Accepted Article

This is the accepted version of the publication as archived with the DLR's electronic library at <https://elib.dlr.de>.
The original publication is available at <https://doi.org/10.1109/CASE49439.2021.9551482>.

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Towards context-aware natural language understanding in human-robot-collaboration

Tobias Haase¹ and Manfred Schönheits¹

Abstract—Human-robot-collaboration still poses a challenge to contemporary robotic systems. Besides functional safety, usability and user experience are topics of current research. This paper describes an approach that combines contextual-awareness and natural language understanding (NLU) to convey human intents to a robotic system. A user study is then conducted using the system in a simplified collaborative assembly scenario and the findings are discussed.

I. INTRODUCTION

Humans use multiple verbal and non-verbal channels in order to communicate with each other. These channels are usually strongly coupled and can be hard to translate to robotic systems. This constitutes a major challenge for human-robot-interaction (HRI). Significant effort is put into emulating these channels in order to make HRI as intuitive, easy and fast as human dialogue.

Communication of the robot’s motion intent through gestures and hint movements has been investigated in [1] and [2]. Visualizing the robot state and communicating its intent using AR/VR is discussed in [3] and [4].

This work is focused on interfacing robots through natural language in industrial scenarios. In [5], Xiaoling Lv et al. in 2008 already demonstrated voice control for robots by controlling a mobile robot through seven speech commands. Tasevski et al. in 2013 introduced a robot that was controlled by simple speech commands that visually determined its workspace and was able to place specified objects correspondingly [6]. A kinematically more complex setup is used by Zhang et al. in [7], where a two-arm humanoid robot is controlled by keywords.

However, these approaches were based more or less on static language. Kapadia et al. [8] used the Amazon Alexa voice assistant to be able to control an industrial robot with natural language without pre-defined keywords. Jungbluth et al. [9] also used Amazon Alexa to develop a voice user interface for an industrial robot. In [10], Eppe et al. developed a natural language interface based on the Embodied Construction Grammar (ECG) framework that allows integration of semantically complex concepts like conditionals and clarification. Paul et al. presented their model of Temporal Grounding Graphs in [11]. Here, the range of language that the robot can understand is expanded using historical information gathered through natural language interactions and visual observations.

In this paper, a natural user interface for robots in industrial applications is developed. The key element under investigation is the inclusion of contextual knowledge such as the user’s position and gaze, which seems suitable to supplement the user’s utterances in order to increase the usability of the system. A user study is conducted in order to evaluate the impact of contextual awareness on the user experience. However, this work does not use other modalities like gestures or body movement. It is planned to do so in future experiments. Furthermore, in this work the signal bandwidth from the user to the robot is significantly wider than vice versa. Truly bilateral communication is beyond the scope of this paper and probably will be covered in future publications.

II. SYSTEM ARCHITECTURE AND IMPLEMENTATION

In this section, a short description of our hardware and software setup is given, and an architecture we designed for this work is introduced.

A. Setup and Preliminaries

Two Universal UR10e robots form the base of our system setup. We chose to use two of them to be able to represent a multi-agent scenario where the user does not always interact with the same robot only, but both of them either alternately or simultaneously. One of the robots is equipped with a Robotiq adaptive 2-Finger-Gripper 140.

The Microsoft HoloLens serves as an input and feedback device for the user. We used it to capture the position and orientation of the user in the first place. Additionally, the HoloLens is used for its speech-to-text capabilities. Only basic visual feedback is displayed via AR.

Initially we have used the Snips platform (discussed in [12]) in order to implement our language understanding components. During our work on the topic Snips was bought by the Sonos company, which subsequently shut down the public access to the console and thus making it impossible to locally train custom sentences and intents. We then proceeded to use Microsoft’s Azure-based LUIS¹.

At first, we investigated using ROS² as a platform to integrate the components. However, we found that while ROS provided a suitable interface for the control of the robots, it did not support the other necessary components properly. Therefore, ROS would limit our choice of other components and also introduce a lot of unnecessary complexity into the infrastructure.

