



university of  
groningen

---

# Detection of a human ready for object transfer to facilitate human-robot interaction

---

Bachelor Integration Project

*Author:*

Daniël Knapper - S4374606

*1st supervisor:*

dr. E. Wilhelm

*2nd supervisor:*

dr M. Taheri

Groningen, June 14, 2024

BSc Industrial Engineering and Management  
Faculty of Science and Engineering  
University of Groningen



## CONTENTS

<b>I</b>	<b>Abbreviations</b>	3
<b>II</b>	<b>INTRODUCTION</b>	5
<b>III</b>	<b>RESEARCH DESIGN</b>	5
III-A	Problem analysis . . . . .	5
III-B	Stakeholder analysis . . . . .	5
III-C	Research objective . . . . .	6
III-D	Research questions . . . . .	6
III-E	Research steps . . . . .	6
<b>IV</b>	<b>LITERATURE REVIEW</b>	6
IV-A	Vision based recognition . . . . .	6
IV-B	Force sensors . . . . .	7
IV-C	Torque sensors . . . . .	7
<b>V</b>	<b>METHODS</b>	7
V-A	OpenMANIPULATOR-X . . . . .	7
V-B	Experiments . . . . .	9
	V-B.1 Tools . . . . .	9
	V-B.2 Methods . . . . .	10
<b>VI</b>	<b>RESULTS</b>	10
<b>VII</b>	<b>DISCUSSION</b>	11
VII-A	Limitations . . . . .	11
VII-B	Further research . . . . .	11
<b>VIII</b>	<b>CONCLUSION</b>	11
<b>IX</b>	<b>ACKNOWLEDGEMENT</b>	11
<b>X</b>	<b>APPENDIX A</b>	14
<b>XI</b>	<b>APPENDIX B</b>	19
<b>XII</b>	<b>APPENDIX C</b>	30

## I. ABBREVIATIONS

Discrete Technology & Production Automation (DTPA)  
ENgineering and TEchnology institute Groningen (ENTEg)  
Degrees of Freedom (DOF)



**Abstract**—The problem for robots is to identify whether humans are ready for object transfer. In this report, the kinematics of the OpenMANIPULATOR-X will be researched to design a torque controller. Experiments are conducted with the torque controller designed in Arduino IDE to see whether this problem can be solved. This method cannot solely be responsible for adequate detection of whether the robot knows the human is ready for object transfer. However, with an extension of sensors, this method can be a suitable method to solve this problem.

**Index Terms**—Human-robot interaction, object handover, torque control

## II. INTRODUCTION

Robots have been developed continuously in the past to aid humans with various tasks [1] and further technological developments are expected in the future. One of these developments is the transfer of objects from robots to humans. This challenge is researched for the past decade and the robot's dexterity has improved significantly [2]. Transferring objects between two entities is an everyday task that is straightforward for humans however, it is challenging for robots to achieve the same fluency and efficiency as humans. The main challenge is obtaining the same robustness and flexibility between the human and the robot as humans have amongst each other [2].

The transfer of objects between a human and a robot is quite challenging. The robot must detect when a human is ready to take the object. This is possible by various methods for example a camera can be used to visualize the human's upper body [3] [4]. This method tries to predict the position and the trajectory of the arm using an algorithm. This method relies on the use of Bayesian statistics, the intention of the human reaching can be predicted [5] [6]. Another method to predict the trajectory of the human arm is with a trajectory generator that translates the trajectory in Cartesian coordinates [7]. Research from the past shows that it is possible to give and take objects from a robot, however, this process is considerably more static than humans interacting with each other [8]. Another method shows that a torque controller can be responsible for the release of an object. A kinematic model of the robot can be created and visualized to identify the torque present in all the joints. Then when the robot reaches a certain torque it can open the gripper and release the object [9].

This research will find out if the robot can detect when a human is ready to take the object by using the torque sensors in the robot's motors. These experiments will be assessed on two main criteria, namely time and the success rate of the experiment. The novelty of this study is that this is not yet done for the OpenMANIPULATOR-X robot. This study will address the key issue of the robot being able to detect when the human is ready for object transfer.

The structure of this report looks as follows. After the introduction, the research design will be presented in which

the problem and research objective will be elaborated on. After that literature review is done on different methods of identifying whether a human is ready to take an object from the robot. The core of this research consists of conducting experiments in which the success rate of the detection of a human ready for object transfer to facilitate human-robot interaction will be researched. The research will be finalized with a discussion and conclusion of this method and the limitations and further research will be discussed.

## III. RESEARCH DESIGN

This chapter provides an overview of the research's general setup. This includes the problem analysis, the stakeholders involved, the objective, the questions, and the steps that need to be taken.

### A. Problem analysis

There are a few challenges in the transfer of objects between humans and robots. The current robots lack the capability to adapt to new tasks or environments and can improvise when these things change. The robots need to be programmed in a certain way so that they can repeat the same task over and over again in the same environments with the same constraints. Humans on the opposite are capable of adapting and improvising with uncertainties [10]. The robot's lack of capability to adapt can be minimized by using different sensors and combining these sensors.

Another significant issue is the high unpredictability of human behavior. Humans may try to grab an object but the second they touch it they change their mind and decide not to grasp the object anymore. This causes the robot to release the object when it identifies the human's intention to take it, however, if the human steps back for example, the object may fall, potentially leading to fatal consequences in factories.

This analysis results in the following problem statement: The challenge lies in the inability of the robot to accurately identify the readiness of a human to take objects, requiring the selection and testing of a single method to address this issue.

### B. Stakeholder analysis

The following stakeholders are involved in the project: RUG, Dr. Elisabeth Wilhelm DTPA, Dr. Mozhdeh Taheri, manufacturing companies, and humans. The stakeholder analysis can be seen in figure 1. The stakeholder analysis is done by following the theory of Ackermann and Eden [11].

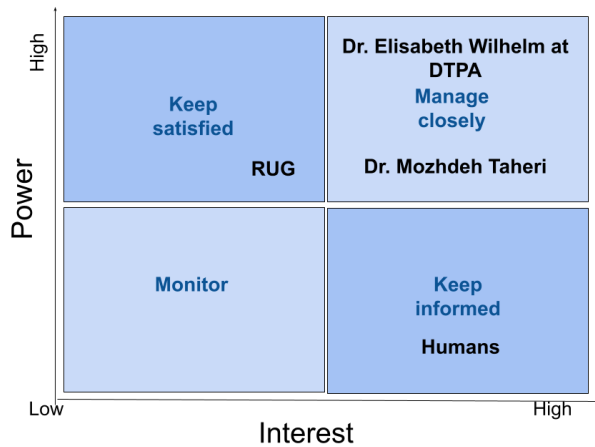


Fig. 1. Stakeholder analysis

**Dr. Elisabeth Wilhelm at DTPA** is a stakeholder with high interest since she is the problem owner, this means that she also has high power. Therefore she is a stakeholder who needs to be managed closely, the same accounts for the 2nd supervisor **Dr. Mozhdeh Taheri**. The **RUG** has some interest in this research, but relatively less than dr. Elisabeth Wilhelm. However, the RUG does have significant power which means the RUG needs to be kept satisfied. The end users of this project are the **humans likely to interact with the robot**. These humans work in the industrial sector where robots can aid humans with their tasks. For example, production line work that is not possible only with robots but humans are still needed. In this case, robots can help humans. These people have high interest but they have no power over this project. That is why they must be informed about the project's progress. They can be informed with updates on the project process through emails.

### C. Research objective

A meaningful research objective needs to be SMART, specific, measurable, attainable, relevant, and time-bound, to provide a clear direction for the planning and implementation of the research [12]. The research objective is to evaluate the effectiveness of a sensor in detecting when a human is ready to take an object. The objective is specific as the end goal is the algorithm for the torque control combined with the robot to enable the robot to do its task. It can be tested according to the error and the time it takes which makes it measurable. The objective is attainable and highly relevant because the use of this human-robot interaction can help humans transfer objects and it also adds benefits to a human-robot production process. The objective is time-bound since it has a time span of 14 weeks with an end deadline on the 14th of June.

### D. Research questions

**Main research question:** *What is the success rate of the robot when using torque control to identify when a human is ready to take the object?*

#### Sub-questions:

- 1) What is the error of the human-robot interaction for the object handover?
- 2) What is the time it takes to complete the object handover?

### E. Research steps

The following steps need to be taken to answer the research questions:

- 1) Understand the robot and its limitations
- 2) Develop a controller in Arduino IDE
- 3) Determine threshold value to use in experiments
- 4) Test the controller by conducting the experiments

## IV. LITERATURE REVIEW

This section introduces different methods and their corresponding sensors. The benefits and drawbacks of each method with their sensors will be elaborated on.

### A. Vision based recognition

The first method is to identify the position of the arm by using a camera. First of all this method identifies the face of a human so that the robot knows it can expect a command and it needs to react. After that, the arm is analyzed based on the information of the region, the periodical motion, and the shape of the arm. This needs configuration to know when the robot needs to act [3]. Another vision-based recognition is using Bayesian-based trajectory prediction. This method uses the Bayesian method to forecast the trajectory of the human arm. It uses a cost function and Markov's hypothesis to limit the amount of data that needs to be used. The drawback of this method is that it is sensitive to noise. Next to that, this method does not need prior memory and only uses the memory stored in the accumulated probability. When the target of motion changes this method is still suitable and does not need any adaptations [5]. In another study, a low-cost camera was used to identify the human hand position and the trajectory. This was done by using different colors. The human had a bracelet with a purple color on it and there was a blue square on the ground. The high-accuracy OptiTrack motion capture system was used to define the position of the human hand [13]. This method is relatively simple however, the advantage is that the equipment needed is significantly less expensive than in other studies. Another disadvantage is that there is only one camera being used which can lead to occlusion and errors in the detection. A notable drawback of cameras being implemented in the workplace is that it decreases the employee's motivation [14].

### B. Force sensors

One of the sensors that can be used to identify the position and trajectory of the human arm are force sensors. Some examples are force, tactile, and optical sensors. The force sensor can be used to measure when a human is grabbing the object held by the robot and applying force on the object by pulling the object towards him. The robot can sense this and will release the object if the human is applying enough force on the object [15]. The following paper has researched using pressure sensors to indicate the release of an object by a certain threshold [16].

Another type of force sensor are tactile sensors. Tactile sensors are force sensors that are attached to the human instead of the robot. Literature shows that this can be used for example on a glove. Then the human wears a glove with tactile sensors on it and the robot knows whether the human wants to take the object. With this glove and the tactile sensors, the robot also can identify whether the object the human is holding is soft or rigid [17].

### C. Torque sensors

Another method is to use torque sensors. Torque sensors can sense the current torque in the motor when a certain force is exerted on the motor. The torque is the force times the length measured perpendicular to the force. This means that at the end of the arm, where the gripper is located, the torque is the most. The following paper presents experiments where force/torque was used to learn more about object transfer [18]. Another research shows that torque sensors and a camera can be used to identify whether to release the object or not [9]. This paper shows the results by only using torque data, only using vision data, and the data for the combination of both.

## V. METHODS

This section discusses the method used for this research and the robot's setup. Furthermore, it will give an in-depth view of the mathematical model of the robot in terms of the kinematics and the experiments that will be conducted will be explained.

Different methods have been proposed and elaborated on and the method using a torque controller algorithm will be used for this research. Past research shows that this method can enable the robot's detection of a human ready for object transfer. However, the difference is that this research will be done for the Openmanipulator-X robot.

This method does not need any additional sensors since the torque controller is based on the sensors the Dynamixel motor has already been equipped with. The benefit is that this makes the research low-cost compared to other methods. No additional sensors are needed since the Dynamixel motors in the robot already have a sensor that

can measure the current.

### A. OpenMANIPULATOR-X

The robot used for these experiments is the OpenMANIPULATOR-X from Robotis. This robot has 4 DOF and a gripper, which means a total of 5 DOF. The robot is equipped with five Dynamixel XM430-W350-T motors that enable the robot to move. The motors have torque, velocity, position, and trajectory sensors that enable the measurement of position, velocity, and current. The dimensions of the robot can be seen in figure 2.

