two points.

By using these functionalities, the neck and shoulders were created in fixed positions following the position of the head, as well as the wrist follows the position of the hand. The elbow sphere was initially created in the middle point between the wrist and the shoulder.

As a first approach for the calculus of the position of the elbow, an important feature was missing. The input information of the hands and the head, coming from the devices, includes position and rotation which was not taken into account from the very beginning. Therefore, when building the corresponding transformation matrices and not including that information, the resultant position of the elbow was not correct. Then, the final solution has been conducted through trigonometry and more specifically with the cosine theorem. Knowing the length of the arm and the forearm, it is possible to estimate the position of the elbow depending on the distance between the wrist and the shoulder (Figure 13).



$$\alpha = \operatorname{acos}\left(\frac{a^2 + b^2 + c^2}{2 * a * b}\right)$$

$$\sin(\alpha) = \frac{z}{b * 0.5}$$

$$z = b * 0.5 * \sin(\alpha)$$

$$\cos(\alpha) = \frac{x}{b * 0.5}$$

$$x = b * 0.5 * \cos(\alpha)$$

Distance to move the point in both axis:

$$distA = z$$

$$distB = a - x$$

FIGURE 13. ELBOW CALCULATION

Next, taking into account the corresponding distances for the translations and being careful with the input rotation, new translation matrices were built. The resulting position for the elbow is checked moving the sphere in this new position. The translations were made according to the axis of the device.

Once Simumatik VR tool was available, the development of the project was imported from HARFANG environment to Simumatik´s. This meant translating libraries for the code to work exactly in the same way as they used to. Previously, in HARFANG environment the implementation of the position of the elbow was fulfilled, therefore, it was necessary to check if that solution could be correctly implemented again in Simumatik. Finally, the behaviour of the position of the elbow was successfully achieved but there was a problem with the translation. Regarding the initial conditions and ergonomics, there was missing a final translation which was primarily rotation. The elbow had between 30º and 45º degrees offset. Finally, throughout quaternions, a rotation matrix was built to get rid of that offset. This is the last part for creating the elbow in VR (Appendix A).

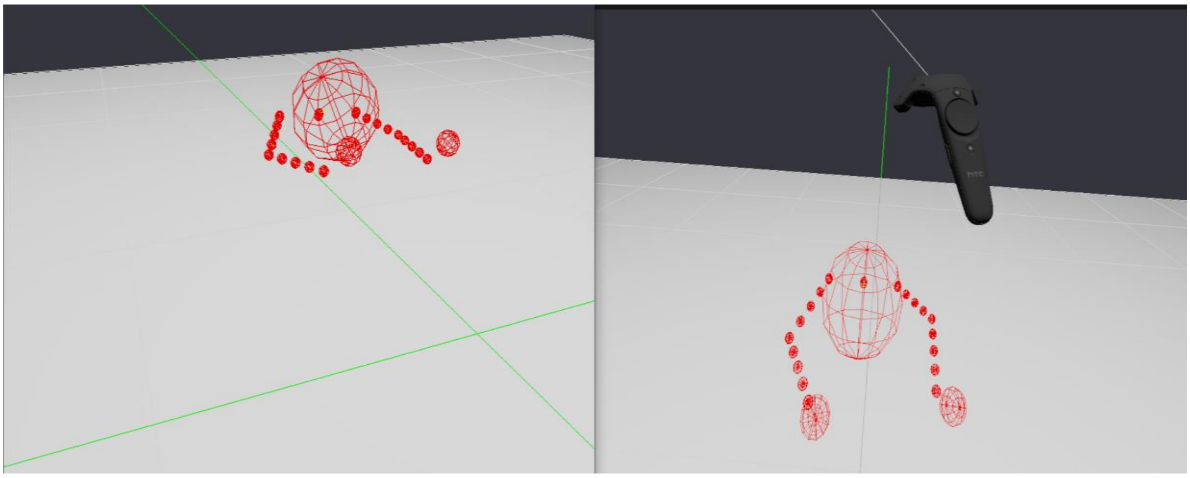


FIGURE 14. THE INITIAL REPRESENTATION OF THE ARMS USING SPHERES

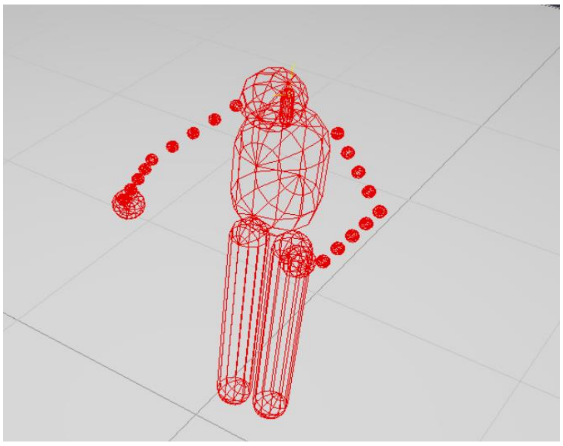


FIGURE 15. INTRODUCTION OF THE TORSO AND LEGS

The goal was to implement cylinders as forearm and arm, but some problems appeared as it was complex to get the rotation for them. Therefore, in the beginning, several balls were created to visualize the entire arm (Figure 14). After that, the torso and legs were implemented following the movement of the neck position, translated in the Z-axis (Figure 15). Estimation of the measures is as seen in Appendix A.

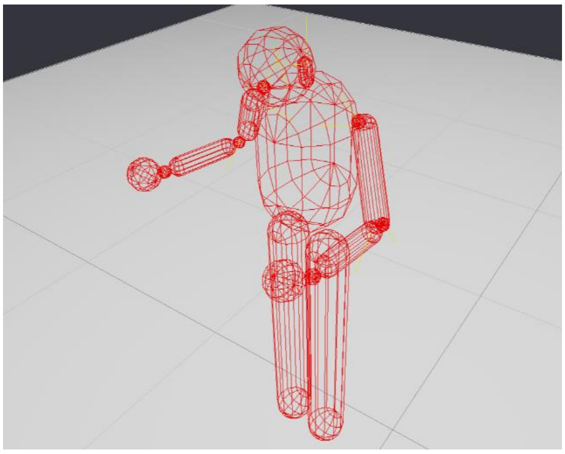


FIGURE 16. COMPLETE DHM

Finally, to create the cylinders, it was needed to find a rotation that had one of the axes, oriented from one point to another. The solution was achieved (Figure 16) after doing projections, calculations, rotations, and translations. To do this, it was necessary to use graphical examples of what it was planned to do, as it is difficult to imagine several vectors moving around the space.

Therefore, using computer-assisted design, experiments with projections were conducted. The main objective was to know the angle between the original vector that was created between the wrist and elbow and its projections in the planes XZ (Figure 17) and XY (Figure 18).

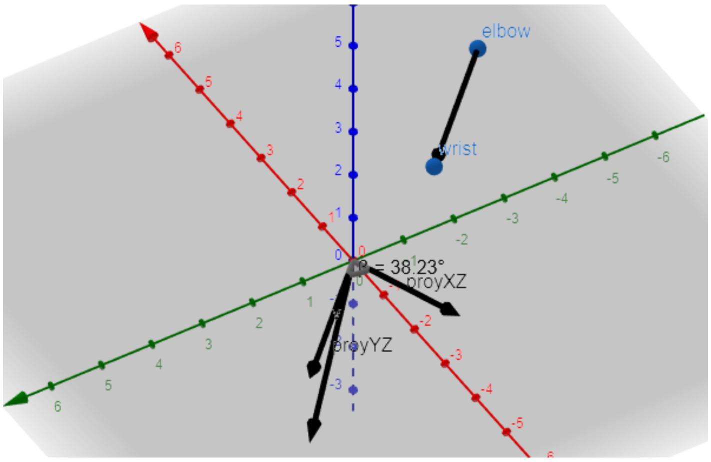


FIGURE 17. PROJECTION XZ

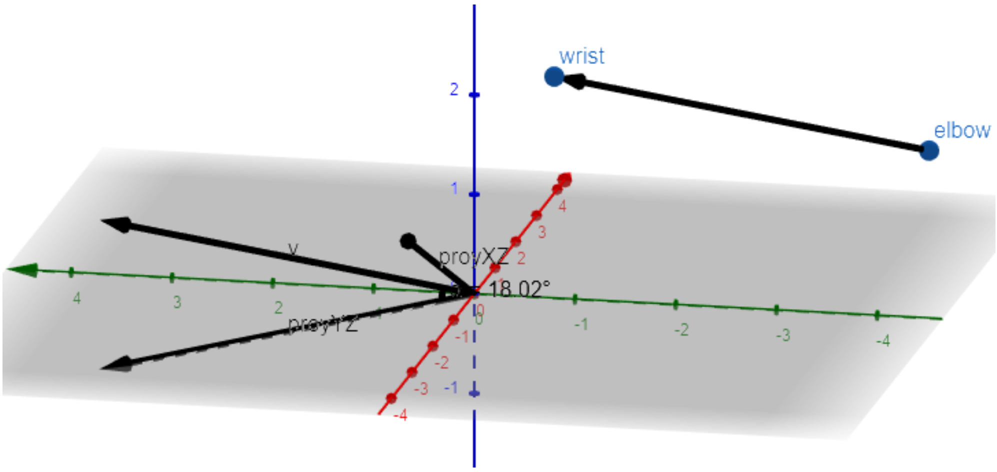


FIGURE 18. PROJECTION XY

Once these angles are known, it was possible to do two rotations in a specific order. Firstly, rotate on the X-axis (Figure 19), and secondly on the Z-axis (Figure 20). Besides, this approach for the orientation of the forearm only worked in 1 quadrant, so it was needed to introduce some conditions to solve this problem.

