RESEARCH PROPOSAL BACHELOR ASSIGNMENT

K.E. Jaramillo Cordova, J. Reuling

March 2021

Contents

1 Introduction

In light of Industry 4.0 the Internet of Things (IoT) creates many new opportunities for production environments and for improving their efficiency and effectiveness. To explore and maximize the effect of the potential of Industry 4.0, an interactive prototyping environment is anticipated. This environment supports the different perspectives of stakeholders and allows the integration of simulation and what-if analysis. These testbeds introduce a platform that is meant to be a flexible, user-friendly, intuitive tool that can help any company in collecting real-time data from all relevant machines and systems. Such a testbed could be essential for companies to predict and test certain changes for their production environments. In this testbed an environment is simulated and data, such as flows between assets, cost, heat output etcetera, are visualized in real time on this testbed. This is done using a Synthetic Prototyping Environment (SPE).(1)

A proof of concept of such an SPE has been realised in the Virtual Reality Lab on the University of Twente. However, this is only version 0.1. Therefore research is necessary to realise a new version of the existing setup. Also a new version will have to be build to realise an even better proof of concept. This proposal will set context for this project.

2 Problem statement

2.1 Relevance

As the vision of the University of Twente implies, it strives on being up-to-date with innovation and development. The university constantly anticipates future developments to be able to respond to it with ground-breaking research and world-class innovations (2). With Industry 4.0 (I4.0) being one of those developments that the rapidly changing world of today offers, factories strive for existing of manufacturing resources like production machinery and other related systems that are "autonomous, capable of controlling themselves in response to different situations, self-configuring, knowledge-based, sensor-equipped and spatially dispersed and that also incorporate the relevant planning and management systems" (3).

As attractive as the idea of I4.0 sounds, the implementation of a digitalized and connected production environment seems to be a challenge that brings a lot of unpredictability with it. This makes moving towards a different production environment difficult and risky. If it's possible to get rid of this uncertainty by being able to know the consequences of change, the decision-making process towards a digitalized and connected production environment is supported. That is where an SPE comes into play. In section 3.1 a literature study has been done on SPE's.

2.2 Problem

In figure 1 the setup as it is anticipated by Damgrave and Lutters in their research is visualized. In this case the final stage of the SPE should consist of a table with physical, possibly 3D printed, models of machines on it. Stakeholders should be able to rearrange and add the models and data should be visualized on the test environment for direct feedback. This visualization could be done by AR, VR and overlay projection or a combination of these. Also, certain setups should be saved during the testing process and these setups should be easily called back. For this system to work a digital twin is run alongside the physical testing environment(1)

This calling back of setups means the table should have a system which automatically rearranges the physical objects to their recalled places. As if it were magic the objects move over the table to their correct places. A small proof of this concept has been made by Daan van Meurs. This setup uses a 3D carving machine called the X-Carve with a togglable magnet, which moves under a glass table. The setup uses Unity3D to build the digital twin and to control the X-Carve. The operation of the current setup in short: Unity3D shows the user how objects are arranged, which matches the physical table. When using Unity3D to move an object, the X-Carve moves to the object, picks up the object by turning on the magnet, moves to the place the user pointed to and lets go of the physical object by turning off the magnet (4).

A proof of concept for image recognition to locate the objects on the table has been made by Timber Halbesma, however due to lockdown circumstances this has not been realized. Image recognition could be used to improve accuracy of the current setup and for the software to calculate and visualize data (5). This project will expand on the current setup made by Daan van Meurs. The current setup certainly is a proof of concept, however a new version has to be realised, because the movement of the current setup is slow and not very accurate.

2.3 Goal

The goal of this project is to improve the movement of the objects on top of the table, with a focus on the movement speed, accuracy and extra options to orient objects on top, such as turning the objects. We aim to do this by improving the software and hardware. Unity3D can be used more efficiently and effectively by improving the interface between the physical and digital environments. The use of X-Carve for the movement is not ideal and could be replaced in the new setup. Furthermore, image recognition to locate models on the table can be worked on when we have sufficient time. In theory it should work already, however it has yet to be realized.

The project will not focus on visualizing data using AR, VR or projection, nor calculating this data. Also the user friendliness of the digital version will not be improved. No new features will be added to the current setup, only improved.