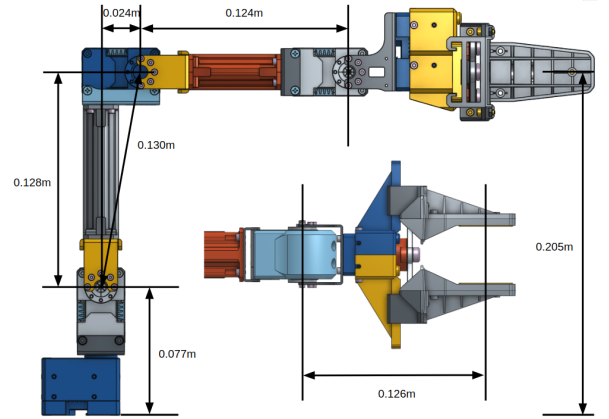


Fig. 2. Dimensions of the OpenMANIPULATOR-X [19]

The robot has four joints that can move in two directions, the fifth motor steers the gripper and enables the gripper to open and close. Torque control is chosen to identify whether the robot can detect when a human is ready to take the object. Therefore a mathematical model needs to be illustrated to identify all the forces and torques present in the robot. As stated before the configuration of the robot while conducting the experiments is shown in figure 2. This means that the main ballast and torque will be on the third joint, which is the joint that makes an almost 90-degree angle. The results of the measured current when applying a force on the gripper that pushes the gripper down can be seen in Appendix B. This supports the statement that the motor that is in the third joint has the highest difference in current. The force applied in this model is at the end of the arm at the gripper. Next to the pulling force on the model, there is a constant gravitational force pulling on the robot. In figure 3 a simplistic model of the robot arm where the moment and the force acting on the robot can be seen. The numbers 1-4 correspond to the four joints the robot arm consists of. Equation 1 shows the general formula for torque, in this formula  $F$  is denoted as all forces acting on the end of the gripper. This is always the gravitational force  $F_g$  and in this case, the pulling force of the human which is denoted as  $F_p$  as well.

$$\tau = F * l \quad \text{with} \quad F = F_g + F_p \quad (1)$$



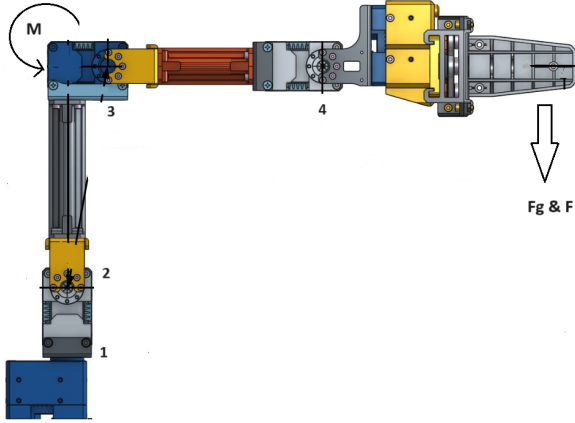


Fig. 3. Mathematical model of the robot arm

To identify the torque in each joint the Denavit-Hartenberg (DH) conventional technique will be used. This is done in previous studies and with the use of this the torque will be calculated [20] [21] [22] [23]. This technique first requires an analysis of the configuration of the robot arm. After this analysis the DH parameters can be determined, the results can be seen in table I. The next step involves calculating the transformation matrix. First the transformation matrices between every joint need to be calculated to obtain the final transformation matrix.

TABLE I  
DH PARAMETERS OF OPENMANIPULATOR-X

Joint	$\theta_i(\text{degree})$	$a_i(\text{degree})$	$a_i(m)$	$d_i(m)$
1	$\theta_1$	90	0	0.077
2	$\theta_2 - \theta_0$	0	0.130	0
3	$\theta_3 + \theta_0$	0	0.124	0
4	$\theta_4$	0	0.1216	0

The general formula for the transformation matrix can be seen in equation 2. This shows the transformation of the i-frame relative to the i-1 frame.

$$T_i^{i-1} = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i)\cos(a_i) & \sin(\theta_i)\sin(a_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(a_i) & -\cos(\theta_i)\sin(a_i) & a_i\sin(\theta_i) \\ 0 & \sin(a_i) & \cos(a_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Now the matrices between each joint,  $T_1^0 * T_2^1 * T_3^2 * T_4^3$  need to be calculated to find  $T_4^0$ . This is done by filling in the DH parameters into equation 2 according to the correct joint. The joints are counted from the bottom of the robot to the joint just before the end effector which counts as the last joint. This results in the following 4 matrices:

$$T_1^0 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

$$T_2^1 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$T_3^2 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

$$T_4^3 = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & a_4\cos(\theta_4) \\ \sin(\theta_4) & \cos(\theta_4) & 0 & a_4\sin(\theta_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

Now all these 4 matrices need to be multiplied with each other to obtain  $T_4^0$  as can be seen in equation 7. It is easier to first calculate  $T_1^0 * T_2^1$  and  $T_3^2 * T_4^3$  and then multiply these outcomes. The important values from matrix  $T_4^0$  are the position vector denoted as the values in the matrix as 14 which is  $p_x$ , 24 which is  $p_y$ , and 34 which is  $p_z$ .

$$T_4^0 = T_1^0 * T_2^1 * T_3^2 * T_4^3 \quad (7)$$

The complete calculations of the transformation matrix  $T_4^0$  can be seen in Appendix A. This transformation is split up into a rotation matrix and a position vector. The position vector is needed to obtain the Jacobian matrix and this vector is a 3x1 column vector. The entries of the vector are  $T_{14}, T_{24}$ , and  $T_{34}$  of the transformation matrix. The next step is to obtain the Jacobian matrix, to do this the entries in the position vector will be differentiated with respect to all four angles as seen in the appendix. After that, the transpose needs to be taken of the Jacobian matrix and this matrix needs to be multiplied with the force vector. This is done by using equation 8. The torque is in this case a vector that includes the four different torques present in the robot.

$$\tau = ({}^0J)^T * F \quad (8)$$

The final equations for the torque in each joint can be seen in Appendix A as well as all the matrix multiplications. This supports the statement that the highest torque will be expected on joint 3 with the robot in its current position. In figure 4 the four motors can be seen excluding the one that only takes care of rotating the arm. Id 12 is responsible for joint 2, id13 is responsible for joint 3, id14 is responsible for joint 4, and 15 represents the gripper. From this figure, it shows that the highest torque will be on the third joint. However, joint 2 will also experience some torque.

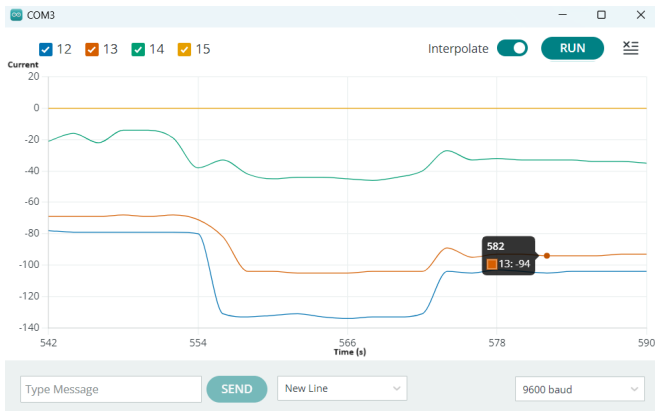


Fig. 4. Current measurements in Arduino for 4 different motors

## B. Experiments

The experiments will only include taking the object from the robot. Since the main goal is to check whether the robot identifies that the human is ready to take the object. This means that the robot is already holding the object, and how the robot grasps the object is out of the scope of this research.

For the experiments, the initial position of the robot is depicted in figure 5. From this initial position, the robot is already holding the object, while a human is trying to take the object from the robot. The desired goal is that the robot will open the gripper after measuring a small torque that translates into the human pulling the object. In more detail, this means that the time it takes for the handover to be complete will be timed using a stopwatch. The time will run from the moment the participant makes contact with the object held by the robot. The time cannot exceed 2 seconds because past 2 seconds the handover will not have the desired smoothness. The error as spoken in subquestion 2 will be defined as the percentage of completed handover tasks. More specifically this means that the human takes the object from the robot's hand and that the object does not fall on the ground. The object can not be caught by the human but the human has to take it out of the robot's fingers. The limit of dropping the object is set to 1 cm, if the object is caught below this the handover is identified as unsuccessful. The objects chosen for the experiments can vary but cannot exceed 500 grams since this is the maximum payload of the robot. Additionally, the objects need to have an industrial application because that is where the robot will be stationed and used.

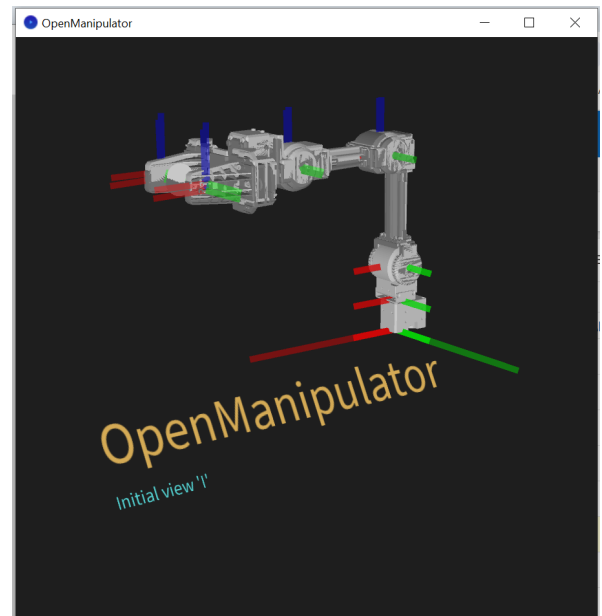


Fig. 5. Initial position of the robot

As discussed earlier the torque cannot be directly measured from the Dynamixel motors, this means that with the use of the current and the performance graph of the motors that can be seen in figure 6 the torque can be calculated. This is needed to establish the threshold that will be used to steer the gripper when the object needs to be released.

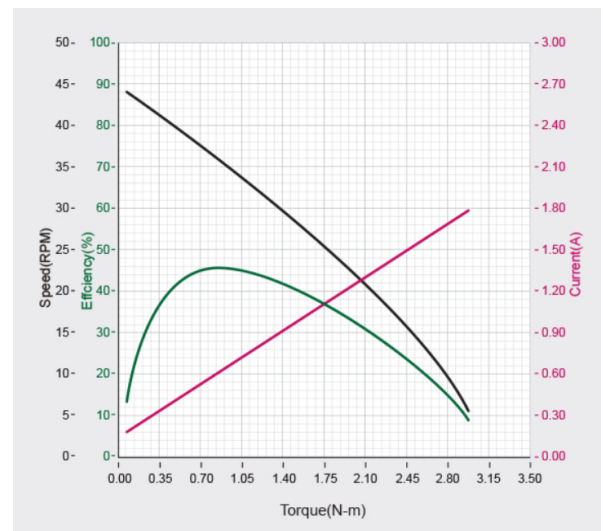


Fig. 6. Performance graph for the XM430-W350 motors [19]

1) *Tools:* The first and most important tool used in this research is **Arduino IDE 2.3.2**. Arduino IDE is a development environment that facilitates writing code on Arduino boards. In this program, the controller for the torque control will be programmed using existing libraries. Another tool that is used and provided by the manufacturers of the robot is **Processing 4.0**. This tool is a demonstration

software that has an easy controller that can be used to steer the robot in the desired position. The last tool used is from the manufacturers of the Dynamixel motors itself and is called **Dynamixel WIZARD 2.0**. This program can be used to measure all data that is measured in the Dynamixel motors. This will mainly be used to measure the current in the motors while conducting the experiments.

2) *Methods*: The first and second subquestions will be answered by conducting experiments and measured by Arduino and Dynamixel Wizard 2.0. This means that when the pulling force of the human is above this standard the human pulls too hard and this does not count as a complete handover. Afterward, conclusions can be drawn about the success rate of the robot and whether the design needs to be improved for future work.

## VI. RESULTS

First of all the threshold value needs to be identified. This has been done with a small amount of tests to see which threshold reacted the best. This threshold defines with what increase or decrease of torque the gripper of the robot opens and thus releases the object. The measured current of the first initial tests can be seen in figure 8 - 16. In these figures, the current can be seen along with the current and goal position of the joint. The torque can be estimated using the performance graph 6 and the current measurements. The expected threshold would be somewhere between the range of 0.3 - 0.6. After conducting the tests the results show that a threshold value of 0.5 is the most suitable in combination with the torque controller.

After the threshold value is set, more experiments are conducted. These experiments consist of testing the torque control algorithm for different objects and different participants. In these experiments three different objects were used: a piece of wood, a screwdriver, and a tape measure. These objects are chosen because they all have industrial appliances. These three objects with their weights can be seen in figure 7. For every object, the participant is asked to grasp the object 6 times in total, to rule out any luck.

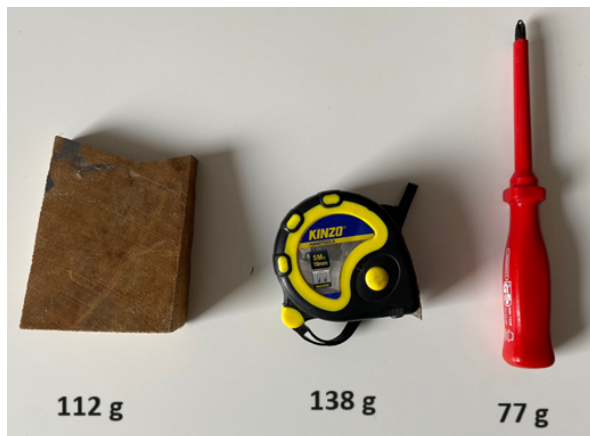


Fig. 7. The three objects used for experiments with corresponding weight

The experiments for each object were conducted 6 times per participant. The time and error were measured in this experiment. The participants were instructed to grasp the object how they would normally do when exchanging objects between humans. Another instruction was that when the object will drop they do not try to catch the object. This means that the robot has released the object too early. The results of the experiments are shown in Appendix B. In the tables, the error is denoted with a 0 or a 1, 0 meaning a fail, and a 1 meaning a successful transfer. A failure in the object transfer can be caused by the object being dropped or not handed over smoothly. Another cause for the failure is when the time it takes exceeds two seconds. This limit is chosen to maintain efficiency in the production process. The success rate of all three individual objects as well as the total success rate are shown in table II. The screwdriver scored the lowest success rate and the piece of wood the highest. The reason for this is probably the difference in the dimensions of the objects. The piece of wood has simple dimensions, nearly square in shape, while the screwdriver has more complex dimensions to grasp and has a more smooth surface. this makes it harder for the robot to correctly measure the torque when interacting with the human.