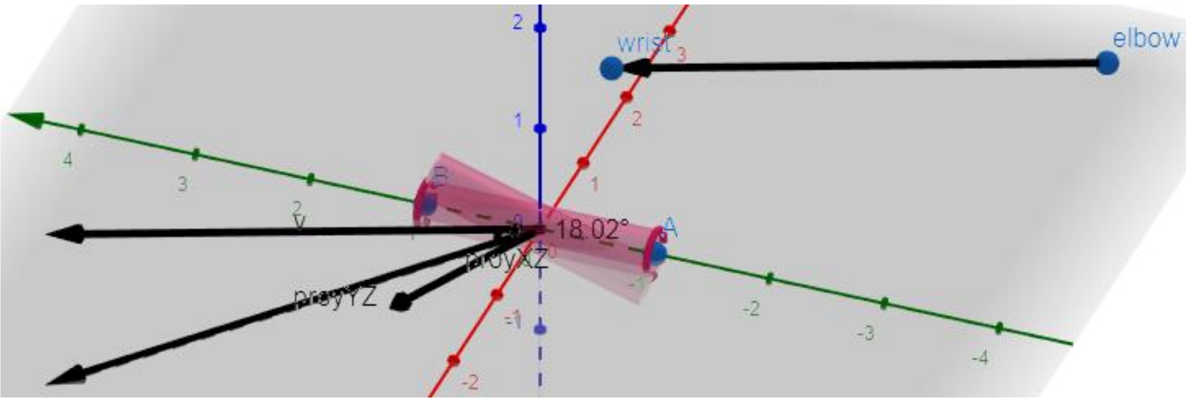
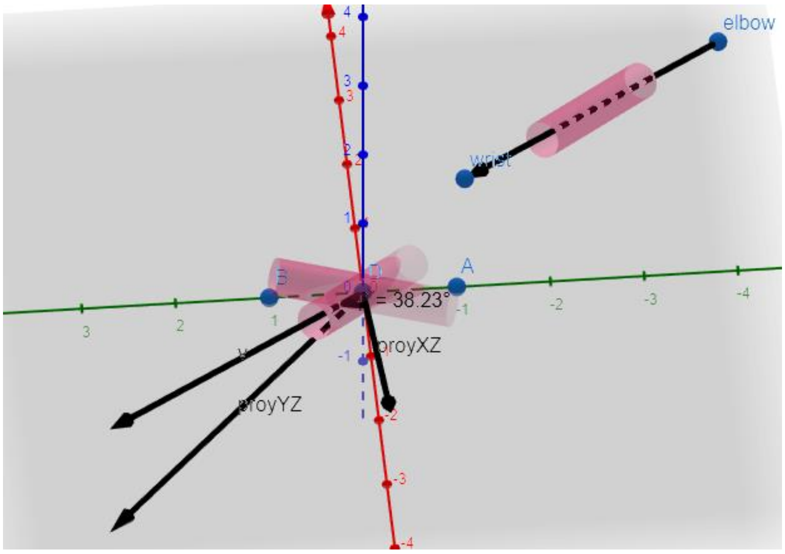


FIGURE 19. FIRST ROTATION ON THE X-AXE



Once the orientation of the cylinder is correct, the last step is to do the translations from its original position to the one of interest. These calculations are applied continuously considering the corresponding quadrant. However, there is a mismatch when the vectors are in parallel with one of the axes.

FIGURE 20. SECOND ROTATION ON THE Z-AXE AND TRANSLATION

6.3. VR Demonstrator

In parallel to the previous development of the project, a demonstrator was built in Simumatik (Figure 21), alongside Codesys and RobotStudio, which are necessary to run the cell and the ABB Robot. This demonstrator aims to simulate a working environment in which an operator can interact with the objects of the cell. Sensors can detect the operator, therefore, if the operator gets in the range of the robot or is placed somewhere it should not be, the system would detect it and react to that, halting the operations working at that moment. This feature is useful to see if the system has been programmed correctly, to check the VC and improve operator's safety. Also, in Simumatik, it is possible to have more than one operator working at the virtual cell, so they will be able to see each other and interact with the same system at the same time.

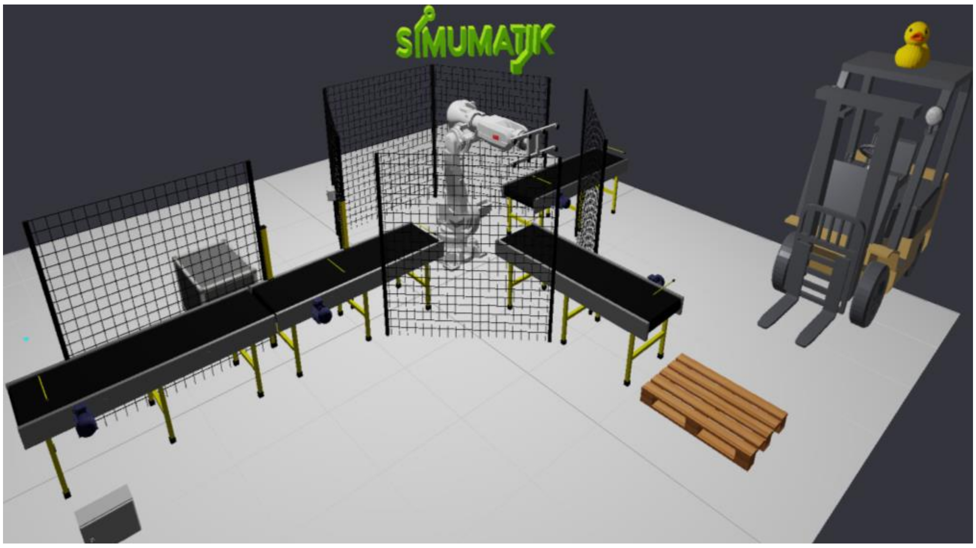


FIGURE 21. VR DEMONSTRATOR

The system's behaviour is simple and works as follows:

Firstly, a product spawns in the first conveyor, this conveyor transports the product until the control stage for the operator to check its quality. Once the product has arrived at the control stage (Figure 22), the operator has to decide if the product fulfils the quality specifications (Figure 23). If the product is good enough the operator only has to leave the product in the same place, but if not, it has to press the panel button (Figure 24) located at the left and leave the product in the conveyor.

