2 DOF robotic manipulator with electromagnet end effector for the SPE testbed

K.E. Jaramillo Córdova

University of Twente, Faculty of Engineering Technology, Drienerlolaan 5, 7522 NB, Enschede, The Netherlands
k.e.jaramillocordova@student.utwente.nl

ABSTRACT: The Synthetic Prototype Environment testbed is a project under development that can be used as tool to modify digital environments with tangible objects. The testbed requires to give to the objects over the table the ability to move around without human interaction. This with the goal of giving to the testbed positional data of equipment for a production environment, and relocate the objects over the table relative to the given data automatically. In this paper a 2 DOF robotic manipulator is developed with the main goal of increasing the speed to move the objects over other solutions. The maximum measured speed of the developed 2DOF is around 2.1 m/s which is over 600% increase from previous solutions. Furthermore, the objects can be moved more precisely the further the trajectory followed by the end effector is from the center of the robot. When the trajectories are close to the center the trajectories require greater jumps in angles of the joints which subsequently require more torque from the motors than initially expected, this makes the end effector not to follow the trajectory satisfactorily. Finally, to cope with this unexpected behaviour some recommendation are given for future versions of robotic manipulators and other types of robotic solutions to move the objects over the table.

Key words: SPE project, controllers, 2 DOF manipulator, magnetic end effector

1 INTRODUCTION

The design process of a production environment has a major impact in the subsequent material flow, product flexibility, product quantity, etc. This is why it is important to spot possible bottlenecks and design weaknesses prior to the construction phase which at the end helps to save time and resources. To help with the design process, the Synthetic Prototype Environment (SPE) is being developed at the university of Twente. The project consist of using Virtual and Augmented reality technology to make a digital copy of an existing or yet to be build production environment, a copy in which tests can be executed to study what are the best arrangements to then be validated and constructed in the real environment. Part of the SPE consist of making editing of 3D environments easy with tangible objects. The SPE test-bed is a table with objects, which are correlated to the VR environments objects, this objects can be moved by hand and by doing so, the virtual environment also changes. The goal of the SPE test-bed is to be used as a tool for team cooperation. A multidisciplinary team could gather around the table, interact with the table and redesign a certain production environment according to everyone's perspective and with help of visualized data projected over the table. Another feature of the SPE test-bed is the ability to make the object move without human interaction. This is desired if auto-optimization features are included, positional data of the equipment can be loaded on the test-bed and the objects will be moved to the relative position automatically. Furthermore, this thesis focus in developing a 2DOF robotic manipulator with an electromagnet end effector. The objects over the table have a magnet at the lower surface, the robotic manipulator then can pick up the object by turning on the electromagnet. Initially the SPE testbed consisted of a XCarve CNC machine with the same electromagnet. However, the XCarve is not designed for this type of application and the maximum speed it can move the objects is considered to be insufficient. One of the main goals of using a robotic manipulator is then, to increase the moving speed of the objects without losing them. Below a more detailed analysis of the XCarve machine is presented together with a set of requirements for the robotic manipulator. Next, the robotic manipulator's design process and construction is presented. Then, testing of the robotic manipulator is executed to analyze if requirements are met. Finally, conclusions and future recommendations are given based on the results.

2 ANALYSIS

In this section an analysis of the test bed with the XCarve is presented. The limitations, problems with software and hardware will be listed. Then the research question and the sub questions are specified. From there a set of requirements are prescribed for the 2 DOF robotic arm based on the previous points.

2.1 *Previous setup analysis*

The previous setup corresponds to a XCarve CNC machine in which the drilling component was exchanged with an electromagnet. The XCarve is placed under a table with a glass surface, on the surface there are 3D printed objects with a magnetic bottom surface. The XCarve then is controlled to go to the objects, activate the electromagnet and drag the object to the desired position. Regarding the software of the set up, Unity 3D is used to create a user interface. The user interface is designed to send GCode through serial communication to the XCarve microcontroller (ATmega328P-Arduino Uno). There are however some problems and limitations with the hardware and software which will be listed below.

2.1.a XCarve

Regarding speed the XCarve can move up to 8000 mm/min [1] or 0.33 m/s, this speed is considered to be insufficient and must be improved in future developments. Another downside of using the XCarve is the use of GCode, stepper motor and Arduino shield for the motor controller. This configuration performs well for CNC applications but it does not give much freedom for robotic applications. In the other hand the current Software generated with C# in Unity 3D suffers from instability, it stops working without warning and the XCarve does not respond to the GCode as desired. The paths generated consist of straight lines which do not avoid collision with other objects, which is an important feature to be included in future developments.