TABLE II  
SUCCESS RATES OF THE INDIVIDUAL OBJECTS IN OBJECT HANDOVER

	Number of trials	Number of successful handovers	Success rate
Piece of wood	36	22	61.11%
Tape measure	36	21	58.33%
Screwdriver	36	20	55.56%
Total	108	63	58.33%

In table III the means of the time it took for the handovers to be completed can be seen. This table shows that the unsuccessful handovers took longer than the successful handovers in most cases. Only the screwdriver has a higher mean for the successful handovers. However, when looking at the raw data it can be seen that the screwdriver has many cases where the time was relatively low, which translates into the screwdriver being dropped or even falling.

TABLE III  
COMPARISON OF THE MEANS OF THE DIFFERENT OBJECTS FOR SUCCESSFUL AND UNSUCCESSFUL HANDOVERS

	Mean unsuccessful handovers (s)	Mean successful handover (s)	Total mean (s)
Piece of wood	1.45	1.38	1.41
Tape measure	1.71	1.55	1.53
Screwdriver	1.51	1.55	1.53
Total	1.56	1.49	1.49

When analyzing the data, it is revealed that one participant achieved only two successful handovers, while another achieved five successful handovers. Both of these participants were conducting the experiments with a piece of wood. Looking into the participant that had only 2

successful handovers it can be seen that two of the times it takes longer than two seconds that is set. This means that the robot releases the object too late causing a fight with the robot and leading to an unsuccessful handover. When looking at the participant who obtained five successful handovers, the time it takes for him is almost every time very close to the mean.

The variance can be calculated for each object to analyze which object was easier to grasp. equation 9 shows the formula to calculate the variance, where  $\bar{x}$  is the sample mean and  $n$  is the sample size. The sample size is 36 for each object and the sample mean is taken from table III. The variances for all three objects can be seen in table 9. A high value for the variance means that there is a larger spread in the data. According to the variances, the tape measure has the highest spread on the data.

$$S^2 = \frac{\sum(x_i - \bar{x})^2}{n - 1} \quad (9)$$

TABLE IV  
VARIANCE FOR ALL THREE OBJECTS

	Variance ( $S^2$ )
Piece of wood	0.3989
Tape measure	0.6424
Screwdriver	0.3576

## VII. DISCUSSION

The success rate of the OpenMANIPULATOR-X equipped with the torque controller was 58.33% for the total handover. This is lower than other studies, for example, one study has a 98% success rate [9]. There are a few differences between their experiments. Their latter system utilized an advanced sensory integration approach, combining joint torque sensors with an RGB-D camera. This allowed for real-time classification of human actions through a CNN-based classifier for torque data and an SSD object detector for visual data, significantly enhancing the robot's decision-making capabilities during handovers. Additionally, the manipulator in this study featured a more dexterous gripper, capable of nuanced control beyond simple opening and closing motions. Their gripper consists of a 3-finger mechanical hand, and the OpenMANIPULATOR-X has only a gripper with two fingers that can only open or close.

### A. Limitations

The experiments are conducted with a limited amount of objects. The main reason is that the gripper has a maximum payload of 500 grams. Besides this limitation, only three objects were used due to time constraints. For better results more objects need to be used and also different kinds of objects. The three objects used were all three rigid and not soft. Soft objects would be harder to grasp and probably harder to use in combination with the torque controller.

Another limitation is the test participants. All participants were younger than the target population of industrial workers. Age is relevant in this context as the reaction time and grip strength deteriorate with age [24]. Another limitation is the small number of subjects that conducted the experiments. There are only 6 participants used for each object. All participants were students aged between 18 and 25. This limits the research since the target group is industrial workers.

### B. Further research

Further research is based on the limitations discussed in the previous section. It would be interesting to implement this controller with a robot that can handle a higher payload. Another question that can be addressed in the future is how to improve the success rate. This means that for future research other sensors can be added so a combination of sensors would be used to identify the success rate. One interesting sensor is the 3d force sensor from the company Contactile [25]. This force sensor can measure all three force components, leading to a precise reading of the slip force generated by pulling the object. Combining this with the torque controller the success rate can be improved.

## VIII. CONCLUSION

The problem of a robot-human handover has not been completely solved since the error rate is still too high. An error rate of 41.67% is not feasible to use in the industries. However, the experiments showed that a handover based on the torque of a single motor of a single joint is in general feasible. The error rate could potentially be improved by implementing all motors in the torque control algorithm. The torque controller is now based on one of the motors that experience the highest torque, however, the other motors also experience torque. If the torque controller can be expanded by adding all motors to control the gripper, the success rate will most likely improve. To do this a threshold for all motors need to be defined then when all threshold values are met the robot can release the object. This will add a more stable algorithm that does not only rely on one motor but on all motors.

## IX. ACKNOWLEDGEMENT

The author of this report would like to thank dr. Elisabeth Wilhelm and dr. Mozhdeh Taheri for the incredible guidance and help. This work is supported by the The Discrete Technology & Production Automation (DTPA) group at Engineering and Technology institute Groningen (ENTEG).

## REFERENCES

- [1] J. Baraglia, M. Cakmak, Y. Nagai, R. P. Rao and M. Asada, "Efficient human-robot collaboration: When should a robot take initiative?" *The International Journal of Robotics Research*, **journal** 36, **number** 5-7, **pages** 563–579, 2017.
- [2] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, **journal** 364, **number** 6446, eaat8414, 2019.
- [3] D. Kim, J. Lee, H.-S. Yoon, J. Kim and J. Sohn, "Vision-based arm gesture recognition for a long-range human-robot interaction," *The Journal of Supercomputing*, **journal** 65, **pages** 336–352, 2013.
- [4] Y.-R. Chen, C.-M. Huang and L.-C. Fu, "Visual tracking of human head and arms with a single camera," *in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems IEEE*, 2010, **pages** 3416–3421.
- [5] A. M. Zanchettin and P. Rocco, "Probabilistic inference of human arm reaching target for effective human-robot collaboration," *in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) IEEE*, 2017, **pages** 6595–6600.
- [6] A. Alshabani, I. Dryden, C. Litton and J. Richardson, "Bayesian analysis of human movement curves," *Journal of the Royal Statistical Society Series C: Applied Statistics*, **journal** 56, **number** 4, **pages** 415–428, 2007.
- [7] A. Kupcsik, D. Hsu and W. S. Lee, "Learning dynamic robot-to-human object handover from human feedback," *Robotics Research: Volume 1*, **pages** 161–176, 2018.
- [8] A. Edsinger and C. C. Kemp, "Human-robot interaction for cooperative manipulation: Handing objects to one another," *in RO-MAN 2007-The 16th IEEE International Symposium on Robot and Human Interactive Communication IEEE*, 2007, **pages** 1167–1172.
- [9] M. Mohandes, B. Moradi, K. Gupta and M. Mehrandezh, "Robot to human object handover using vision and joint torque sensor modalities," *in International Conference on Robot Intelligence Technology and Applications Springer*, 2022, **pages** 109–124.
- [10] Y. Wang, D. Ewert, R. Vossen and S. Jeschke, "A visual servoing system for interactive human-robot object transfer," *Journal of Automation and Control Engineering Vol.*, **journal** 3, **number** 4, 2015.
- [11] F. Ackermann and C. Eden, "Strategic management of stakeholders: Theory and practice," *Long range planning*, **journal** 44, **number** 3, **pages** 179–196, 2011.
- [12] O. Ogbeiw, "Why written objectives need to be really smart," *British Journal of Healthcare Management*, **journal** 23, **number** 7, **pages** 324–336, 2017.
- [13] V. Petit-Magat, H. Lhachemi, C. Stoica, A. Thakker and M. Da Silva, "Enthusiasing engineering students over multi-agent systems control via human-robot interaction," *in 2023 27th International Conference on System Theory, Control and Computing (ICSTCC) IEEE*, 2023, **pages** 24–29.
- [14] K. Ball and others, "Electronic monitoring and surveillance in the workplace," *European Commission Joint Research Centre*, 2021.
- [15] A. Gómez Eguíluz, I. Rañó, S. A. Coleman and T. M. McGinnity, "Reliable robotic handovers through tactile sensing," *Autonomous Robots*, **journal** 43, **pages** 1623–1637, 2019.
- [16] W. P. Chan, C. A. Parker, H. M. Van der Loos and E. A. Croft, "A human-inspired object handover controller," *The International Journal of Robotics Research*, **journal** 32, **number** 8, **pages** 971–983, 2013.
- [17] A. Mazhitov, T. Syrymova, Z. Kappassov and M. Rubagotti, "Human-robot handover with prior-to-pass soft/rigid object classification via tactile glove," *Robotics and Autonomous Systems*, **journal** 159, **page** 104 311, 2023.
- [18] A. Al-Yacoub, Y. Zhao, W. Eaton, Y. M. Goh and N. Lohse, "Improving human robot collaboration through force/torque based learning for object manipulation," *Robotics and Computer-Integrated Manufacturing*, **journal** 69, **page** 102 111, 2021.
- [19] ROBOTIS, *Robotis e-manual version 2024*, <https://emanual.robotis.com/>, Accessed: 2024-06-12, 2024.
- [20] H. Z. Ting, M. Hair, M. Zaman, M. Ibrahim and A. Moubark, "Kinematic analysis for trajectory planning of open-source 4-dof robot arm," *International Journal of Advanced Computer Science and Applications*, **journal** 12, **number** 6, **pages** 769–777, 2021.
- [21] M. A. M. Adzeman, M. H. M. Zaman, M. F. Nasir, M. Faisal and S. M. M. Ibrahim, "Kinematic modeling of a low cost 4 dof robot arm system," *International Journal of Emerging Trends in Engineering Research*, **journal** 8, **number** 10, 2020.
- [22] Z. Zhou, "A mobile robot with a manipulator to alleviate the shortage of health workers in hospitals in covid 19," *Highlights in Science, Engineering and Technology*, **journal** 39, **pages** 872–880, 2023.
- [23] I. A. Rozaini, N. E. N. A. Zakey, M. H. M. Zaman, M. F. Ibrahim, S. M. Mustaza and A. M. Moubark, "Bilateral teleoperation with a shared design of master and slave devices for robotic excavators in agricultural applications," *Int. J. Exp. Res. Rev.*, **journal** 35, **pages** 119–127, 2023.
- [24] J. L. Fozard, M. Vercruyssen, S. L. Reynolds, P. Hancock and R. E. Quilter, "Age differences and changes in reaction time: The baltimore longitudinal study of aging," *Journal of gerontology*, **journal** 49, **number** 4, P179–P189, 1994.

[25] Contactile, *Contactile*, <https://contactile.com/>, Accessed: 2024-06-15.

X. APPENDIX A

$$T_1^0 * T_2^1 = \begin{bmatrix} \cos(\theta_1) & 0 & \sin(\theta_1) & 0 \\ \sin(\theta_1) & 0 & -\cos(\theta_1) & 0 \\ 0 & 1 & 0 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2 \cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2 \sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \quad (10)$$

$$\begin{bmatrix} \cos(\theta_1)\cos(\theta_2) & -\cos(\theta_1)\sin(\theta_2) & \sin(\theta_1) & a_2\cos(\theta_1)\cos(\theta_2) \\ \sin(\theta_1)\cos(\theta_2) & -\sin(\theta_1)\sin(\theta_2) & -\cos(\theta_1) & a_2\sin(\theta_1)\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2\sin(\theta_2) + d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$T_3^2 * T_4^3 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a_3\cos(\theta_3) \\ \sin(\theta_3) & \cos(\theta_3) & 0 & a_3\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & a_4\cos(\theta_4) \\ \sin(\theta_4) & \cos(\theta_4) & 0 & a_4\sin(\theta_4) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \quad (12)$$

$$\begin{bmatrix} \cos(\theta_3 + \theta_4) & -\sin(\theta_3 + \theta_4) & 0 & a_4\cos(\theta_3)\cos(\theta_4) + a_3\cos(\theta_3) - a_4\sin(\theta_3)\sin(\theta_4) \\ \sin(\theta_3 + \theta_4) & \cos(\theta_3 + \theta_4) & 0 & a_4\sin(\theta_3)\cos(\theta_4) + a_4\cos(\theta_3)\sin(\theta_4) + a_3\sin(\theta_3) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

$$T_4^0 = T_0^1 * T_1^2 * T_3^2 * T_4^3 = \left[ \begin{array}{ccc|c} & & & R \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (14)$$

