

51st CIRP Conference on Manufacturing Systems

Assisted assembly process by gesture controlled robots

Tamás Csértég^a, Gábor Erdős^{a,b}, Gergely Horváth^{a,b,*}^aInstitute for Computer Science and Control, Hungarian Academy of Sciences, Budapest 1111, Hungary^bDepartment of Manufacturing Science and Engineering, Budapest University of Technology and Economics, Budapest 1111, Hungary^{*} Corresponding author. Tel.: +36 1 279 6181; E-mail address: gergely.horvath@sztaki.mta.hu**Abstract**

During repetitive tasks, human worker is prone to inaccuracies like switching up assembly steps. These errors can be prevented by a robot handing over the appropriate tools or workpieces, indicating the forthcoming step in the process plan. A general handover model has been specified, taking into consideration fundamental safety measures, as well as basic comfort requirements. By utilizing *gesture communication*, we established a method, that is capable of dynamically negotiating the necessary conditions of the handover task. The proposed approach has been evaluated in laboratory environment, supporting its technical feasibility.

© 2018 The Authors. Published by Elsevier B.V.

Peer-review under responsibility of the scientific committee of the 51st CIRP Conference on Manufacturing Systems.

Keywords: gesture control, cyber physical system, human robot collaboration**1. Introduction**

Although automation and robotization is a key component in successful production, human-robot hybrid assembly systems have numerous advantages over fully automated systems [1]. Automation does not fulfill all the expectations attributed to it for some companies. Reasons include the inflexibility of these facilities with respect to changing lot sizes, increasing complexity, shorter product life cycle or other conditions. Another group of reasons stems from economical measures. These include initial investment, complete production cost of highly automated production and inflexibility regarding new production schemes. It is clearly shown that hybrid assembly lines, humans and robots working cooperatively, lead to the increased efficiency where robots can assist the human worker [1].

To support the design of such collaborative workplaces, multiple factors have to be taken into consideration. One such factor is the information transfer between the human and robot member of a workcell. *Gesture recognition* is a widely used communication interface between human operators and computers. Wang and Liu discuss gesture recognition generally and its use in human-robot collaboration in their review [2]. They define gesture recognition as the mathematical formulation and capturing of the human motion, using a computational device. Robots have to interpret human gestures and act accordingly in order to be able to work in collaboration with human operators. The gesture recognition model, based on the general human information processing model has 5 major component such namely: *data collection*, *gesture identification*, *gesture track-*

ing, *gesture classification* and *gesture mapping*.

In an earlier work [3] this structure has already been used, where a gesture control system, regarding CPPS has been presented. Current paper is a follow-up research for [3], making it the starting point of the system presented in the current paper, albeit other papers also address the problem of human-robot collaboration (e.g. [4,5]). As the method utilized only static gestures, *gesture tracking* has been omitted. Another difference is the extension of *data collection* with a preprocessing phase.

Erden et al. [6] worked on a 2D tool handover scenario. They used a multi-agent approach, where fuzzy control have been applied to the different robot parts. Although the solution seems promising, it lacks any real word implementation and its hand following ability in 3D implementation wasn't shown.

A recurring problem during human-robot collaboration is the mental stress effecting the human operator. Both [7,8] discuss the possible issues with workcells considering the well being of the human operator. Both list a number of details to consider when setting up HRC work places, like maximum speed of the robot, or avoiding the unexpected movement of the robot.

Strabala et al. [9] made a general model of object handover. During the realization of our system we used this model. The details are laid out in Section 4.

In this paper we present a general model and a proof of concept implementation on human-robot tool handover scenario. We base our model on the one presented in [9], eliminating the need to preset the position of the handover. To achieve these goals we utilize *gesture communication*.

2. Problem statement

A collaborative robot cell with a 3D capable depth sensor, and a virtual model of the cell as a CAD model are given. This CAD model is the so-called *as-design* model of the workcell. Equipment, tools or workpieces are laid out for the robot within its workplace. A method for precise pick and place is assumed to be available, therefore problems and difficulties of this task are not addressed in this article.