2.2 *Research questions*

Using a XCarve CNC for moving objects over the test bed does not perform as desired for reasons explained in the previous section. Then in this paper the focus of the research will be on answering the main question:

- Is it possible to use a 2 DOF robotic manipulator for the SPE test bed?

In order to answer the main question the following sub question must subsequently be answered as well:

- How fast can objects be moved with the manipulator without losing track of the objects?

- Is it possible to make trajectories to avoid other objects and move the objects with a desired accuracy?

2.3 *List of requirements*

After analysing the problems with the XCarve setup and considering the research question the following requirements for the 2DOF robotic manipulator can be seen in table 1:

Table 1: Functional requirements

Maximum linear speed	7.6/3 = 2.53 m/s
Reach dimension	45 cm
Accuracy	+- 1cm
Payload	Magnet weight (350 gr)

The linear speed 2.53 m/s is chosen based in the Scara 3DOF robot, which is a well known, high performance robotic manipulator, it can achieve up to 7.6 m/s in the X-Y-axis in the end effector, it is desired to move the object from one position to another in less than 5 seconds regardless the distance. The accuracy for locating objects for this application is not required to be high: +- 1 cm is concluded to be enough.

3 DEVELOPMENT AND PROTOTYPING

In this section the concept idea is developed and explained. The robot is chosen to have 2 DOF because the working environment is a table, then 2DOF should be able to reach any point over the surface inside a certain radius. Also, it is chosen to be serial manipulator over planar because this type of manipulator will deliver higher speed, since the maximum speed at the end effector is the summatory of the speed of both motor speeds time their link length, this allows to have a motor with relatively low speed: under 500 rpm.

3.1 *Electronics*

The electronics used are shown in table 2. They are chosen due to low cost, low energy consumption and meet initial requirements of torque and speed from quick calculation. The downside of using the JGA25-370 motor is the backlash +- 2°. The electromagnetic is reused from the previous set up with the XCarve machine.

Table 2: Electronics

Motor X2	JGA25-370
Motor drivers	L298n
Power supply X2	12 V
Electromagnet	25kg Electric Solenoid
Micro controller	Arduino Mega
Electromagnet relay	SRD-05VDC-SL-C
Electromagnet power supply	HF25W-SL-12

3.2 Cad design

The model is shown in figure 8. The model is designed to be mostly 3d printed with PLA. It consists of two joints, the first joint has an extra gear ratio of 13.2:1, the gears have a modulus of 1.5 which is finely produced by a standard 3d printer. A gear system was necessary at first to reduce the torque felt at the motor, since initially steel masses of around 1 kg were used to counterbalance the manipulator, however the masses were later replaced by support that are in continuous contact with the table that generate counter balance forces, this design lower the inertia felt at the joints and the stiffness required for the whole structure thus the 3d printed parts can perform well without excessive deflection. The counter support can be seen in figure 1 as the circular pieces in opposite direction to the electromagnet. Furthermore, the links are aluminum bars which are chosen for low weight and soft enough to be drilled with manual power tools. The gears are mounted on the piece seen in figure 2, the piece as well as the gears have circular engraving to place steel balls to avoid friction between plastic and plastic.. The design is under the table, with the first joint in line with the center of the table. The design summary is shown in table 3

Table 3: Cad model

Gears ratio 1	13.1
Gears module 1	1.5
Link 1	24 cm
Link 2	21.5 cm
Reach radius	45.5 cm

