2022 IEEE/SICE International Symposium on System Integration (SII)
January 9-12, 2022. Narvik, Norway

2022 IEEE/SICE International Symposium on System Integration (SII) | 978-1-6654-4540-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/SII52469.2022.9708896

Integration of a reconfigurable robotic workcell for assembly operations in automotive industry

Bojan Nemec, Matija Mavsar, Mihael Simonič, Matevž Majcen Hrovat, Jure Škrabar+, Aleš Ude

Abstract— **This paper deals with the integration of a flexible, reconfigurable work cell performing assembly of parts in the automotive industry. The unique feature of the developed cell is that it can function in two modes: a) entirely autonomously or b) in cooperation with a human, where the operation of the robot dynamically adapts to human actions. We have implemented technologies for online recognition of human intention and for real-time learning of robust assembly policies to achieve the desired outcome. This challenging goals dictate the integration of modern deep learning algorithms, statistical learning, and compliant robot control into a unique ROS-based robot control system.**

I. Introduction

Flexibility supported by fast reconfiguration ability and human-robot collaboration (HRC) is among the prominent topics of the emerging Industry 5.0 agenda [1]. One of the important goals is to implement collaborative production processes for small lot size production, where a production environment cannot be prepared for the desired task. Consequently, classical industrial robotics approaches that rely on carefully prepared production environments are not applicable for this type of production. We propose to make small lot size production economically viable by introducing a flexible multi-purpose robotic cell, in which human workers can participate when necessary. The motivations for human participation in industrial production cells are diverse. For example, some operations are too demanding for a robot and can therefore be more efficiently performed by a human worker. Even if the robot can perform the task on its own, it might still occasionally require human help, either when solving unforeseen situations or when it is beneficial to perform the task in collaboration with the human worker, thus avoiding a bottleneck in the production process. In our work, we focus on collaborative problem solving, where the robot learns from a human worker how to deal with unforeseen situations. By acquiring new knowledge, the robot can become more and more autonomous over time. Another guideline in our cell design is easy deployment, aiming at decreasing the setup time and lower the required skill level of the operators. To this end, we utilize self-contained hardware modules [2] and modular software architecture [3].

All of the above technologies have been the subject of intensive research in the past [4], [5]. Regardless of this, their implementation in production processes remains a challenge [6]. Recently, we have witnessed successful applications of reconfigurable robotic cells [7], [8] and the introduction of intelligent, collaborative systems [9] into production. This paper describes the integration of advanced technologies of human-robot collaboration [10], learning from demonstration [11] and exception strategy learning into a flexible and reconfigurable robotic work cell.

In Section II, we explain the application of a flexible robotic cell in a production process that requires human-robot collaboration, such as the automated assembly of automotive components. The main technological issues include learning of exception strategies and accomplishment of human-robot collaboration, which are presented in Section III and IV, respectively. One of the important issues addressed here is the online determination of the human worker's intent. The overall system integration is described in Section V. Our future work is finally explained in Section VI.

II. Problem description

Many industrial production processes cannot be fully automated. One example is the assembly of car starters (see Fig, 1). In this task, the assembly process starts with the insertion of copper sliding rings into metal molds, which are then transferred to the casting machine. Currently, the insertion of sliding rings into the molds is performed manually and is considered hard to automate. Besides inserting the sliding rings, the human operator simultaneously performs also other production tasks. Initially, the sliding rings are distributed on a plate at random positions and orientations, see Fig.1 left. During the insertion into the mold (see Fig. 1 center), the operator has to take care of the correct orientation of the contact "wings" of the copper sliding rings. When the casting molds are filled with all four sliding rings, the robot transports the whole mold pallet to the casting machine. The finished part is shown in Fig. 1 right.

The assembly process is demanding due to the high flexibility and elasticity of the sliding rings. Previous automation attempts failed for two reasons: 1) it is sometimes not possible to grasp the rings because they are often stuck together on the transport plate; 2) it is not possible to assure the required success rate of insertion due to the flexibility and elasticity of the rings.