The focus of this paper is the seamless tool handover between a robot and a human operator. During the handover operation we expect the following:

- the robot follows the movement of the operators giving/receiving hand. Continuous following is favorable, as unpredictable robot motion may increase mental strain on the human operator [7].
- the robot approaches the operator in a non-intrusive manner, avoiding direct contact, still staying in arms-reach of the operator,
- leaving the initiation of the handover to the human operator,
- require the operator to commit to the handover to avoid dropping of tools or equipment,
- the robot takes trivial safety measures during the process (e.g. do not hit the operator, desk or other obstacles, do not drive itself into unresolvable joint states etc).
- during the process, gesture communication is realized.

To give a viable solution to the task at hand, the following workflow is followed:

- creation of a fully functional, digital workcell model,
- calibration of said model,
- realizing operator tracking and following,
- recognizing initiation of handover operation and acting on it.

3. Digital twin

An actualized digital representation of the workplace should be developed. This digital representation should provide the actual and accurate position of the objects in the surrounding of the robot, the position of the human operators and the interaction channel with the operator. This representation needs a virtual model capable of capturing the actual state of the workcell, with sufficient level of detail. Loading real time data to the digital model leads to the creation of the *digital twin* [10]. With the digital twin we have an always updated digital model, which is capable of receiving (using sensors, databases etc.) and distributing data (to databases, decision logic, controllers etc.) in the workcell.

The model of the robot workcell is created with LinkageDesigner [11], which is an add-on package for Wolfram Mathematica™. This package uses a graph representation for mechanism or linkage definition, where the nodes of the graph are the various links (or bodies) building up the linkage, while the edges between two nodes are joints, connecting the bodies corresponding to the given nodes. Joints are equations constraining the relative position and orientation of the links in

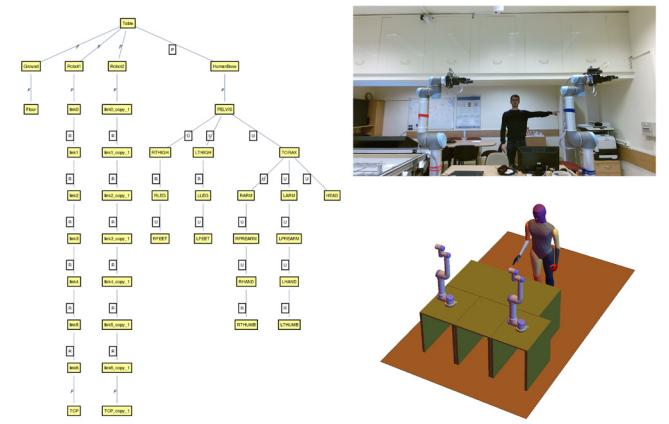


Fig. 1. Linkage graph (left), photo (top right) and virtual model (bottom right) of the workcell [3].

question. A pair of links and a joint between them is usually referred to as a *kinematic pair*. Such a graph can be seen on Figure 1. Linkage definition captures the following properties:

- Features of the links
 - link geometry (via 3D triangle mesh),
 - local link reference frame,
- Kinematic constraints between the links
 - fixed transformations,
 - parameterized variable transformations representing the moving kinematic pairs.

The information captured by the linkage definition is capable of modeling not only a single mechanism, but making it well suited for modeling a whole workcell. It is possible to model the robot, human operators, tables and other relevant equipment in the workcell. It is also possible to collect the relative positions of the objects in space, as joint data. Benefit of this approach is the potential to update these joints based on real world measurements. Incorporating sensor data, the digital twin can be updated and kept synchronized with reality, thus providing an invaluable base data for various planning and decision making processes. This virtual model is called the digital twin [10].

3.1. Calibration

Calibration itself deals with the difference of the reference CAD model and the reality [12]. Usually the calculation of joint and link dimension errors is considered as calibration, which corresponds to precise pick and move scenarios. On the other hand, calculation of transformation between the local reference frame and some pre-defined global reference frame can also be viewed as calibration.