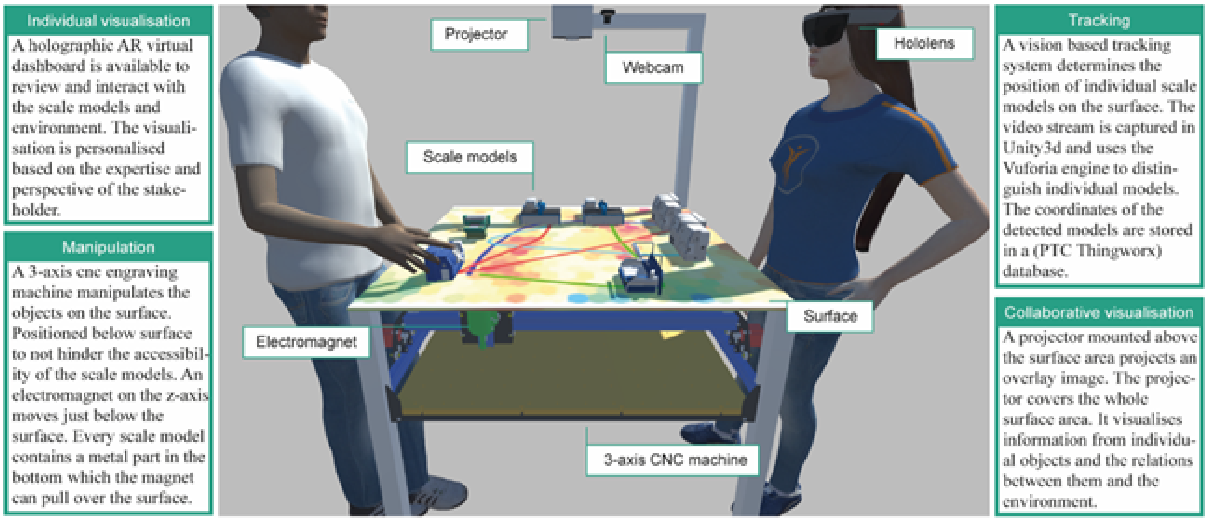

Figure 1: A representation of the SPE setup with the currently used hardware (1)

3 Literature research

To get a grip on what is the state of the art for different aspects of the project, a literature research has been done. First it is important to understand the SPE and it's relevance. The project focuses on the movement, so research has been done to 2 Degrees Of Freedom (DOF) robotics and the movement of objects over a planar plate. At last the software side of the SPE has been researched.

3.1 Synthetic Prototype Environment

A short introduction of the SPE has been provided in section 2. A prototype environment is used to facilitate the decision-making process towards a realisation. In this case this realisation is a production process. These environments combine real and simulated information, and therefore allow for utilising multiple timescales, stakeholders, elements and locations to understand the impact of future manufacturing environments. This instrumentalizes an environments' unpredictability, uncertainty and risks of change. A *testbed* is such an environment. This testbed is used to test new layouts, assets etc. on, and see what these changes do to certain aspects in real time (6).

To bridge the gap between a complete virtual environment and a real environment, a *synthetic* prototype environment is introduced. The environment consists of a scale model of the current or potential environment. The scaled environment is of such a size it can be easily supervised by multiple people at the same time, and all models are within reach, as can be seen in Figure 1. Since there is a shared visible environment, each participating stakeholder knows what other stakeholders can see. There is no need to visualize, for example, a single virtual system, nor to communicate and ask for confirmation of this (eg "you can see...")

Every change made to the environment is represented in both the physical and the virtual environment; this also allows the solution to switch between being a representation of the digital twin and the digital prototype; the digital twin is the digital copy of the production environment and the digital prototype is the digital copy of the try-out of the production environment.

The proposed physical scale model environment should be seen as a tool for interacting with a testbed; The testbed can represent an alternative to an existing environment, or an instance of an existing environment, which is filled with different data to view the capabilities of the current configuration. Trying out new configurations and the feedback it provides can be directly used as feedback and input to change the real environment. In this way, an environment can be achieved in which stakeholders with different expertises can review the consequences at the same time (1).

3.2 Boundaries & requirements

At a certain moment some boundaries and requirements (e.g. number of objects, minimum size, resolution/accuracy of positioning, etc.) need to be set. Relevant research has been conducted in order to set these boundaries and requirements.