Our aim was thus to develop a robotic system that operates autonomously or together with the operator. The operator and the robot execute the same operations; the work-sharing is assigned dynamically. The task of the operator and the robot is to grasp the rings from the transport pallet and

Humanoid & Cognitive Robotics Lab, Department of Automatics, Biocybernetics and Robotics, Jožef Stefan Institute, Ljubljana, Slovenia, e-mail: {bojan.nemec, matija.mavsar, mihael.simonic, matevz.majcen, ales.ude}@ijs.si
+Kolektor Group d.o.o., Idrija, Slovenia e-mail: {jure.skrabar@kolektor.si

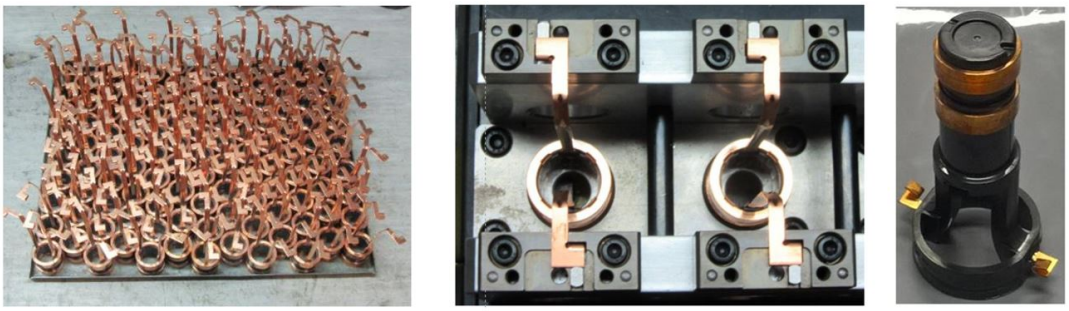

Fig. 1. Left: Sliding rings placed on a transport plate. Center: Sliding rings correctly inserted into the pallet. Right: The finished part after injection molding.

insert them on the casting pallet. The human co-worker also supervises the assembly performed by the robot and corrects it if necessary. Based on these corrections, the robot gradually improves its operation and becomes more reliable. Finally, the human operator can leave the cell as their help is not needed any more.

The developed work cell is shown in Fig. 2. It is based on a modular design where each module shares the same archetypical design: a steel frame that provides rigidity, with an aluminum work surface on top that enables easy mounting of module-specific equipment, e.g., robots, sensors, and auxiliary devices [2]. Modules are connected with Plug-and-Produce (PnP) connectors that provides informational connectivity via Ethernet, mechanical coupling, and electric and pneumatic power supply. In this work cell, we integrated two archetypical modules. The first module supports 7 degree of freedom collaborative robot Franka-Emika Panda with control computers located in the module's base. It is attached to the module equipped with sensors and cameras to support the assembly process. Two cameras located on the pillar monitor the motion of a human collaborator. The first is the motion tracker OptiTrack Trio and the second is the Intel RealSense D435i RGB-D camera. A linear stepper drive is installed on the work surface of the module. It is used for positioning another Intel RealSense RGB-D camera, which is used to supervise inserting sliding rings into the casting molds.

Placed along both modules is a conveyor belt that transports pallets with casting molds. A tray with sliding rings is located on the right. It is mounted on a passive reconfigurable fixture in the form of the Stewart platform. The robot is in charge of moving this parallel mechanism. The robot adjusts the position of the mechanisms each time a new human worker starts working in the cell. In this way, we make sure that the tray is accessible to the robot and, at the same time, provide an ergonomically optimal position for the human co-worker. Algorithms for calculating the optimal position of the passive mechanism are presented in [12].

The initial robot trajectories are specified by kinesthetic teaching and refined by Iterative Learning Control (ILC). They are encoded with dynamic motion primitives (DMPs) [13]. Please refer to our previous work [14] for the details of this learning framework. Even though the robot has carefully learned and optimized the trajectories, the insertion of rings into molds is often not successful. There are several reasons for this. Firstly, the insertion tolerances are very small, below 0.1 mm, which the robot often does not achieve due to uncompensated and configuration-dependent friction in its joints. Another problem are the gripping tolerances and deformations of the sliding rings. We cannot predict all unforeseen situations in advance. Therefore we have developed a procedure where the operator helps the robot to resolve unforeseen situations. The robot learns new actions from this process and eventually becomes more and more autonomous.