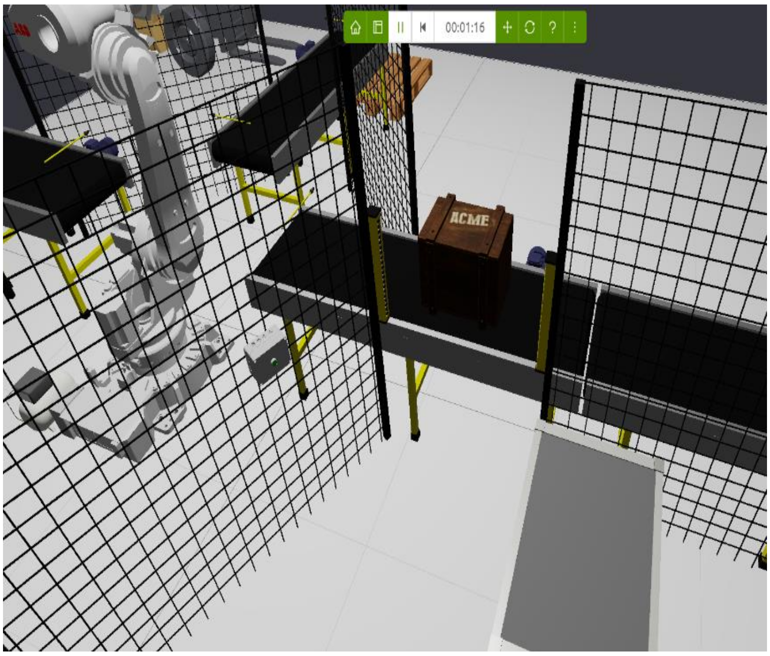


FIGURE 22. CONTROL QUALITY STATION

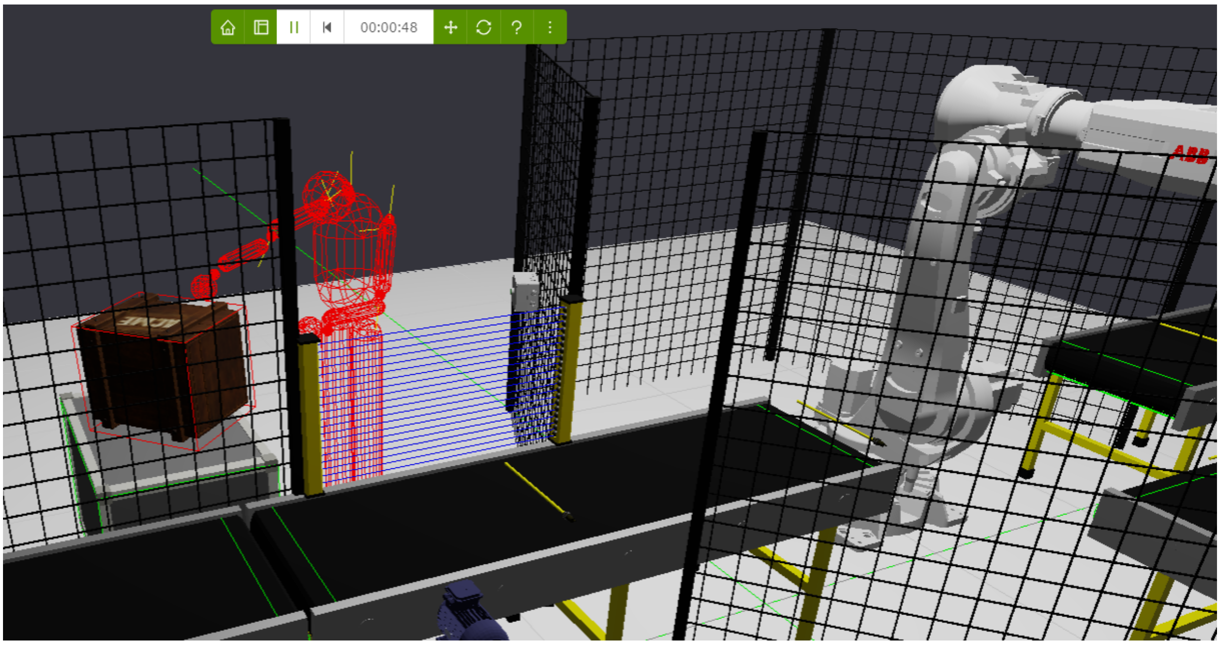
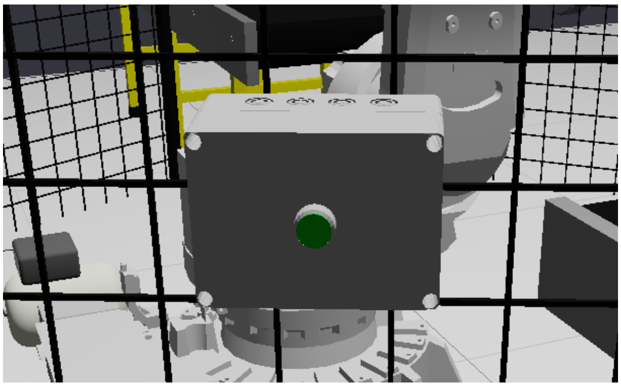


FIGURE 23. DHM IN THE WORKSPACE



Finally, if the button has been pressed, the robot will put the product on the conveyor as seen in Figure 26, and the operator will put it on the bin. If the button has not been pressed, the robot will put the product on the other conveyor as seen in Figure 25, and the operator will put it on the pallet.

FIGURE 24. PANEL BUTTON

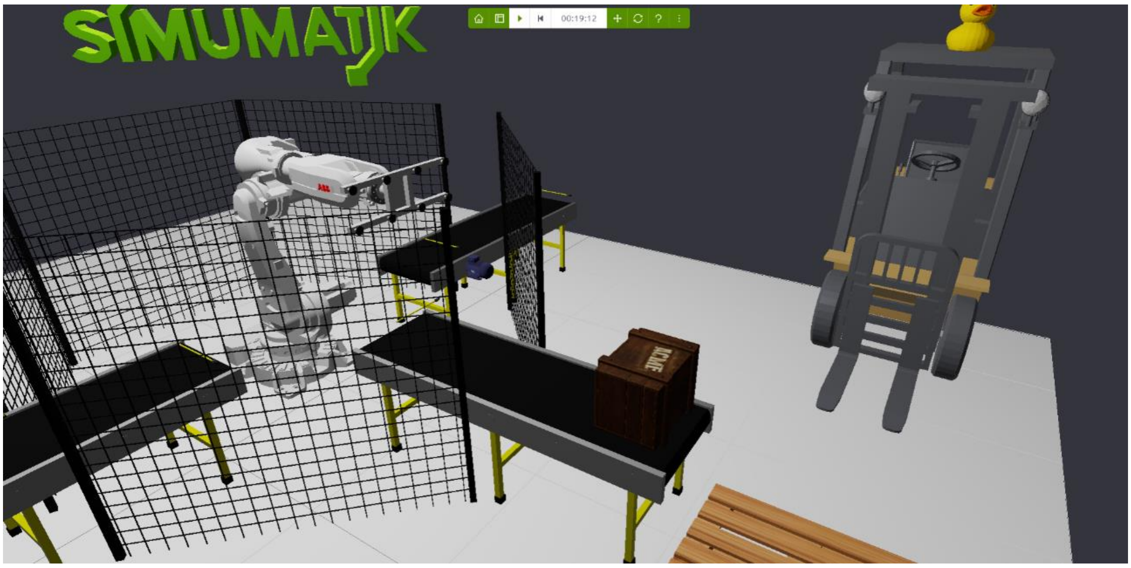


FIGURE 25. QUALITY CONTROL PASSED

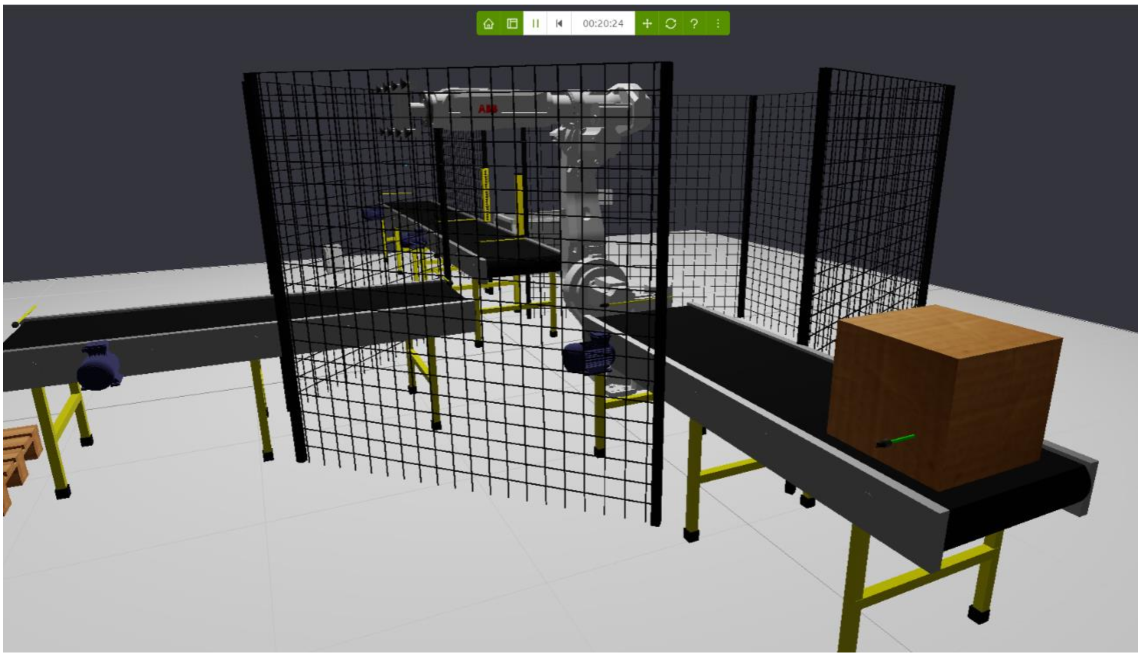


FIGURE 26. QUALITY CONTROL FAILED

7. RESULTS

The results obtained according to the aim and objectives previously stated are the following.