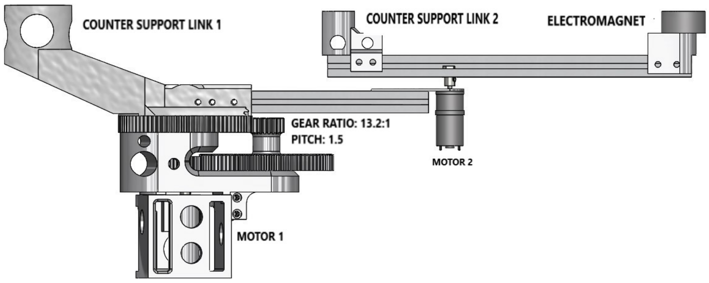

Fig. 1: 3D CAD design of manipulator

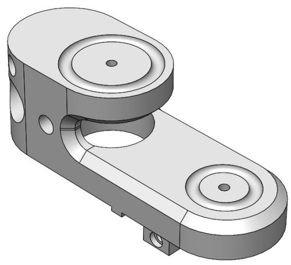

Fig. 2: Gears support

3.3 Controllers Design

PID controller with a low pass filter over Feed forward is chosen because the requirement for error is not high and it is simple to program in the Arduino micro-controller. In order to obtain the PID values for the system. A chirp signal is generated with Simulink and send over serial communication to Arduino (following the tutorial in [2]), the output is then recorded and used to obtain a second order transfer function, this with the help of a Matlab toolbox for control design. The function used function in Matlab is: tfest(iddata(y, u, Ts), 2), where 'y' is the output from the robot, 'u' is the input signal, Ts is the sample time 0.01 seconds and '2' resembles a second order transfer function, which is chosen for simplicity. Once the transfer function is found, the Simulink block PID Controller together with the auto-tune option is used to find appropriate values for the PID controller which will then be imported to the Arduino script. The block diagram for the tuning is shown in figure 3

The final values for the PID controllers of the motors are shown in table 4

3.4 Software and Communication

In this section the software that is used to control the robot from the user interface will be explained. 3 Different programming languages are being used with two communication protocols between them (Serial

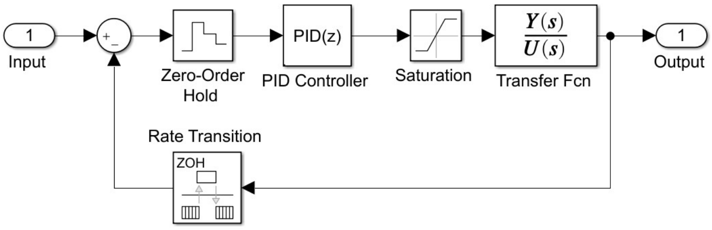

Fig. 3: Block Diagram for tuning PID controller

Table 4: Cad model

	Motor 1	Motor 2
P	101.76	22.270
I	186.85	59.24
D	0.87	0.539
N	42.13	17.45

and Socket). Serial is used for Arduino communication for its simplicity and well documented working principle. Socket is used for Unity and python communication, for being widely used for such purposes and is also well documented. Furthermore, in python a Server socket is running and listening for updates in the Unity interface, with python the trajectory is generated and sent through serial communication to the Arduino board. A class diagram can be seen in figure 4 with the main classes and functions.Further below a short explanations of the scripts in Python, C# and the Arduino is explained.

3.4.a Arduino and Micro controller for robot

The Arduino board Mega is programmed with the standard Arduino IDE, the open source library PID_H1 [3] written by Brett Beauregard is used to control both DC motors of each joint. The sample time for the PID controllers is chosen to be 0.02 seconds, it was found that it is the shortest sampling time that delivers good performance.

3.4.b Unity

The user interface can be seen in figure 5. The big square represents the glass table on a scale of 1:1. The middle circle represents the reach radius of the robot. The small cylinders around the middle circle represent the objects that must be moved from one position to another, the color of the cylinders can change to black if they are clicked upon and the gray cylinder follows the mouse position so that the new location of the picked cylinder can be chosen. The links of the robot could have been chosen to be longer to reach the corners as well, but then there is the risk of collision