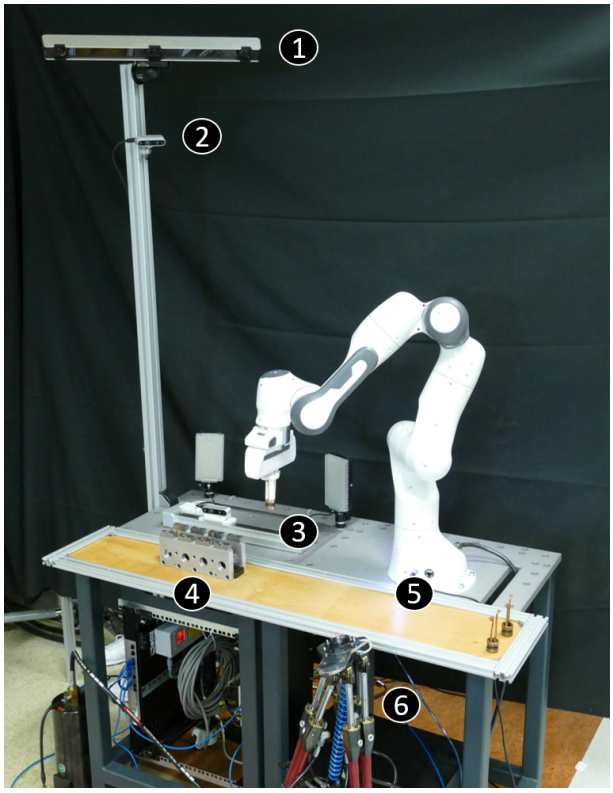

Fig. 2. Modular collaborative assembly cell is composed of 1) motion tracker, 2) overhead human surveillance camera, 3) linear drive with assembly control camera, 4) casting molds, 5) collaborative robot, and 6) passive reconfigurable fixture (the transport tray for sliding rings is not mounted on the fixture in this picture).

III. Exception strategy learning

The general idea of learning an exception strategy is that when an unforeseen situation occurs, the operator resolves the situation with the manual guidance of the robot. In doing so, the robot remembers the context and the demonstrated action. Context is built from sensory information that is relevant to a particular application. Once a robot has defined a sufficiently large database consisting of contexts and associated actions, it can use statistical learning to predict appropriate action to resolve an unforeseen situation based on the current context. Note that it is essential to define the context as a low-dimensional entity to apply statistical learning successfully. More details of this framework are in [15].

We used this concept to solve errors when inserting a sliding ring into a casting mold. In our case, there are two major types of errors. The first type is when the base of the ring is not properly seated into the model (See Fig. 3) left. The second type of error occurs when the sliding ring wings have not properly seated the mold as shown in Fig. 3 right. Both types of error can be reliably determined by observing the z coordinate of the robot. In the first case, we determine the context based on the ring base's x and y positions. The position of the ring base is obtained by image segmentation of the RGB-D camera. We applied the state-of-the-art YOLACT [16] image segmentation algorithm based on a deep neural network to determine the y coordinate, as shown in the Fig 3 left. The x coordinate can be estimated from the depth information of the camera. The context is thus composed of vector $\zeta_1 = (x, y)$.

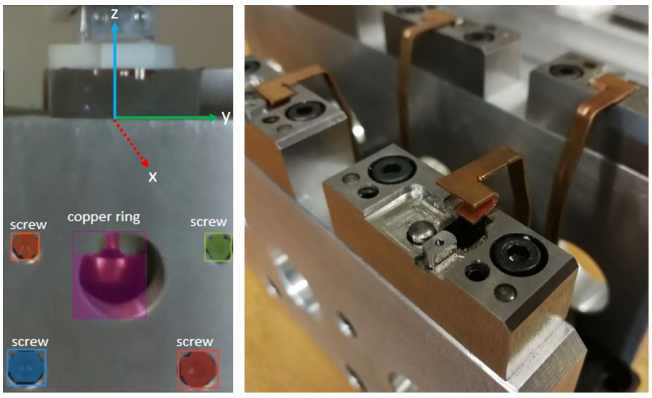

Fig. 3. Left: Result of the segmentation using YOLACT NN. The figure shows a part of a casting mold and the robot gripper while inserting the copper sliding ring. The position of the bounding box of the sliding ring is given with reference to the screws of the casting mold. Right: Improperly inserted sliding ring wings.

