

STANDARDIZED DATA FORMATS FOR FLEXIBLE HUMAN-ROBOT-COLLABORATION AT A MULTI-WORKSTATION ASSEMBLY TABLE

Daniel BRAND^{1*}, Niels BENTHIN¹, Marvin BOHL¹

^{1*} City University of Applied Sciences, Nature and Engineering, Mechanical Engineering, IAT, Neustadtswall
30, Bremen, 28199, Germany, e-mail: brand-daniel@gmx.de

(Received 24 September 2019, Accepted 21 October 2019)

Abstract: In the process of industrial automatization, collaborative robots are becoming an increasingly important tool in a wide range of industries. The robot assists humans in strenuous and repetitive tasks, in order to increase the productivity and well-being of the worker. The concept presented in this paper focuses on a flexible assistance function of a robot for multiple workstations simultaneously and incorporates different human-machine-interaction approaches. Therefore, an assembly table (Figure 2) with the possibility to study different concepts for one robot at the same time is designed and constructed. Aspects like ergonomics and safety are considered in the design. Additionally, a standardized data format that enables users and developers to control the robot through peripheral devices is created. As an example, gesture and voice control are discussed and evaluated.

Keywords: human-robot-collaboration, assembly, data format, ergonomic

1. INTRODUCTION

The society is increasingly influenced by robots which interact with humans. The industrial or high technology company sector has long since ceased to be the only operator of human-robot-collaboration (HRC). For example, at home or in the agricultural sector robots collaborate with humans in teams [1-4]. Despite the improvements in robotics, a full replacement of the human part is highly improbable because of the robot restrictions (e.g. the robot needs to be maintained). Therefore, HRC is in a steady progress [4,5]. To allow such flexible teams, the robot needs different safety requirements as for instance force limiting. Those requirements, which are specified from the International Organization for Standardization (ISO) would allow a fenceless interaction with robots in the workplace [4,6,7]. To get an insight in the field of collaborative robotics, the City University of Applied Sciences offers inter alia the lecture/module Interdisciplinary Project (INTP) for students in the master program mechanical engineering and aerospace technologies. Students will acquire knowledge and skills in the field of

cooperating work and networking on selected (research) projects of the institutes for mechanical engineering [8]. One of those research projects, with the subject “Collaborative Robotics in Assembly”, offers students, even beyond one semester, the opportunity to deal with robotics, digital production planning and machine learning. The focus is on concept development and the associated practical implementation for industrial purposes [9].

This paper includes the results of four student projects during this (INTP-) research project. While working in groups on different tasks the aim is to design a multi-workstation assembly table (MWAT) for a HRC workspace (see section 3.1.) with (currently) three robot control types (see section 3.3.). The design considerations are important for further development in the “Collaborative Robotics in Assembly”-project. Therefore, ergonomic design considerations were in focus.

The deployed robot from Universal Robots (UR5) is integrated in a network consisting of a server (to remotely control the UR5) and a database (to store information and robot positions). In order to adapt the robot controller for this research, a separate operating interface is developed. In this interface the technician

is able to operate the robot with a selected control type (or device). Due to these different types of control and the constantly changing tasks and functions for the robot during this and further projects, a standardized data format (SDF) is defined (see section 3.2.). This format allows quick and easy changes or simple implementation of new functions for the robot control. In addition, it is possible to realize an operation of different other robot models (instead of the actual robot) in a short period of time. Nevertheless, the peripheral controls benefit from the SDF due to – again – quick and easy changes and uniform commands for the server. Due to the control peripherals, the technician is able to interact with the Robot freehand.

The MWAT design, the SDF and the robot control peripherals are discussed in the following sections.

2. PROJECT OBJECTIVES AND GOALS

First of all, the MWAT is planned to be a workplace for multiple HRC-purposes. It needs the flexibility to allow the setup of different and depending on the task(s) adaptable workstations. Additionally, minimum space requirements and lightweight materials are predefined. Nevertheless, the workspace needs to be ergonomic for the technicians.