Calibration of the digital twin—as stated in Section 3—is a key requirement. It means that the *as-design* and *as-build* states of the workcell need to be matched (see Figure 2). The reason behind the difference might be caused by multiple reasons, like cables, grippers or other equipment which may be excluded from the *as-design* model. Both models have their own local reference frame, which needs to be merged. This will give us an accurate virtual model—the digital twin itself—that contains the current, updated state of the workcell, capable of serving as

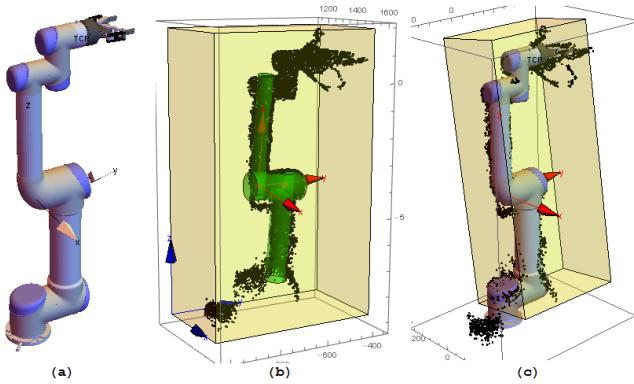


Fig. 2. CAD model and feature frame Frame_{CAD} (a), point cloud, branch cylinders and feature frame Frame_{PC} (b), aligned connected subset and CAD model (c) [12].

an input for any kind of workcell control.

4. Proposed system

For describing our approach the model created by Strašála et al. [9] has been chosen. They put forth, that any handover situation is based on the negotiation of the *what*, *when* and *where*. According to their model, previous approaches—where the robot brings the tool to a fix position—negotiated the *where* in advance, rather than delegating it to the participating parties. Examples of such method are [13,14]. In our concept, negotiation takes place immediately before the handover itself, making it more adaptive and comfortable.

In our case *what* is not negotiated by the participants of the handover scenario, because the robot has prior knowledge of the forthcoming step in the assembly sequence. The availability of this knowledge is based on [15]. For demonstration purposes in this paper, a human gesture is used to initiate the first pickup (subsequent ones started when handover of previous element is finished). Our previous work describes the used (static) gestures [3].

First the robot picks up the subsequent tool and starts to follow the operator's hand. The way the robot moves after the hand should be steady and unmistakable. The use of hand following is detailed in Section 4.1.

By initiating the hand following sequence, the robot signals its readiness to handover, so the operator can take the tool anytime. Using any additional signal would take away the fluency of the handover operation, therefore reaching toward the tool indicates that the operator wishes to accept the tool *now*.

In theory the robot should stop immediately, which is also the negotiation of *where*, but the exact position of the handover depends on the implementation.

Figure 3 shows our method in the context of the described model.

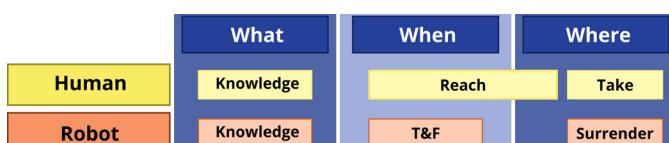


Fig. 3. The handover model (T&F means track and follow)

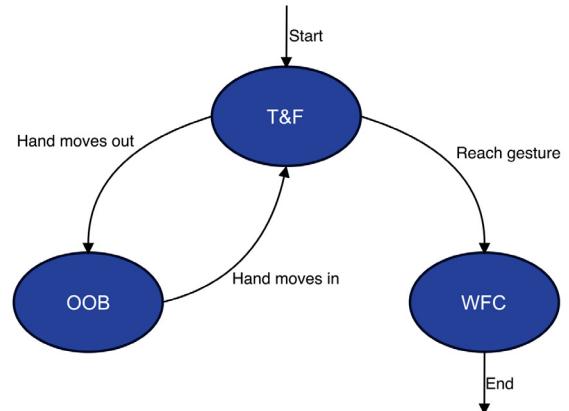


Fig. 4. State machine of the handover process

4.1. In depth description of model

Hand tracking and following has been chosen to be the basis of the handover process. The robot picks up the next tool/workpiece while the operator is working on the previous step of the assembly and starts to follow his hand. With the continuous following the given object is always within arm's reach of the human operator, shrinking the time frame of tool/workpiece choosing and picking. This approach is ergonomic in a way that, with the continuous movement, sudden starts/stops can be avoided, that may cut the worker's mental stress [8].