On the contrary, we cannot use a camera to solve the second case, as the gripper covers the wings of the ring during the insertion. Instead, we use the estimated forces and torques on the gripper and the robot's pose to calculate the vector on which the contact point between the robot and the casting mold lies [17], [18]. The context ζ_2 is thus given with x and y coordinate of the contact point

$$\boldsymbol{\gamma} = \frac{\mathbf{F} \times \mathbf{M}}{\|\mathbf{F}\|^2} + \alpha \frac{\mathbf{F}}{\|\mathbf{F}\|} + \mathbf{p}_r, \tag{1}$$

where $\mathbf{F} \in \mathbb{R}^3$ and $\mathbf{M} \in \mathbb{R}^3$ are vectors of measured forces and torques, respectively, and $\mathbf{p}_r \in \mathbb{R}^3$ is the robot position vector. Scalar α is chosen so that the z component of $\boldsymbol{\gamma}$ equals the z component of the top of the casting mold. The robot's action, necessary to resolve an exception, is described by sequences of forces and torques. Namely, due to the robot's compliance and friction with the environment, the robot cannot apply small movements that would cause the sliding ring to settle into the mold. Therefore, instead of the position trajectory, it is necessary to capture the forces and torques applied by the operator during the demonstration. The sequence of forces and torques are encoded as phase-dependent radial bases function (RBFs), where matrix $\mathbf{W} \in \mathbb{R}^{7,N}$ denotes the weights of each radial basis function, and N is the number of radial basis functions [14].

When a new error situation occurs, we capture the camera image and robot sensors and calculate current context ζ_i. From the current context and the database of K previous contexts and associated action we calculate weights of the RBF of the generalized action as

$$\mathbf{W}_i = \mathtt{LWR}(\zeta_\mathtt{i}, \zeta_\mathtt{k}, \mathbf{W}_\mathtt{k},\ \mathtt{k} = \{\mathtt{1}, \ldots, \mathtt{K}\}) \tag{2}$$

$\mathtt{LWR}$ denotes statistical learning procedure Locally Weighted Regression [19]. From obtained parameters $\mathbf{W}_i$ we calculate forces and torques as weighted sum of radial basis functions at each sampling instance [14].

The trajectory composed of the reference positions and orientations obtained from the DMP integration and forces and torques calculated from (2) is fed to the passivity-based impedance control law [20], which produces the corresponding robot joint torque. The details of the applied control law can be found in [20], [21].

We composed a database composed of $(\zeta_k, \mathbf{W}_k)$ with $K = 50$ demonstrations of exception policies, 25 for each error type. We achieved 98% success in 476 insertions of rings to the cast mold trials with the above-described framework. Note that without the exception strategy learning, the success rate drops to 79%.

IV. Human robot collaboration

As mentioned earlier, a robot and a human can simultaneously insert sliding rings into a casting mold. Therefore, it is essential to prevent possible conflicts and collisions between the robot and the operator. This section briefly presents our framework for recognizing human intentions and modifying the robot policy based on this. Intention recognition is a vital part of HRC, enabling the robot to recognize and anticipate human actions. In our case, this is necessary to prevent possible collisions between the robot and the human. In doing so, we want to determine the human task as quickly and reliably as possible and change the robot's actions accordingly. For this purpose, we use an RGB-D camera installed above the operator's workspace and observe