Number of objects Damgrave and Lutters state in (1): "To provide a complete and coherent overview of a testbed the scaled models of all assets can be made physical." Therefore, multiple objects need to be moved. The amount of objects on the table differs from company to company.

Size This scaled environment is of such size that it can be easily overseen by multiple people at the same time, and that all the models are within an arm-reach distance (1).

Accuracy The accuracy and quality of individual parts can be changed throughout the life-cycle of the solution. To quantify accuracy a grid size could be determined.

Speed Stakeholders will try out change or explore variants of their company. Moreover, every exploration of alternatives comes with a roll-back or undo function over a history path of changes(1). This means the movement should be sufficiently fast to keep up with the pace of the stakeholders and not slow down their iterative process. Slow movement could also distract stakeholders and disturb the state of flow.

3.3 Movement

The first version of the testbed by Daan van Meurs this project focusses on moves very slow, therefore the current setup should be redesigned. Different ways of moving objects over a plane have to be researched. This section has been divided into 2 subsections: Physical actuator and an embedded actuator in objects. The physical actuator takes into account robots under the table and without physical touch, but also robots with physical touch above the table. The emedded actuator consists of movement actuated by the objects itself.

3.3.1 Physical actuator

A physical actuator could exist of a design of a robot which moves in 2 DOF. The actuator could be placed under the table to move objects without physical touch, or it could act as a pick and place manipulator to move objects with physical touch. Many different performance criteria have to be taken into account, such as accuracy, stiffness, speed, etc. When the robot has to perform a task requiring a very small position error, it is necessary to use external sensors (vision sensors for instance) to control the robot. The first pair displayed in Fig. 2 consists of a parallel robot with four links of lengths l_1, l_2, l_3, l_4 and a serial planar robot with two links of lengths l_1, l_2. In (7), Briot and Bonev ask the question: "Are parallel robots more accurate than serial robots?" In their introduction, they quote numerous papers or books asserting that parallel robots have better accuracy. This common opinion is based on the fact that the joint errors are averaged for parallel mechanisms, while they are accumulated for serial robots. Our design will be taking this research as a starting point.

The current setup with the X-Carve should be tested as well before we decide on a new design. User "ChrisTait" says on discuss.inventables.com the X-Carve can be sped up to 12000mm/min.[1]. However, testing it ourselves should be step one. In this case 12000mm/min still would not be fast enough, but if the objects on top can not follow this speed, a redesign is unnecessary.

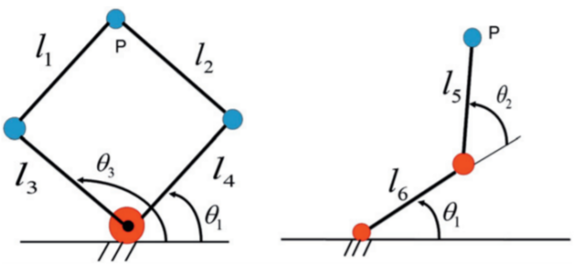

Figure 2: Parallel robot (left), serial robot (right) (8)

3.3.2 Movement of objects without physical touch

Before we design a new robot which is faster than the X-Carve, we have to research if the movement over a planar plate (the table) of the objects can be faster. In the current setup,

[1]Retrieved from https://discuss.inventables.com/t/feeds-pushed-to-the-limit-how-fast-can-the-x-carve-move/39736 on 11-3-2021

when the X-Carve goes any faster it loses the object on the table, because of the friction.

Magnet In the current setup a magnet is used to move the objects on the table as if it were magic. To bridge the gap through the plate on the table a long magnetic field is required. Electromagnets, which can be turned on and off, generally have a different magnetic field than permanent "on" magnets. The electromagnet has a relatively 'flat' magnetic field compared to a permanent magnet, which naturally has a 'longer' magnetic field (4).

That is why in the current setup *magnetic* electromagnets are used. In this kind of electromagnets the power supply on the coil around the permanent magnets cancel out the magnetic field. This allows to loose the load. This means that the electromagnet is magnetic when it is disconnected from power, and it's not magnetic when it is connected to power. These magnets have a longer magnetic field and therefore bridge the gap of the table more easily (4).

The problem with using a magnet below the testbed is the friction from the objects onto the table. This limits the speed the magnet can move in order to carry the object. Therefore literature research about different ways to decrease friction is done in the following paragraphs.