- Define a model: it was planned to achieve the parts of the body as follows (Table 2):

TABLE 2. THE APPROACH OF THE DHM

Part of the body	Approach
Head	The position comes from the headset device.
Neck	Translation on the Z-axis of the device´s coordinate system.
Shoulder	Translation on the X-axis in both senses of the device´s coordinate system.
Wrist	It is planned to be following the movement of the controllers.
Elbow + Forearm + Arm	The elbow should be calculated, and then the cylinders orientated, as forearm and arm.
Hands	The position comes from the controller devices.
Torso	Translation on the Z-axis of the system.
Legs	Translation on the Z-axes of the system.

- Implement the model: The body's parts position and orientation have been implemented using Python 3.7 while receiving the position and orientation of the headset and controller devices. The model has been portrayed in Simumatik using geometrics forms. The neck, shoulders, torso, and legs have been calculated through forward kinematics by measuring a human being and having as base the headset. The forearm and arm have been represented by calculating the elbow, using inverse kinematics and trigonometry, between the shoulder and the controllers.

- Test the model: Firstly, a quantitative comparison between the real world and the virtual world has been done. When creating a working cell in Simumatik, the floor is generated with lines every square meter, therefore, by measuring 1 meter in reality with the devices position using a metric, it has been checked that in the virtual world, it correspond to 1 meter distance. By doing this comparison, the movement of the virtual and real body will be of the same distance. Also, creating several cylinders and rotating them with fixed angles, make possible to test that the rotations work perfectly.

  Now, to check that the results obtained fulfilled the specifications, every part of the body has been checked as shown in Table 3:

TABLE 3. TEST OF THE DHM

Part of the body	Testing	Results
Head	Test that the geometric form created, follows the position of the headset device.	Follows the movement as expected
Neck	Look up and down by moving only the head, checking that the position of the neck remains the same.	The position remains almost the same
Shoulder	Looking at the front, move the controller to the shoulder, and check that it matches with the real position of the shoulder.	The virtual shoulder is in the position of the real one
Wrist	Rotate the real wrist, and check that the virtual wrist is not moving.	The position remains almost the same
Elbow + Forearm + Arm	Test that the movements of the virtual arms, follow as accurately as possible the movement of the real arms, feeling comfortable.	The position of the virtual elbow looks realistic as it is close to the real one
Hands	Test that the geometrics form created, follow the position of the controllers.	Follows the movement as expected
Torso	Move the controller to the waist, and check that it matches close to the real waist.	The torso height fits the one of the user
Legs	While standing, check that the legs are not trespassing the floor	The legs do not pass through the floor and has cohesion with the torso

The functionalities that the body was supposed to achieve in the virtual environment have been checked by creating a demonstrator. Two operators, who can see each other, perform different jobs such as picking up boxes, pressing buttons and activating sensors.

8. DISCUSSION

Throughout the investigation for the literature review, many articles were found relevant for this project. These articles provided insight, clarification, and knowledge for this thesis. Besides, the methodology used to conduct the research uses the information for making advances and backtracking when necessary allowing to perform this project in good terms. Analysing the Strengths, Weaknesses, Opportunities and Threats (SWOT), the reader can see the qualities of this project:

- Strengths: The code is understandable, and it is properly commented. Moreover, it is written in a single python script. Considering the limitations of this project, it has been possible to create a human being from geometric forms such as spheres and cylinders.
- Weaknesses: As the only input information comes from three devices, it was necessary to do some assumptions and eliminate some DoF, therefore some movements are limited and the feeling of lack of fidelity might appear. But this situation can only appear if the user is not holding the controllers properly losing realism regarding ergonomics. Also, there is a small mismatch in this project when changing quadrant while rotating.
- Opportunities: Nowadays, VR is being developed and it is a promising field of research and technology. VC is becoming more important every year and surely VR technologies will be there. This project is just a step into this world constantly increasing and improving.
- Threats: VR is relatively new, this could lead to a lack of information or applications at the moment.

In comparison, there are articles and companies that have work with VR in VC, but none of them provides the code or resources to work with, but mainly information about what has been done, and the advantages and disadvantages of their work. For example, Siemens is a company that provides many very accurate systems of human factors and ergonomics for simulation in VR, but it is needed to purchase a license to use them. Our project provides a simple code and it is open-source, even if it is not as accurate as others projects.

Everything that was planned to be implemented can be described as successful. Currently, the values of the height are fixed for an average human body following ergonomics, thus a person that is not on these average values could feel that the movements are not precise.

The assumptions that have been made on this project are:

- The neck is unable to turn to the sides, it is fixed respect to the shoulders.
- The wrists are fixed in rotation, so it is possible to move the hands, but not to rotate them.
- The torso and the legs are fixed, following the orientation of the head.

The first assumption has been made because there is no way of knowing exactly the position of the shoulders without any extra sensor, however in some studies, it has been possible to estimate the position of the shoulder depending on the position and orientation of the head and the hands, but it would require more calculations, and more assumptions about how would a human behave in a comfortable way, but still, it would not be an exact solution.

The second assumption has been made because trying to estimate the position of the entire arm, without fixing any DoF, would give infinite solutions, therefore other assumptions about the ergonomics of the human arm would have to be done. This project is supposed to be for workers that do not do complex wrist rotation, but only normal jobs moving the arms and body.

The third assumption is due to the lack of information coming from these parts of the body. It is impossible to know the position of the torso and the legs, so these parts have been made mainly for visual aspects.

9. CONCLUSION

The project has been carried out successfully. In the beginning, the first part of the project was a bit complicated because there existed the need to learn about VR and to write code in Python. VR applications are common in video games and for medical purposes, it is now when the interests of this technology aim for VC. The aim and objectives were not clear at the beginning but after building and following the methodology, the ideas became clearer. The development of the project started a bit late because of the lack of the simulation environment, so the aim and objectives had to be changed a bit throughout the development of the project. But once the development started, the project advanced in good terms. The parts of the body were successfully represented one by one while the interaction within the system was being developed and, therefore, implemented. Besides, building the demonstrator was easier than expected due to the experience gained in earlier versions of Simumatik (Simumatik 3D) and courses in the University of Skövde such as robotics.

Simumatik had in mind the implementation of a simple digital human model for their VC platform. The requests of the company were to create arms and the rest of the body which allowed the user to interact with Simumatik´s environment. The input information comes from the head and the hands, therefore, building an entire body had to be done making assumptions and working with restrictions. Without a sensor tracking the elbow, it is almost impossible to know its position and orientation, there exist infinite solutions for this problem. The solution this project offers is to create the arms in Simumatik in a position in which the operator finds comfortable to perform the tasks that could happen in a workspace in a real factory. The resulting digital human model fulfils the needs of the company, so now Simumatik has a new tool for VC.

To sum up, the methodology used for this project allowed to create a structure to follow and gain insight into what it was needed to do and how to do it. Simumatik has the digital human model that wanted for their VC platform, OT in virtual environments is one step further to become reality.

10. FUTURE WORK

Despite the results achieved in this project by fulfilling all the objectives successfully, there is still open the possibility of improvement. Firstly, the arm could count with more DoF and, therefore, calculate the position and orientation of the elbow more accurately with fewer restrictions. Moreover, for the orientation of the cylinders that represent the arms, there could be a more precise way to implement them. When the vector that represents the orientation changes of quadrant, sometimes appear a mismatch of the orientation of the arms. Also, it would be possible to make the lower part of the body able to bend depending on the high of the head, so it would be visually more realistic, even if it is not precise. Finally, introduce a test which asks the user to adopt certain positions for the system to calibrate the length of every part of the body.

11. REFERENCES

Al-Ahmari, A., Abidi, M., Ahmad, A. and Darmoul, S., 2016. Development of a virtual manufacturing assembly simulation system. Advances in Mechanical Engineering, 8(3), p.168781401663982.

Al-Mashhadany, Y., (2011). Virtual Reality for real planer motion of human arm. Conference of Engineering and technology symposiums, Volume 4. April 2011

Dombrowski, U., Stefanak, T. and Perret, J., 2017. Interactive Simulation of Human-robot Collaboration Using a Force Feedback Device. Procedia Manufacturing, 11, pp.124-131.

Dombrowski, U., Stefanak, T. and Reimer, A., 2018. Simulation of human-robot collaboration by means of power and force limiting. Procedia Manufacturing, 17, pp.134-141.

Hoffmann, P., Maksoud, T., Schumann, R. and Premier, G., 2010. Virtual Commissioning Of Manufacturing Systems A Review And New Approaches For Simplification. ECMS 2010 Proceedings edited by A Bargiela S A Ali D Crowley E J H Kerckhoffs.

Mainprice, J. and Berenson, D., 2013. Human-robot collaborative manipulation planning using early prediction of human motion. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems.

Marr, B., 2018. What Is Industry 4.0? Here's A Super Easy Explanation For Anyone. [online] Forbes. Available at: <https://www.forbes.com/sites/bernardmarr/2018/09/02/what-is-industry-4-0-heres-a-super-easy-explanation-for-anyone/#5568a5ea9788> [Accessed 1 April 2020].

