Design and Development of a Cost-Effective Robotic Arm

Christian Oberhauser and Yeongmi Kim (supervisor)

***Abstract*—Caring for the elderly is becoming increasingly important in view of the demographic shift toward a much older society. Service robots exist for this target group, but they often lack the dexterity to assist them in their daily tasks. Existing commercial robotic arms are often too expensive for the elderly to purchase without government support. In this paper, a cost-effective robotic arm for a smart care bot is proposed.**

The robotic arm has four degrees of freedom and an additional degree for gripping objects. Since gripping different objects is difficult, a soft gripper with three fingers that bend according to the gripping geometry is chosen. In addition, the graspable objects are identified with a depth camera. This enables the use of a motion planning framework.

The control structure is implemented using robot operating system on a Nvidia Jetson Nano. The position of the arm's DC motors is then controlled using a PID controller with the feedback being the angle difference. When the target position is reached, the soft gripper can be closed via an smartphone application and the arm can be driven to a desired position manually.

The developed robotic arm with a payload of 1.5 kg was demonstrated in experiments that it can pick up objects such as cups and bottles. However, the used depth camera introduces a large measurement ripple to the system, which in turn affects the position estimation. Further improvements can be made by replacing the motors and depth camera. With a total cost of less than € 1000, the developed robotic arm is a good basis for development.

***Index Terms*—Service robot, ROS, soft gripper, depth camera, PID controller, robotic arm.**

Y. Kim is with the Department of Medical Technologies, MCI, Innsbruck, Austria, e-mail: yeongmi.kim@mci.edu.

I. Introduction

THE world's population continues to age and the proportion of people over 60 is expected to reach 21.1% of the world's population by 2050. This leads to an increasing need for eldercare support. The older population faces a variety of cognitive and physical challenges and limitations in activities of daily living (ADL). The need for caregivers and family support is therefore of utmost importance to maintain a good quality of life. The development of automation and robotics has opened up many opportunities to support the care of older people, enabling them to manage their daily tasks more confidently and independently, while relieving the burden on family and caregivers [1].

As the cost of industrial robotics decreases and machine learning algorithms improve dramatically, the range of applications for robotic systems in various sectors is expanding. This will lead to a higher demand of robotic systems in nearly every industries [2]. Nevertheless, there are still risks, especially safety is still a big issue. With the growing market of collaborative robots and the development of better collision avoidance, e.g. with series elastic actuators and object recognition algorithms, humans are expected to work even closer with robots [3].

Considering these aspects for the world's social development is the motivation for the cost-efficient intelligent care robot arm. With the help of open source systems like robot operating system (ROS), it is possible to develop a normally complicated control system with object recognition and inverse kinematics (IK) calculations much more conveniently. There are many open source robot projects on websites. e.g. GitHub and hackaday.com, and since they are available to the public, the projects are reviewed and improved by many individuals. This makes open source projects interesting for companies as they reduce the costs for the early development phase.

II. System

A robotic arm often has multiple degree of freedom (DOF), a gripper and a sensor to ensure precise interaction with objects. The number of DOF varies and depends on several factors, such as task complexity, range of motion, object manipulation, and cost constraints. The higher the number of DOF, the more complex the control structure becomes. In addition, the cost of a robotic arm increases with each axis, as more hardware, namely actuators, is required. The gripper used as the end-effector of a robotic arm is designed to lift and hold different objects. Sensor modules are needed to sense the environment and perform the tasks with appropriate precision. For this reason, an RGB depth camera is used to image the environment and additionally obtain the XYZ-data with respect to the camera of the target to be gripped using an object recognition algorithm. Everything is then controlled by ROS-packages named MoveIt, Darknet 3D, a self written PID controller and additional packages for drivers.

A. Kinematics

The kinematics of the robotic arm form the basis of the desired range of motion, as well as the required motion patterns for the tasks that shall be executed. There are several different existing configurations of robotic arms, including both translational and rotational joints, as well as soft robotic features. The choice of joint type as well as joint placement is important for optimizing the functionality and efficiency of the robotic arm.