Air bearing Different researches show potential for a *planar air-bearing.* This is a way to produce an air cushion below the objects. One creates airflow through the grid of holes in the flat table surface similar to an air-hockey table. In this case there must be an air-supply system connected to the reservoir under the table surface. This kind of facility is presented in Technion University distributed space system laboratory (9) and in the Keldysh Institute of Applied Mathematics RAS (10). However, no literature is found about moving objects on top using a magnet of some sort with this planar air-bearing.

Ball bearings Also a bearing could be added to the objects on top. This bearing could consist of omnidirectional wheels on a carrier, which carries the scaled machines. This may reduce friction with the table contact patch.

3.3.3 Embedded actuator in objects

Movement of the objects could also come from the objects itself. In (11) A. Özgür, W. Johal and P. Dillenbourg present an omnidirectional ball wheel drive design that utilizes a permanent magnet as the drive roller to generate the contact force. They claim it is particularly interesting for novel human-mobile robot interaction scenarios where the users are expected to physically interact with many palm-sized robots, and the design combines simplicity, low cost and compactness. Their design uses 3 ball wheels, on which a motor acts. The design can be seen in Figure 3.

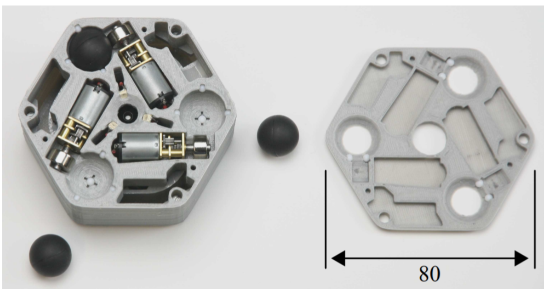

Figure 3: Ball drive implementation, size given in mm. Main body (on the left) rests bottom-side-up. Bottom "lid" opened (on the right) and two ball wheels removed from enclosures for better visibility of internals. In the center, camera lens aperture and 3 exposure LEDs used for localization are seen. (11)

The design is almost completely made with low cost off-the-shelf components. It has naturally compact geometry that enables it to fit inside a palm-sized robot with such components. The design could be used for our case as a carrier for the scaled machines. Aside from this it is not suitable for high-precision applications. This can be overcome by using object tracking using a camera on top (Section 3.4).

3.4 Object tracking

Prior in the SPE project(1) Timber Halbesma focused his final assignment (5) in studying different method to track objects, in his thesis it is discussed extensible two categories for this matter: the use of camera and the use of sensors(without camera). For tracking without camera, it is considered: Radio frequency Identification and Near field communication. Then, for tracking with camera, it is considered: Qr codes fiducial markers and Image recognition part of computer vision. Furthermore, after analysis it is concluded that the most appropriate method for the SPE is the use of camera and fiducial markers. In this papers the use of camera for tracking objects is elaborated further, specially the method for image recognition and color classification to detect objects. The first method requires to create a model as starting point, then train data for a specific application using videos and images, finally the model can be used to recognize objects with camera feed. There are multiple algorithms that can be used for these matters, for example Image AI and You Only Look Once v3 (YOLO v3) algorithms. In (12) H.S. Tarimo W and Sabra M present a powerful Deep Learning Neural Network technique that combines ImageAI deep learning libraries and the YOLO v3 method. This technique concludes with a 65% of accuracy, which is claimed to be more compared to YOLO v3 or ImageAI alone. The second method "color classification" case does not require any predefined model, it consists in filtering colors from camera feed, in this way it is possible to make the object to be tracked of a distinguishable color to make filtering more efficient, a similar approach Tim Halbesma uses in his thesis (5) where the fiducial markers are recognized with color classification, however color classification can be used by itself without the use fiducial markers that add complexity to the system. Furthermore, both cases make use of the *OpenCV* library, which is a library of programming functions mainly aimed at real-time computer vision.