The following should happen only in the *shared workspace* between the operator and the robot (which is one of the basic assumptions of HRC). If the operator's hand leaves this area during the handover process (not the whole of his workspace is shared), it has to be indicated through a well defined signal. As two-third of human communication is non-verbal communication [16], special robot movement patterns can be used to communicate its state to the worker. With the gesture control this bilateral information transition is what we call *gesture communication*.

4.2. State machine representation of said model

The previously described model can be represented as a finite-state-machine (*FSM*), as can be seen on Figure 4. Three states are defined:

- Track and Follow (*T&F*),
- Out of Boundary (*OOB*),
- Wait for Commit (*WFC*).

The transitions between these states are also defined:

- between *T&F* and *OOB*:
 - moving out of the shared workspace,
 - moving in the shared workspace,
- between *T&F* and *WFC*:
 - moving the tracked hand towards the robot.

The handover process starts with picking up the proper tool,

then with starting the hand following (*T&F*) the robot enters the state machine. If the operator's tracked hand moves out of the shared workspace the state is shifted to the *OOB* state and vice versa. In this state a small, continuous robot activity is defined which shows the operator that his hand is out of the shared workspace (*gesture communication*). In *T&F* state dynamic gesture recognition is used to determine when the operator wants to take the tool. If his hand moves towards the robot, the state is shifted to *WFC*. The *WFC* state needed only for safety reasons, because the dropping of equipment/workpieces must be avoided, therefore after grasping it the operator has to apply a small force to the robot. After this last commitment the robot surrenders the tool, then exits from the state machine. Based on the assembly task sequence, it enters an idle state or picks up the upcoming object.

5. Proof of concept

Skeletal fitting algorithms may place more than one joint on the hand and for the *T&F* state one of them has to be selected. The criteria are the following: the joint has to be at the end of the hand and the signal's noise should be the lowest possible. Depending on the Signal-to-Noise Ratio (*SNR*), filtering of the joint's signal may be considered. The filtering algorithm is expected to smooth the original signal and having a low phase delay is key.

To achieve safe hand follow we can not drive the robot directly to the position of the hand. The robot's Tool Center Point (*TCP*) has to be shifted with a convenient distance from the followed joint.

The aim is that the robot stops, if the tracked hand moves towards it. The operator's intent is indicated by his hand movement—i.e. the speed of his hand. The speed vector and the spatial vector that directs from the hand to the gripper have to be compared. If these two vectors are co-directional within sensible threshold, it means that the operator wants the tool and the state can shift to *WFC*.

The entry action of *WFC* is to stop the robot. However if it only halts, it is unclear that an error occurred or stopped because reach gesture has been recognized. Therefore a small robot gesture is defined which suggests that the operator can take or give the tool (*gesture communication*).

After this gesture the robot waits for the last commitment, overcoming a force threshold. The operator has to hold the tool to apply the given force on the robot, thereby dropping the tool can be avoided.

5.1. Implementation

The sample workcell consists of an UR5 robot and a Microsoft Kinect sensor connected to a workstation computer. The robot is about 1 m away from the Kinect sensor, while the human operator is about 1 m more away from it. The sample task was for the robot to hand over different cubes (in size and colour) symbolizing different tools. Their sequence and pick up position was previously defined.