$$R = \begin{bmatrix} \cos(\theta_1)\cos(\theta_2) * \cos(\theta_3 + \theta_4) + & \cos(\theta_1)\cos(\theta_2) * -\sin(\theta_3 + \theta_4) + & \sin(\theta_1) \\ (-\cos(\theta_1)\sin(\theta_2)) * (\sin(\theta_3 + \theta_4)) & -\cos(\theta_1)\sin(\theta_2) * \cos(\theta_3 + \theta_4) & \\ \sin(\theta_1)\cos(\theta_2) * \cos(\theta_3 + \theta_4) + & \sin(\theta_1)\cos(\theta_2) * -\sin(\theta_3 + \theta_4) + & -\cos(\theta_1) \\ -\sin(\theta_1)\sin(\theta_2) * \sin(\theta_3 + \theta_4) & -\sin(\theta_1)\sin(\theta_2) * \cos(\theta_3 + \theta_4) & \\ \sin(\theta_2) * \cos(\theta_3 + \theta_4) + & \sin(\theta_2) * -\sin(\theta_3 + \theta_4) + & 0 \\ \cos(\theta_2) * \sin(\theta_3 + \theta_4) & \cos(\theta_2) * \cos(\theta_3 + \theta_4) & \end{bmatrix} \quad (15)$$

$$P = \begin{bmatrix} (\cos(\theta_1)\cos(\theta_2)) * (a_4\cos(\theta_3)\cos(\theta_4) + a_3\cos(\theta_3) - a_4\sin(\theta_3)\sin(\theta_4)) + \\ (-\cos(\theta_1)\sin(\theta_2)) * (a_4\sin(\theta_3)\cos(\theta_4) + a_4\cos(\theta_3)\sin(\theta_4) + a_3\sin(\theta_3)) + a_2\cos(\theta_1)\cos(\theta_2) \\ \sin(\theta_1)\cos(\theta_2) * (a_4\cos(\theta_3)\cos(\theta_4) + a_3\cos(\theta_3) - a_4\sin(\theta_3)\sin(\theta_4)) + \\ -\sin(\theta_1)\sin(\theta_2) * (a_4\sin(\theta_3)\cos(\theta_4) + a_4\cos(\theta_3)\sin(\theta_4) + a_3\sin(\theta_3)) + a_2\sin(\theta_1)\cos(\theta_2) \\ \sin(\theta_2) * (a_4\cos(\theta_3)\cos(\theta_4) + a_3\cos(\theta_3) - a_4\sin(\theta_3)\sin(\theta_4)) \\ + \cos(\theta_2) * (a_4\sin(\theta_3)\cos(\theta_4) + a_4\cos(\theta_3)\sin(\theta_4) + a_3\sin(\theta_3) + a_2\sin(\theta_2) + d_i \end{bmatrix} \quad (16)$$

$$\tau = ({}^o J)^T * F, \text{ where } J \text{ is the Jacobian matrix and } F \text{ is the force vector} \quad (17)$$

$$J(\theta_1, \theta_2, \theta_3, \theta_4) = \begin{bmatrix} \frac{\partial y_1}{\partial \theta_1} & \frac{\partial y_1}{\partial \theta_2} & \frac{\partial y_1}{\partial \theta_3} & \frac{\partial y_1}{\partial \theta_4} \\ \frac{\partial y_2}{\partial \theta_1} & \frac{\partial y_2}{\partial \theta_2} & \frac{\partial y_2}{\partial \theta_3} & \frac{\partial y_2}{\partial \theta_4} \\ \frac{\partial y_3}{\partial \theta_1} & \frac{\partial y_3}{\partial \theta_2} & \frac{\partial y_3}{\partial \theta_3} & \frac{\partial y_3}{\partial \theta_4} \end{bmatrix} \quad (18)$$

$$J_{11} = \frac{\partial y_1}{\partial \theta_1} = (-\sin(\theta_1) * \cos(\theta_2)) * (a_4\cos(\theta_3)\cos(\theta_4) + a_3\cos(\theta_3) - a_4\sin(\theta_3)\sin(\theta_4)) + \\ (\sin(\theta_1)\sin(\theta_2)) * (a_4\sin(\theta_3)\cos(\theta_4) + a_4\cos(\theta_3)\sin(\theta_4) + a_3\sin(\theta_3)) + -a_2\sin(\theta_1)\cos(\theta_2) \quad (19)$$



$$J_{12} = \frac{\partial y_1}{\partial \theta_2} = (\cos(\theta_1) * -\sin(\theta_2)) * (a_4 \cos(\theta_3) \cos(\theta_4) + a_3 \cos(\theta_3) - a_4 \sin(\theta_3) \sin(\theta_4)) + (-\cos(\theta_1) \cos(\theta_2)) * (a_4 \sin(\theta_3) \cos(\theta_4) + a_4 \cos(\theta_3) \sin(\theta_4) + a_3 \sin(\theta_3)) + a_2 \cos(\theta_1) * -\sin(\theta_2) \quad (20)$$

$$J_{13} = \frac{\partial y_1}{\partial \theta_3} = (\cos(\theta_1) * \cos(\theta_2)) * (a_4 * -\sin(\theta_3) \cos(\theta_4) + a_3 * -\sin(\theta_3) - a_4 \cos(\theta_3) \sin(\theta_4)) + (-\cos(\theta_1) \sin(\theta_2)) * (a_4 \cos(\theta_3) \cos(\theta_4) + a_4 * -\sin(\theta_3) \sin(\theta_4) + a_3 \cos(\theta_3)) + a_2 \cos(\theta_1) * \cos(\theta_2) \quad (21)$$

$$J_{14} = \frac{\partial y_1}{\partial \theta_4} = (\cos(\theta_1) * \cos(\theta_2)) * (a_4 * \cos(\theta_3) * -\sin(\theta_4) + a_3 * \cos(\theta_3) - a_4 \sin(\theta_3) \cos(\theta_4)) + (-\cos(\theta_1) \sin(\theta_2)) * (a_4 \sin(\theta_3) * -\sin(\theta_4) + a_4 * \cos(\theta_3) \cos(\theta_4) + a_3 \sin(\theta_3)) + a_2 \cos(\theta_1) * \cos(\theta_2) \quad (22)$$

$$J_{21} = \frac{\partial y_2}{\partial \theta_1} = (\cos(\theta_1) * \cos(\theta_2)) * (a_4 \cos(\theta_3) * \cos(\theta_4) + a_3 \cos(\theta_3) - a_4 \sin(\theta_3) * \sin(\theta_4)) + (\cos(\theta_1) \sin(\theta_2)) * (a_4 \sin(\theta_3) \cos(\theta_4) + a_4 \cos(\theta_3) \sin(\theta_4) + a_3 \sin(\theta_3)) + a_2 \cos(\theta_1) \cos(\theta_2) \quad (23)$$

$$J_{22} = \frac{\partial y_2}{\partial \theta_2} = (\sin(\theta_1) * -\sin(\theta_2)) * (a_4 \cos(\theta_3) * \cos(\theta_4) + a_3 \cos(\theta_3) - a_4 \sin(\theta_3) * \sin(\theta_4)) + (-\sin(\theta_1) \cos(\theta_2)) * (a_4 \sin(\theta_3) \cos(\theta_4) + a_4 \cos(\theta_3) \sin(\theta_4) + a_3 \sin(\theta_3)) + a_2 \sin(\theta_1) * -\sin(\theta_2) \quad (24)$$

$$J_{23} = \frac{\partial y_2}{\partial \theta_3} = (\sin(\theta_1) * \cos(\theta_2)) * (a_4 * -\sin(\theta_3) * \cos(\theta_4) + a_3 * -\sin(\theta_3) - a_4 \cos(\theta_3) * \sin(\theta_4)) + (-\sin(\theta_1) \sin(\theta_2)) * (a_4 \cos(\theta_3) \cos(\theta_4) + a_4 * -\sin(\theta_3) \sin(\theta_4) + a_3 \cos(\theta_3)) + a_2 \sin(\theta_1) * \cos(\theta_2) \quad (25)$$

$$J_{24} = \frac{\partial y_2}{\partial \theta_4} = (\sin(\theta_1) * \cos(\theta_2)) * (a_4 * \cos(\theta_3) * -\sin(\theta_4) + a_3 * \cos(\theta_3) - a_4 \sin(\theta_3) * \cos(\theta_4)) + (-\sin(\theta_1) \sin(\theta_2)) * (a_4 \sin(\theta_3) * -\sin(\theta_4) + a_4 * \cos(\theta_3) \cos(\theta_4) + a_3 \sin(\theta_3)) + a_2 \sin(\theta_1) * \cos(\theta_2) \quad (26)$$

$$J_{31} = \frac{\partial y_3}{\partial \theta_1} = \sin(\theta_2) * (a_4 \cos(\theta_3) \cos(\theta_4) + a_3 \cos(\theta_3) - a_4 \sin(\theta_3) \sin(\theta_4)) + \cos(\theta_2) * (a_4 \sin(\theta_3) \cos(\theta_4) + a_4 \cos(\theta_3) \sin(\theta_4) + a_3 \sin(\theta_3) + a_2 \sin(\theta_2) + d_i) \quad (27)$$

$$J_{32} = \frac{\partial y_3}{\partial \theta_2} = \cos(\theta_2) * (a_4 \cos(\theta_3) \cos(\theta_4) + a_3 \cos(\theta_3) - a_4 \sin(\theta_3) \sin(\theta_4)) + \sin(\theta_2) * (a_4 \sin(\theta_3) \cos(\theta_4) + a_4 \cos(\theta_3) \sin(\theta_4) + a_3 \sin(\theta_3) + a_2 \cos(\theta_2) + d_i) \quad (28)$$

$$J_{33} = \frac{\partial y_3}{\partial \theta_3} = \sin(\theta_2) * (a_4 * -\sin(\theta_3) \cos(\theta_4) + a_3 * -\sin(\theta_3) - a_4 \cos(\theta_3) \sin(\theta_4)) + \cos(\theta_2) * (a_4 \cos(\theta_3) \cos(\theta_4) + a_4 * -\sin(\theta_3) \sin(\theta_4) + a_3 \cos(\theta_3) + a_2 \sin(\theta_2) + d_i) \quad (29)$$

$$J_{34} = \frac{\partial y_3}{\partial \theta_4} = \sin(\theta_2) * (a_4 * \cos(\theta_3) * -\sin(\theta_4) + a_3 * \cos(\theta_3) - a_4 \sin(\theta_3) \cos(\theta_4)) + \cos(\theta_2) * (a_4 \sin(\theta_3) * -\sin(\theta_4) + a_4 * \cos(\theta_3) \cos(\theta_4) + a_3 \sin(\theta_3) + a_2 \sin(\theta_2) + d_i) \quad (30)$$

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} & J_{14} \\ J_{21} & J_{22} & J_{23} & J_{24} \\ J_{31} & J_{32} & J_{33} & J_{34} \end{bmatrix}, J^T = \begin{bmatrix} J_{11} & J_{21} & J_{31} \\ J_{12} & J_{22} & J_{32} \\ J_{13} & J_{23} & J_{33} \\ J_{14} & J_{24} & J_{34} \end{bmatrix} \quad (31)$$

$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} = \begin{bmatrix} J_{11} * F_x + J_{21} * F_y + J_{31} * F_z \\ J_{12} * F_x + J_{22} * F_y + J_{32} * F_z \\ J_{13} * F_x + J_{23} * F_y + J_{33} * F_z \\ J_{14} * F_x + J_{24} * F_y + J_{34} * F_z \end{bmatrix} \quad (32)$$

$$\tau_1 = J_{11} * F_x + J_{21} * F_y + J_{31} * F_z \quad (33)$$

$$\tau_2 = J_{12} * F_x + J_{22} * F_y + J_{32} * F_z \quad (34)$$

$$\tau_3 = J_{13} * F_x + J_{23} * F_y + J_{33} * F_z \quad (35)$$

$$\tau_4 = J_{14} * F_x + J_{24} * F_y + J_{34} * F_z \quad (36)$$

## XI. APPENDIX B

This appendix consists of tables and figures that show the results of the experiments. In the tables, the error and the time can be seen for each experiment. Each participant conducted the experiment 6 times for each object. In the tables, a 0 or a 1 can be seen for the error. The 0 means the handover was unsuccessful and 1 means a successful handover. Next to the error the time in seconds can be seen for each handover.