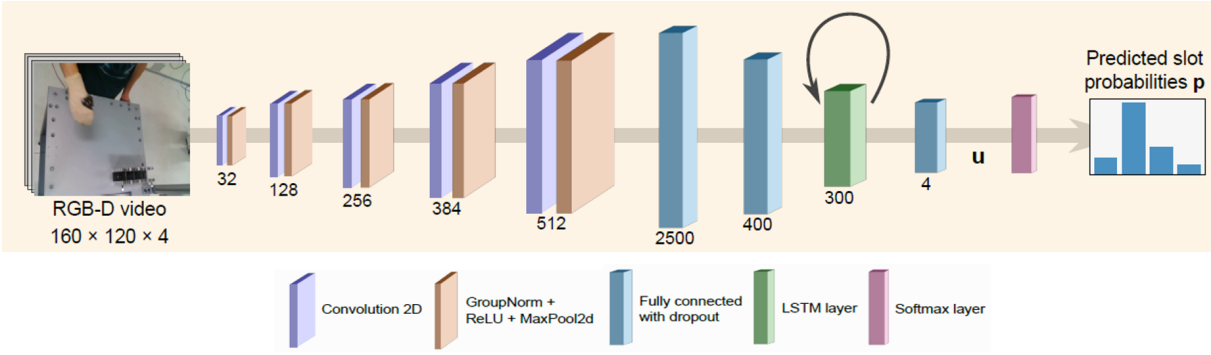

Fig. 4. The proposed recurrent neural network architecture

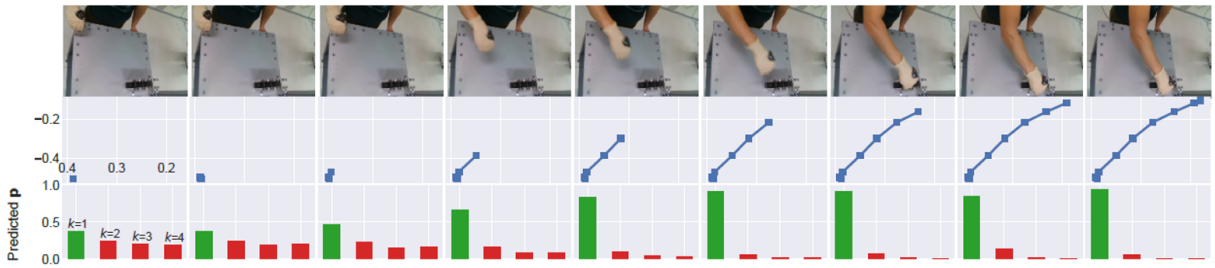

Fig. 5. Example predictions of probability distributions across four goal slots during a task. The upper part of the figure shows the RGB image, the middle part is predicted motion in the robot coordinate system, and the bottom part shows the prediction probability for each slot of the casting mold. Green blocks denote the predicted probability for the first (ground true) slot and red blocks probabilities for the other three slots, respectively.

the motion of the human hand. For predicting the human hand trajectory, we applied recurrent neural networks (RNN). RNNs are especially useful for the intention recognition from a sequence of RGB-D frames due to their structure, consisting of memory units that allow storing information dependent on inputs from previous states. However, classic RNNs can suffer from vanishing or exploding gradients [22], which instigated the development of LSTM networks [23]. LSTM memory units are composed of a cell, an input gate, an output gate, and a forget gate. The cell memorizes values over an arbitrary number of time intervals, where the three gates regulate the flow of information into and out of the cell. During training, errors flow backward through several virtual layers unfolded through time, and the LSTM can thus learn tasks that require memories of events from several time steps earlier [24].

The structure of the applied network named *HandNet* [25] is shown in Fig. 4. It consists of fully connected, dropout, LSTM, softmax layers, and additional input convolutional layers combined with group normalization, non-linear and max-pooling layers. The LSTM layers enable processing sequences of input data and make predictions even after a single input sample is processed. This property enables us to recognize the human's intention before the entire motion sequence is available to the network. The developed RNN was implemented using PyTorch [26] and an NVIDIA GeForce GTX 1080 graphics processing unit. Both architectures were trained using the RMSprop optimization algorithm [27] with a learning rate of $1e^{-4}$ and a batch size of 20. The training stops after 40 consecutive epochs of no mean accuracy improvement on the validation set. The framework was evaluated on the test database of 100 motion samples, which were not used during the training phase. The input samples are formed of sequences of RGB-D camera frames to obtain the predicted intention of the human worker. The predicted intention is the label of the target slot of the casting mold, where the worker is inserting the sliding ring. After each element of the input sample is processed, the networks output a probability distribution across four target slots. With each new position measurement or camera frame, the predicted probabilities are updated, thus allowing online acquisition of the worker's intention as the motion is being carried out. Fig. 5 shows prediction for an example motion sample.