First step of implementation is the creation and calibration of the digital twin. As described in Section 3 the model itself is created in Wolfram Mathematica™, using the LinkageDesigner package. Calibration was done with a captured pointcloud of

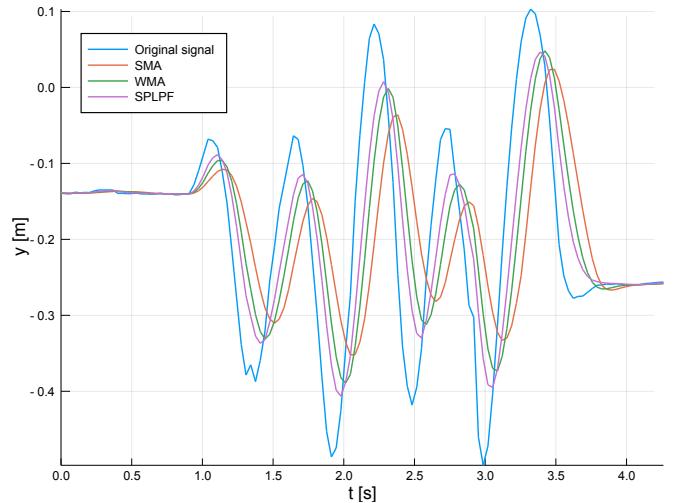


Fig. 5. y coordinate of the *hand* joint during the experiment

the Kinect sensor which was merged with the CAD model (see Figure 2). The result of the calibration is the exact coordinate transformation between the robot's and sensor's coordinate system. To keep the digital twin always up-to-date, the current state of the robot and human operator have to be queried. The actual status of the robot can be acquired by *Real Time Data Exchange* (RTDE) protocol designed by the manufacturer of the robot. A Python script using a third-party library [17] handles this protocol and sends the for the digital twin. The position of the human operator is updated exploiting its skeletal model, provided by the Kinect sensor.

The skeletal tracking algorithm—provided by the manufacturer—places three joints to the end of the arm, these are: *wrist*, *hand* (palm) and *thumb*. The accuracy of the algorithm is closely tied to the accuracy of the sensor, which means it can always be boosted by utilizing higher accuracy sensors. As proprietary skeletal tracking method is used (the official Kinect2 SDK), the pros and cons of it are beyond the scope of this paper. Although it was sufficiently reliable for our proof of concept implementation, it can be arbitrarily substituted with more suitable systems if the need arise. It is also worth mentioning that, our proof of concept does not really require accuracy higher than 1 cm, which can be easily satisfied with the Microsoft Kinect sensor. To fulfil the mentioned requirements—Section 5—an experiment was carried out, where the operator was moving his hand while the signal of the three joints was recorded. To determine the most suitable joint, standard deviation of every series is compared, because its magnitude ties in with the noise. Standard deviation uses the mean of the given series, but in this case by reason of the dynamic move it would give faulty result. Instead of the mean value, central moving average is used because it shows the mean of a dynamic function at the given point. That joint have been chosen which has the lowest standard deviation, in our case that is the *hand* joint.

Because of the noisy sensor signal a filter should be used to suppress the high frequency noises, therefore low-pass and moving average filters were considered. First we chose the algorithm depending on the phase delay, then tuned that filter to match the noise cancellation goal. We tested single pole low-pass filter (SPLPF), simple and weighted moving average

(SMA and WMA) algorithms on the *hand* joint's data series and experienced that SPLPF has the smallest delay. Therefore we tuned it to have the lowest noise possible. Figure 5 shows the phase delay difference between the different algorithms.

For shifting the position of the robot TCP—with respect to the hand of the operator—both the direction and distance of the shifting have to be determined. During direction specification it is assumed, that the operator is executing the assembly process in front of him. We noticed, that the operator prefers to have the robot slightly on the side of his field of view, rather than in front of him. To satisfy this need, the TCP is shifted towards the robot (from the hand) and the distance is chosen to be constant 0.2 m. On Figure 7 the realized shifting can be seen. The robot moves directly to this shifted position therefore the above mentioned noise filtering also smooths the robot's path.

The state machine and every Kinect related part are implemented in the digital twin itself, running on the workstation. The UR5 robot is programmed in its own programming language called UR script. The digital twin controls the robot in a master-slave relation and their communication is established over TCP/IP protocol. The flow chart of the implementation can be seen on Figure 6.