¹Center for Lightweight Production Technology (ZLP), German Aerospace Center (DLR), 86159 Augsburg, Germany {tobias.haase, manfred.schoenheits}@dlr.de

¹Language Understanding Intelligent Service, <https://www.luis.ai/>

²Robot Operation System, <https://www.ros.org/>

Finally, we decided to build a custom communication infrastructure based on the Message Queuing Telemetry Transport (MQTT) protocol, that is already used by snips and other voice assistants.³ This so-called Hermes protocol defines JSON-based messages sent via MQTT. This allows us to build a loosely coupled, modular system with a common communication stream. Participants in this system can be producers or consumers of information or both. We use the InfluxDB database system and Grafana for monitoring and visualization, which come with out-of-the-box MQTT support and integrates easily with the described setup.

B. Communication Design

An outline of our communication architecture is shown in Figure 1. This design was inspired by [3], whose authors also used a central broker to establish communication between robot and interaction layer. Except for the so-called "RoboBroker" all the components basically represent independent services, each of them with a limited scope. The RoboBroker talks to all the components through MQTT, combining their services into application logic. It also listens to all other components and keeps track of the state of the environment.

Fig. 1. System communication

The advantage of this architecture lies in its flexibility. All components are connected and able to receive every information provided by other components. This enables some more features like displaying the robot position inside the HoloLens. Additionally, the RoboBroker is the only proactive component, which means that no unwanted actions are triggered without it.

The independent observer is not a specific component, but a general substitute for many optional components. These components can subscribe to one or more topics for their own purposes.

The original Hermes protocol is excellently suited for communication in the areas of speech-to-text, natural-language-understanding and text-to-speech. However, we had to extend it with message definitions from the robotics domain.

C. RoboLens

In order to provide position and orientation of the user as well as the utterances by the user, we have written RoboLens,

which is an application running on the HoloLens. Speech-to-text processing is activated by saying the keyword "Listen", the result is sent to the RoboBroker as a string. To archive this, the build-in dictation library was used. Furthermore, the RoboLens application handles the translation between its own coordinate system and the so-called world coordinate system. Each time the HoloLens reboots, it creates a new origin at its current position. However, the rest of the system uses a fixed coordinate system that has its origin at a defined position. In order to translate between these two systems, the user must place a virtual origin in the origin of the world. This manual step corresponds to our needs at the moment, but can be replaced in the future by the spatial mapping capabilities of the HoloLens.

Since the HoloLens does not provide MQTT support, we have developed a simple communication protocol based on socket communication. The HoloLens Connector receives these custom packets and forwards them to the MQTT broker.

D. RoboLuis

RoboLuis is a component that we have developed to work as a gateway to the Microsoft LUIS Service. It simply obtains a query, translates it into a HTTP Request for LUIS and posts back the received intent to the broker using the hermes protocol.

We chose Luis because of the easy-to-use API and the excellent test results. Ball et al. evaluated several frameworks in 2018 [13]. Luis got the best result, followed by Snips and Rasa NLU.

RoboLuis is able to detect an intent, but it can also recognize slots. The intent describes the overall intention of the user. Slots can contain additional information needed to fulfill the user's request. For example, the utterance "Turn on the light in the living room" would lead to the intent "Turn-on-lights", if this was trained before. A possible slot can be {location: living room}.

E. URConnector

There are two instances of the URConnector, one for each robot. The URConnector is a MQTT interface for the Universal Robots that we have created. It provides current information about the status of a robot and forwards motion commands. Each URConnector is responsible for one of our two robots. It connects to its robot via a well documented tcp interface [14]. On the one hand, this makes it possible to receive status updates about the robots. On the other hand, this way a predefined robot program can be sent to the robots. The URConnector has access to skeletons of robot programs, which can be completed with coordinates. The coordinates are received via MQTT. In this way the URConnector can generate complete robot programs and send them to its robot.

F. RoboBroker

Once the RoboBroker has received the user intent, it adds missing information using its knowledge about the environment. Then it generates a program containing the

³<https://github.com/snipsco/hermes-protocol>

corresponding logic and actions for the respective robot and sends it to the URConnector.

The RoboBroker starts its process by receiving a transcribed utterance from the RoboLens application via MQTT. Then the RoboBroker generates a request to RoboLuis, which detects an intent. If the matching score is high enough, the RoboBroker generates an action. Each action contains an optional robot movement and a spoken response for the user. If necessary information is missing, the RoboBroker tries to extract this information from the environment. This way some intents can be used to create a valid action even if some important slots are missing. The generated action is executed by sending a text to RoboLuis and optionally movements to the corresponding robots.