TABLE V  
RESULTS FOR A PIECE OF WOOD

Participant 1		Participant 2		Participant 3		Participant 4		Participant 5		Participant 6	
Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)
0	2.79	0	0.97	0	0.23	1	1.26	1	0.77	0	0.46
1	1.72	1	1.43	1	1.52	1	1.63	1	1.87	1	1.42
1	0.58	0	2.11	0	1.98	0	1.75	0	1.64	0	1.01
0	0.11	1	1.13	1	1.04	0	2.21	1	1.52	0	0.42
1	1.03	0	2.97	1	1.67	1	1.94	1	1.62	1	1.64
1	1.08	0	1.66	1	1.09	1	1.38	1	1.34	1	1.72

TABLE VI  
RESULTS FOR TAPE MEASURE

Participant 1		Participant 2		Participant 3		Participant 4		Participant 5		Participant 6	
Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)
1	1.10	0	2.56	1	1.67	1	1.58	0	2.73	1	1.97
0	0.22	1	1.53	0	0.31	1	1.83	1	1.47	1	1.43
0	3.46	0	1.67	0	2.78	1	1.46	1	1.97	1	1.64
0	3.31	1	1.12	1	1.64	0	1.24	0	1.34	0	0.88
1	1.52	0	0.48	1	1.37	1	1.59	1	1.67	1	1.84
1	0.89	1	1.79	1	1.49	0	1.01	0	0.49	0	3.12

TABLE VII  
RESULTS FOR SCREWDRIVER

Participant 1		Participant 2		Participant 3		Participant 4		Participant 5		Participant 6	
Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)	Error	Time (s)
0	3.01	0	0.42	1	1.37	1	1.97	1	1.94	1	1.67
0	2.24	1	1.45	1	1.14	0	1.23	1	1.67	0	1.13
1	1.18	0	1.24	1	1.57	0	0.46	0	1.06	0	1.79
1	1.41	0	2.77	0	0.89	0	2.64	1	1.42	1	1.65
1	1.08	1	1.37	1	1.63	1	1.84	1	1.39	1	1.82
0	1.91	1	1.97	0	1.14	1	1.37	0	1.87	0	0.39