1) Tasks to assist the eldery: Firstly, the main tasks to be solved include the assistance of picking up objects, such as a cup or a bottle, for the user. The goal is then to make a robotic arm that can assist the elderly person in both picking up and carrying such items.

Secondly, the robotic arm should be able to pick up items from the ground if they are out of reach for the user. The average elderly person experiences daily challenges from having to crouch their bodies in order to fetch items that either have their designated place on ground level or fall to the ground by accident. A robotic arm with the reach to fetch and deliver such items to the user will be able to significantly ease the daily life of an elderly person.

Lastly, the robotic arm should be able to carry items with the weight of 1.5 kg and low-friction over longer distances without letting them slip or fall. The reason for the payload of 1.5 kg is that it is the weight of a regular 1.5 L water bottle and such a bottle is considered as a heavy daily object.

2) Choice of degree of freedom: Considering these requirements, a choice of four degrees of freedom has been chosen. All joints are rotational, to maintain simplicity and usability while also keeping costs to a minimum. The configuration and placement of the joints can be seen in figure 1.

The placement of the joints is motivated by the required tasks to be solved. The robotic arm is inspired by a human arm, consisting of a shoulder joint in the base, with the first two degrees of freedom, and an elbow joint containing the last two degrees of freedom. In addition to these four degrees of freedom, a fifth degree of freedom is considered for the soft gripper, which is discussed in detail in section II-B2.

The shoulder joint (joints 1 and 2) serves the purpose of making the robotic arm able to perform with a 360° motion in the horizontal plane, as well as up and down on the Z-axis. The shoulder joint enables the robotic arm to reach objects independently of their placement, be it in front or behind the defined direction of the robot.

The elbow joint (joints 3 and 4), on the other hand, gives the arm the ability to rotate objects while holding them. This is useful for a task like pouring

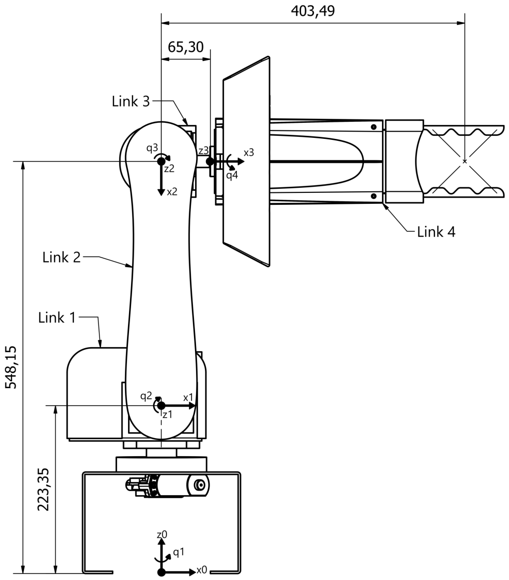

Figure 1: Zero position of the four DOF robotic arm with dimensions in mm.

fluids from a container. It also yields more flexibility in terms of distance from the base of the robot to the object that is to be picked up. The configuration and placement of the joints can be seen in the drawing of the robotic arm in figure 1.

B. Mechanical design

The mechanical design of the robotic arm creates the bridge between the desired kinematics and the resulting structural design. The main points that are taken into consideration in the design process are material choices, structural integrity, cost-efficiency as well as the availability of the components and materials.

1) Robotic arm: The housing has the task of protecting the electronics and ensuring safe handling for the user. In addition, for service reasons, each part of the housing is connected with screws and can be easily dismantled. For safety reasons, each drive motor of each joint has a built-in worm gear; in other words, the robot arm cannot be moved backwards and holds its position even in the event of a power failure.

The motors of joint 1 and 4 are placed directly onto the joint, to maintain a simple design. This is not the case for the motor driving joint 2 and 3. The motor of joint 2 has a 3D-printed planetary gear with a ratio of 1:15.75 attached [4]. The combination of gearbox and motor theoretically delivers a torque of 45.7 Nm, which is according to the calculation

$$\begin{aligned} T &= m_{arm} \cdot g \cdot \frac{1}{2} \cdot l + m_{payload} \cdot g \cdot l \\ T &= 19.65\,\text{Nm} \leq 45.7\,\text{Nm} \end{aligned} \quad (1)$$

sufficient for the task. The estimation of $m_{arm} = 2.5\,\text{kg}$ is extracted from the CAD of the robotic arm and a homogeneous distribution along $l = 0.72829\,\text{m}$ of the weight is assumed. The output shaft of the gearbox is made of 42crmo4 steel for structural reasons.