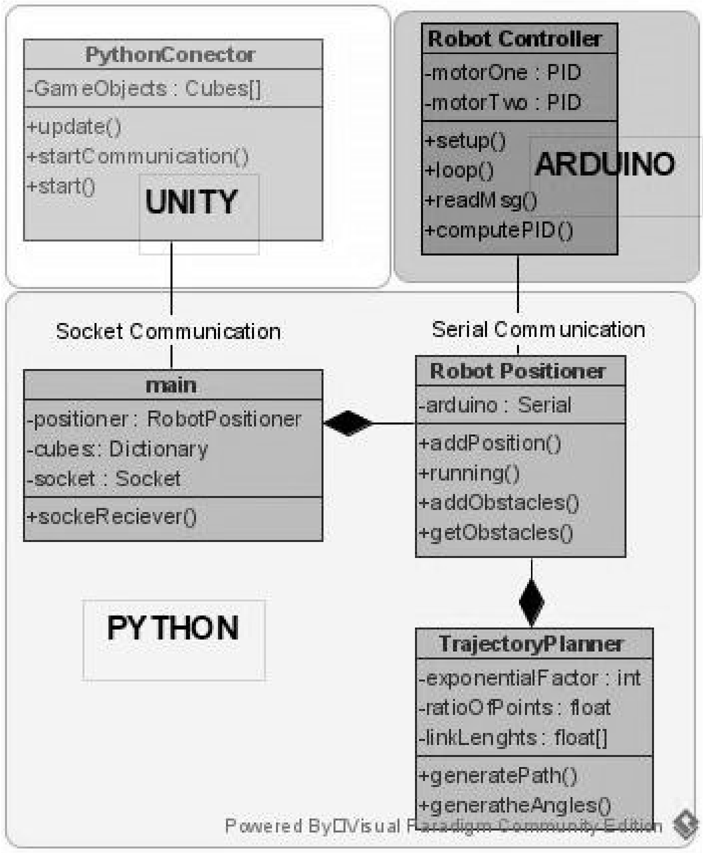

Fig. 4: Class diagram of the software

with the walls of the table, such risk was concluded to be unnecessary to take for a first version of 2DOF manipulator for the SPE.

3.4.c Python

In figure 4 the box named as python takes most of the space, this is because the logic to control the systems is written there. Python was chosen because it is considered as an easy to write and easy to handle data programming language. Furthermore, most of the functions can be seen already in figure 4. Also, in order to have socket communication with unity and serial communicating with Arduino, multi-threading is used with the library ”threading” [4], then it is possible to have parallel communication with unity and Arduino.

3.5 Trajectory generation

In figure 4 it can be seen in a class called **TrajectoryPlanner**, this class is created to generate the angles in time that are passed on to the Arduino board to drive the DC motors. There are multiple possibilities to generate the trajectories, it is possible to use open source library Robotic toolbox for python or create a

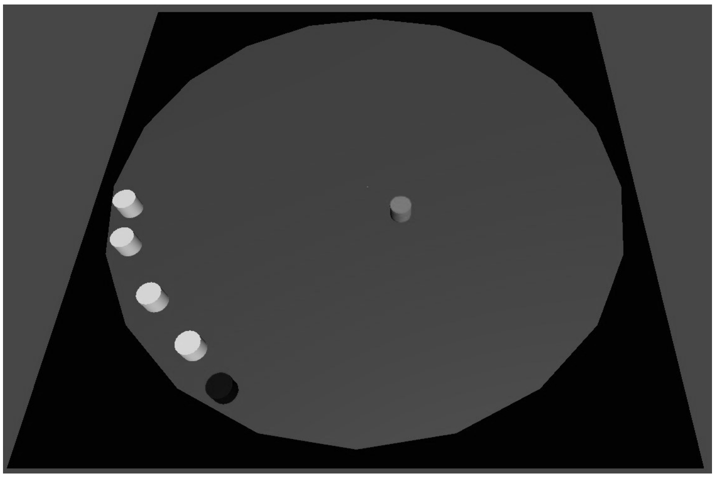

Fig. 5: User Interface

generator from scratch that matches the requirements of the SPE. Below both options are explained further. However, the final chosen method is the second one: create a generator from scratch, for reasons explained below.

3.5.a Robotic toolbox for python

The robotics toolbox for python is an extensive library for robotics application [5]. For the SPE testbed, it is possible to import the library to python and create and object representing a 2DOF planar robot with the code line **rtb.models.DH.Planar2()**, this object for instance, have many available function to perform inverse kinematics, forward kinematics, acceleration and velocity control, etc. Also from the same library it is possible to generate angles trajectory with a function that takes as input the desired initial and final angles, which are then interpolated and the output is ready to be sent to the Arduino microcontroller. This process is relatively simple and does not require writing extensive code. However, the disadvantage is that there is not much control on how the angles are calculated, for example: If it desired to change the angle from 0°to -90°, the library will create a path from 0°to 270°, this mean it takes the longest path rather than shorter path. This is only one example but there are more situations in which the library does not perform as expected. It could be possible to optimize it going through the source code and make the necessary changes, however instead of trying to optimize the library to match the requirements, it is preferred to create a generator from scratch, which will be explained in the next subsection.

3.5.b Preferred method

The preferred method to generate trajectories must fulfill the following requirements:

- All the objects should at all times keep a certain distance from each other to avoid attraction between objects due to electromagnetic forces. This means, there is a radius of each object that no other object should come into.
- When a object is moved from one point to another, the path should avoid to intersect other objects radius to avoid attraction due to electromagnetic forces
- In order to move the object as fast as possible without losing contact, the path should be planned to accelerate and decelerate slowly and reach its maximum speed in the middle of the path.