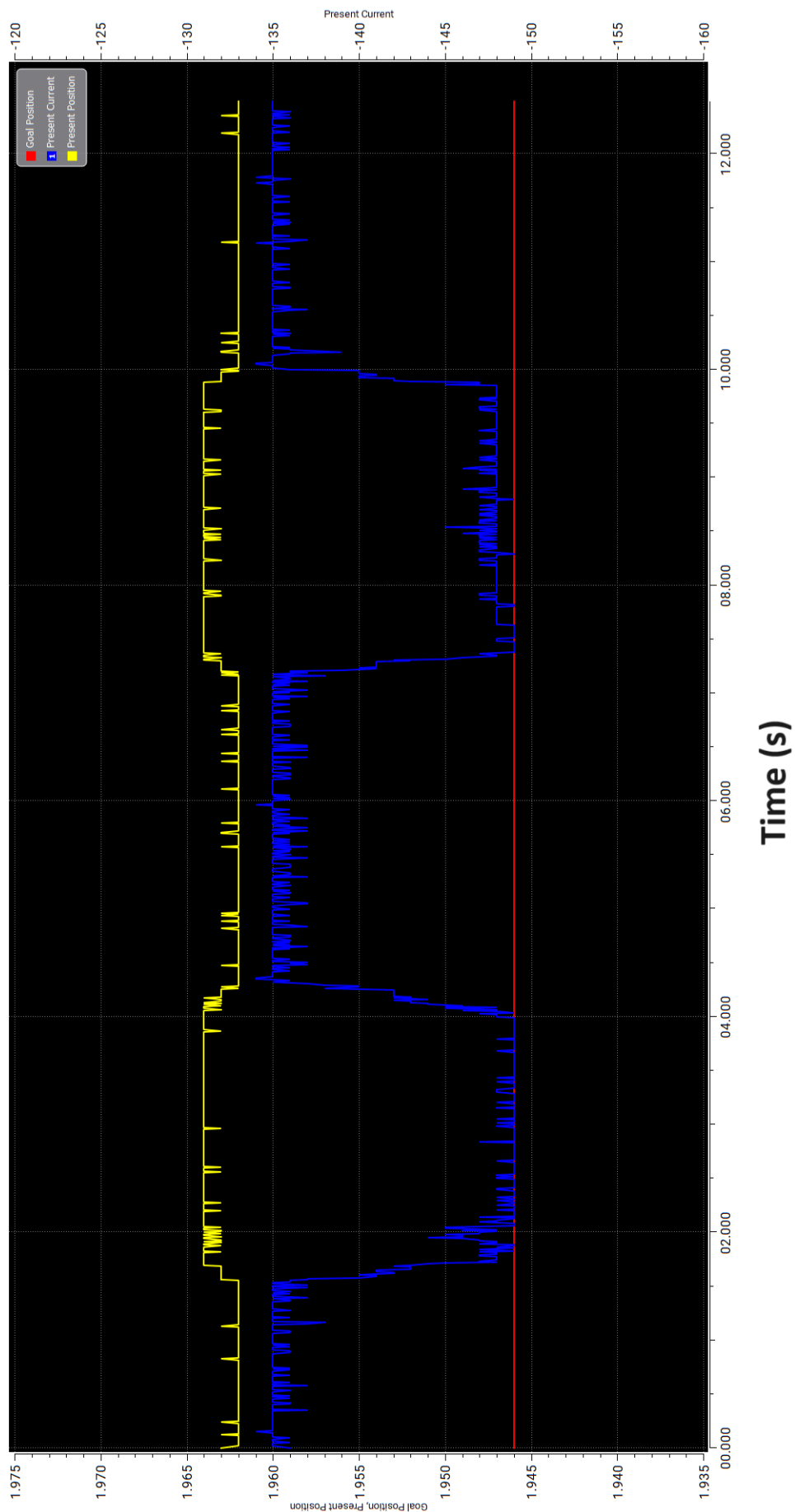


Fig. 8. Current measurements

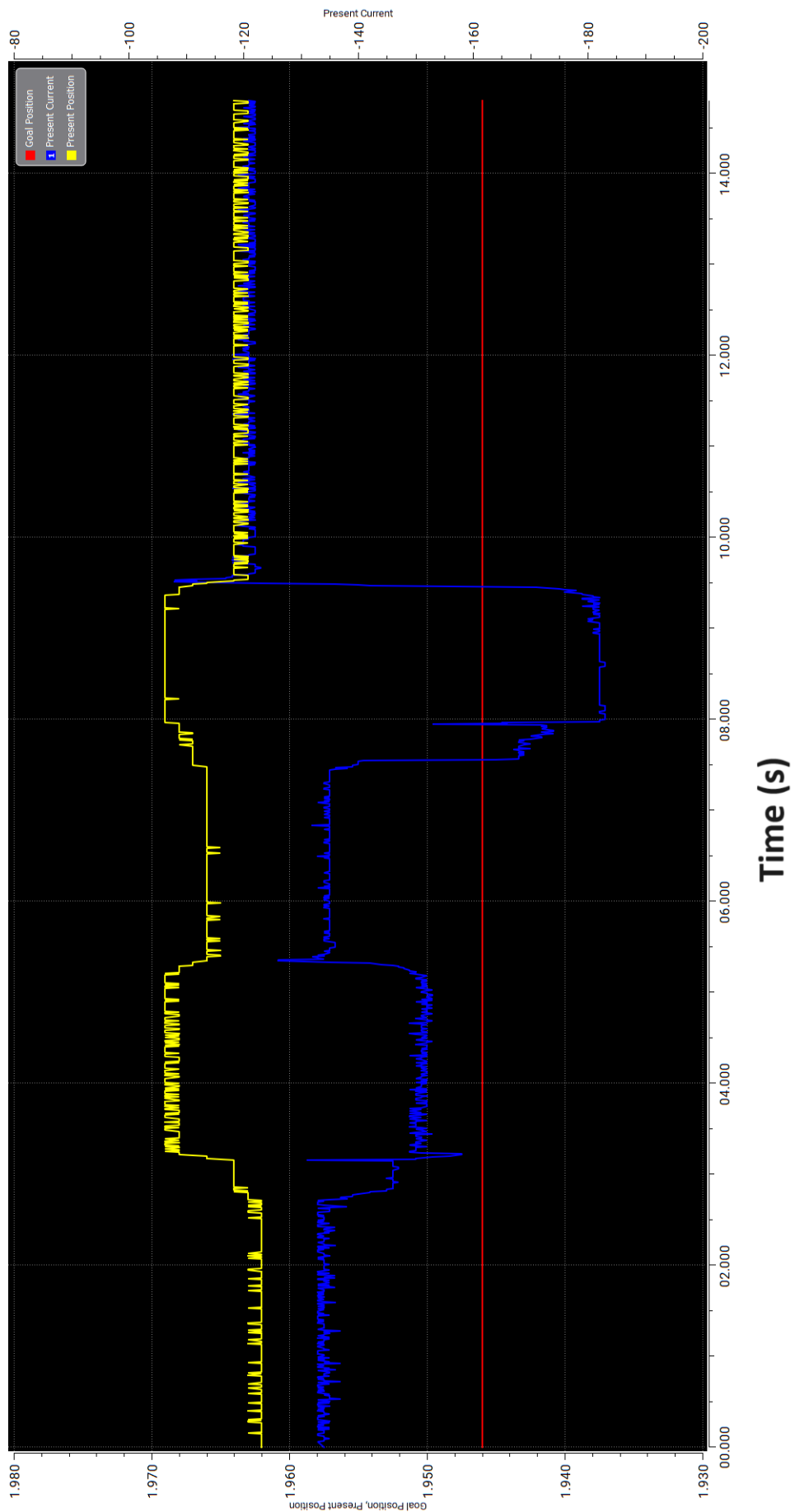


Fig. 9. Current measurements

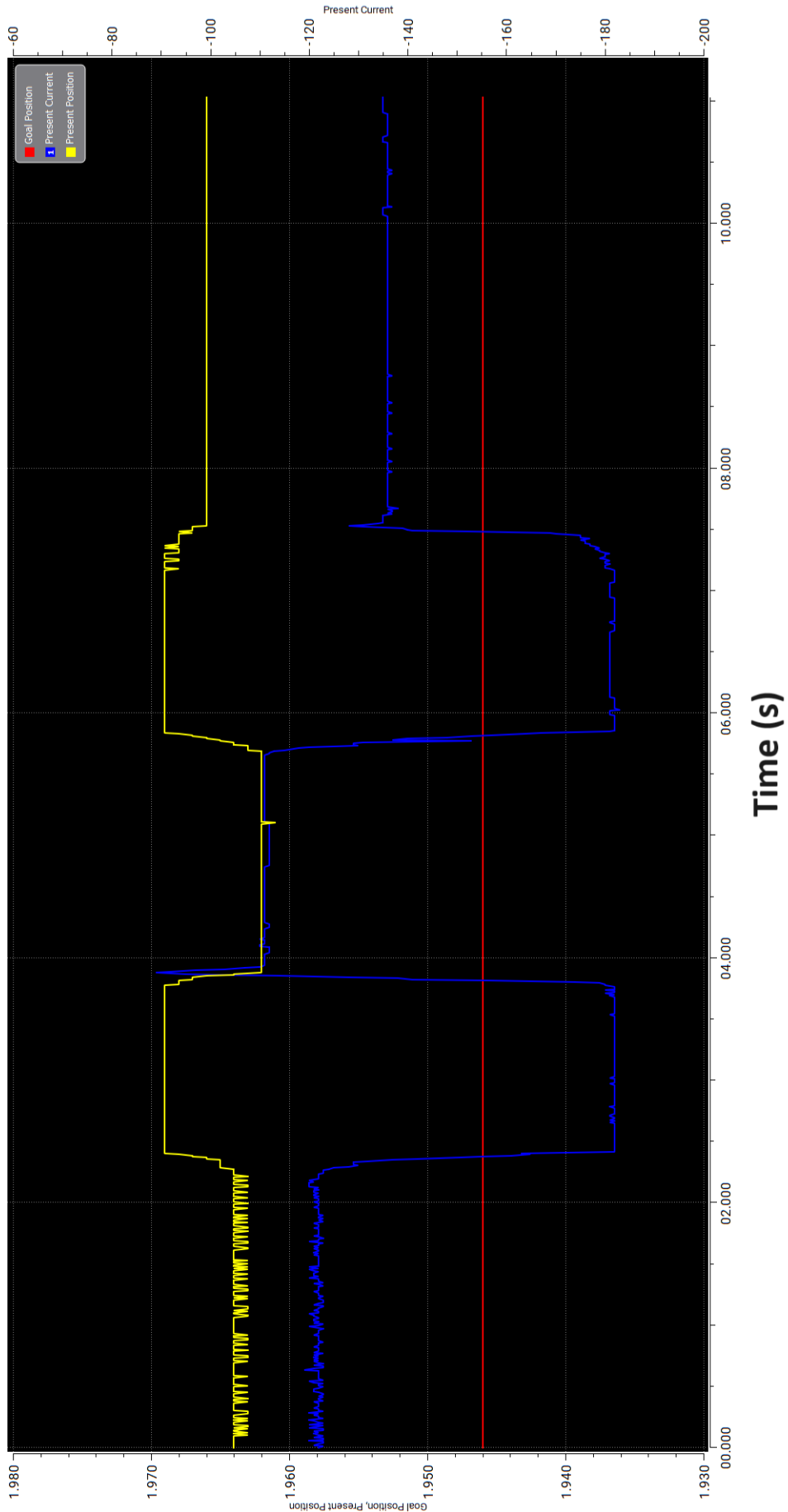


Fig. 10. Current measurements



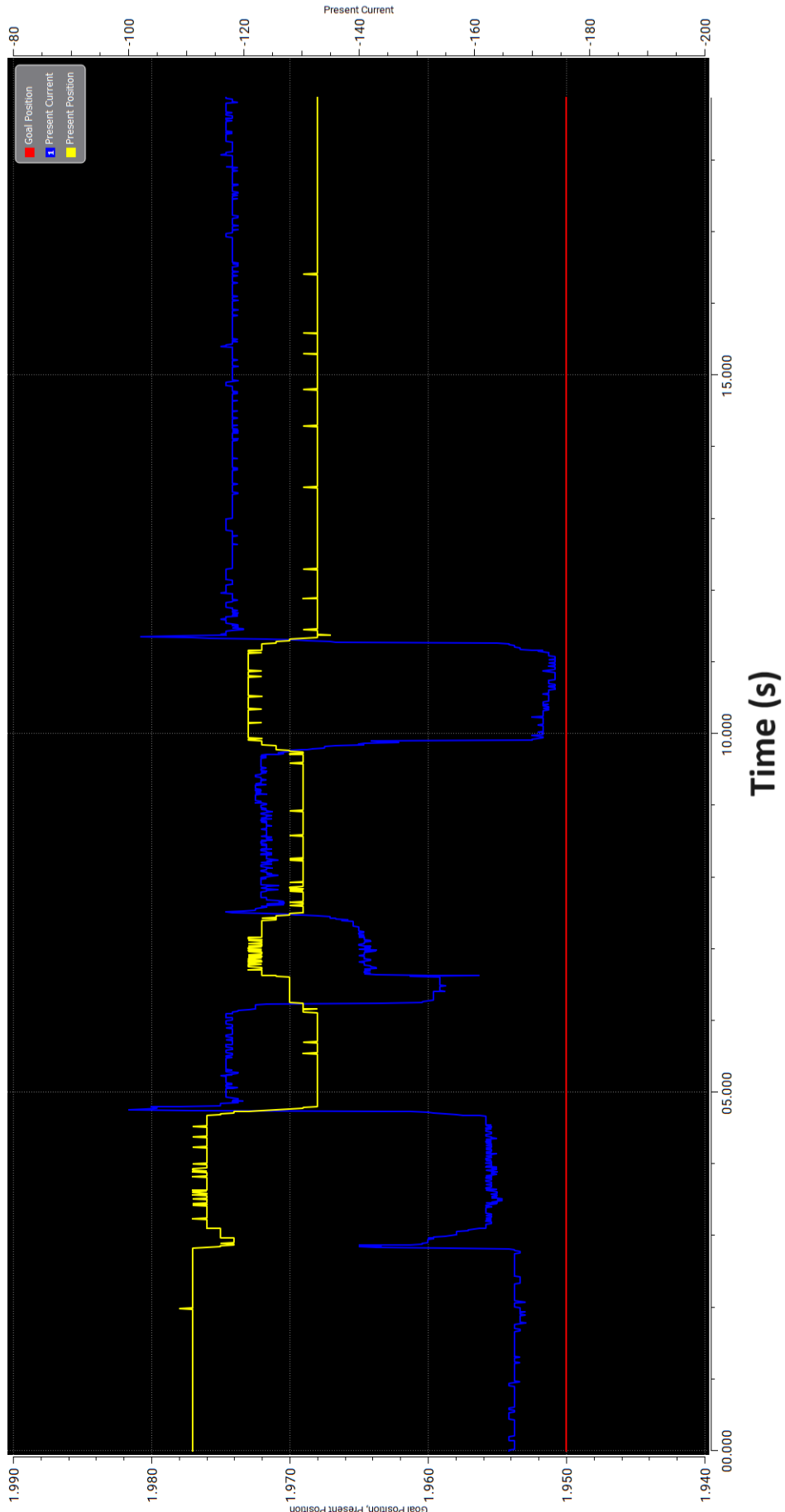


Fig. 11. Current measurements

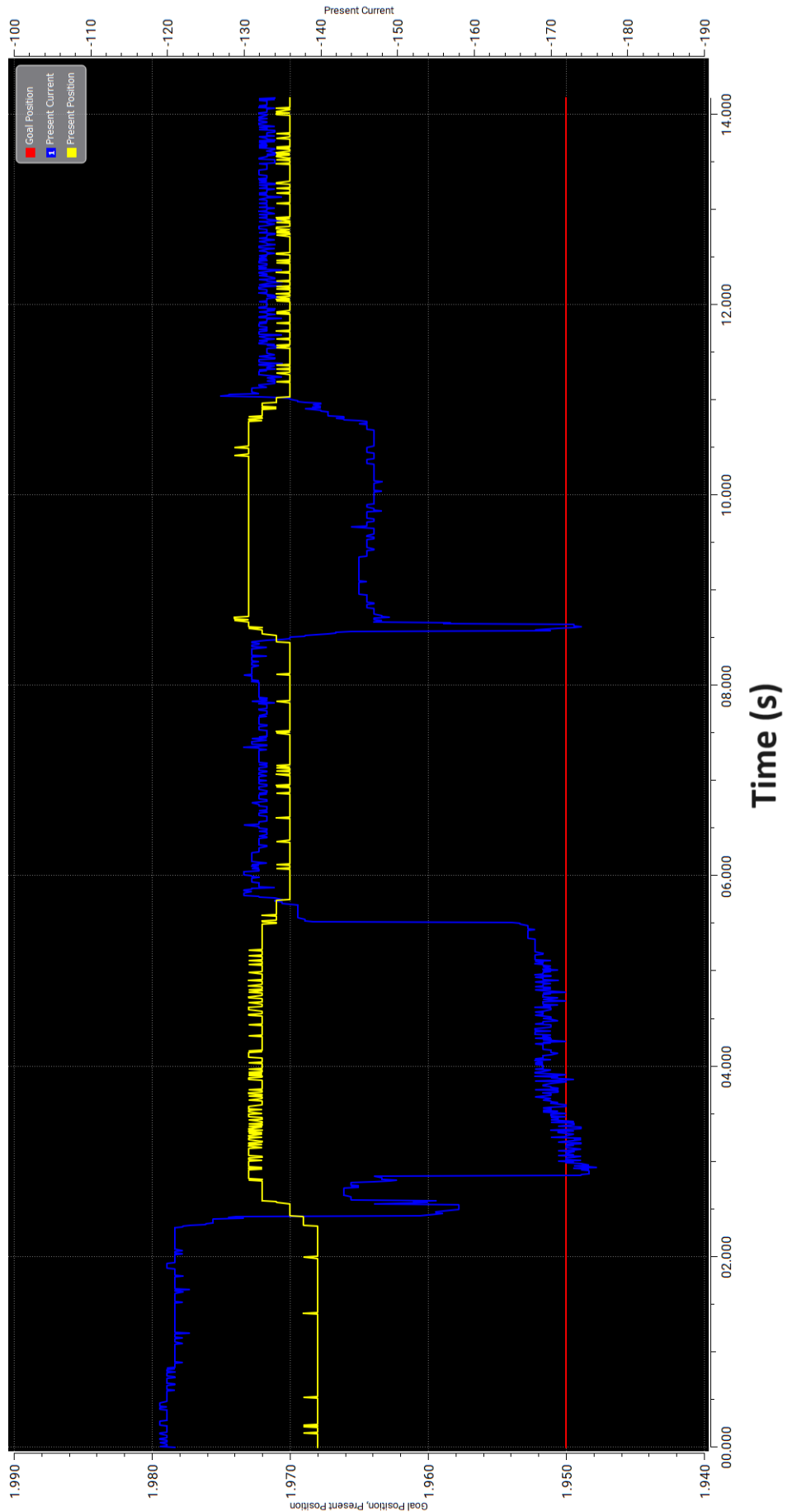


Fig. 12. Current measurements

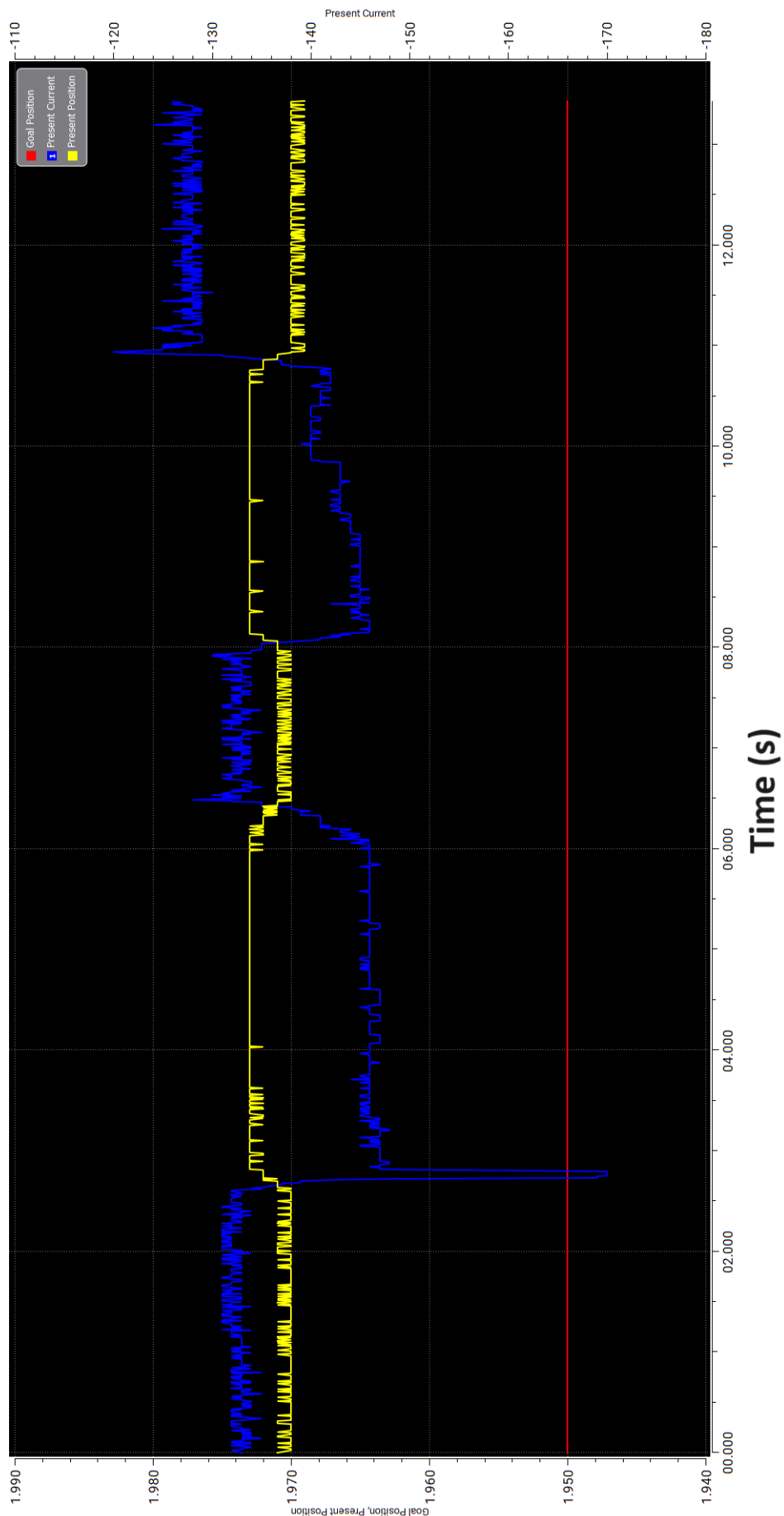


Fig. 13. Current measurements

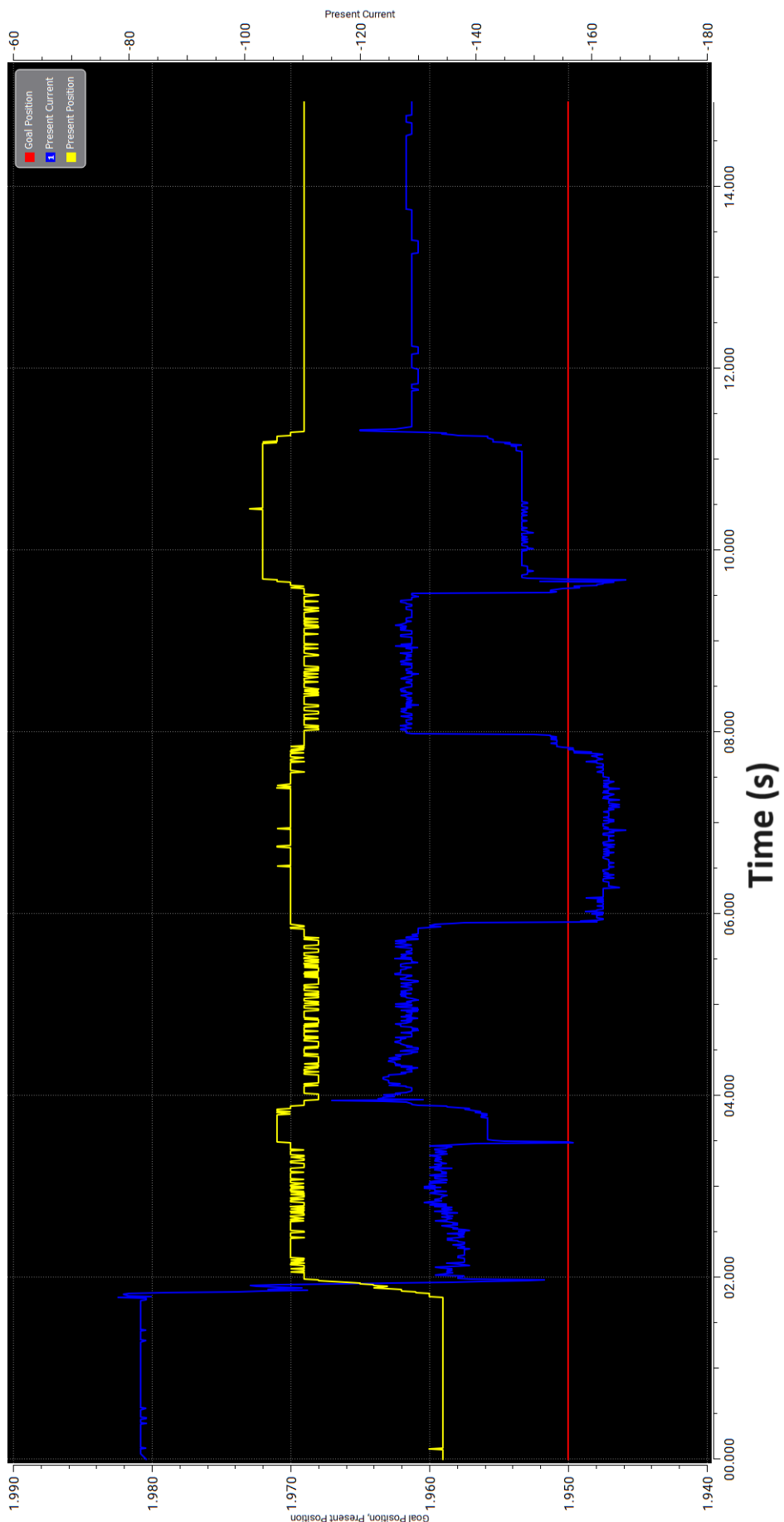


Fig. 14. Current measurements

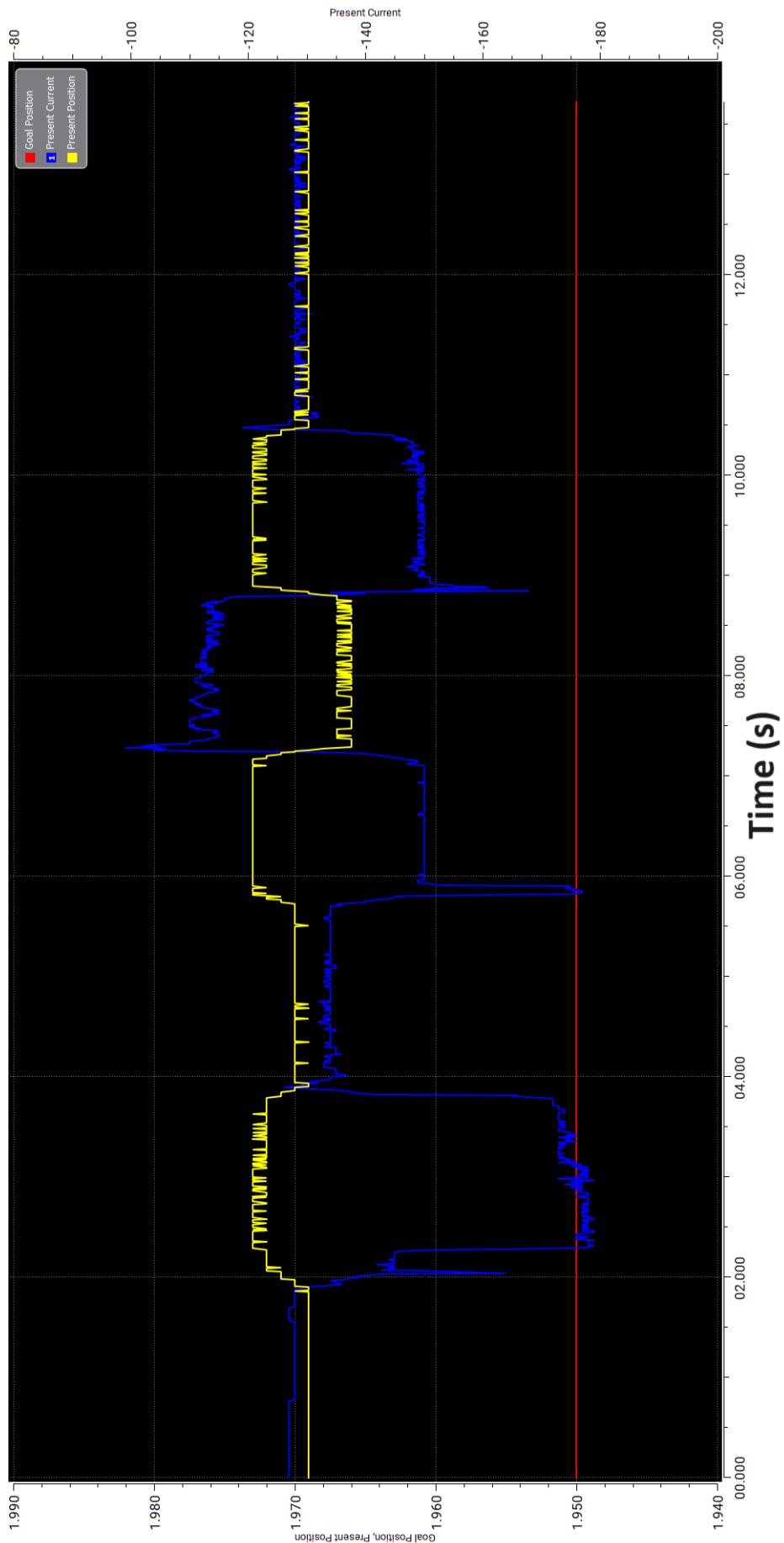


Fig. 15. Current measurements

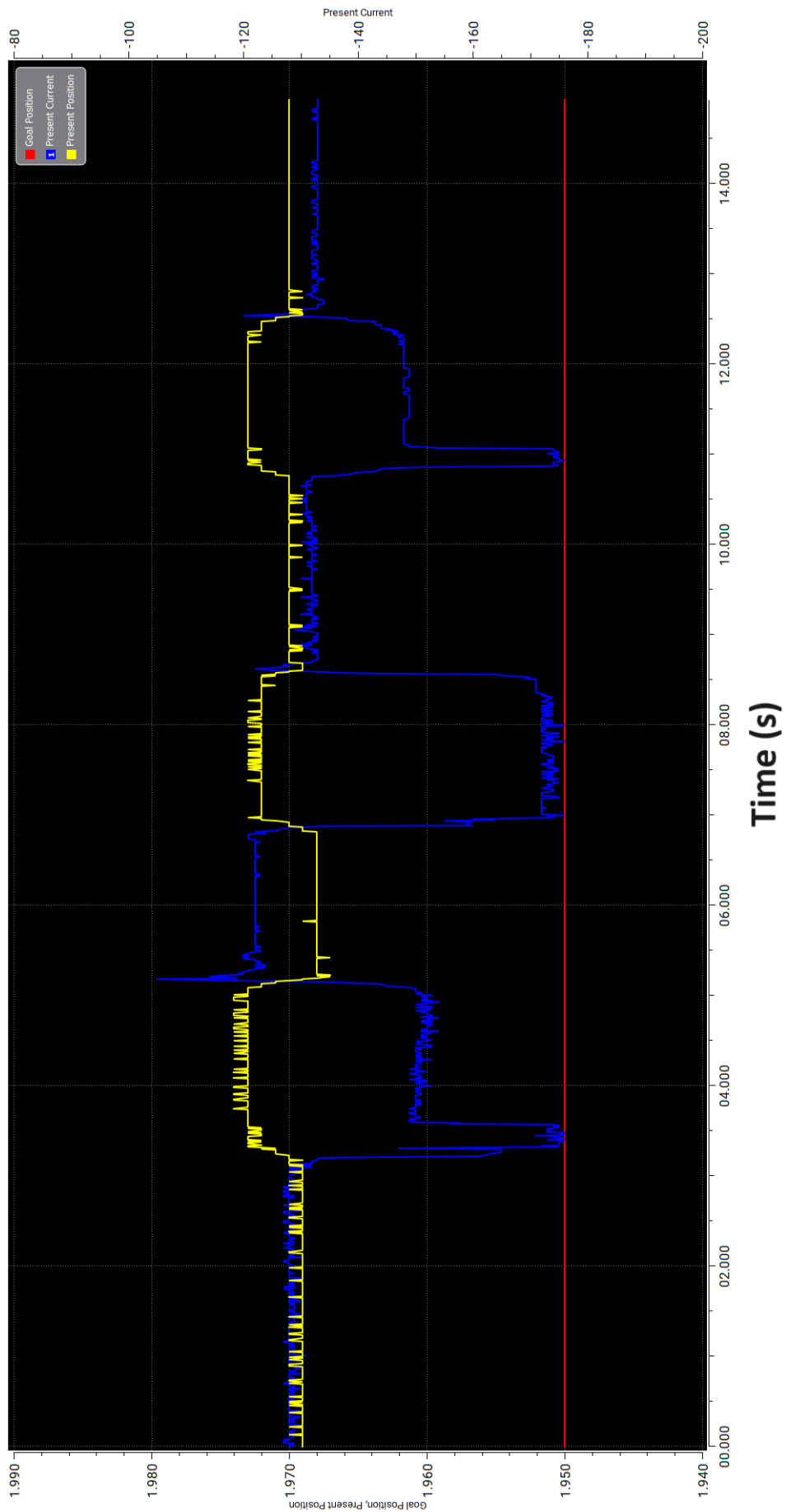


Fig. 16. Current measurements

## XII. APPENDIX C

This appendix contains the Arduino code for the torque controller used in all the experiments. This code is developed in Arduino IDE version 2.3.2.

```
1
2 #include <DynamixelSDK.h>
3
4
5 // Control table address
6 #define ADDR_PRO_TORQUE_ENABLE          64           // Control table address
   is different in Dynamixel model
7 #define ADDR_PRO_GOAL_POSITION         116
8 #define ADDR_PRO_PRESENT_POSITION     132
9 #define ADDR_PRO_present_current      126 // present current id
10 #define ADDR_PRO_goal_current         102 // goal current id
11
12 // Protocol version
13 #define PROTOCOL_VERSION               2.0          // See which protocol
   version is used in the Dynamixel
14
15 // Default setting
16 #define BAUDRATE                       1000000
17 #define DEVICENAME                     "OpenCR_DXL_Port" // This definition only
   has a symbolic meaning and does not affect to any functionality
18
19 #define TORQUE_ENABLE                   1           // Value for enabling
   the torque
```

```
20 #define TORQUE_DISABLE 0 // Value for disabling
    the torque
21 #define DXL_ID_012 12 // Dynamixel ID: 12
22 #define DXL_ID_013 13 // Dynamixel ID: 13
23 #define DXL_ID_014 14 // Dynamixel ID: 13
24 #define DXL_ID_015 15 // Dynamixel ID: 13
25 #define DXL_012_MINIMUM_POSITION_VALUE 2000 // 013 MINIMUM VALUE
26 #define DXL_012_MAXIMUM_POSITION_VALUE 2000 // 013 MAXIMUM VALUE
27 #define DXL_013_MINIMUM_POSITION_VALUE 2000 // 013 MINIMUM VALUE
28 #define DXL_013_MAXIMUM_POSITION_VALUE 2000 // 013 MAXIMUM VALUE
29 #define DXL_014_MINIMUM_POSITION_VALUE 2000 // 014 MINIMUM VALUE
30 #define DXL_014_MAXIMUM_POSITION_VALUE 2000 // 014 MAXIMUM VALUE
31 #define DXL_015_MINIMUM_POSITION_VALUE 2000 // 015 MINIMUM VALUE
    gripper 1600 is bijna helemaal open
32 #define DXL_015_MAXIMUM_POSITION_VALUE 2000 // 015 MAXIMUM VALUE
    gripper 2300 is helemaal dicht
33
34 #define DXL_MOVING_STATUS_THRESHOLD 20 // Dynamixel moving
    status threshold
35 #define DXL_CURRENT_STATUS_THRESHOLD 0.8 // Dynamixel current
    status threshold voor de while loop
36
37
38 #define ESC_ASCII_VALUE 0x1b
39
40 #define CMD_SERIAL Serial
41
42 int getch()
```



```
43 {
44   while(1)
45   {
46     if( CMD_SERIAL.available() > 0 )
47     {
48       break;
49     }
50   }
51
52   return CMD_SERIAL.read();
53 }
54
55 int kbhit(void)
56 {
57   return CMD_SERIAL.available();
58 }
59
60 void setup()
61 {
62   Serial.begin(115200);
63   while(!Serial);
64
65   Serial.println("Start..");
66 }
67 }
68
69 void loop()
70 {
```

```
71 // Initialize PortHandler instance
72 // Set the port path
73 // Get methods and members of PortHandlerLinux or PortHandlerWindows
74 dynamixel::PortHandler *portHandler = dynamixel::PortHandler::getPortHandler(
    DEVICENAME);
75
76 // Initialize PacketHandler instance
77 // Set the protocol version
78 // Get methods and members of Protocol1PacketHandler or Protocol2PacketHandler
79 dynamixel::PacketHandler *packetHandler = dynamixel::PacketHandler::
    getPacketHandler(PROTOCOL_VERSION);
80
81 int index = 0;
82 int dxl_comm_result = COMM_TX_FAIL;           // Communication result
83 int DXL_012_GOAL_POSITION[2] = {DXL_012_MINIMUM_POSITION_VALUE,
    DXL_012_MAXIMUM_POSITION_VALUE};
84 int DXL_013_GOAL_POSITION[2] = {DXL_013_MINIMUM_POSITION_VALUE,
    DXL_013_MAXIMUM_POSITION_VALUE};           // Goal position
85 int DXL_014_GOAL_POSITION[2] = {DXL_014_MINIMUM_POSITION_VALUE,
    DXL_014_MAXIMUM_POSITION_VALUE};           // Goal position
86 int DXL_015_GOAL_POSITION[2] = {DXL_015_MINIMUM_POSITION_VALUE,
    DXL_015_MAXIMUM_POSITION_VALUE};           // Goal position
87
88 uint8_t dxl_error = 0;                       // Dynamixel error
89 int32_t dxl_present_position_12 = 0;         // Present position
90 int32_t dxl_present_position_13 = 0;         // Present position
91 int32_t dxl_present_position_14 = 0;         // Present position
92 int32_t dxl_present_position_15 = 0;         // Present position
```

```
93
94 int16_t dxl_present_current_12 = 0;           // Present position
95 int16_t dxl_present_current_13 = 0;           // Present position
96 int16_t dxl_present_current_14 = 0;           // Present position
97 int16_t dxl_present_current_15 = 0;           // Present position
98
99
100 // Open port
101 if (portHandler->openPort ())
102 {
103     Serial.print ("Succeeded_to_open_the_port!\n");
104 }
105 else
106 {
107     Serial.print ("Failed_to_open_the_port!\n");
108     return;
109 }
110
111 // Set port baudrate
112 if (portHandler->setBaudRate (BAUDRATE))
113 {
114     Serial.print ("Succeeded_to_change_the_baudrate!\n");
115 }
116 else
117 {
118     Serial.print ("Failed_to_change_the_baudrate!\n");