Maurya, C.M., Karmakar, S. & Das, A.K. Digital human modelling (DHM) for improving work environment for specially-abled and elderly. SN Appl. Sci. 1, 1326 (2019). https://doi.org/10.1007/s42452-019-1399-y

McLean, S., 2012. Strategies for Gathering Reliable Information. Successful Writing. pp. 558-584.

Oates, B., 2006. Researching Information Systems And Computing. London: SAGE.

Ohio State University, 2018. Choosing & Using Sources: A guide to Academic Research. July 2018, pp. 76-77.

Our Common Future, 1987. Chapter 2: Towards Sustainable Development. Report of the world Commission of Environment and Development: Our Common Future. 20 March 1987.

Parger, M., Mueller, J., Schmalstieg, D. and Steinberger, M., 2018. Human upper-body inverse kinematics for increased embodiment in consumer-grade virtual reality. Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology - VRST '18, pp.1-10.

Reinhart, G. and Wünsch, G., 2007. Economic application of virtual commissioning to mechatronic production systems. Production Engineering, 1(4), pp.371-379.

Rehiara, A., 2011. *Kinematics Of AdeptThree Robot Arm*. INTECH Open Access Publisher.

Rouse, M., 2018. What Is Virtual Commissioning? - Definition From Whatis.Com. [online] SearchERP. Available at: <https://searcherp.techtarget.com/definition/virtual-commissioning> [Accessed 1 April 2020].

Seidel, R. and Chatelier, P., 1997. Virtual Reality, Training's Future?. New York: Plenum Press.

Shiratuddin, M.F., and Zulkifli, A.N. (2001) Virtual reality in manufacturing. In: Management Education for the 21st Century, 12 - 14 September, Ho Chi Minh City, Vietnam

Streefkerk, R., 2019. Qualitative Vs. Quantitative Research | Differences & Methods. [online] Scribbr. Available at: <https://www.scribbr.com/methodology/qualitative-quantitative-research/> [Accessed 20 February 2020].

Tao, W., Leu, M.C., and Lai, Z., 2019. Manufacturing Assembly Simulations in Virtual and Augmented Reality. July 2019.

Tolani, D. and Badler, N., 1996. Real-Time Inverse Kinematics of the Human Arm. Presence: Teleoperators and Virtual Environments, 5(4), pp.393-401.

Tudero, A. and Azkue, J., 2017. 'Emulation of a manufacturing process. Focus on maintenance and operator training'. Bachelor's degree project. University of Skövde, Sweden.

Turnbull, C., 2019. What Is Virtual Commissioning? - Virtual Commissioning. [online] Virtual Commissioning. Available at: <https://virtualcommissioning.com/what-is-virtual-commissioning/> [Accessed 1 April 2020].

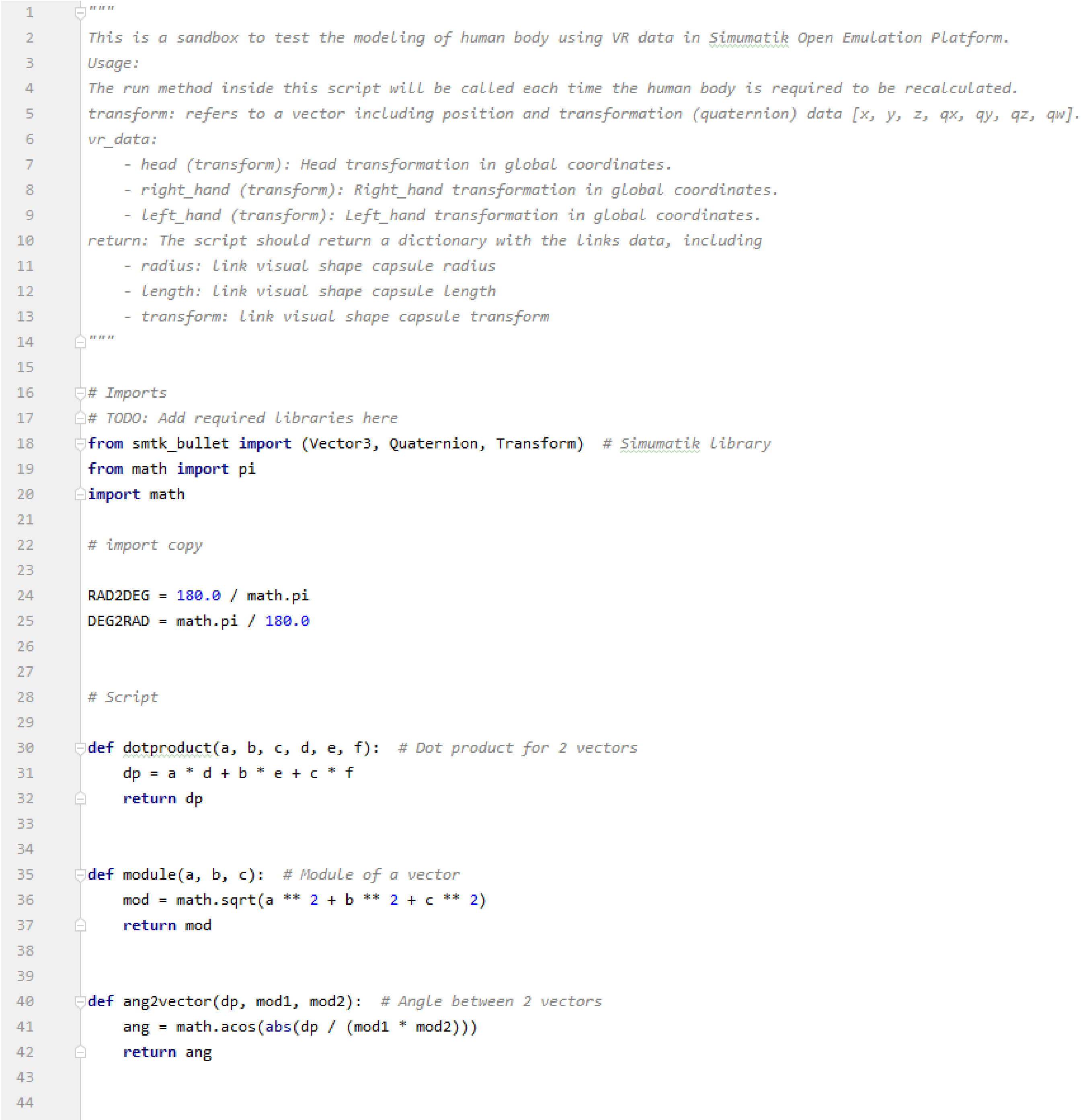
Okapuu-von Veh, A., Marceau, R., Malowany, A., Desbiens, P., Daigle, A., Garant, E., Gauthier, R., Shaikh, A. and Rizzi, J., 1996. Design and operation of a virtual reality operator-training system. IEEE Transactions on Power Systems, 11(3), pp.1585-1591.

Virtual Reality Society: What is Virtual Reality? 2017. [online], available at: https://www.vrs.org.uk/virtual-reality/what-is-virtual-reality.html [Accessed 2 March 2020]