For completing the requirements of the gesture communication, a gesture is defined for the *OOB* state. We chose a continuous back and forth move along one horizontal axis, which suggests to the operator that the robot can not go past a virtual vertical plane.

For the last state transition, dynamic gesture recognition is used to determine if the direction of the hand's speed vector and the spatial vector that directs from the hand to the gripper are co-directional. For speed calculation we used a smooth noise-robust differentiator algorithm described in [18]. The robot's actual TCP position is acquired by the aforementioned RTDE protocol. After normalizing the vectors, their scalar product is calculated for the comparison. If the result is near 1—larger than a threshold value, which we chose to be 0.9—then the *gesture identification* part of the sequence is done.

Gesture tracking is needed to validate the operator's intention and filter possible noise from the derivation. For this purpose a clear up counter unit is used. The responsiveness and the noise-robustness can be tuned with the threshold value. In our case the value is 4 which gives approximately 0.13 second delay—with this Kinect sensor—between the start of the gesture and the reaction to it. *Gesture classification and mapping* are merged into one step. If the threshold is reached, the state shifts to *WFC*. For entry action the robot stops and with a small, single move shows that the state has been changed. After stopping the robot waits for a last commitment from the operator. The operator has to act a force—larger than a threshold—on the tool wherewith he ensures that he has seized the tool. When it happens, the *WFC* state ends, and as an exit action the robot releases the tool, moves back in its idle pose or picks up the next tool.

Figure 7 shows a capture of the *T&F* state of the handover process. A larger cube can be seen in the gripper of the robot and other smaller, coloured ones on a palette. The implementation meets all expectations laid down in Section 2.

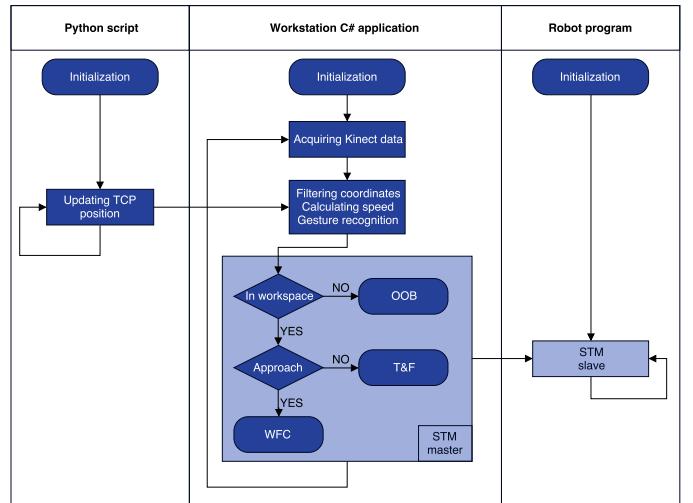


Fig. 6. Sequence diagram

5.2. Latency evaluation

In HRC scenarios low latency is key, therefore we evaluated our implementation in this view. We measured an average 3 millisecond communication time between the digital twin and the robot. A timer was started after acquiring new sensor data then the processed data was sent to the robot. The timer stopped when an acknowledge message arrived from the robot. That way we measured a 6 millisecond delay half of which was taken as communication delay. With the Kinect sensor's 30 Hz sampling frequency this gives a worst case 36 millisecond reaction time.

We also recorded how fast the robot reached a given position. The operators followed hand and the robot's position was recorded simultaneously as can be seen on Figure 8. Based on this measurement an average 104 millisecond has been determined as moving delay. As it does not contain the processing time the overall hand following delay is 140 millisecond.

As described above, most of the delay depends on the robot's configuration (speed, acceleration) and the sensor's sampling frequency. These are implementation specific attributions and can be improved with other devices.