Additionally, the RoboBroker is able to compensate some services. For example, if LUIS is not available, the RoboBroker can use a keyword-based intent recognition system instead of natural language understanding. We achieved this by matching the spoken text against possible phrases and calculating the respective difference.

G. Additional services

There is the possibility to use an optional motion planning service. This service knows the robots' positions and calculates a trajectory to the target position preventing any collisions. We decided not to use this service since there are only short movements necessary to fulfill the given task. Such an extension might be useful if the task required more complex trajectories, e.g. because of obstacles that need to be avoided. Furthermore, we configured the influxdata TICK Stack⁴ to monitor our system. Grafana Dashboards were used to display important information.

III. EXPERIMENT

For experimental evaluation we conducted a user study. The user study was designed around a use case that on the one hand is realistic enough in the sense that it might actually be implemented in a comparable form in the industry. On the other hand we were restricted by time and budget constraints and wanted to stick to the collaborative robots. The user study involved 24 people from our institute and related organizations, all of whom came from research backgrounds and had varying levels of experience with robotics. Aside from a similar work environment, participants are sufficiently varied to allow for a good balance.

A. Task

The use case we came up with for our user study is a simple assembly scenario, where participants have to perform a couple of steps in order to assemble a computer case. In particular, they have to install a hard drive and read out a serial number printed on the motherboard. One of the robots named "Tom" has a gripper that can hold a flashlight. The other robot "Jerry" has the computer case mounted to its flange and serves as a positioning device. Figure 2 shows an example configuration during one of the tests.

Fig. 2. Collaborative robotic setup (example configuration)

At the beginning, both robots are in a defined home position, which is hard coded into our system. All the steps that have to be performed in detail are as follows:

- 1) Move the robots into a comfortable working position
- 2) Open the computer case by removing 4 screws
- 3) Give Tom the flashlight
- 4) Bring the flashlight into a suitable position to illuminate the inside of the case
- 5) Read out the serial number
- 6) Install the hard drive
- 7) Close the computer case
- 8) Make Tom release the flashlight again
- 9) Move both robots back into home position

B. Study Design

One of our objectives was to analyze the impact of different levels of support for natural language understanding. Therefore the system was set up to switch between 4 different levels of NLU assistance. This assistance level also represents the only independent variable controlled by us.

- L1: Zero voice assistance, conventional programming with teach pendant
- L2: Fixed voice commands with predefined sentences
- L3: NLU without contextual awareness
- L4: NLU with contextual awareness

L1 means that there is no voice control at all. Both of the robots are conventionally controlled using only the teach pendants. At level L2, the robots can be controlled with fixed, hard-coded speech commands only. For instance, to make Jerry go to its home position, the user has to say the exact phrase "Jerry, move home!". Since the RoboBroker is able to substitute natural language understanding with a built-in pattern matching algorithm, it was easy to enable this mode. On level L3, NLU is enabled allowing more inaccuracy for voice input but potentially also introducing higher risk of misinterpretation. For example, the two sentences "Jerry, go home" and "Jerry, move to home position" should both lead to a deduction of the "Home" intent. On both L2 and L3, context information such as the position and gaze direction of the user is not available to the system. That means that

⁴<https://www.influxdata.com/time-series-platform/>

the system cannot know which robot the user is looking at, thus making it necessary to explicitly say the corresponding robot's name for each input. The respective MQTT topics are simply ignored within the RoboBroker when it has switched to this mode. Since no contextual information is used in these levels, a microphone could theoretically be used instead of the HoloLens for speech input. However, for reasons of consistency we decided not to do so. Level L4 has NLU enabled as well as full access to context information. With Level L4, it is enough to look at Jerry and say e.g. "Go home" or "Home position" to make Jerry go to its home position. Furthermore, the intents "Look" and "Come" are available in L4, which makes the collaboration more convenient. Table I shows the list of defined intents we have trained LUIS for.