119     return;
120 }
```

```
121 |
122 | // Enable Dynamixel Torque 012
123 | dxl_comm_result = packetHandler->write1ByteTxRx(portHandler, DXL_ID_012,
124 | ADDR_PRO_TORQUE_ENABLE, TORQUE_ENABLE, &dxl_error);
125 | if (dxl_comm_result != COMM_SUCCESS)
126 | {
127 |     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
128 | }
129 | else if (dxl_error != 0)
130 | {
131 |     Serial.print(packetHandler->getRxPacketError(dxl_error));
132 | }
133 | else
134 | {
135 |     Serial.print("Dynamixel_has_been_successfully_connected_\n");
136 | }
137 | // Enable Dynamixel Torque 013
138 | dxl_comm_result = packetHandler->write1ByteTxRx(portHandler, DXL_ID_013,
139 | ADDR_PRO_TORQUE_ENABLE, TORQUE_ENABLE, &dxl_error);
140 | if (dxl_comm_result != COMM_SUCCESS)
141 | {
142 |     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
143 | }
144 | else if (dxl_error != 0)
145 | {
146 |     Serial.print(packetHandler->getRxPacketError(dxl_error));
147 | }
```

```
147 else
148 {
149     Serial.print("Dynamixel_has_been_successfully_connected_\n");
150 }
151 // Enable Dynamixel Torque 014
152 dxl_comm_result = packetHandler->write1ByteTxRx(portHandler, DXL_ID_014,
153     ADDR_PRO_TORQUE_ENABLE, TORQUE_ENABLE, &dxl_error);
154 if (dxl_comm_result != COMM_SUCCESS)
155 {
156     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
157 }
158 else if (dxl_error != 0)
159 {
160     Serial.print(packetHandler->getRxPacketError(dxl_error));
161 }
162 else
163 {
164     Serial.print("Dynamixel_has_been_successfully_connected_\n");
165 }
166 // Enable Dynamixel Torque 015
167 dxl_comm_result = packetHandler->write1ByteTxRx(portHandler, DXL_ID_015,
168     ADDR_PRO_TORQUE_ENABLE, TORQUE_ENABLE, &dxl_error);
169 if (dxl_comm_result != COMM_SUCCESS)
170 {
171     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
172 }
173 else if (dxl_error != 0)
```

```
173 | {
174 |     Serial.print(packetHandler->getRxPacketError(dx1_error));
175 | }
176 | else
177 | {
178 |     Serial.print("Dynamixel_has_been_successfully_connected_\n");
179 | }
180 |
181 |     //////////////////////////////////////
182 |     // TORQUE ENABLE END
183 |     //////////////////////////////////////
184 |
185 |
186 |
187 | while(1)
188 | {
189 |     /*
190 |     Serial.print("Press any key to continue! (or press q to quit!)\n");
191 |     if (getch() == 'q')
192 |         break;
193 |     */
194 |     // Write goal position 012
195 |     dx1_comm_result = packetHandler->write4ByteTxRx(portHandler, DXL_ID_012,
196 |         ADDR_PRO_GOAL_POSITION, DXL_012_GOAL_POSITION[index], &dx1_error);
197 |     if (dx1_comm_result != COMM_SUCCESS)
198 |     {
199 |         Serial.print(packetHandler->getTxRxResult(dx1_comm_result));
```

```
200     else if (dxl_error != 0)
201     {
202         Serial.print(packetHandler->getRxPacketError(dxl_error));
203     }
204
205     // Write goal position 013
206     dxl_comm_result = packetHandler->write4ByteTxRx(portHandler, DXL_ID_013,
        ADDR_PRO_GOAL_POSITION, DXL_013_GOAL_POSITION[index], &dxl_error);
207     if (dxl_comm_result != COMM_SUCCESS)
208     {
209         Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
210     }
211     else if (dxl_error != 0)
212     {
213         Serial.print(packetHandler->getRxPacketError(dxl_error));
214     }
215     // Write goal position 014
216     dxl_comm_result = packetHandler->write4ByteTxRx(portHandler, DXL_ID_014,
        ADDR_PRO_GOAL_POSITION, DXL_014_GOAL_POSITION[index], &dxl_error);
217     if (dxl_comm_result != COMM_SUCCESS)
218     {
219         Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
220     }
221     else if (dxl_error != 0)
222     {
223         Serial.print(packetHandler->getRxPacketError(dxl_error));
224     }
225
```

```
226 // Write goal position 015
227 dxl_comm_result = packetHandler->write4ByteTxRx(portHandler, DXL_ID_015,
        ADDR_PRO_GOAL_POSITION, DXL_015_GOAL_POSITION[index], &dxl_error);
228 if (dxl_comm_result != COMM_SUCCESS)
229 {
230     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
231 }
232 else if (dxl_error != 0)
233 {
234     Serial.print(packetHandler->getRxPacketError(dxl_error));
235 }
236
237 do
238 {
239     // Read present position
240     dxl_comm_result = packetHandler->read4ByteTxRx(portHandler, DXL_ID_012,
        ADDR_PRO_PRESENT_POSITION, (uint32_t*)&dxl_present_position_12, &dxl_error
        );
241     if (dxl_comm_result != COMM_SUCCESS)
242     {
243         Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
244     }
245     else if (dxl_error != 0)
246     {
247         Serial.print(packetHandler->getRxPacketError(dxl_error));
248     }
249
250     //Serial.print("[ID:"); Serial.print(DXL_ID_012);
```



```
251 //Serial.print("] GoalPos:"); Serial.print(DXL_012_GOAL_POSITION[index]);
252 //Serial.print(" PresPos:"); Serial.print(dxl_present_position_12);
253 Serial.println("_");
254 Serial.println("\n");
255
256 // Read present current 12
257 dxl_comm_result = packetHandler->read2ByteTxRx(portHandler, DXL_ID_012,
        ADDR_PRO_present_current, (uint16_t*)&dxl_present_current_12, &dxl_error);
258 if (dxl_comm_result != COMM_SUCCESS)
259 {
260     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
261 }
262 else if (dxl_error != 0)
263 {
264     Serial.print(packetHandler->getRxPacketError(dxl_error));
265 }
266