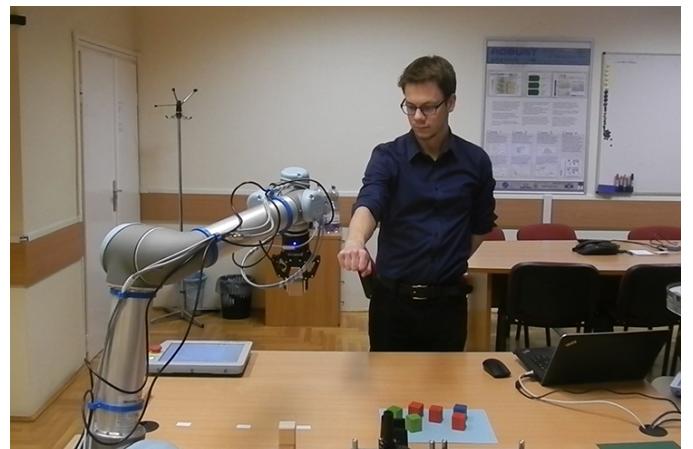


Fig. 7. Picture of the workcell during handover process

6. Conclusion and future work

The use-case in Section 5.1 demonstrates the feasibility of the proposed method, realizing a non-obtrusive, robot assistance workcell, for human-robot collaboration. Testing it with multiple, untrained user, the feedback was mostly positive regarding the user-friendliness of the system. We used virtual sensors on the digital twin of the workcell to follow the movement of the proper joint of the human operator. Further virtual sensors could be added, for instance a collision detection between the robot and the model of the workpieces, or the captured image of non-modeled objects in the workcell.

The system also can be extended with other *T&F* operations, based on the knowledge of the task sequence—e.g. hovering a heavy object over its intended position, while the operator executes sub-assembly tasks.

In our proof of concept we did not implement collision detection because the shared workspace was empty. However in the *T&F* state the robot movements are not predefined, which means our model should be extended with this functionality. It is possible that in cluttered workcells this can be a resource intensive calculation, which can increase latency between detection and movement.

7. Acknowledgement

This research has been supported by the GINOP-2.3.2-15-2016-00002 grant on an "Industry 4.0 research and innovation center of excellence". This research has also been supported by the EU H2020 Grant SYMBIO-TIC No. 637107.