Considering the points above, the path is designed to consist of straight lines only, this with the objective of having control in the distance from object to another. In figure 6 it can be seen the path generated considering the points above. The algorithm works as follow:

- Process a straight line from the initial to the final position
- Give as input the location of the objects and draw a circle around the obstacles
- Check if the initial line intersect with the circles
- If the line intersects, draw a line perpendicular the **initial line** with the middle in the center of the radius of the object
- Then, draw straight lines taking the furthest points of the perpendicular line
- Join the lines avoiding objects with the initial and final point
- Calculate the inverse kinematics for all the points in the path (for this purpose the lesson available at [6] was used)

To generate the straight lines the well known equations of the line 1 is used.

$$y = m \cdot x + b \qquad (1)$$

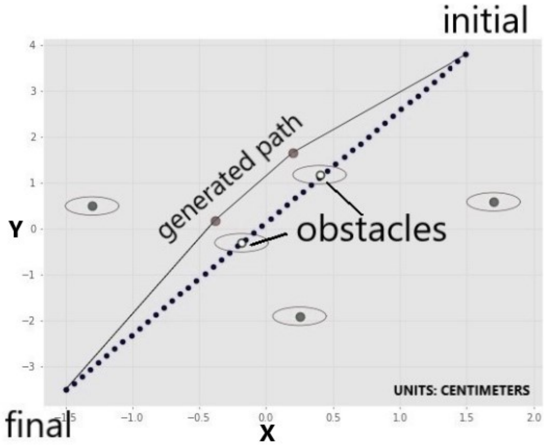

Fig. 6: Path planning

The path shown in figure 6 does not consider in any form acceleration. The path as it is will send position to which the robot will try to go as fast as possible.
To deal with acceleration the following approach is used:

- The use of the exponential function (equation 2) is used to generated exponentially distributed numbers from 0 to 1

$$y = a + b * \exp(-x) \tag{2}$$

- Initial values for the x-axis are generated with the standard linspace(initial point, final point, number of points) function
- The points from the linspace are interpolated with the exponential function, 0 being the beginning of the line and 1 being the end of the line
- The interpolated points are then passed onto the equation of the line 1 to generate the y-axis points
- For each of the points in the x and y-axis are calculated the inverse kinematic which then can be passed on to the motors

A line interpolated with exponential function line can be seen in figure 7. Some of the configuration of the line can be changed to achieve the desired speeds and acceleration.
In the next section different configuration of the trajectory are changed and analysed to see what are the limits and performance in moving object with the current configuration.

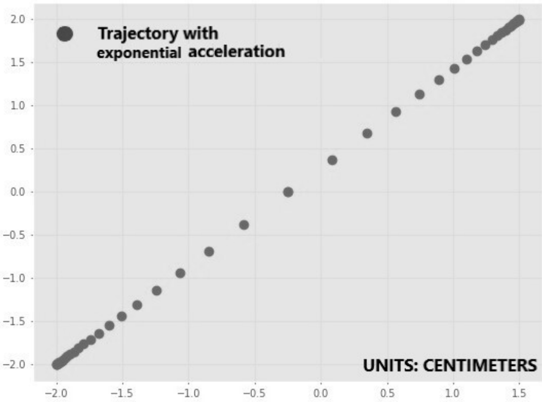

Fig. 7: Path with acceleration

4 RESULTS

The 2DOF robotic manipulator went through some iterations but the final prototype is shown in figure 8. A list of remarks for design are presented below.

- Cables represent a big challenge for this concept since one of the requirements is that the manipulator must be able to turn multiple turns. This can cause for cable to get stuck between the gears. The approach to cope with this is to pass the wires to the back of the lower links and with software, make the robot return to its zero position once it has completed three turns.
- There is backlash in the gears and the motors themselves, which makes the end effector to have a play of +- 2 cm
- Both robot joints are back drivable in spite of the many parts being causing friction: gears against gears, gears with bearing and supports, supports with glass surface,