TABLE I
DEFINED INTENTS

Intent	Description
Move	Makes the robot move into specified direction
Turn	Makes the robot rotate into specified direction
Home	Makes the robot move to its home position
Freedrive	Activates the freedrive mode of the robot
Open	Opens the gripper (only available for Tom)
Close	Closes the gripper (only available for Tom)
Stop	Stops the current action
Come	Makes the robot come to the user
Look	Makes the robot orient itself towards the user

The next step was to determine which tests each of the participants had to do. To allow reasonable comparisons between L2, L3, and L4 and to ensure the independence of the measurements, each participant was allowed to test only one of these levels. On the other hand, the difference between speech-based interaction in contrast to the conventional method using only the teach pendant should be measured. We settled for each participant doing two runs — one with Level L1 and another with L2, L3 or L4. To further balance the impact of the training effect on the results, we alternated the order. With each participant, the following schedule was used:

- 10 minutes general introduction to use case
- 5 minutes training level A
- Test using level A
- 5 minutes training level B
- Test using level B

The short training periods have been included for mainly two reasons. On the one hand, the participants had varying experience with the Universal Robots platform and / or voice assistance, which should be mitigated a bit in this way. On the other hand, the participants have different abilities to learn and adapt, so a short training should have a beneficial effect on the overall fluency of the process, probably leading to more objective and better comparable results.

To evaluate the system, we decided to measure usability, which is composed of efficiency, effectiveness and user experience [15]. Efficiency is measured by the time required

for a task. Effectiveness is determined by how few errors occur during the task. As errors we counted wrongly deducted intentions and false negatives. To measure the user experience, the users were asked to fill out a system usability scale (SUS) form, which is a 10 item questionnaire with five response options ranging from "strongly disagree" to "strongly agree" [16]. Furthermore, the participants could add qualitative remarks to the form as free text. Additionally, general process parameters such as the position of the user and the robots, the speech-to-text results, the derived intents and errors were permanently logged in the database.

All tests were performed in an industrial environment, where the microphone of the HoloLens was exposed to considerable background noise such as other robots and various industrial machines. Therefore, realistic conditions can be assumed.

C. Results And Discussion

The average SUS scores across all levels are depicted in figure 3, with L2 having an average score of 81.25%, L4 of 75.06%, L1 of 74.38% and L3 of 70.00%.

Fig. 3. Mean and standard deviation of SUS score per level

While the SUS values of all other levels vary widely between 45% and 95% it can be clearly seen that the values for L2 are constantly high. It is reasonable to assume that this is due to the fact that the system is very simple and intuitive to use. L3 got the lowest SUS score of all levels. This could be the result of the increased error rate in combination with the language barrier, since most users' native language is German. As in L2, all necessary information has to be given by voice, which is uncomfortable for the users and can rather lead to errors. Users of L4 can select the desired robot by looking at it. Additionally, they can tell the robot to move to them, which may be a complex movement. These abilities reduce the amount of necessary information of the utterance and render the whole system more robust against misunderstandings. It can be seen that the use of environmental knowledge drastically reduces the negative impact on the SUS score of L3.

The differences in error rates can be seen in figure 4. The average error rates are 2.13 for L4, 2.5 for L2 and 3.25 for L3. Since we defined errors as misunderstandings, we could not measure any errors in L1. The measured duration times can be seen in figure 5. The shortest average duration was achieved with L1 at 311s, followed by L2 at 346s, L4 at 379s

Fig. 4. Mean and standard deviation of error rate per level

and L3 at 387s. Looking at these figures, it can be seen that a shorter duration does not directly lead to lower error rates. Although L4 has a longer average duration, it actually has the lowest error rate and the smallest error margin.

Fig. 5. Mean and standard deviation of duration per level

As mentioned earlier, users have to utter the keyword "Listen" to activate voice control. The recognition of the keyword as well as the language input was signaled to the user with a confirmation sound. This handshake-like scheme introduced a delay in the interaction of a couple of seconds each, as the HoloLens need some time to switch between the different interfaces for keyword and dictation. We have monitored that this had a strong negative influence on the duration and the user experience.

As one could guess from the previous figures, there are minor but relevant differences between the different levels. The calculation of multivariate analysis of variance (MANOVA) proves this, as the table II shows. We used two different MANOVA tests, as both have their own advantages. Since MANOVA does not allow repeated measurements, we used only the results of L2, L3 and L4.

TABLE II
MANOVA OVER L2, L3 AND L4 AND DIFFERENT VARIABLES

Test	Df	test stat	approx F	num Df	den DF	Pr(>F)
Wilks	2	0.30484	2.5958	10	32	0.01974
Pillai	2	0.8947	2.7522	10	34	0.01339

We calculated T-tests to prove the significant difference between the given levels and found that the only significant

difference between L2 and L3 is a p-value of 0.08995. These figures are shown in table III.