References

- [1] Bley, H., Reinhart, G., Seliger, G., Bernardi, M., Korne, T.. Appropriate human involvement in assembly and disassembly. *CIRP Annals* 2004;53(2):487 – 509. doi:[https://doi.org/10.1016/S0007-8506\(07\)60026-2](https://doi.org/10.1016/S0007-8506(07)60026-2).
- [2] Liu, H., Wang, L.. Gesture recognition for human-robot collaboration: A review. *International Journal of Industrial Ergonomics* 2017;doi:<https://doi.org/10.1016/j.ergon.2017.02.004>.
- [3] Horváth, G., Erdős, G.. Gesture control of cyber physical systems. *Procedia CIRP* 2017;63(Supplement C):184 – 188. doi:<https://doi.org/10.1016/j.procir.2017.03.312>; manufacturing Systems 4.0 Proceedings of the 50th CIRP Conference on Manufacturing Systems.
- [4] Tsarouchi, P., Athanasatos, A., Makris, S., Chatzigeorgiou, X., Chryssolouris, G.. High level robot programming using body and hand gestures. *Procedia CIRP* 2016;55:1 – 5. doi:<https://doi.org/10.1016/j.procir.2016.09.020>; 5th CIRP Global Web Conference - Research and Innovation for Future Production (CIRPe 2016).
- [5] Tsarouchi, P., Matthaiakis, S., Makris, S., Chryssolouris, G.. On a human-robot collaboration in an assembly cell 2016;30:1–10.
- [6] ERDEN, M.S., LEBLEBICIOGLU, K., HALICI, U.. Multi-agent system based fuzzy controller design with genetic tuning for a service mobile manipulator robot in the hand-over task. *IFAC Proceedings Volumes* 2002;35(1):503 – 508. doi:<https://doi.org/10.3182/20020721-6-ES-1901.00896>; 15th IFAC World Congress.
- [7] Michalos, G., Makris, S., Tsarouchi, P., Guasch, T., Kontovraklis, D., Chryssolouris, G.. Design considerations for safe human-robot collaborative workplaces. *Procedia CIRP* 2015;37(Supplement C):248 – 253. doi:<https://doi.org/10.1016/j.procir.2015.08.014>; cIRPe 2015 - Understanding the life cycle implications of manufacturing.
- [8] Arai, T., Kato, R., Fujita, M.. Assessment of operator stress induced by robot collaboration in assembly. *CIRP Annals* 2010;59(1):5 – 8. doi:<https://doi.org/10.1016/j.cirp.2010.03.043>.
- [9] Strabala, K., Lee, M.K., Dragan, A., Forlizzi, J., Srinivasa, S., Cakmak, M., et al. Towards seamless human-robot handovers. *Journal of Human-Robot Interaction (JHRI)* 2013;2.
- [10] Rosen, R., von Wichert, G., Lo, G., Bettenhausen, K.D.. About the importance of autonomy and digital twins for the future of manufacturing. *IFAC Papers Online* 2015;48(3):567–572.
- [11] Erdős, G.. Linkagedesigner, the mechanism prototyping system website. <http://www.linkagedesigner.com>; 2005. Accessed: 2016-12-30.
- [12] Horváth, G., Erdős, G.. Point cloud based robot cell calibration. *CIRP Annals* 2017;66(1):145 – 148. doi:<https://doi.org/10.1016/j.cirp.2017.04.044>.
- [13] Huber, M., Lenz, C., Rickert, M., Knoll, A., Brandt, T., Glasauer, S.. Human preferences in industrial human-robot interactions. *Proceedings of the International Workshop on Cognition for Technical System 2008*;
- [14] Cakmak, M., Srinivasa, S.S., Lee, M.K., Forlizzi, J., Kiesler, S.. Human preferences for robot-human hand-over configurations. *IEEE International Conference on Intelligent Robots and Systems* 2011;:1986–1993doi:10.1109/IROS.2011.6048340.
- [15] Kardos, C., Kovács, A., Váncza, J.. Decomposition approach to optimal feature-based assembly planning. *CIRP Annals* 2017;66(1):417 – 420. doi:<https://doi.org/10.1016/j.cirp.2017.04.002>.
- [16] Hogan, K., Stubbs, R.. Can't get through : eight barriers to communication. Pelican Pub. Co.; 2003. URL: <http://www.worldcat.org/oclc/51769254>.
- [17] Roulet-Dubonnet, O.. Urx python library. 2012. URL: <https://github.com/SintefRaufossManufacturing/python-urx>; accessed: 2017-10-30.
- [18] Holoborodko, P.. Smooth noise robust differentiators. <http://www.holoborodko.com/pavel/numerical-methods/numerical-derivative/smooth-low-noise-differentiators/>; 2008.

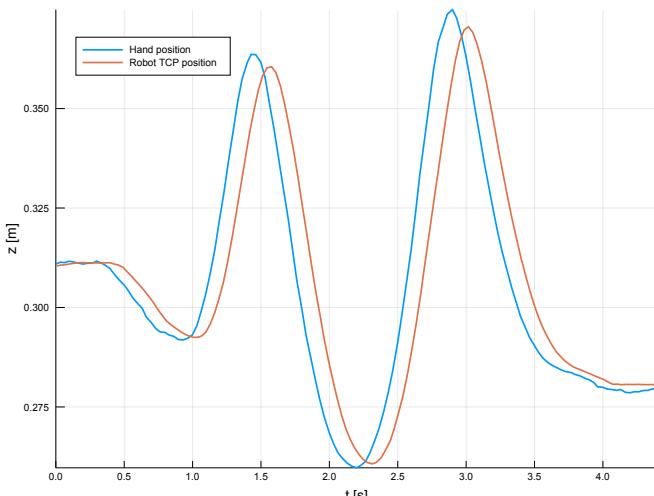


Fig. 8. Recorded robot and operator hand positions