2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 23-27, 2022, Kyoto, Japan

2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-7927-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IROS47612.2022.9982004

Robot Companion, an intelligent interactive robot coworker for the Industry 5.0*

F. Gosselin, S. Kchir, G. Acher, F. Keith, O. Lebec, C. Louison, B. Luvison, F. Mayran de Chamisso, B. Meden, M. Morelli, B. Perochon, J. Rabarisoa, C. Vienne, G. Ameyugo

Abstract— **To overcome the limitations of the so-called Industry 4.0 focusing on mass production and full automation, a novel paradigm was recently introduced, namely Industry 5.0, which aims at an increased collaboration between humans and machines, and particularly robots, instead of replacing the former with the latter. This challenge requires novel interactive intelligent robots able to perform complex tasks easily and efficiently and to collaborate on the fly with humans whenever required, be it for training or working. In this work, the Robot Companion, a novel demonstrator of this paradigm, is introduced. It combines robotics, Artificial Intelligence, software engineering and embedded systems technologies, and targets industrial assembly tasks. First tests show that this robot can efficiently assemble a representative gear system autonomously or in collaboration with human operators.**

I. Introduction

Since the middle of the 2010s, most developed countries have launched large initiatives aiming at reshoring factories to regain control of their industrial production and increase their economic growth potential. These initiatives were only marginally focused on traditional mass production, for which developed countries can hardly compete with low wage countries. On the contrary, their goal was to answer an ever-increasing demand for personalized products, still at a competitive cost. Despite various names used in different countries, all these initiatives can be framed within the generic trend "Industry 4.0", i.e. the combination of IoT, data science and automation targeting, amongst others, mass customization. Almost 10 years later however, these initiatives, targeting full automation, have failed to reach the small and medium enterprises (SMEs). The main perceived barriers are the considerable initial investments required, the limited flexibility, incompatible with shorter product life cycles, smaller lot sizes, and mass-customized products and the need for highly skilled people to operate the robots [1].

More recently, a novel paradigm, namely Industry 5.0, has been proposed to tackle these drawbacks. While keeping the benefits of Industry 4.0 coming from connectivity and Artificial Intelligence (AI), Industry 5.0 also aims at integrating human intelligence in the loop. The main revolution is now the co-working between humans and robots for the final goal of customizing products at a mass scale, combining human creativity with the precision, speed and reliability of robots. The quality of industrial production is therefore improved, while at the same time it becomes possible to have robots applied to smaller lot sizes. Indeed, another essential point of Industry 5.0 is the potential to redeploy robots to different tasks thanks to a much easier interaction between the operator and the machine. The paradigm also shows the potential for frugality, adjusting rapidly to different needs and saving resources. The collaboration between the human and the robots leads to flexible business models, adjusting to demand in real-time and further reducing waste and overproduction.

To create this new generation of robots that can increase the flexibility and productivity of manufacturing, three main requirements can be identified: a user-friendly human-machine interface (HMI), allowing a non-expert user to easily define the robot task; an increased capacity to react and adapt to unexpected situations; and a cloud infrastructure allowing a more reliable and efficient shared training and use of intelligent functions and skills. To date however, no solution fully integrates these capacities. Industrial collaborative robots do not perform complex tasks (they are mostly used in palletizing or simple machine tending operations, see e.g. www.universal-robots.com/case-stories) while laboratory prototypes integrating the numerous functions necessary for realizing them often come in the form of humanoid or torso robots [2] [3] which suffer from an insufficient robustness for real-life applications. In addition, despite technological progress, advanced prototypes perform several times slower than humans, preventing a favorable return on investment (see e.g. video media in [4]). Simple robots are more robust, but they hardly prove to be really interactive, losing the interest of human-robot collaboration.

In this work, we introduce the Robot Companion first demonstrator. This robot integrates software engineering and AI functionalities for performing much more complex tasks than collaborative robots. Yet, contrary to industrial robots, it integrates collaborative robotics functionalities and multimodal HMIs for an efficient collaboration with humans, those functions being embodied in a simple design, much more compact and reliable than state of the art humanoid robots. The Robot Companion is capable of performing industrial assembly tasks in autonomous and interactive modes, with the aim in a longer term to reach and even surpass the speed of human operators.

This paper is organized as follows: related work is presented in section II, the Robot Companion setup and its application are then presented in section III, and its main components in section IV. First validation is introduced in section V, considering both autonomous and interactive modes. These results are discussed in section VI, which outlines perspectives for future work. Finally, section VII concludes the paper.