TABLE III
T-TESTS OVER SUS SCORE OF L2, L3 AND L4

Groups	Variable	t	df	p
L2-L3	SUS	1.8594	10.981	0.08995
L2-L4	SUS	0.94364	10.339	0.3669
L3-L4	SUS	-0.64195	13.854	0.5314

It can be seen that the decrease in usability due to the higher complexity of NLU can be compensated by adding environmental awareness to the system. We expect the SUS score to further increase once we have addressed user feedback and added additional intents that benefit from this awareness.

TABLE IV
T-TESTS OVER DURATION OF L2, L3 AND L4 IN COMPARISON WITH L1

Groups	Variable	t	df	p
L2-L1	Duration	1.8831	7	0.1017
L3-L1	Duration	2.0498	7	0.07956
L4-L1	Duration	2.0898	7	0.07499
L2to4-L1	Duration	3.6333	23	0.001392

We have calculated paired T-tests with repeated measurements over the levels with voice assistance compared to the respective L1 measurements. While other variables such as the SUS score did not differ significantly across all experiments, we found that the duration of the L1 tests is always less than that of the respective other test. Accordingly, the overall comparison between tests with and without voice assistance shows the same effect. The results are shown in table IV. We expect to decrease this effect by improving the flow of interaction. Furthermore, the UR robots are very easy to navigate as they can be moved by hand using the so-called Freedrive mode. Conventional Cartesian movements are much slower because they are less intuitive. Kapida et al. observed a decrease in average times in their user study [8]. However, the system they used to compare with the NLU-based system was more complicated and less comfortable to use than the UR robots without language support.

We analyzed the free text remarks we got from the participants and tried to gather issues that were mentioned by multiple people. The following list shows a number of points with the highest number of occurrence:

- The ability to use both hands during the task is beneficial (mentioned 15x times)
- Teach pendant is uncomfortable, heavy, need to change pendants (one pendant per robot) is cumbersome (11x)
- HoloLens feels heavy and uncomfortable (10x)
- Keyword-based activation of language understanding delays interaction (10x)
- Insecurity in interaction, incorrect understanding (7x)
- Voice interaction is intuitive and easy to learn (6x)

The first two points prove that there is great potential for natural language understanding when working with robots. The user can use both his hands for the actual task, while the robots can be controlled asynchronously in a natural way. Even though we have received a lot of positive feedback, there is still room for improvement.

IV. CONCLUSIONS AND FUTURE WORK

This paper introduced an approach to control industrial robots through a natural user interface. Different inputs such as the user's position and orientation, as well as natural language understanding, are combined into a virtual assistant that enables collaboration between a human and multiple robots. Furthermore, a communication architecture based on MQTT is presented. This architecture is expandable to integrate additional sensor interfaces or more complex robots and grippers in future experiments. We investigated the impact of contextual awareness to voice-based control by conducting a user study. We have found there is remarkable potential for NLU combined with environmental knowledge in industrial HRI applications. Especially when the robots have to perform complex movements that are not easily accessible via their programming device, the natural user interface has a great benefit. Many of our users enjoyed being able to collaborate with multiple robots while using both hands for their actual work.

We also have identified challenges that we want to investigate in future experiments. Future work could include some of the following items. The need of a keyword delays the interaction and disturbs the natural feeling while working with the robots. In the future we plan to drop the generic keyword in favor of the names of the robots. We could go even further and activate voice control simply by looking at a robot, as one would do with human co-workers. This can be supplemented by additional visual feedback, e.g. highlighting the currently active robot on the HoloLens. However, it is necessary to determine the risk of false positives following this approach.

We have discovered that while the voice is an excellent tool for activating interaction, it is not very natural to embed all the necessary information of an assignment in it. Furthermore, we have shown that HRI benefits even more from complex commands than from simple motion commands. Accordingly, we try to minimize the amount of information that needs to be embedded in the voice command. However, if an important part of the command is missing, there should be a possibility for the system to ask for it. This is another important part of natural language interaction and of future voice assistants.