The motor of joint 3 has been moved further down the link towards the base with a belt drive, to enhance the weight distribution of the arm. The belt drive has a ratio of 1:3.2. Moving this motor allows the payload of the robotic arm to be increased without a large trade-off in terms of weight or strength.

The materials that are chosen for the mechanical design are as mentioned mainly steel and aluminum plates and 3D printed PLA shells. These materials are quite accessible, affordable, and most importantly easy to process, shape, and machine for the required purpose.

As shown in figure 2 the overall look of the robotic arm is simple, with a focus on functionality and usability. The proportions of the links are designed to maximize the range of motion and keep the weight-to-strength ratio as low as possible. The skeleton design allows a better configuration of the motors and sensors.

2) Soft gripper: A soft gripper with three fingers is chosen for the gripping mechanism of the robot arm. The design is inspired by a human finger. Therefore, each finger has three phalanges with the

Figure 2: The finished robotic arm holding a cup.

is used to pull the steel cable. This results in a similar behavior as a tendon of a real finger. To ensure that the steel cables are properly seated and do not get tangled, two rollers are mounted inside the housing. The rollers consist each of a steel tube with a diameter of 5 mm and two ball bearings mounted inside the housing. The main advantage of a soft gripper compared to a conventional gripper is that the fingers of the gripper can bend according to the object geometry. This leads to a broader range of applications.

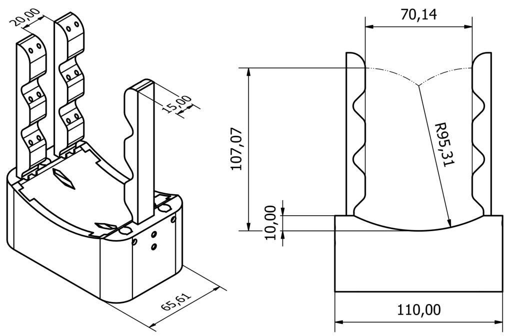

Figure 3: CAD model of the soft gripper with the specification of the workspace in mm.

same length of 15 mm separated by a 45° cut. The total length of a finger is 135 mm and between the phalanges the thickness measures 4.5 mm. The lower part of the finger is firmly connected to the housing with M2 screws, and the thickness of this part is only 2.429 mm. This results in better bending behavior, as the lower part bends before the upper part.

The fingers are 3D printed from Filaflex 70A material and arranged in such a way that each finger can be fully closed. The CAD model of the soft gripper can be seen in figure 3. The maximum allowable diameter of an object is 70.14 mm. The gripping mechanism is implemented with a steel cable with a diameter of 2 mm inserted into the holes of the fingers. A motor with a winch attached is mounted on the bottom of the end-effector housing. The winch

C. Electrical design

The electrical design requires on one hand the handling of the DC-Motor currents from the motor drivers. On the other hand also the I^2C-bus connection from the processing unit the Nvidia Jetson Nano development kit to the AS5048B encoders and to the PCA9685 16 channel PWM board. To reduce, interference the two types of connections, the cable used for the I^2C-bus connection is shielded. As three different types of motors are used, they all have different nominal currents. The nominal voltage for the motors is 12 V and is supplied by an external battery. In table I the selected motors with their expected currents and cabling can be seen.

Table I: Electrical information with selected wiring and position of motors.

Motor position	Type	Nominal Current / A	Cable / mm^2
Joint 1	CYTRON RB-Cyt-176	15	2.5
Joint 2	CYTRON RB-Cyt-176	15	2.5
Joint 3	Nikou B0875MK6W6 20RPM	1.6	1.5
Joint 4	Nikou B0875VF1LF 12RPM	1.6	1.5
Gripper	RB350030-0A101R	2.1	1.5

The addresses of the I^2C hardware can be changed using the pins A1 and A2 of the AS5048B encoders [5]. This gives a different combination for 4 devices. As the motor of the end-effector only needs to be turned on or off the measurement of the current position is not needed. For the parallel connection of the I^2C terminals, SDA and SCL additional pull-up resistors with a value of $10\,k\Omega$ are included.