According to the predicted slot of the casting mold, we update the robot trajectory. The DMP goal of the robot trajectory is set to the most distant slot regarding the predicted slot. Additionally, the robot has to select a new slot even if it has already started the motion. In this case, we have to provide for the smooth switch between the trajectories. To accomplish this task, we implemented techniques for smooth sequencing of DMP trajectories [28].

V. System Integration

One of the main issues of Industry 4.0 and 5.0 is the fast and cost-efficient deployment of new solutions that integrate heterogeneous high-tech components for human worker friendly industrial production. The developed cell is a good example of the integration of advanced technologies to implement a production environment suitable for human-robot

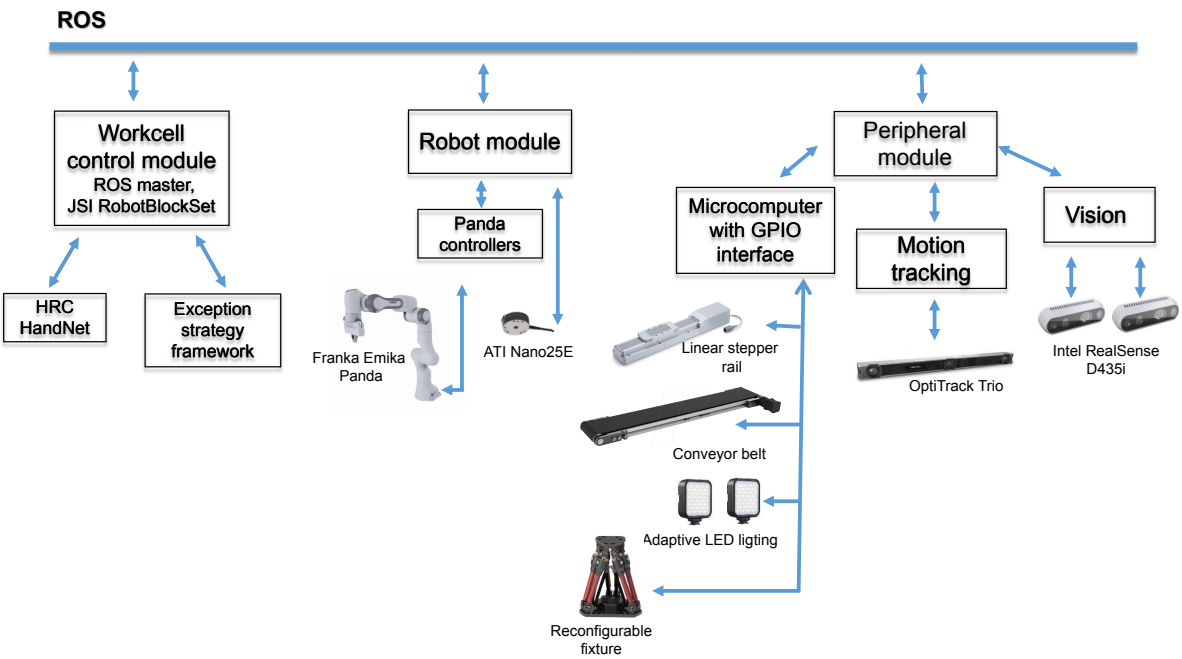

Fig. 6. Block scheme of the cell for the assembly of parts in automotive industry

collaboration. For such environments it is crucial not to hard-code the desired solution. Instead, it should be possible to adapt the required robot operations whenever necessary. The underlying software and hardware environment must support such adaptation processes. To facilitate implementation, the proposed collaborative learning and adaptation strategies were integrated into a modular software architecture based on Robot Operating System (ROS) [3]. ROS provides a suitable framework for deploying various software components that cooperate and exchange data over the shared network [29]. The block scheme of the work cell and connectivity of individual modules via ROS is shown in Fig. 6.

The cell consists of three main modules - 1. workcell control module that synchronizes different operations in the cell, 2. robot module and 3. peripheral module. The workcell control module acts as the ROS master and hosts the high-level workcell software that controls the processes taking place in the cell, including human-robot collaboration and exception strategies framework. The software is based on RobotBlockSet, which is a library of high-level, robot-agnostic controllers developed at our institute. It features commands for various motion interpolation schemes, it supports policy encoding and decoding using DMPs and RBFs, kinematic transformations, motion optimization, reinforcement learning, and iterative learning control algorithms, kinesthetic teaching, and more.