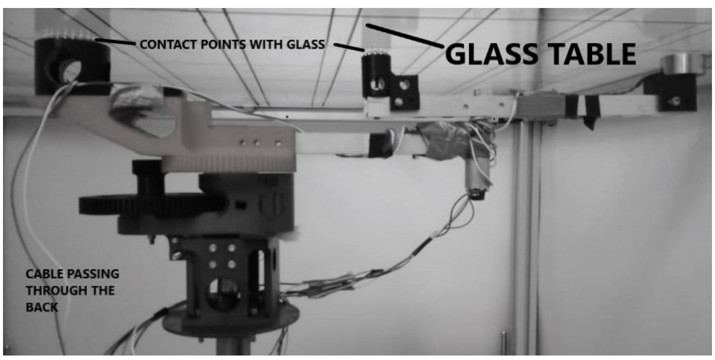

Fig. 8: Real prototype

4.1 Testing

To study if the robotic manipulator meets the requirements the generated and real trajectories are saved and plotted in figure 9 and 10. It is important to notice that the PID controllers have been optimized as best as possible following the procedure shown in section 3.3. Figures 9 and 10 show two different scenarios, the first scenario (figure 9) represents a straight trajectory and the angles from point to point change gradually and does not represent a problem for the robotic manipulator, the difference between the intended path and the one followed by the robot can be considered to be sufficient. On the other hand, in figure 10 it can be seen that the error has a particular increase in error, it follows an almost circular trajectory instead of a linear one, this error is denominated as ”special error”. This is because the angles required to follow that trajectory do not change gradually but have abrupt jumps, for example from 30 to 180 degrees. The robot is not able to move fast enough to cope with such jumps, the circular behaviour is the response of the manipulator trying to follow the trajectory but the actuator do not now have enough power to achieve the required speed. The so-called special error happens and increases the closer the trajectory gets closer to the center of the table.

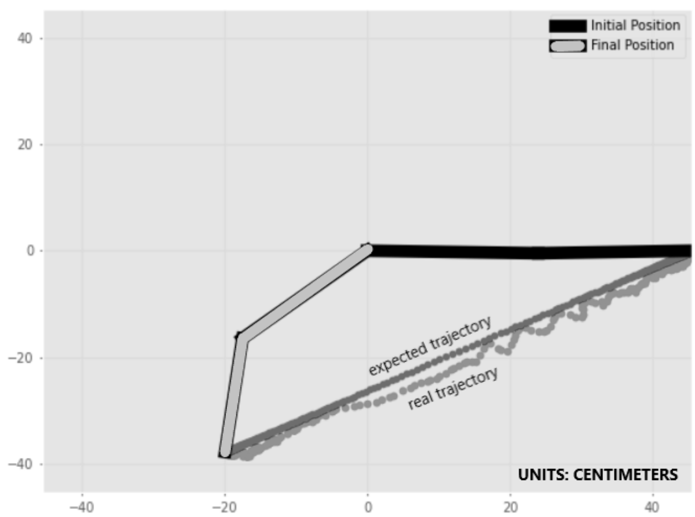

Fig. 9: Error while following a straight path

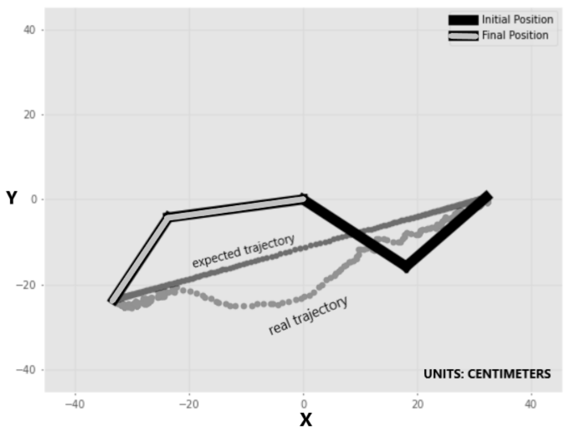

Fig. 10: Increased error by special case

To test the maximum speed that the robotic manipulator is able to move the objects over the table. Some parameters of the trajectory are changed so that it requires the minimum time to be sent to the robot. The expected and real speed is recorded and shown in figure 11.

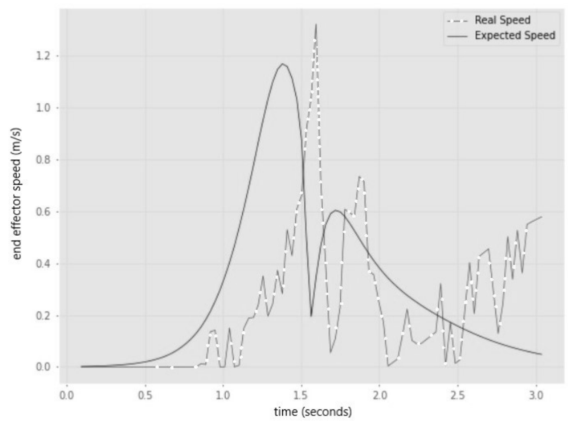

Fig. 11: Increased error by special case

Multiple tests were executed similarly as in figure 11, the max speed recorded is around 2.1 m/s. It can be seen that the velocity profile does not follow the expected velocity profile, this is again because the DC motors are not able to deliver enough torque and the amount of friction that exists in the system causes the manipulator to have inconsistent velocity profile.
Regardless of the DC motor not being able to follow the trajectory very smoothly and accurately during the so called ”special cases”, the manipulator can be considered to be stable for the following reasons:

- The objects can be moved without being lost
- The object are kept track of after many pick and place operations. Meaning the software as

well the hardware are well coordinated regarding communication and position tracking.