D. Object detection

To detect objects of interest to the user, an object detection algorithm is used in combination with a depth camera. The algorithm used is You Only Look Once (YOLO) v3 and the depth camera used is the Microsoft Kinect v1.

There are a few reasons to use an older depth camera instead of the newer version. For one, the Kinect v2 weighs 610 g without cable and power supply. Since weight distribution, especially on the end-effector, is of great importance, the Kinect v1 is the better choice with only 430 g. Second, the closest measurement distance of the Kinect v2 is 500 mm and that of the Kinect v1 is 400 mm. Since the end-effector is only 338.19 mm long and the camera is placed as far down as possible on joint 4, at 40 mm, the smaller measuring distance is advantageous. Third, the higher resolution of 1920 x 1080 px means that the object detection algorithm has more pixels to process [6]. This ultimately makes object detections slower. The trade-off between better but slower object detection and good weight distribution is considered. To enable the camera within ROS the freenect_camera package is needed. This package uses the libfreenect camera driver from OpenKinect and provides an ROS interface to the Microsoft Kinect v1. The topics of interest are the \camera\rgb\image_color for YOLO v3 and \camera\depth_registerd\points for the depth information.

The object detection algorithm YOLO v3 is used in a ROS package called darknet_ros [7]. The information is then published as a bounding box with the X and Y coordinates of the received image to the darknet_ros node. As YOLO only provides two dimensional information, another ROS-package is used to combine the measured depth values with the object detection. This package is called darknet_ros_3d [8].

E. Control system

To allow a control of the robotic arm, the motors listed in Table I needed to be controlled. The considered solution is to use a PCA9685 16 channel PWM board which generates five PWM signals with changeable duty cycle. These PWM signals are than used to drive five Cytron MD13-S DC-Driver Boards each connected to a single motor. By varying the duty cycle of the PWM signal the DC voltage applied to the single motors is varied, which results in a controlled speed of those. As this speed is strongly dependent on the applied load a understanding and testing of the system is necessary. The influence of the applied load is of major interest especially when changing the direction of motion of the motors. The change of the direction, e.g. clockwise (CW) to counterclockwise (CCW), is implemented using additional channels of the PWM Board connected to the direction pin of the Cytron MD13-S DC-Driver Boards. As a maximum duty cycle of about 95 % is possible additional smoothing capacitors are used to manipulate the PWM to be a simple 5V high signal, or when changing the direction to be at ground level, low level.

1) Closed loop control: The AS5048B magnetic angle sensors mounted on the output shaft of motor joints 1, 3 and 4 as well as on the motor shaft of joint 2 measure the angle. With their help, the joints can be controlled to a specific angle. The specific angle is determined by the ROS MoveIt package and given in radian. This package calculates the IK from the starting point to the determined depth values of the gripping object. The last motor, which is responsible for closing and opening the gripper, is only switched on or off.

2) Software: The software used to control the robot arm is implemented with various ROS packages, some of which are open source and some of which are self-written. The computing device used is an Nvidia Jetson Nano development kit with Ubuntu 18.04 and the ROS distribution used is named Melodic. The reason for using an older version of ROS is that Nvidia has its own software development kit (SDK), called JetPack, and here the latest version is installed with Ubuntu 18.04. Another reason to use an older ROS version is that the ROS community is mostly working with the first version at the moment, which makes debugging more convenient. The main advantage of a Jetson Nano processor is that it has a built-in GPU, and with the CUDA API it is possible to process machine learning algorithms like YOLO at much higher rate [9]. The first thing that has to be done is to get the CAD model of the robot into ROS. This is done with the universal robot description file (URDF). Then the ROS MoveIt package is set up for motion planning from the URDF using the setup wizard. Next, the various ROS nodes are compiled as a C++ script to; read the encoder values, set the PWM board sample rates, control the angular position with manual buttons or PID controller, connect the camera link to joint 4 for 3D mapping, get target values from MoveIt, get the XYZ values from object detection with respect to the base and finally execute a movement.