To collaborate with the robot, a network with a server, database and different control peripherals (see Figure 1) need to be developed. This allows to control the robot in real-time, store data e.g. specific positions of the robot and share information with superiors e.g. the actual status of the task-progress. Therefore, a large amount of communication data is expected. The robot control itself has multiple command-functions to use (e.g. move, stop) and monitoring data (e.g. motor current, motor temperature). To simplify this amount of data, a SDF shall be developed, which allows every robot command to be written in one single array with an identical structure.



Fig. 1. Robot control peripherals; Left: Graphical User Interface; Middle: Gesture Control (Myo) [10]; Right: Voice Control (RealWear - HMT1) [11]

The robot control itself shall be realized with three different peripheral devices (see Figure 1). Main control device is the graphical user interface (GUI). Server settings and the robot control shall be implemented in the GUI. Furthermore, connection options enable the other robot control devices. This prevents the simultaneous control from different devices. Robot control by gesture recognition is planned with a Myo (Thalmic Labs) which recognizes muscle contraction patterns and the arm motions. The signals of this armband will be evaluated and converted to specific commands which are send to the server (or to the robot) by the SDF. Another option to control the robot is via voice control (VC). Implementation of this control shall be realized by a RealWear – HMT1 smart-glass. It combines the control with the voice and shows important data on the corresponding display.

3. RESULTS AND DISCUSSION

The mentioned project objectives and goals are realized as described. The following sections are subdivided in the construction of the MWAT, the SDF and the robot control devices. After this, a small overview of application and testing of the main objects is given.

3.1. Assembly Table Design

An assembly table as seen in Figure 2 with four workstations which have an angular offset of 90 degrees from each other, is constructed. At these four assembly workstations the following concepts will be implemented:

- manual assembly,
- HRC,
- automated assembly / material staging,
- Robot-Robot-Collaboration.



Fig. 2. CAD-Model of the multi-workstation assembly table with shown work area (light blue sphere) of the robot

During the designing process of the assembly table, workplace safety and ergonomic aspects were of high importance. To assure optimal ergonomic design for the workplaces, the following parameters as shown in Figure 3 were considered [12]:

- table height (TH): 725 mm,
- table depth (TD): 900 mm,
- legroom depth (LD): 600 mm,
- footwell depth (FD): 800 mm.

The framework of the assembly table is realized using “Item”-aluminium profiles of different lengths. These specific profiles are chosen, as they are light and can be mounted very quickly. Furthermore, these aluminium profiles can be added or exchanged without much effort. To build a stable assembly table, numerous struts are used to create a stiff truss construction.

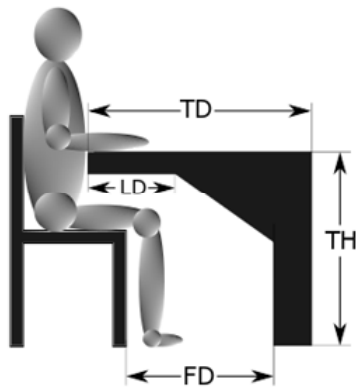


Fig. 3. Design considerations for an ergonomic work place

This truss construction (Figure 2) consists of five table legs. Four of which are placed outside the table top, that extend upwards to build a fastening structure for several devices. For maximum stability a fifth shorter leg is placed in the middle of the construction. The workstation provides the benefit, due to different workstation equipment, to keep the workplace design modular and thus adapted to the technician or actual task. Among other things the following devices can be easily attached to the outside fastening structure:

- depth cameras,
- displays / touchscreens,
- clipboards,
- mini-PC's,
- pocket projectors,
- holder for tools.

Furthermore, a round table top is mounted to the truss construction that can be sequentially rotated by 90 degrees using an electric motor, which is located near the middle under the workstation surface. The rotation of the table top is realized by eight heavy duty castors. These are positioned on the truss construction under the table top. The centring of the table top is

realized by side rollers. These side rollers are mounted on the fastening structure.