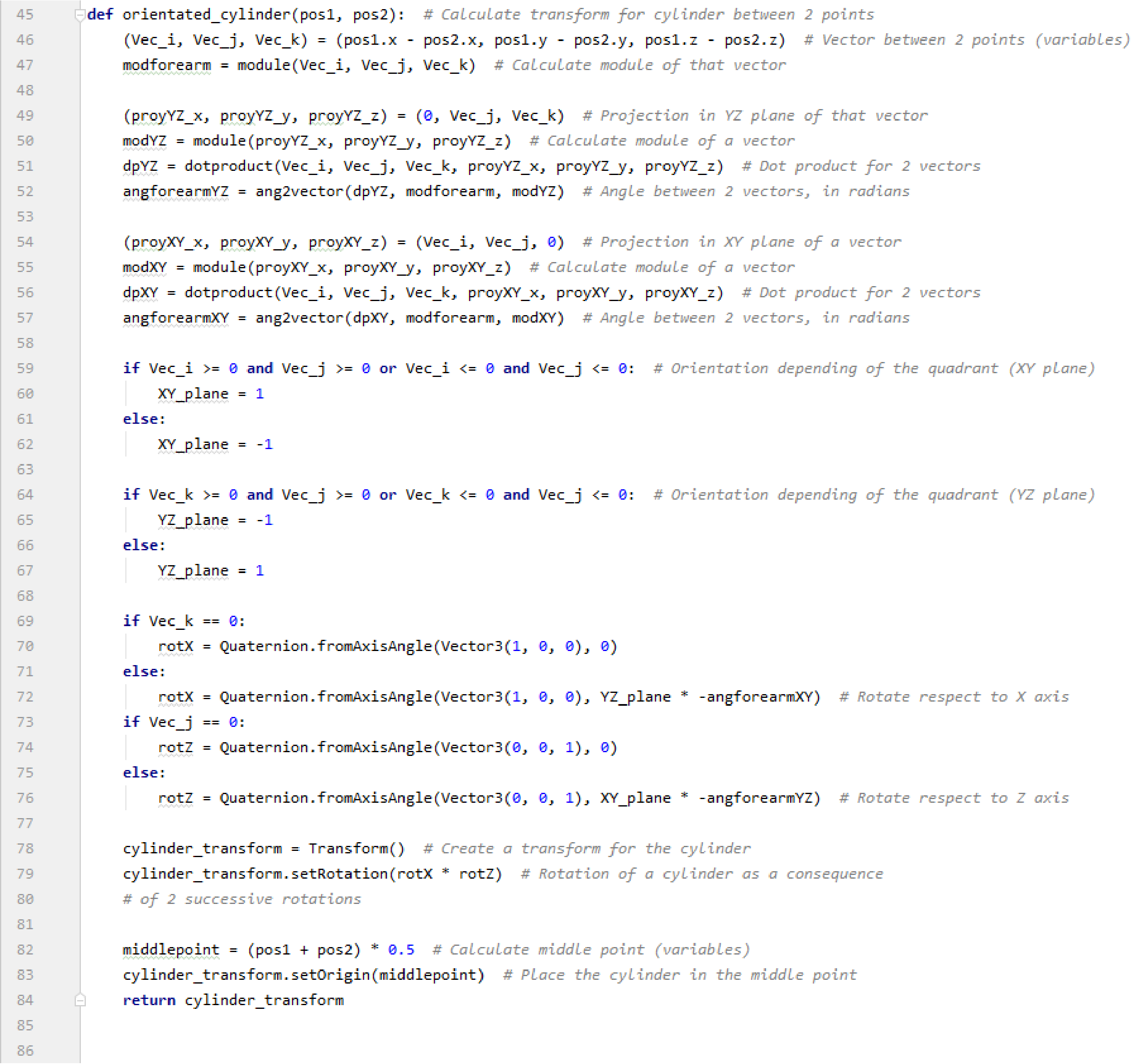
Wangel, J. and Leap, G., 2015. Introduction to Sustainable Development and Design. [online], available at: http://cemusstudent.se/wp-content/uploads/2012/03/2.-JosefinWangel-CEMUS-SDesign.-Sustainable-development-and-design-2015-01-28.pdf [Accessed: 20 February, 2020]

Wilczek, K., 2011. 'Research strategy guide for finding quality, credible sources'. [online], available at: https://journalistsresource.org/tip-sheets/research/research-strategy-guide/ [Accessed: 20 February, 2020]