3) Human-Machine Interface: To ensure that a non-technical person can control the robot arm, a Human-Machine Interface (HMI) has to be built. The advantage of a system like ROS is that any input can be realized, as only the published topics need to be of a certain type. The HMI used is called ROS-Mobile and can be downloaded from the Google Playstore to an Android smartphone [10]. The Android app can publish and subscribe to various nodes. For this to work, the smartphone has to be on the same network as the ROS master.

After the connection setup is done the app can be configured. The input signals for the direct motor control are ten bool buttons and additional the camera feed can be shown. To start the planned movement to grip a detected object an start movement button is implemented.

III. Evaluation

The built robot arm is evaluated by analyzing different experiments.

A. Payload test

The payload is evaluated with the robot arm extended in a horizontal position and the tested joint is then moved into vertical position. In the horizontal position the maximum torque is applied to the tested joint. The test object to lift is a regular 1.5 L plastic bottle filled with water. Joint 2 is moved, as this joint can apply the greatest theoretical torque and has the longest lifting arm with 728.29 mm. The arm can pick up different types of bottles and cups, with a tested payload of 1.605 kg.

B. Repeatability test

The repeatability of the planned movement is checked by setting two different positions of the robot arm and letting the arm move repeatedly from one point to the other. The positions are chosen in a way that all joints have to be driven in order to move from one position to the next. The movement is then repeated for 20 times with the x, y and z positions being logged by a laser tracking device named LEICA Absolute Tracker AT402 with a of $MPE = \pm 0.015\,\mathrm{mm} + 0.006\,{}^{\mathrm{mm}}/_{\mathrm{m}}$. The reflector used is a tooling ball reflector with a diameter of 0.5 in. The reflector support is then placed on the end-effector of the robotic arm. To calculate the average value and the standard deviation of the movement a MATLAB code is used. The results of the repeatability test are shown in figure 4 as a histogram plot with a normal distribution on top. The standard deviation of the error is $\sigma = 4.05\,\text{mm}$.

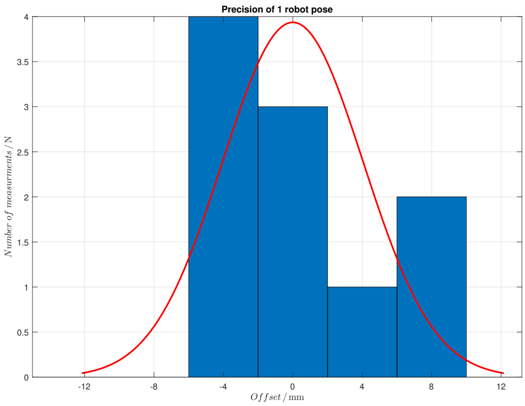

Figure 4: Normal distribution with histogram plot of repeatability test position 1.

C. *Gripping objects*

For the test of the task of picking up a daily object, a 0.39 kg aluminum bottle is chosen. For the task, the test object is placed around 400 mm on the left side of the robot arm at the same height as the housing. Thereafter, the robot arm is moved manually with the HMI to a position where the camera can see the object. Next, the planned gripping movement is initialized with the "start movement" button on the HMI. When the arm has reached the object, the gripper is then closed with another button.

The results of the gripping object test can be seen in figure 5. Figure 5a is the starting position for the test. Figure 5b is the end position after the planned gripping movement is finished. The test object in figure 5a is placed at the coordinates $x = 500\,\text{mm}$ and $y = 0\,\text{mm}$ on the left side of the robotic arm. The task is to grasp the body of the object around Z=130 mm. The object is grasped correctly, as can be seen in figure 5b, with an overshoot in X direction of 100 mm.

(a) Gripping object test starting position.

(b) Gripping object test end position.

Figure 5: Start and end position of gripping object test.