- The magnet is turned on and off where and when it is required
- The magnetic forces of the magnet do not interact with the magnet when the magnet is turned off. This means there are not undesirable attracting between the electron-magnet and magnetic objects.

Next, testing of trajectories to avoid objects are presented in figures 12 and 13

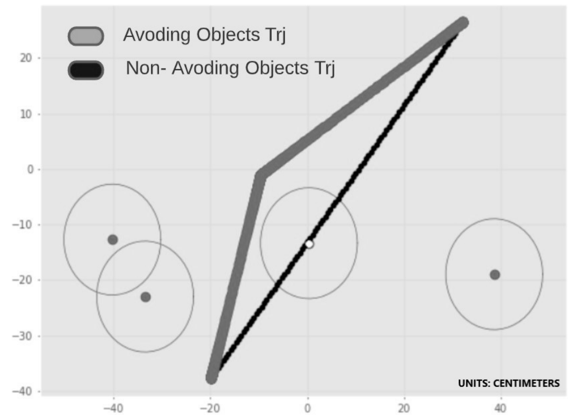

Fig. 12: Increased error by special case

In figure 12 the generated trajectory avoids the object that a straight trajectory would make contact with. In figure 13 the actual trajectory followed by the robot can be seen. It is seen that going from left to right, the first section of the trajectory is followed correctly, while the second section of the trajectory has an almost circular distortion which corresponds with the so denominated special error explained previously. This is again due to insufficient power that the motor can deliver as also seen in figure 10

5 CONCLUSIONS

The 2DOF robotic manipulator for the SPE testbed is capable of moving objects with a maximum speed of around 2.1 m/s. Compared to the previous set up with the X Carve CNC machine 0.3 m/s, the speed in moving objects has increased by more than 600 % percent, then regarding speed the robotic manipulator is significantly better. Also, during travelling the robotic manipulator does not lose the objects regardless of the speed, because the trajectory is generated to

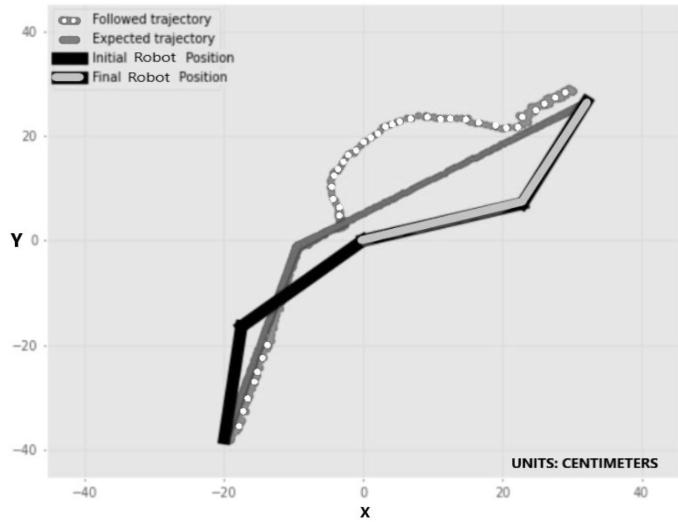

Fig. 13: Increased error by special case

include acceleration and deceleration. However, the behaviour for following trajectories is not satisfactory for the so called ”special cases’ ’, this due to the motors not being capable to deliver the amount of torque required to make the abrupt rotations. The same can be said regarding the speed profile which does on have smooth behaviour.

6 REFLECTIONS AND RECOMMENDATIONS

A robotic manipulator for the SPE testbed is capable of moving magnetic objects over a table with high speed without losing contact with them. There are however improvements that could be made if a future version of a robotic manipulator is made. The main recommendation is to replace the DC motors for higher torque and higher speed motors, the recommended motors are brushless DC motors like MIT Mini Cheetah [7] which is able to deliver 7 Nm. Those motors will be able to match the trajectory profile required for the SPE testbed. Also the lower the torque requirements, the aluminum links could be replaced by a more lightweight material, such as: fiberglass/carbon, plastic, etc. Another feature that could be improved is to add self homing features. On the other hand some feature that are recommended to keep are:

- To counter balance the links, the structure with the steel balls making contact with the bed is a reliable solution that does not add extra inertia joints of the manipulator.