In the hole in the middle of the table top a collaborative robot is directly attached to the truss construction (currently a UR5 robot). As a result, the collaborative robot will not rotate while the table top rotates. This allows a fixed coordinate system for the whole construction.

This design provides a good initial condition for further researches to HRC in the mentioned project of the City University of Applied Sciences. Therefore, the SDF is developed, to simplify the connection to the robot.

3.2. Standardized Data Format

To archive the possibility to control a given robot (e.g. UR5) with the gesture- or voice-control (GC, VC), the commands for the robot shall be standardized. First of all, every needed movement type and setting have to be defined in this data format. Additionally, a GUI communicates with the robot (or exactly with the server) with the same format.

First, a data format has to be defined. Therefore, an array-type is chosen, where all commands have a fixed place in the array. These commands are documented with the corresponding input, the location in the array, the (detailed) equivalent robot motion of this command and at least one example. This form needs to be updated every time a change in the format is made to prevent errors in the communication.

An example of this data format with a description of the elements is shown in Figure 4. The first element describes the movement command. In this example a motion, relative to the actual position of the robot (which is not important for the data format) is displayed. It additionally signalized to move in the tool-coordinate system. This characteristic is created due to the definition which will be described in detail later. The currently existing coordinate systems (base and tool) and their positive directions are shown in Figure 5.

The following three values in Figure 4 define the distance for the robot movement in millimetre and the selected coordinate system (X-Axis = 80.0 mm, Y-Axis = 90.0 mm, Z-Axis = 70.0 mm in tool-coordinate system directions).

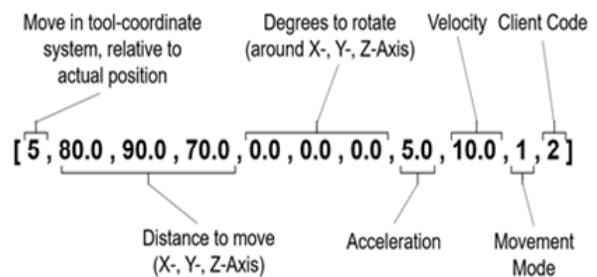


Fig. 4. Example of a SDF command as an array object

The chosen values at the fifth, sixth and seventh position transmit, that the robot should not rotate around the corresponding axis. As the distances described before, the order of value input is the rotation angle in degrees around the X-, Y- and Z-axis of the selected coordinate system.

According to these values, the next two objects in the array describe the acceleration (5 mm/s^2) and the maximum velocity (10 mm/s). If the distance is too short, it is possible that the robot cannot achieve the maximum velocity because the deceleration (when reaching the position) is the same as the (negative) acceleration value.

The movement mode (second last digit in Figure 4) is set to a circular movement mode of the system (move joints freely) and the last integer represents the client code, which indicates in this case the command comes from the VC.

The complete array object has to be converted from the user into a "JSON"-String and send as a byte format to the server. The server converts the data back to an array object and filter the necessary input and (if every input is correct) moves the robot as described above.

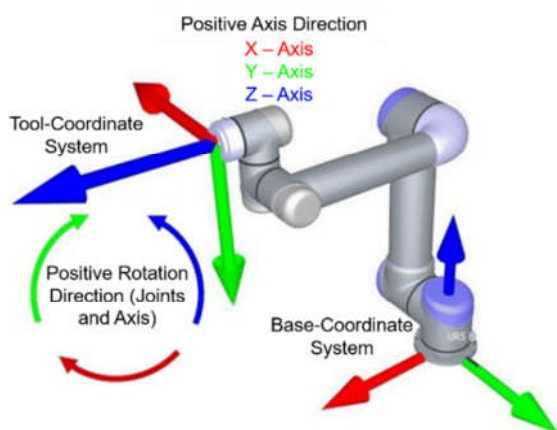


Fig. 5. Illustration of the coordinate systems (base and tool) and the positive definition of the axis of the UR5 robot of the City University of Applied Sciences. (Arranged screenshot of robot from RoboDK-Software)