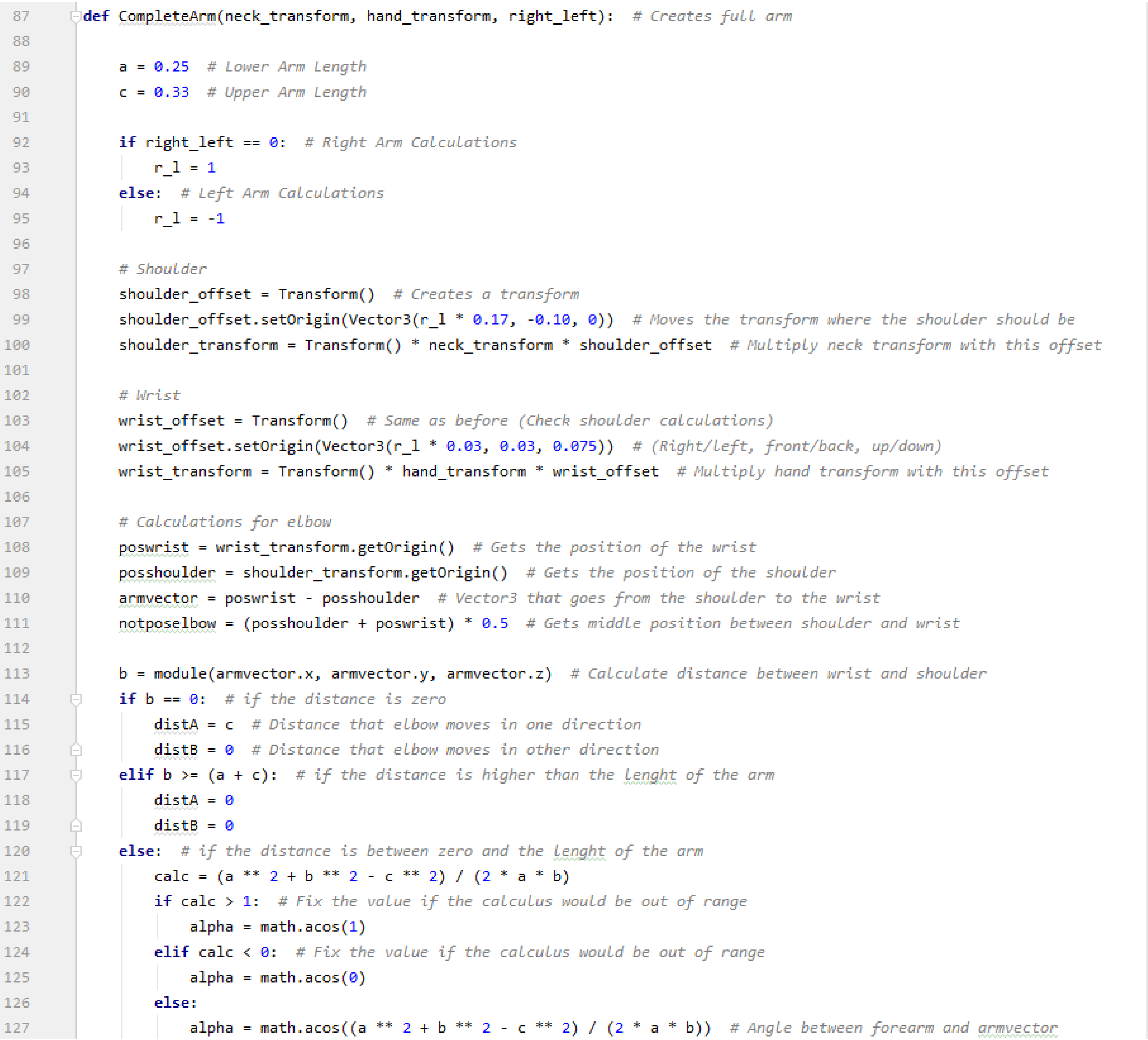
Appendix A: Code



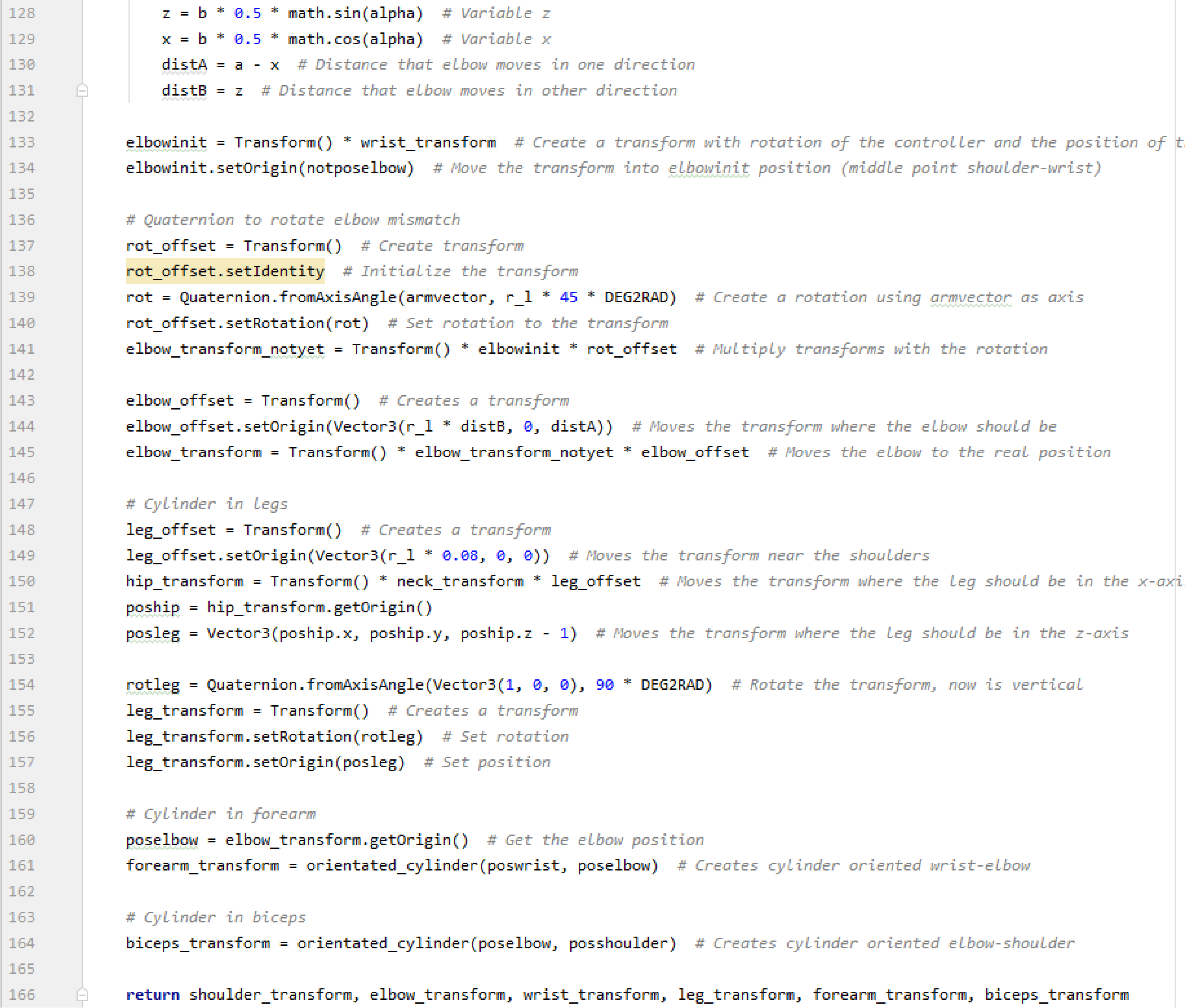
```python
"""
This is a sandbox to test the modeling of human body using VR data in Simumatik Open Emulation Platform.
Usage:
The run method inside this script will be called each time the human body is required to be recalculated.
transform: refers to a vector including position and transformation (quaternion) data [x, y, z, qx, qy, qz, qw].
vr_data:
    - head (transform): Head transformation in global coordinates.
    - right_hand (transform): Right_hand transformation in global coordinates.
    - left_hand (transform): Left_hand transformation in global coordinates.
return: The script should return a dictionary with the links data, including
    - radius: link visual shape capsule radius
    - length: link visual shape capsule length
    - transform: link visual shape capsule transform
"""

# Imports
# TODO: Add required libraries here
from smtk_bullet import (Vector3, Quaternion, Transform)  # Simumatik library
from math import pi
import math

# import copy

RAD2DEG = 180.0 / math.pi
DEG2RAD = math.pi / 180.0


# Script

def dotproduct(a, b, c, d, e, f):  # Dot product for 2 vectors
    dp = a * d + b * e + c * f
    return dp


def module(a, b, c):  # Module of a vector
    mod = math.sqrt(a ** 2 + b ** 2 + c ** 2)
    return mod


def ang2vector(dp, mod1, mod2):  # Angle between 2 vectors
    ang = math.acos(abs(dp / (mod1 * mod2)))
    return ang


```



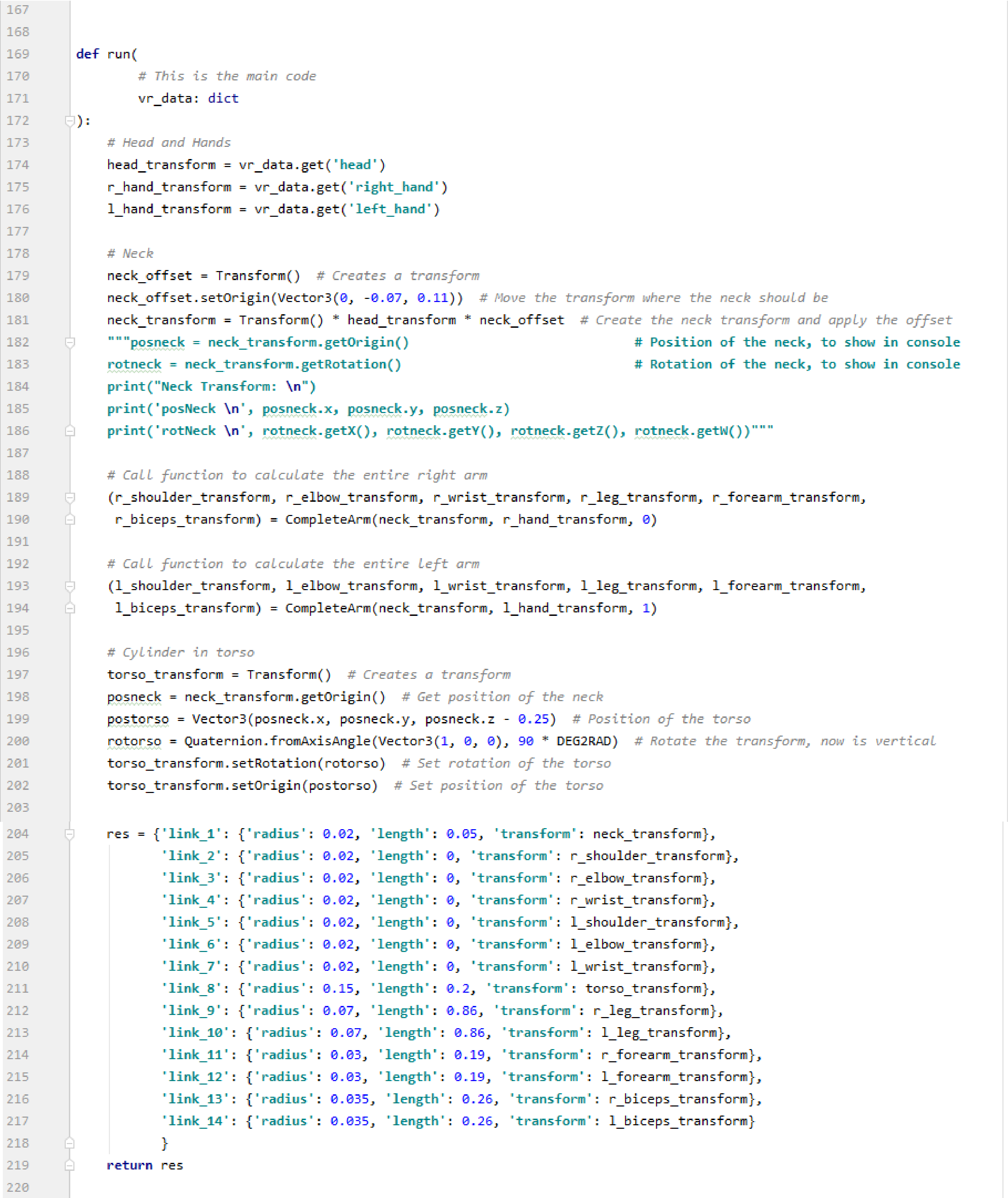
```python
def orientated_cylinder(pos1, pos2):  # Calculate transform for cylinder between 2 points
    (Vec_i, Vec_j, Vec_k) = (pos1.x - pos2.x, pos1.y - pos2.y, pos1.z - pos2.z)  # Vector between 2 points (variables)
    modforearm = module(Vec_i, Vec_j, Vec_k)  # Calculate module of that vector

    (proyYZ_x, proyYZ_y, proyYZ_z) = (0, Vec_j, Vec_k)  # Projection in YZ plane of that vector
    modYZ = module(proyYZ_x, proyYZ_y, proyYZ_z)  # Calculate module of a vector
    dpYZ = dotproduct(Vec_i, Vec_j, Vec_k, proyYZ_x, proyYZ_y, proyYZ_z)  # Dot product for 2 vectors
    angforearmYZ = ang2vector(dpYZ, modforearm, modYZ)  # Angle between 2 vectors, in radians

    (proyXY_x, proyXY_y, proyXY_z) = (Vec_i, Vec_j, 0)  # Projection in XY plane of a vector
    modXY = module(proyXY_x, proyXY_y, proyXY_z)  # Calculate module of a vector
    dpXY = dotproduct(Vec_i, Vec_j, Vec_k, proyXY_x, proyXY_y, proyXY_z)  # Dot product for 2 vectors
    angforearmXY = ang2vector(dpXY, modforearm, modXY)  # Angle between 2 vectors, in radians

    if Vec_i >= 0 and Vec_j >= 0 or Vec_i <= 0 and Vec_j <= 0:  # Orientation depending of the quadrant (XY plane)
        XY_plane = 1
    else:
        XY_plane = -1

    if Vec_k >= 0 and Vec_j >= 0 or Vec_k <= 0 and Vec_j <= 0:  # Orientation depending of the quadrant (YZ plane)
        YZ_plane = -1
    else:
        YZ_plane = 1

    if Vec_k == 0:
        rotX = Quaternion.fromAxisAngle(Vector3(1, 0, 0), 0)
    else:
        rotX = Quaternion.fromAxisAngle(Vector3(1, 0, 0), YZ_plane * -angforearmXY)  # Rotate respect to X axis
    if Vec_j == 0:
        rotZ = Quaternion.fromAxisAngle(Vector3(0, 0, 1), 0)
    else:
        rotZ = Quaternion.fromAxisAngle(Vector3(0, 0, 1), XY_plane * -angforearmYZ)  # Rotate respect to Z axis

    cylinder_transform = Transform()  # Create a transform for the cylinder
    cylinder_transform.setRotation(rotX * rotZ)  # Rotation of a cylinder as a consequence
    # of 2 successive rotations

    middlepoint = (pos1 + pos2) * 0.5  # Calculate middle point (variables)
    cylinder_transform.setOrigin(middlepoint)  # Place the cylinder in the middle point
    return cylinder_transform