267 Serial.print("_present_Current_12:"); Serial.print(dxl_present_current_12);
268
269 // Read present current 13
270 dxl_comm_result = packetHandler->read2ByteTxRx(portHandler, DXL_ID_013,
        ADDR_PRO_present_current, (uint16_t*)&dxl_present_current_13, &dxl_error);
271 if (dxl_comm_result != COMM_SUCCESS)
272 {
273     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
274 }
275 else if (dxl_error != 0)
276 {
```

```
277     Serial.print(packetHandler->getRxPacketError(dx1_error));
278 }
279
280 Serial.print("_present_Current_13:"); Serial.print(dx1_present_current_13);
281
282 // Read present current 14
283 dx1_comm_result = packetHandler->read2ByteTxRx(portHandler, DXL_ID_014,
        ADDR_PRO_present_current, (uint16_t*)&dx1_present_current_14, &dx1_error);
284 if (dx1_comm_result != COMM_SUCCESS)
285 {
286     Serial.print(packetHandler->getTxRxResult(dx1_comm_result));
287 }
288 else if (dx1_error != 0)
289 {
290     Serial.print(packetHandler->getRxPacketError(dx1_error));
291 }
292
293 Serial.print("_present_Current_14:"); Serial.print(dx1_present_current_14);
294
295 // Read present current 15
296 dx1_comm_result = packetHandler->read2ByteTxRx(portHandler, DXL_ID_015,
        ADDR_PRO_present_current, (uint16_t*)&dx1_present_current_15, &dx1_error);
297 if (dx1_comm_result != COMM_SUCCESS)
298 {
299     Serial.print(packetHandler->getTxRxResult(dx1_comm_result));
300 }
301 else if (dx1_error != 0)
302 {
```

```

303     Serial.print(packetHandler->getRxPacketError(dx1_error));
304 }
305
306 Serial.print("_present_Current_15:"); Serial.print(dx1_present_current_15);
307
308 //////////////////////////////////////////////////////////////////// release based on
    current ////////////////////////////////////////////////////////////////////
309 Serial.println("_____");
310
311 int previousCurrent;
312 Serial.println(previousCurrent);
313
314 static unsigned long lastMillis = 0;
315 unsigned long interval = 500;
316 if (millis() - lastMillis >= interval)
317 {
318     lastMillis = millis();
319     Serial.print("Current_difference_=_");
320     Serial.println(dx1_present_current_13 - previousCurrent);
321
322     Serial.println("\n");
323     float diff = fabs((float(dx1_present_current_13 - previousCurrent) / float(
        previousCurrent)));
324     Serial.println("diff:");
325     Serial.println(diff, 6);
326
327     previousCurrent = dx1_present_current_13;
328

```

```

329     delay (1000);
330     if (diff >= DXL_CURRENT_STATUS_THRESHOLD) {
331         Serial.println("als_je_dit_ziet_is_de_if_statement_true");
332         dxl_comm_result = packetHandler->write4ByteTxRx(portHandler,
333             DXL_ID_015, ADDR_PRO_GOAL_POSITION, 1600, &dxl_error);
334         //dxl_comm_result = packetHandler->write1ByteTxRx(portHandler,
335             DXL_ID_015, ADDR_PRO_TORQUE_ENABLE, TORQUE_DISABLE, &dxl_error);
336         //torque uit schakelen voor gripper
337     } //probleem is dat die nu gelijk de torque eraf gooit al voordat die
338         begint omdat die eerste waarde te hoog is gelijk
339     ; //die line met torque disable erachter heeft geen zin want die is dan
340         sws uit voor de hele tijd
341     }
342
343     //
344     //////////////////////////////////////////////////////////////////////////////////////////////////////
345
346     delay(1000); // delay readings
347     Serial.print(dxl_present_current_13);
348     }while((abs(DXL_012_GOAL_POSITION[index] - dxl_present_position_12) >
349         DXL_MOVING_STATUS_THRESHOLD)); //???
350
351     // Change goal position
352     if (index == 0)
353     {
354         index = 1;
355     }

```

```
349     else
350     {
351         index = 0;
352     }
353 }
354
355 // Disable Dynamixel Torque 012
356 dxl_comm_result = packetHandler->write1ByteTxRx(portHandler, DXL_ID_012,
        ADDR_PRO_TORQUE_ENABLE, TORQUE_DISABLE, &dxl_error);
357 if (dxl_comm_result != COMM_SUCCESS)
358 {
359     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
360 }
361 else if (dxl_error != 0)
362 {
363     Serial.print(packetHandler->getRxPacketError(dxl_error));
364 }
365 // Disable Dynamixel Torque 013
366 dxl_comm_result = packetHandler->write1ByteTxRx(portHandler, DXL_ID_013,
        ADDR_PRO_TORQUE_ENABLE, TORQUE_DISABLE, &dxl_error);
367 if (dxl_comm_result != COMM_SUCCESS)
368 {
369     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
370 }
371 else if (dxl_error != 0)
372 {
373     Serial.print(packetHandler->getRxPacketError(dxl_error));
374 }
```

```
375 // Disable Dynamixel Torque 014
376 dxl_comm_result = packetHandler->write1ByteTxRx(portHandler, DXL_ID_014,
    ADDR_PRO_TORQUE_ENABLE, TORQUE_DISABLE, &dxl_error);
377 if (dxl_comm_result != COMM_SUCCESS)
378 {
379     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
380 }
381 else if (dxl_error != 0)
382 {
383     Serial.print(packetHandler->getRxPacketError(dxl_error));
384 }
385
386 // Disable Dynamixel Torque 015
387 dxl_comm_result = packetHandler->write1ByteTxRx(portHandler, DXL_ID_015,
    ADDR_PRO_TORQUE_ENABLE, TORQUE_DISABLE, &dxl_error);
388 if (dxl_comm_result != COMM_SUCCESS)
389 {
390     Serial.print(packetHandler->getTxRxResult(dxl_comm_result));
391 }
392 else if (dxl_error != 0)
393 {
394     Serial.print(packetHandler->getRxPacketError(dxl_error));
395 }
396
397
398 // Close port
399 portHandler->closePort();
400
```