Some of the commands affect the following input of the array, because different values for different movement types are needed. The inputs in the array (commands) are determined as integers (except of movement distances, velocity and acceleration values). With this, the robot gets exact information about its current task. The actual status of this SDF offers different types of movement and settings for the robot. This format is described in the following in more detail for different movements using the example of the UR5-Robot of the University of Applied Sciences Bremen.

It is necessary to define all inputs with a value. Actual a fully completed chain of commands (full array) needs eleven inputs (see Figure 4). The first integer defines the movement type. With an integer from one to five it is possible to select one of the following movement commands:

1. Movement command (select integer):
 - 1.1. Move the joints relative to the actual robot position.
 - 1.2. Move in the direction of the base-coordinate system of the robot, relative to the actual position.
 - 1.3. Move to defined position in the base-coordinate system of the robot.
 - 1.4. Move to defined position with specified joint position.
 - 1.5. Move (and rotate) in the tool-coordinate system of the robot, relative to the actual position.

The next six floats describe (depending on the chosen command before) the new position or the distance to be covered from the current position (zero means no movement in the direction) in millimetres or rather degrees:

2. Float for: Joint one / coordinate system X-value.
3. Float for: Joint two / coordinate system Y-value.
4. Float for: Joint three / coordinate system Z-value.
5. Float for: Joint four / coordinate system rotate around X-value.
6. Float for: Joint five / coordinate system rotate around Y-value.
7. Float for: Joint six / coordinate system rotate around Z-value.

After the movement and the distances/position coordinates of the robot are defined, additional settings for the robot are required. Therefore, the next three array positions are reserved for it:

8. Float for: Acceleration.
9. Float for: Velocity.
10. Movement mode (select integer).
 - 10.1. Circular.
 - 10.2. Linear.

Finally, a client definition is necessary. Because of (actual) two different control peripherals, a query is made of which control peripheral is allowed to control the robot. This prevents errors related to duplicate control attempts. To differentiate between VC and GC the last integer describes from which peripheral the command is transmitted:

11. Client code (select integer).
 - 11.1. Gesture-control (GC).
 - 11.2. Voice-control (VC).

This format is modular, so that new functions can be added with little effort. To stop the robot (e.g. for safety reasons) the data format is simplified. In this case an array with only two commands is necessary.

The first position is the stop-signal (0), which is followed by the client code (see above).

3.3. Robot Control

The robot can be controlled in three different ways (Figure 1). Main control is obtained by a GUI on a computer.

All required settings can be made here. Additionally, the connection between the peripherals and the robot is controlled via the GUI. A control from the peripherals is only possible, if the user allows a connection from a peripheral control system to the server. Nevertheless, the GUI can always disconnect the user from the peripheral system or stop the robot immediately for safety reasons. Additionally, it is possible to run the robot only on a simulation software or save and run to different robot positions, which can be saved local or on a database. While the GUI use buttons to operate the robot, it is possible to change to a VC or GC mode. An individual adapting user interface for each technician – which reducing the information amount to a required minimum – results of the modular software design.

The development of using gestures for commanding the robot is done by using a gesture recognizing armband named Myo [13]. The Myo is made by Thalmic Labs who has now ceased production. With eight electromyography sensors it measures the electricity that runs through the muscles of the arm due contracting and shifting while performing gestures with the hand [13]. The signals are processed parallel and translated to the gestures, which will be used to perform different tasks. The following aspects are particularly important.