- A modulus of 1.5 for the gears presents reliable solutions and sufficient accuracy. It is recommended to use the same or smaller modulus.
- Pass the wires of the motors and electromagnet to the back of the robot to avoid contact with gears.

A further reflection is given regarding the use of other types of robotic solutions for the SPE-test-bed. A cable driven robot is a parallel type robot, in which two or more cables are attached to the end effector and serve as a driven mechanism by pulling on them. The cables are pulled by rotational motors and a system of pulleys. This type of robot may better fit the SPE test-bed for the following reasons:

- The reach area of the robotic arm presented in this paper is circular with a radius of 45.5 cm. To reach the corners the links will have to be longer thus if care is not taken, the robotic manipulator can make undesirable contact with the walls of the structure. A cable driven motor has a square reach, able to get to the corners.
- Trajectories are desired to be straight to avoid collisions, but as shown in this paper it is complex for the robot to follow such trajectories without adding complexity to the system mainly; higher power requirements for sudden jumps in angles.
- Friction between gears and between the support and the table are eliminated.
- Simplified trajectory without adding extra complexity like sudden jumps in angle complexity.
- Use of cables allow for no-backlash end effector which represents extra accuracy.
- Simplified design because the use of cables as driving bodies.

Another solutions is the use of planar motors, which has been recently developed by Planar Inc [8] and used for packaging application. It consists of a body with magnets and a table section with coils inside; the coils generate magnetic fields that subsequently move the magnetic objects over the table in the desired direction and speed. Some of the advantages are:

- The precision it can achieve is in the nanometers range [9] which is more than enough for the SPE test-bed
- Has 6 degrees of freedom
- There is no limit in the amount of objects it can move at once. This opens to new possibilities in how the SPE testbed performs. Not only data can be projected over the table but also it can be combined with multiple moving objects representing some other data point

The disadvantage is the complexity of developing such a system for the SPE test-bed from scratch. As mentioned in the video [9] the system also needs a cooling system to dissipate heat, which adds extra complexity and with complexity also costs are increased.
To conclude, if a new version of a 2 DOF robotic manipulator is built, it is strongly recommended to use brushless MIT Cheetah motor, this would improve performance for the so called "special errors". But, it is better recommended to replace the serial manipulator for a 2 DOF cable driven parallel robot, which has the same number of motors, no complexity to generate trajectories when getting close to the center of the table, no backlash from gears and square reachable area. Furthermore, the best solution would be the use of planar motors, because not just multiple objects can be moved at the same time, but also each object have 6 degrees of freedom, the downside is the complexity of the system, which also needs a cooling systems and other features specially for the SPE testbed, that is why it is also expected to remarkably higher cost that the other two solutions.

ACKNOWLEDGEMENTS

I would like to thank DR. ROY DAMGRAVE for allowing me to be part of the SPE test bed, and also giving me good advice and insights along the way.
I would also like to thank the people working on the workshop in the Horst Building at the University of Utwente, who helped me during the construction of the prototype as well as giving design advices.

REFERENCES

1. Bart Dring, What is maximum feed?, `https://discuss.inventables.com/t/what-is-maximum-feed/8657`.
2. Leonardo Mariga, Simulink-arduino-serial, `https://github.com/leomariga/Simulink-Arduino-Serial`.
3. Brett Beauregard, Arduino pid library - version 1.2.1, `https://github.com/br3ttb/Arduino-PID-Library`.
4. Python, threading.py, `https://github.com/python/cpython/blob/3.9/Lib/threading.py`.
5. Peter Corke, Robotics toolbox for python, `https://github.com/petercorke/robotics-toolbox-python7`.

6. Peter Corke, Inverse kinematics for a 2-joint robot arm using geometry, https://robotacademy.net.au/lesson/.
7. Ben Katz, The mini cheetah robot, https://build-its.blogspot.com/2019/12/the-mini-cheetah-robot.htmlhttp://www.test.org/doe/, (2019).
8. Planar Inc, https://www.planarmotor.com/, (2021).
9. Das Portal für Smart Production Solutions, Acopos 6d: Markus sandhöfner, geschäftsführer br deutschland, erläutert details, https://youtu.be/4UF7vRaMoHI, (2021).