The robot module is dedicated to the real-time control of the Franka Emika Panda collaborative robot. Within the ros_control framework, we implemented passivity-based impedance controller and force controller, where it is possible to set the robot's compliance in an arbitrary direction in space. Implementation details can be found in [21]. Although the Panda robot can estimate the TCP forces and torques using internal joint torque sensors, we mounted an additional ATI universal force-torque sensor in the robot wrist to get more precise measurements for contact tasks.

The peripheral module contains various peripheral devices and sensors to aid the sliding ring insertion process. The peripheral devices include a linear rail for camera placement, conveyor belt, adaptive LED lighting, and a reconfigurable fixture. They are controlled using GPIO interface on a Rasberry Pi 3 microcomputer. ROS integration is provided by the "Equipment Server" and "Equipment Manager", which are both provided by the raspi_ros package [3]. Additionally, two Intel RealSense RGB-D cameras are mounted on the module. Using the realsense_ros package, the image stream is provided for the image segmentation (YOLACT) and intention recognition (HandNet) software running on the workcell control module. Finally, human motion tracking is provided by the OptiTrack Trio motion tracker. It is used to validate the intention recognition framework. ROS integration for the tracker is provided by the Motive and mocap_optitack packages.

VI. Conclusions

In this paper we presented the development of a production cell for the assembly of components in the automotive industry. The cell has been realized based on a modular hardware and software platform that supports rapid development of new solutions and easy inclusion of new modules. In this way, we can shorten the preparation of entirely new solutions and the maintenance and augmentation of the existing cell with new functionalities. The main challenge was to integrate advanced machine learning technologies, advanced control,

and learning exception strategies into a production environment that requires human-robot collaboration. Currently, the cell is in the testing and fine-tuning phase. Its installation in the production line is scheduled for April 2022. In the future, most of our efforts will be directed towards increasing the reliability of assembly in new, unforeseen situations. Another partially solved problem that requires improvement is the reliable picking of sliding rings from the transport plate.

ACKNOWLEDGMENT

The research leading to these results has received funding from the Horizon 2020 RIA CoLLaboratE, grant no. 820767, and H2020 DIH network TRINITY, grant no. 825196.

REFERENCES

[1] K. A. Demir, G. Döven, and B. Sezen, "Industry 5.0 and human-robot co-working," *Procedia Computer Science*, vol. 158, pp. 688–695, 2019.

[2] P. Radanovič, J. Jereb, I. Kovač, and A. Ude, "Design of a modular robotic workcell platform enabled by plug & produce connectors," in *20th International Conference on Advanced Robotics (ICAR)*, Ljubljana, Slovenia, 2021.

[3] M. Simonič, R. Pahič, T. Gašpar, S. Abdolshah, S. Haddadin, M. G. Catalano, F. Wörgötter, and A. Ude, "Modular ros-based software architecture for reconfigurable, industry 4.0 compatible robotic workcells," in *20th International Conference on Advanced Robotics (ICAR)*, 2021.

[4] V. Villani, F. Pini, F. Leali, and C. Secchi, "Survey on human-robot collaboration in industrial settings: Safety, intuitive interfaces and applications," *Mechatronics*, vol. 55, pp. 248–266, 2018.

[5] R. Goel and P. Gupta, *Robotics and Industry 4.0*. Cham: Springer International Publishing, 2020, pp. 157–169.

[6] J. Zhang and X. Fang, "Challenges and key technologies in robotic cell layout design and optimization," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 231, no. 15, pp. 2912–2924, 2017.

[7] T. Gašpar, B. Ridge, R. Bevec, M. Bem, Ž. Gosar, I. Kovač, and A. Ude, "Rapid hardware and software reconfiguration in a robotic workcell," in *18th International Conference on Advanced Robotics (ICAR)*, Hong Kong, 2017, pp. 229–236.