The gestures for commanding the robot should be very intuitive. For this reason, they are chosen out of the context of the specific task. For example, the robot will move upwards by lifting the arm. In the workspace the armband will be worn during performing different tasks, like assembling complete subassemblies. To ensure that gestures will not be recognized and processed by accident during this task a safety procedure is developed. This precaution is illustrated in the upper part of Figure 6. The lower section images the control options simplified. In the following the procedure to send a signal to the robot/server is specified.

First, the gesture of a double tap with two fingers is used to initialize the software. In [14] is determined that this gesture has the worst recognition rate, so the change of doing this gesture accidentally has the smallest probability to occur. Following a fist gesture, combined with a roll rotation of the forearm to the left or to the right is needed, to get into the next step of the procedure. Because of safety reasons a minimum rotation angle of 30 degrees have to be reached in order to advance. This action needs to be done in five seconds. If not, the double tap gesture has to be used

again to initialize. This reduces the possibility of accidentally activate the communication.

If the arm is rotated to the left in time, the user can choose between several predefined programs/positions using different gestures. The following gestures are implemented for exemplary and (currently) theoretical tasks:

- wave left: the robot grabs material for the next working step,
- wave right: the robot grabs the completed work piece and puts it away,
- finger spread: the robot moves to its predefined home position.

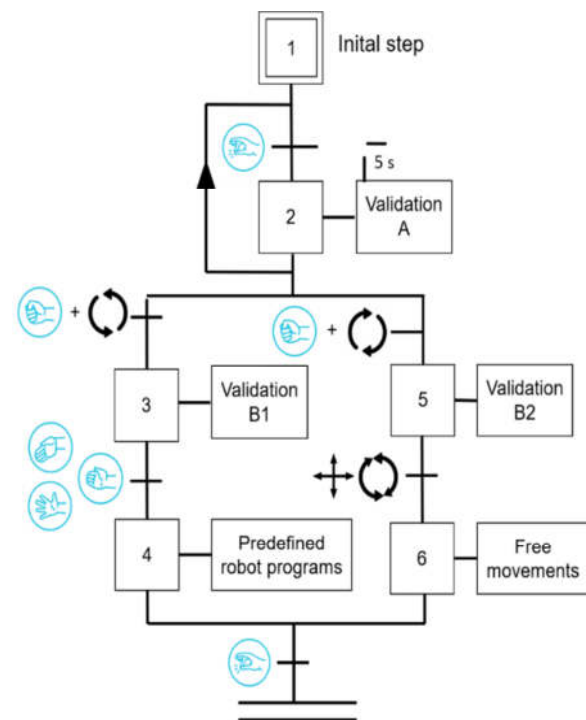


Fig. 6. Flow schematic for the Myo armband robot control peripheral

If the arm is rotated the other way around it is possible to guide the robot using the position angle of the arm. Therefore, a visualization interface is created to observe the actual arm position/angle. The robot moves upwards for example, when the forearm is at an angle of above 30 degrees to the top (pitch). If the arm is rotated to the left or to the right around the yaw axis, the robot is turning around its base, which enables that every possible position of the round table is reachable. Turning the forearm around its roll axis to the left or to the right, it is possible to move the robot along its tool centre point of the Z-axis. Again, to move the robot a minimum angle of the arm must be guaranteed. This concept avoids accidentally movements and compensates little inaccuracies of the armband. The velocity and acceleration of the robot are predefined and cannot be changed with higher angles of the arm position. By performing the double tap gesture the procedure ends. In this mode (initial step (1) as seen in

Figure 6) the Myo can be worn without moving the robot accidentally.

To calculate the position angle of the forearm an internal measurement unit, containing magnetometer, accelerometers and gyroscopes, which is integrated in the GC armband is used. Because all of these three sensors have different weaknesses and limitations a complementary filter is used to provide good results. Even in static situations accelerometers are sensitive to vibration and mechanical noise. Gyroscopes are not free from noise but less sensitive for linear motions as accelerometers and not influenced by external influences as magnetometers [15].