```



```python
def CompleteArm(neck_transform, hand_transform, right_left):  # Creates full arm

    a = 0.25  # Lower Arm Length
    c = 0.33  # Upper Arm Length

    if right_left == 0:  # Right Arm Calculations
        r_l = 1
    else:  # Left Arm Calculations
        r_l = -1

    # Shoulder
    shoulder_offset = Transform()  # Creates a transform
    shoulder_offset.setOrigin(Vector3(r_l * 0.17, -0.10, 0))  # Moves the transform where the shoulder should be
    shoulder_transform = Transform() * neck_transform * shoulder_offset  # Multiply neck transform with this offset

    # Wrist
    wrist_offset = Transform()  # Same as before (Check shoulder calculations)
    wrist_offset.setOrigin(Vector3(r_l * 0.03, 0.03, 0.075))  # (Right/left, front/back, up/down)
    wrist_transform = Transform() * hand_transform * wrist_offset  # Multiply hand transform with this offset

    # Calculations for elbow
    poswrist = wrist_transform.getOrigin()  # Gets the position of the wrist
    posshoulder = shoulder_transform.getOrigin()  # Gets the position of the shoulder
    armvector = poswrist - posshoulder  # Vector3 that goes from the shoulder to the wrist
    notposelbow = (posshoulder + poswrist) * 0.5  # Gets middle position between shoulder and wrist

    b = module(armvector.x, armvector.y, armvector.z)  # Calculate distance between wrist and shoulder
    if b == 0:  # if the distance is zero
        distA = c  # Distance that elbow moves in one direction
        distB = 0  # Distance that elbow moves in other direction
    elif b >= (a + c):  # if the distance is higher than the lenght of the arm
        distA = 0
        distB = 0
    else:  # if the distance is between zero and the lenght of the arm
        calc = (a ** 2 + b ** 2 - c ** 2) / (2 * a * b)
        if calc > 1:  # Fix the value if the calculus would be out of range
            alpha = math.acos(1)
        elif calc < 0:  # Fix the value if the calculus would be out of range
            alpha = math.acos(0)
        else:
            alpha = math.acos((a ** 2 + b ** 2 - c ** 2) / (2 * a * b))  # Angle between forearm and armvector
```



```
        z = b * 0.5 * math.sin(alpha)  # Variable z
        x = b * 0.5 * math.cos(alpha)  # Variable x
        distA = a - x  # Distance that elbow moves in one direction
        distB = z  # Distance that elbow moves in other direction

    elbowinit = Transform() * wrist_transform  # Create a transform with rotation of the controller and the position of the wrist
    elbowinit.setOrigin(notposelbow)  # Move the transform into elbowinit position (middle point shoulder-wrist)

    # Quaternion to rotate elbow mismatch
    rot_offset = Transform()  # Create transform
    rot_offset.setIdentity  # Initialize the transform
    rot = Quaternion.fromAxisAngle(armvector, r_l * 45 * DEG2RAD)  # Create a rotation using armvector as axis
    rot_offset.setRotation(rot)  # Set rotation to the transform
    elbow_transform_notyet = Transform() * elbowinit * rot_offset  # Multiply transforms with the rotation

    elbow_offset = Transform()  # Creates a transform
    elbow_offset.setOrigin(Vector3(r_l * distB, 0, distA))  # Moves the transform where the elbow should be
    elbow_transform = Transform() * elbow_transform_notyet * elbow_offset  # Moves the elbow to the real position

    # Cylinder in legs
    leg_offset = Transform()  # Creates a transform
    leg_offset.setOrigin(Vector3(r_l * 0.08, 0, 0))  # Moves the transform near the shoulders
    hip_transform = Transform() * neck_transform * leg_offset  # Moves the transform where the leg should be in the x-axis
    poship = hip_transform.getOrigin()
    posleg = Vector3(poship.x, poship.y, poship.z - 1)  # Moves the transform where the leg should be in the z-axis

    rotleg = Quaternion.fromAxisAngle(Vector3(1, 0, 0), 90 * DEG2RAD)  # Rotate the transform, now is vertical
    leg_transform = Transform()  # Creates a transform
    leg_transform.setRotation(rotleg)  # Set rotation
    leg_transform.setOrigin(posleg)  # Set position

    # Cylinder in forearm
    poselbow = elbow_transform.getOrigin()  # Get the elbow position
    forearm_transform = orientated_cylinder(poswrist, poselbow)  # Creates cylinder oriented wrist-elbow

    # Cylinder in biceps
    biceps_transform = orientated_cylinder(poselbow, posshoulder)  # Creates cylinder oriented elbow-shoulder

    return shoulder_transform, elbow_transform, wrist_transform, leg_transform, forearm_transform, biceps_transform
```



```python


def run(
        # This is the main code
        vr_data: dict
):
    # Head and Hands
    head_transform = vr_data.get('head')
    r_hand_transform = vr_data.get('right_hand')
    l_hand_transform = vr_data.get('left_hand')

    # Neck
    neck_offset = Transform()  # Creates a transform
    neck_offset.setOrigin(Vector3(0, -0.07, 0.11))  # Move the transform where the neck should be
    neck_transform = Transform() * head_transform * neck_offset  # Create the neck transform and apply the offset
    """posneck = neck_transform.getOrigin()                                # Position of the neck, to show in console
    rotneck = neck_transform.getRotation()                                 # Rotation of the neck, to show in console
    print("Neck Transform: \n")
    print('posNeck \n', posneck.x, posneck.y, posneck.z)
    print('rotNeck \n', rotneck.getX(), rotneck.getY(), rotneck.getZ(), rotneck.getW())"""

    # Call function to calculate the entire right arm
    (r_shoulder_transform, r_elbow_transform, r_wrist_transform, r_leg_transform, r_forearm_transform,
     r_biceps_transform) = CompleteArm(neck_transform, r_hand_transform, 0)

    # Call function to calculate the entire left arm
    (l_shoulder_transform, l_elbow_transform, l_wrist_transform, l_leg_transform, l_forearm_transform,
     l_biceps_transform) = CompleteArm(neck_transform, l_hand_transform, 1)

    # Cylinder in torso
    torso_transform = Transform()  # Creates a transform
    posneck = neck_transform.getOrigin()  # Get position of the neck
    postorso = Vector3(posneck.x, posneck.y, posneck.z - 0.25)  # Position of the torso
    rotorso = Quaternion.fromAxisAngle(Vector3(1, 0, 0), 90 * DEG2RAD)  # Rotate the transform, now is vertical
    torso_transform.setRotation(rotorso)  # Set rotation of the torso
    torso_transform.setOrigin(postorso)  # Set position of the torso

    res = {'link_1': {'radius': 0.02, 'length': 0.05, 'transform': neck_transform},
           'link_2': {'radius': 0.02, 'length': 0, 'transform': r_shoulder_transform},
           'link_3': {'radius': 0.02, 'length': 0, 'transform': r_elbow_transform},
           'link_4': {'radius': 0.02, 'length': 0, 'transform': r_wrist_transform},
           'link_5': {'radius': 0.02, 'length': 0, 'transform': l_shoulder_transform},
           'link_6': {'radius': 0.02, 'length': 0, 'transform': l_elbow_transform},
           'link_7': {'radius': 0.02, 'length': 0, 'transform': l_wrist_transform},
           'link_8': {'radius': 0.15, 'length': 0.2, 'transform': torso_transform},
           'link_9': {'radius': 0.07, 'length': 0.86, 'transform': r_leg_transform},
           'link_10': {'radius': 0.07, 'length': 0.86, 'transform': l_leg_transform},
           'link_11': {'radius': 0.03, 'length': 0.19, 'transform': r_forearm_transform},
           'link_12': {'radius': 0.03, 'length': 0.19, 'transform': l_forearm_transform},
           'link_13': {'radius': 0.035, 'length': 0.26, 'transform': r_biceps_transform},
           'link_14': {'radius': 0.035, 'length': 0.26, 'transform': l_biceps_transform}
           }
    return res

```