[8] T. Gašpar, M. Deniša, P. Radanovič, B. Ridge, T. R. Savarimuthu, A. Kramberger, M. Priggemeyer, J. Roßmann, F. Wörgötter, T. Ivanovska, S. Parizi, Žiga Gosar, I. Kovač, and A. Ude, "Smart hardware integration with advanced robot programming technologies for efficient reconfiguration of robot workcells," *Robotics and Computer-Integrated Manufacturing*, vol. 66, p. 101979, 2020.

[9] A. Ajoudani, P. Albrecht, M. Bianchi, A. Cherubini, S. Del Ferraro, P. Fraisse, L. Fritzsche, M. Garabini, A. Ranavolo, P. H. Rosen, M. Sartori, N. Tsagarakis, B. Vanderborght, and S. Wischniewski, "Smart collaborative systems for enabling flexible and ergonomic work practices [industry activities]," *IEEE Robotics Automation Magazine*, vol. 27, no. 2, pp. 169–176, 2020.

[10] L. Wang, S. Liu, H. Liu, and X. V. Wang, "Overview of human-robot collaboration in manufacturing," in *Proceedings of 5th International Conference on the Industry 4.0 Model for Advanced Manufacturing*, L. Wang, V. D. Majstorovic, D. Mourtzis, E. Carpanzano, G. Moroni, and L. M. Galantucci, Eds. Cham: Springer International Publishing, 2020, pp. 15–58.

[11] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2-3, pp. 109–116, jun 2004.

[12] T. Gašpar, I. Kovač, and A. Ude, "Optimal layout and reconfiguration of a fixturing system constructed from passive stewart platforms," *Journal of Manufacturing Systems*, vol. 60, pp. 226–238, 2021.

[13] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal, "Dynamical movement primitives: Learning attractor models for motor behaviors," *Neural Computation*, vol. 25, no. 2, pp. 328–73, 2013.

[14] F. J. Abu-Dakka, B. Nemec, J. A. Jorgensen, T. R. Savarimuthu, N. Krüger, and A. Ude, "Adaptation of manipulation skills in physical contact with the environment to reference force profiles," *Autonomous Robots*, vol. 39, no. 2, pp. 199–217, 2015.

[15] B. Nemec, M. Simonič, and A. Ude, "Learning of exception strategies in assembly tasks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6521–6527.

[16] D. Bolya, C. Zhou, F. Xiao, and Y. Lee, "Yolact: Real-time instance segmentation," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9156–9165.

[17] K. Kitagaki, M. Fujiwara, T. Suehiro, and T. Ogasawara, "Pseudo contact point monitoring for contact state estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyongju, Korea, 1999, pp. 832–837.

[18] B. Nemec, K. Yasuda, and A. Ude, "A virtual mechanism approach for exploiting functional redundancy in finishing operations," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 4, pp. 2048–2060, 2021.

[19] C. G. Atkeson, A. W. Moore, and S. Schaal, "Locally Weighted Learning," *Artificial Intelligence Review*, vol. 11, no. 1-5, pp. 11–73, 1997.

[20] A. Albu-Schaffer, C. Ott, and G. Hirzinger, "A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots," *The International Journal of Robotics Research*, vol. 26, no. 1, pp. 23–39, 2007.

[21] M. Simonič, L. Žlajpah, A. Ude, and B. Nemec, "Autonomous learning of assembly tasks from the corresponding disassembly tasks," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 230–236.

[22] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[25] M. Mavsar, M. Deniša, B. Nemec, and A. Ude, "Intention recognition with recurrent neural networks for dynamic human-robot collaboration," in *20th International Conference on Advanced Robotics (ICAR)*, 2021.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, vol. 32, pp. 8024–8035.

[27] G. Hinton, "Neural Networks for Machine Learning, Lecture 6e, rmsprop: Divide the gradient by a running average of its recent magnitude," https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, 2012, [Online, accessed 12-November-2021].

[28] B. Nemec and A. Ude, "Action sequencing using dynamic movement primitives," *Robotica*, vol. 30, no. 05, pp. 837–846, oct 2011.

[29] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibsz, E. Bergery, R. Wheeler, and A. Ng, "**ROS:** an open-source robot operating system," in *ICRA Workshop on Open Source Software*, Kobe, Japan, 2009.