For high-dynamic situations the gyroscope is used to smooth the signal and filter the noise from the other sensor. Because the gyroscopes measure the angular velocity, the measured signal needs to be integrated to calculate the actual angle-offset from the start. This result to, that the additive error will rise after a few seconds due to the integration, which will make the computed angle unusable. That is why a sensor fusion is necessary [15].

For static-situations the accelerometer is used to calculate the current tilt-angle using the gravity acceleration vector on its three axes. This enables the determination of the pitch- and roll-angle. To identify the yaw-angle the magnetometer, which measures the magnetic field on the earth, can be used. To avoid the mentioned disadvantages, a complementary filter is used to combine the accelerometer and magnetometer signals (for low frequency tilt angle calculation) with the gyroscope signals (for high-frequency tilt angle calculation) [15].

Another way to control the robot is realized by using a hand-free wearable rugged Android-Smart glass (Realwear - HMT1). This device allows the technician to navigate through a developed graphical user interface using voice commands and display the results (and the GUI in general) in a lens. This GUI is only visible when the user focus on the lens. Simultaneous the user sees the work area. By filling a form which covers all axis of the robot it is possible to reach all positions of the robot workspace. This procedure can also be used to perform teaching tasks. The entered waypoints can be stored in the database.

All three robot control methods are based on the developed SDF which allows nearly equal commands and programming structures. The following section describes the overall connection between control peripherals, server and robot motion.

3.4. Application and Testing

After developing the data format, the control methods are tested. The procedure of operating the robot is shown in Figure 7 as a flow chart.

First, the server (marked with red borders) is started. All necessary actions run on the server. This includes the connection to the robot, the connection to

the database, the conversion of the data format for the robot and the response for the user e.g. visualize a successful connection or the actual robot position with optical or haptic (e.g. vibration) perception.

Next, the GUI is started. In this interface the user is able to set the configurations and select the control peripheral. The GUI use optical visualization elements to call attention of the user e.g. display a red label when robot is moving. Even an acoustic signal is possible. The user is able to select the control of the robot with voice or gesture and unlock the signals to the server of the selected peripheral. This action disables the GUI-robot control functions except for a stop command and disconnecting the selected peripheral. The VC uses optical visualization and acoustic signals to guide the user through the control. Obviously, the GC cannot use an optical visualization. A haptic perception is implemented (vibration) and an acoustic signal is conceivable.

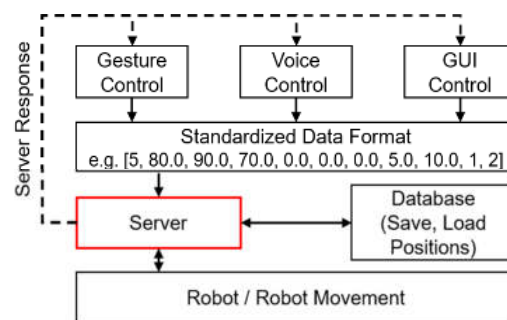


Fig. 7. Flow chart of possible robot controls

Testing the SDF on the robot (UR5) showed a difficulty concerning to the input of distances which are discussed in the following. The robot has, due to its construction, a certain maximum arm length. As described, in the data format for moving the robot relative to its actual position, the user gives a distance value to move. If the value is outside of the limits of the robot an error occurs. When this error occurs, the system divides the input value by two and try again to move the robot. This loop is done as long as the value is not small enough (see Figure 8).

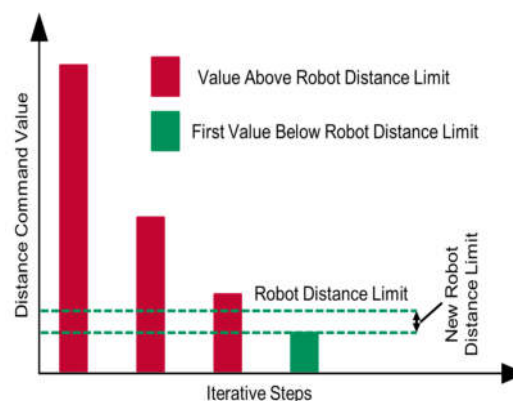


Fig. 8. Illustration of distance error handling for too big user distance value inputs

This means – theoretically – that the robot would move (near the limits of the robot) in infinitesimal small steps to the limit. In practice the motor current is too small to move the robot.