3.4.1 Camera coordinates to manipulator coordinates

The final objective to track objective using camera feed is to be able to manipulate the objects being tracked, it means the manipulator must be able to understand where in space a specific object is located. In video series called "Robotics 1"(13) , Angela Sodeman (assistant Professor at Arizona State University) explains the use matrix transformations to transform the reading from camera feed in pixel to manipulator coordinates in meters. The transformation is show in figure 4.

$$\underbrace{\begin{pmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{pmatrix}}_{\text{Manipulator Cordinates}} = H_c^o \underbrace{\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix}}_{\text{Camera coordinates}} ; H_c^o = \underbrace{\begin{pmatrix} r_{11} & r_{12} & r_{13} & dx \\ r_{21} & r_{22} & r_{23} & dy \\ r_{31} & r_{32} & r_{33} & dz \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{Transformation Matrix from camera to manipulator cordinates}}$$

Figure 4: Camera coordinates to manipulator coordinates

3.5 Trajectory generation for improved motion of manipulator

In order to have a smooth movement of a manipulator/robotic arm and move objects in an efficient way, it is necessary to use path planning and thus use *trajectory generation*. A manipulator without trajectory generation has undesired acceleration and speeds, this can have negative effects on the structure of the robot or the objects being manipulated. Furthermore to generate trajectories complex algorithms can be written such as the ones presented in (14). On the other hand there are open source libraries such as: *PythonRobotics* (15) and *Robot Tool Box for Python* (16), these libraries offer a variety of algorithms that can output reliable trajectories for a robotic model and are relatively easier to implement.

3.6 Connecting Unity to real world movement

Unity is a game engine in which 2D or 3D virtual reality applications can be developed. For the SPE Unity is the tool chosen to develop the virtual environment. Unity uses the programming language `C#`. Python on the other hand has a variety of tools and libraries that can be easily integrated to perform for example object recognition, trajectory planning for robots, manipulation of articulated robots, etc. That is why it is interesting to use python as an interface between the real world movement and the virtual environment. Consequently, it is important to find a way to connect Unity (`C#`) with python, for this purpose two different methods are presented: *Python for Unity*(17) which is an extension in the Unity environment and *Socket Communication*(18)(19) which allows byte array exchange between platforms. Python for Unity requires a longer process to set up, but also provides better functionality and communication between platforms. Socket Communication is limited to send and receive bytes, which can be enough for some applications.

4 Research questions

4.1 Research question

- To what extent is it possible to improve the movement of the real world representation of a dynamic virtual world?

4.2 Sub-questions

- How can tangible objects be moved over a table, with sufficient speed, accuracy and user-friendliness?
  - How can objects be moved by an embedded actuator?
  - How can a robot arm be designed to improve the speed with respect to the X-Carve?
  - How can the friction of objects on the table be decreased to improve the speed of the objects on the table?
  - What magnets can be used, to move objects without physical touch?
  - What is an effective way of connecting the movement to Unity3d, to improve accuracy?
- How can object tracking with camera recognition be used for the SPE?
  - To what extend is it possible to provide Unity3d with feedback on the position of a physical object/different physical objects?

5 Approach

To gain answers to the research questions, as mentioned in section 4, an aproach has been anticipated.

Research variations A decision was made to work in a team of 2 to work on one specific aspect of the SPE project with a different focus. The focus of Kelvin Jaramillo Cordova will be more on the software side; object tracking, Unity and Python integration. While the focus of Jacco Reuling will be more on the design side; robot design and movement. However, it is necessary to work together very closely, because the two variations are very intertwined.

Test current setup and find bottlenecks First, we need to know the bottlenecks of the current setup. This can be done by measuring the speed, accuracy, communication between systems, user interface,etc.Also some usability testing has to be done, it is important to find what parts of the current software can be used or edited, this to avoid wasting time in building something that has already been done. Furthermore, after the current set up has been tested thoroughly, the bottlenecks should be made clear. The X-Carve if possible will be tested to see if the new set up will indeed have better performance, also the maximum speed of the objects on top will be tested with different bearings described in section 3.3.2.

Design and build an improved setup After the testing phase, a new design will have to be realized with certain requirements kept in mind. In this ideation phase new ideas will be developed and generated. First the design should have less friction, if needed a faster robot, and it should include trajectory generation for smooth movements. For this a morphological chart can be used. All relevant aspects of the design should be calculated. The design can be made in SolidWorks. Next, the setup will be build as a proof of concept.

Test the improved version Lastly the new setup should be tested to improve its design ans well as the software, ideally the new set up must be build in such a way that error that can be made are as minimum as possible.

6 Conditions

A number of conditions should be taken into account before and during the research.