Furthermore, we plan to make the interaction multimodal and bilateral, i.e. incorporate concepts like gestures and body movement and combine them with natural language understanding in order to define complex tasks for the robot in an intuitive, natural way. For this purpose we plan to add AR-based models to convey robot intent to the user.

REFERENCES

- [1] T. Nakata, T. Sato, T. Mori, and H. Mizoguchi, "Expression of emotion and intention by robot body movement," in *Proceedings of the 5th international conference on autonomous systems*, 1998.
- [2] L. Takayama, D. Dooley, and W. Ju, "Expressing thought: Improving robot readability with animation principles," in *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2011, pp. 69–76, ISSN: 2167-2121.
- [3] J. Guhl, S. Tung, and J. Kruger, "Concept and architecture for programming industrial robots using augmented reality with mobile devices like microsoft HoloLens," in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Sep. 2017, pp. 1–4.
- [4] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris, and S. Tellex, "Communicating Robot Arm Motion Intent Through Mixed Reality Head-Mounted Displays," in *Robotics Research*, N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, Eds. Cham: Springer International Publishing, 2020, pp. 301–316.
- [5] Xiaoling Lv, Minglu Zhang, and Hui Li, "Robot control based on voice command," in *2008 IEEE International Conference on Automation and Logistics*, Sep. 2008, pp. 2490–2494.
- [6] J. Tasevski, M. Nikolic, and D. Miskovic, "Integration of an industrial robot with the systems for image and voice recognition," *Serbian Journal of Electrical Engineering*, vol. 10, no. 1, pp. 219–230, 2013. [Online]. Available: <http://www.doiserbia.nb.rs/Article.aspx?ID=1451-48691301219T>
- [7] Y. Zhang, Z. Lu, C. Wang, C. Liu, and Y. Wang, "Voice control dual arm robot based on ROS system," in *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, Aug. 2018, pp. 232–237.
- [8] R. Kapadia, S. Staszak, L. Jian, and K. Goldberg, "EchoBot: Facilitating data collection for robot learning with the Amazon echo," in *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, Aug. 2017, pp. 159–165.
- [9] J. Jungbluth, R. Krieger, W. Gerke, and P. Plapper, "Combining Virtual and Robot Assistants-A Case Study about Integrating Amazon's Alexa as a Voice Interface in Robotics," in *Robotix-Academy Conference for Industrial Robotics (RACIR) 2018*. Shaker, 2018, p. 5.
- [10] M. Eppe, S. Trott, and J. Feldman, "Exploiting deep semantics and compositionality of natural language for Human-Robot-Interaction," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, South Korea: IEEE, Oct. 2016, pp. 731–738. [Online]. Available: <http://ieeexplore.ieee.org/document/7759133/>
- [11] R. Paul, A. Barbu, S. Felshin, B. Katz, and N. Roy, "Temporal Grounding Graphs for Language Understanding with Accrued Visual-Linguistic Context," *arXiv:1811.06966 [cs]*, Nov. 2018, arXiv: 1811.06966. [Online]. Available: <http://arxiv.org/abs/1811.06966>
- [12] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips Voice Platform: an embedded Spoken Language Understanding system for private-by-design voice interfaces," *arXiv:1805.10190 [cs]*, May 2018, arXiv: 1805.10190. [Online]. Available: <http://arxiv.org/abs/1805.10190>
- [13] A. Ball, C. Doumouro, and J. Dureau, "An Introduction to Snips NLU, the Open Source Library behind Snips Embedded Voice Platform," Jun. 2018. [Online]. Available: <https://medium.com/snips-ai/an-introduction-to-snips-nlu-the-open-source-library-behind-snips-embedded-voice-platform-b12b1a60a41a>
- [14] Universal Robots A/S, "Remote Control Via TCP/IP - 16496," 2020. [Online]. Available: <https://www.universal-robots.com/how-tos-and-faqs/how-to/ur-how-tos/remote-control-via-tcpip/>
- [15] DIN e.V. (Hrsg.), "Ergonomie der Mensch-System-Interaktion - Teil 11: Gebrauchstauglichkeit: Begriffe und Konzepte (ISO 9241-11:2018); Deutsche Fassung EN ISO 9241-11:2018," Beuth Verlag GmbH, Tech. Rep., Nov. 2018. [Online]. Available: <https://www.beuth.de/de/-/-/279590417>
- [16] J. Brooke, "SUS-A quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996.