4. CONCLUSION AND OUTLOOK

The paper presents a new designed multi-workstation assembly table for different tasks in combination with manual assembly, HRC, automated assembly and robot-robot collaboration. Due to the stable construction, regarding ergonomic aspects, all four possible users are able to assemble even heavy components without any harm to their health. Additionally, the aluminium profiles allow a light weight construction and (if necessary) a fast rebuild to other rooms or something similar. The electric drive offers the possibility to rotate the table top to realize a kind of a circular assembly line. For further researches, the City University of Applied Sciences plans to add different workstation concepts to the MWAT. This possibility results from the aluminium profiles, which allow the attachment of several devices.

In the second part of this paper a robot control SDF is presented. It has been shown, that a control is done with three different peripherals (GUI, Gesture, Voice). The first application steps showed, that the data format is easy to modify or customize for different tasks. Nevertheless, difficulties have to be solved with detailed functions from the server side.

With this (or similar) SDFs a lot of work can be saved, when new peripherals or (robot) functions are needed.

The project offers a lot of possibilities to adapt and improve the assembly table and the data format. New acquired components, sensors and actuators allow nearly any limits to ideas for researches in this workspace. Therefore, a fourth robot control peripheral, which shall be realized by a depth camera, is already in planning. Here also the SDF shall be used and may be modified for this. For the UR5-robot which is actual in use at the City University of Applied Sciences a two-finger gripper and a force-moment sensor is acquired. The associated functions must be considered in the data format in further projects.

Further projects for robot safety as for instance collision detection with joint motor current monitoring in combination with machine learning are in progress, to allow a fenceless human robot collaboration.

Acknowledgements

The authors want to thank the City University of Applied Sciences Bremen especially, because the support was given in all respects. Special thanks to Mr. Felix Riedel who was a great support for this

paper. Finally, special thanks to all students who helped in this project.

Nomenclature

Symbols

<i>FD</i>	– Footwell Depth, mm
<i>LD</i>	– Legroom Depth, mm
<i>TD</i>	– Table Depth, mm
<i>TH</i>	– Table Height, mm

Acronyms

GC	– Gesture Control
GUI	– Graphical User Interface
HRC	– Human-Robot-Collaboration
ISO	– International Organization for Standardization
MWAT	– Multi-Workstation Assembly Table
SDF	– Standardized Data Format
UR5	– Universal Robots 5 (Robot in use)
VC	– Voice Control