- The research has to be executed in a time frame of 11 weeks.
- A decision was made to work in a team of 2 to work on a specific aspect of the SPE project. However, 2 Different research papers have to be delivered.
- To build the current setup a budget will be made available, the amount of this budget has not been decided yet.
- A close relation will be held with the Virtual Reality and Smart Industry Lab. The setup will be build in the Virtual Reality Lab.
- The project will be supervised by Roy Damgrave.

7 Planning

A global schedule for the assignment is shown in Figure 5. On the left side of the table the task that have to performed are shown, the cells highlighted in yellow means that the task has be done with the two members of the team. During the "design" phase it is important that the team keeps close contact to ensure the hardware and software perform as expected, some decisions have to be taken as a group, for example the actuators to be used, the dimensions of the manipulator links, etc. Furthermore, the "building" phase is done by both members since each of hem knows how their part is supposed to be assembled. After the setup is build and running, each member has to consider in detail the "testing procedures" to be done in each of their parts. Finally, the testing phase is to see what is not behaving as expected or can be improved so that it can be further developed in the further "improvement" phase.

Week #	1	2	3	4	5	6	7	8	9	10	11
setup and document results											
Design of new setup: Software & Hardware											
Building new setup											
Write testing procedure											
Testing new setup											
Further improve of set up based on testing											
Write thesis											
& results											
Prepare presentation											

Figure 5: Global planning

References

[1] R. G. Damgrave and E. Lutters, “Synthetic prototype environment for industry 4.0 testbeds,” *Procedia CIRP*, vol. 91, pp. 516–521, 2020.

[2] “Our mission and vision.” https://www.utwente.nl/en/organisation/about/shaping2030/mission/. Accessed: 10-3-2021.

[3] “Industrie 4.0.” https://www.iese.fraunhofer.de/en/innovation_trends/industrie40.html. Accessed: 10-3-2021.

[4] D. V. Meurs and I. Design, “Synthetic prototype environment,” 2020.

[5] T. Halbesma, “Connecting tangible and virtual objects in a Synthetic Prototyping Environment,” vol. 1, no. 1, pp. 1–8, 2020.

[6] E. Lutters and R. G. Damgrave, “Smart Industry Testbed,” *Procedia CIRP*, vol. 84, pp. 387–392, 2019.

[7] S. Briot, I. Bonev, S. Briot, I. Bonev, A. Parallel, R. More, and S. Robots, “Are Parallel Robots More Accurate than Serial Robots?,” vol. 31, no. 4, pp. 445–456, 2009.

[8] R.-m. A. Nzue, J.-f. Brethé, E. Vasselin, and D. Lefebvre, “Comparison of serial and parallel robot repeatability based on different performance criteria,” *MAMT*, vol. 61, pp. 136–155, 2013.

[9] S. Segal, A. Carmi, and P. Gurfil, “Stereovision-based estimation of relative dynamics between noncooperative satellites: Theory and experiments,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 568–584, 2014.

[10] D. S. Ivanov, M. D. Koptev, Y. V. Mashtakov, M. Y. Ovchinnikov, N. N. Proshunin, S. S. Tkachev, A. I. Fedoseev, and M. O. Shachkov, “Laboratory Facility for Microsatellite Mock-up Motion Simulation,” vol. 57, no. 1, pp. 117–132, 2018.

[11] A. Özgür, W. Johal, and P. Dillenbourg, “Permanent magnet-assisted omnidirectional ball drive,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1061–1066, 2016.

[12] H. S. Tarimo W, Sabra M, “Real-time deep learning object detection framework,” pp. 1829–1836, 2020.

[13] “Robotics 1.” https://www.youtube.com/user/asodemann3. Accessed: 11-03-2021.

[14] M. Švejda; Tomáš Čechura, “Interpolation method for robot trajectory planning,” 2015.

[15] A. Sakai, D. Ingram, J. Dinius, K. Chawla, A. Raffin, and A. Paques, “Pythonrobotics a python code collection of robotics algorithms,” 2018.

[16] J. Haviland and P. Corke, *Robotics Toolbox for Python*, 2021.

[17] “Python for unity.” https://docs.unity3d.com. Accessed: 11-03-2021.

[18] “Socket.cs.” https://referencesource.microsoft.com/. Accessed: 11-03-2021.

[19] “Socket.py.” https://github.com/python/cpython/blob/master/Lib/socket.py. Accessed: 11-03-2021.