IV. Conclusion

The main objective of this paper is to design and develop a cost-effective robotic arm for a mobile platform. The reason for a cost-effective solution is that many modern robotic arms in the healthcare sector cost in the range of € 10,000, which is inconsequential as many open-source projects exist. A technical person can build the robot developed in this paper with a 3D printer, some machining tools and a budget of € 797.53. A crucial step in the development of a robotic arm is the selection of DOF and the placement of the appropriate joints. The number of DOF has a considerable impact on the cost of a robotic arm, as each additional actuator increases the budget. However, each DOF also increases the working range and the application possibilities. This trade-off between usability and cost is a key factor in any technical development and the four DOF chosen are more on the side of cost-effectiveness. Using worm geared motors as actuators has the advantage that they are not backdrivable and save power from the battery. In addition, the cost per Nm is relatively low compared to other DC motors. Another important step is the selection of the control system. With the use of ROS and the large online community surrounding it, it is possible to use complicated object detection algorithms like YOLO in combination with motion planning packages like MoveIt and many other. This increases the efficiency of development because one does not have to start from scratch.

The developed robot arm is able to lift a weight of 1.605 kg with a maximum reach of 728.29 mm it is in the middle to upper range compared to other robotic systems like [11] and [12]. However, the application possibilities are relatively simple in comparison. The main advantage is cost efficiency and production effort, since most parts can be 3D printed and the source code used is made available to the public.

REFERENCES

[1] M. Kyrarini, F. Lygerakis, A. Rajavenkatanarayanan, C. Sevastopoulos, H. R. Nambiappan, K. K. Chaitanya, A. R. Babu, J. Mathew, and F. Makedon, "A survey of robots in healthcare," *Technologies*, vol. 9, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2227-7080/9/1/8

[2] S. Korus, "Industrial robot cost declines should trigger tipping points in demand," *ARK Investment, Accessed Oct*, vol. 6, p. 2021, 2019.

[3] L. Gualtieri, E. Rauch, and R. Vidoni, "Emerging research fields in safety and ergonomics in industrial collaborative robotics: A systematic literature review," *Robotics and Computer-Integrated Manufacturing*, vol. 67, p. 101998, 2021.

[4] A. Ostler and D. Felbermayer, "Maturation of a two-stage planetary gearbox by 3d printing," MCI - Management Center Innsbruck, 02 2023.

[5] AMS, "User manual – as5048-ab-v1.1," 2014, [Accessed: 26.09.2023]. [Online]. Available: https://ams.com/documents/20143/36005/AS5048_UG000223_1-00.pdf/72e1b407-4bd2-9ab4-0fdf-acbafa41c083

[6] Tölgyessy M, Dekan M, Chovanec Ľ, Hubinský P., "Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2," *Sensors (Basel)*, vol. 21(2), no. 413, 2021.

[7] M. Bjelonic, "YOLO ROS: Real-time object detection for ROS," https://github.com/leggedrobotics/darknet_ros, 2016–2018.

[8] F. M. Rico and F. G. Ramos, "gb_visual_detection_3d," https://github.com/IntelligentRoboticsLabs/gb_visual_detection_3d/tree/melodic, 2020.

[9] NVIDIA Jetson team, "Jetson nano developer kit - user guide," 2020, [Accessed: 26.09.2023]. [Online]. Available: https://developer.nvidia.com/embedded/downloads#search=Jetson%20Nano%20Developer%20Kit%20User%20Guide

[10] N. Rottmann, N. Studt, F. Ernst, and E. Rueckert, "Ros-mobile: An android application for the robot operating system," *arXiv preprint arXiv:2011.02781*, 2020.

[11] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich, "The design of stretch: A compact, lightweight mobile manipulator for indoor human environments," *CoRR*, vol. abs/2109.10892, 2021. [Online]. Available: https://arxiv.org/abs/2109.10892

[12] A. Campeau-Lecours, H. Lamontagne, S. Latour, P. Fauteux, V. Maheu, F. Boucher, C. Deguire, and L.-J. L'Ecuyer, "Kinova modular robot arms for service robotics applications," *International Journal of Robotics Applications and Technologies*, vol. 5, pp. 49–71, 07 2017.

Christian Oberhauser is a master student of the "Mechatronics & Smart Technologies" program at the MCI Innsbruck/Austria Additionally he works as a software engineer with the focus on robotics at Micado GmbH.