References

- Hoffman, G., Breazeal, C. (2004) Collaboration in Human-Robot Teams. In: AIAA 1st Intelligent Systems Technical Conference. 20 - 22 September 2004, Chicago, Illinois. DOI: 10.2514/6.2004-6434
- Vasconez J. P., Kantor G. A., Auat Cheein F. A. (2018). Human-robot interaction in agriculture: A survey and current challenges. In: Biosystems Engineering Volume 179, March 2019, Pages 35-48.
- Teiwes J., Bänziger T., Kunz A., Wegener K. (2016). Identifying the potential of human-robot collaboration in automotive assembly lines using a standardised work description. In: 2016 22nd International Conference on Automation and Computing (ICAC). DOI: 10.1109/ICAC.2016.7604898
- Thomas C., Matthias B., Kühlenkötter B. (2016) Human-Robot-Collaboration – New Applications in Industrial Robotics. In: International Conference on Competitive Manufacturing
- Tsarouchi P., Matthaiakis A.-S., Makris S. Chryssolouris G. (2016). On a human-robot collaboration in an assembly cell. In: International Journal of Computer Integrated Manufacturing Volume 30, Pages 580-589
- Maurtua I., Ibaruren A., Kildal J., Susperregi L., Sierra B. (2017). Human-robot collaboration in industrial applications: Safety, interaction and trust. In: International Journal of Advanced Robotic Systems July-August 2017: 1–10. DOI: 10.1177/1729881417716
- Mohammed A., Schmidt B., Wang L. (2016). Active collision avoidance for human-robot collaboration driven by vision sensors. In: International Journal of Computer Integrated Manufacturing. Volume 30, 2017. Pages 970-980
- City University of Applied Sciences. [Online] Available: <<http://www.fbm.hs-bremen.de/modul/beschreibung.aspx?modul_id=699>> Accessed on: 12.04.2019
- City University of Applied Sciences. [Online] Available: <<https://www.hs-bremen.de/internet/de/forschung/projekte/detail/index_65597.html>> Accessed on: 12.04.2019
- Robyn. MacSources. 03.2015 [Online] Available: <<<https://macsources.com/myo-armband-available-on-amazon/>>> Accessed on: 12.04.2019
- Realwear, „HMT-1 Bedienungsanleitung (Deutsch) Version: 2“. [Online]. Available: <<https://www.realwear.com/pdf/HMT1_UG_German.pdf>> Access on: 12.04.2019

12. Bullinger H. J. (1994). *Ergonomie – Produkt und Arbeitsplatzgestaltung*. Vieweg + Teubner Verlag, DOI: 10.1007/978-3-663-12094-0
13. A. Sacco. IDG Communications, Inc. [Online] Available: <<<https://www.cio.com/article/2935585/tha-lmic-labs-myo-review-gesture-control-armband.html>>> Accessed on: 12.04.2019
14. Hettig J. Verwendbarkeit von myoelektrischer Gestensteuerung im Operationssaal. [Online] Available: <<http://isgwww.cs.uni-magdeburg.de/cas/pub/Masterarbeit_Julian_Hettig.pdf>> Accessed on: 12.04.2019
15. Patonis P., Patias P., Tziavos I. N., Rossikopoulos D., Margaritis K. G. (2018). A Fusion Method for Combining Low-Cost IMU/Magnetometer Outputs for Use in Applications on Mobile Devices. In: *Sensors* (Basel, Switzerland) vol. 18,8 2616. 9 Aug. 2018, DOI:10.3390/s18082616

Biographical notes



Daniel Brand decided in 2014, after his successful training as a carpenter, to start studying Aerospace Technology at the City University of Applied Sciences Bremen. During his studies he worked as an assistant student at the “Bremer Institut für angewandte Strahltechnik” (BIAS) and wrote his thesis in the field of laser chemistry.

After graduating with a B.Eng. degree in 2017, he immediately began his M.Sc. studies in Aerospace Technology at the City University of Applied Sciences Bremen and specialized in the space sector. He also works as an assistant student at the university itself. Graduation is expected at the beginning of 2020.



Niels Benthin received his B.Sc. degree in Mechanical Engineering with a specialization in “Development and Design” at the University of Applied Sciences Lübeck. At present, he is enrolled at the Department of Nature and Engineering at the Hochschule Bremen City University of Applied Sciences and expects to attain his

M.Eng. degree in Mechanical Engineering by 2020. Currently, he is collaborating on a research project about the digitization of production together with fellow students and research assistants. Specifically, the focus of the project is on robotics and digital production planning - from concept development through to practical implementation.



Marvin Bohl received his B. Eng. degree in Mechanical Engineering from Hochschule Bremen, City University of Applied Sciences, in 2018. He is currently in the process of preparing his master thesis in cooperation with ArianeGroup. Before studying he completed a professional training as an industrial mechanic and worked in this profession for more than

six years. Besides the studies he has been a research assistant at the German Research Center for Artificial Intelligence (DFKI).