* This research was supported by the "Agence Nationale de la Recherche" (Carnot Institute financial support N°19 CARN 0006 01 and N°20 CARN 0006 01).

All authors are with the Université Paris-Saclay, CEA, LIST, F-91120 Palaiseau, France (e-mail : florian.gosselin@cea.fr).

II. Related Work and Targeted Applications

Long limited to the realization of repetitive tasks in controlled environments in the context of large factories automation, robotics has recently made huge progress, allowing it to adapt more simply to novel situations and to perform a much larger set of tasks.

A major field of research in the Industry 4.0 focuses on how already existing equipment and robots can be connected and used to construct intelligent production cells [5] [6] [7]. The goal is to allow seamless integration of various technologies, to optimally share knowledge and data treatment between the edge and the cloud (in a context were time constraints are of primary importance for the real-time control of the equipment) and to promote communication standards (e.g. OPC-UA), aiming at plug and produce solutions. This approach was adopted in the context of several recent challenges. Regarding high-mix low-volume picking in logistics warehouses for example, academic developments were boosted by the Amazon challenge [8], with either collaborative robots or robots endowed with sensors allowing them to react to environment changes, equipped with dexterous grippers or dedicated tools and complemented with vision sensors and artificial intelligence algorithms. This work has allowed impressive results regarding the diversity of the objects the robots are now able to grasp. To date however, robotized picking remains much slower than when performed by a human. As a consequence, the picking itself remains largely manual, robotic assistance focusing more on how to deliver bins to human operators (see e.g. solutions from AutoStore, www.autostoresystem.com, or Exotec, www.exotec.com). The same holds for assembly challenges, among which the World Robot Summit's Assembly Challenge [9]. Robotizing complex assembly tasks is a very challenging work and robots remain to date slower than humans. As a consequence, most such work remains manual today, whether considering motors final assembly in the automotive or aerospace industries, household appliances and consumer electronics, electric equipment, or even textile manufacturing. Dedicated solutions seem feasible, requiring however considerable investments which make them not accessible to SMEs. Whether considering small series (on demand production, mass customization) or even large productions, such assembly tasks are easy for humans but still difficult for robots. Indeed, they require a high level of expertise which is difficult to model and transfer to a robot.

Collaborative robots, or cobots, are without doubt a key element for such tasks and markets. While still a niche market, recent advances in actuators, materials and security sensors make possible the use of cobots without expensive safety measures and cages. Their deployment, redeployment and reconfiguration automatically becomes cheaper and faster, allowing various specific manufacturing applications [10]. Cobots are however limited to simple tasks. As an example, Universal Robots are mostly used in palletizing or simple machine tending operations (www.universal-robots.com/case-stories). To go further, more complex machines integrating functions such as sensing, reasoning, action planning and monitoring, mobility and manipulation, and human-robot interactions, are required. In the literature, such machine often come in the form of advanced humanoid or torso robots like for example Armar 6, HRP-4 or TORO [2] [3]. While less mature than industrial robots and cobots, such systems are beginning to enter our work environment for preliminary tests of their ability to perform complex tasks usually made by humans. Their performances remain however much lower than their human counterparts. Also, their robustness is limited, both considering their mechanical resistance to impacts and fall and their ability to adapt to harsh conditions (i.e. poor illumination) and to react to unanticipated situations. As a consequence, except for such preliminary tests, their usage remains limited to research laboratory environments to date. One of the sole exception is the use of a bimanual torso robot in an industrial setting in the Glory factory in Saitama, Japan, where Kawada Nextage robots are used to assemble money-handling machines [11].

As intelligent production cells and factories are too expensive, while cobots are limited to simple tasks and advanced humanoid robots are still too complex and not robust enough, another kind of robots is required to answer the challenge of the Industry 5.0, at least until more advanced robots become reliable enough. In this paper, the proposed solution to focus on is an intelligent yet interactive robot, called Robot Companion, capable of performing complex tasks autonomously but also to collaborate with human operators whenever required.

III. The Robot Companion Setup

Our first use case focuses on the assembly of a gear system used in assembly challenges [9]. It is composed of 11 parts, a base plate, 2 threaded shafts, 2 spacers, 2 gears mounted on ball bearings, 2 washers and 2 nuts. The goal is to screw the axes on the base plate, then to insert a spacer, a gear and a washer on each axis, and finally to screw both nuts. This task is challenging for vision, with small and shiny parts similar in size and shape. It is also demanding for robotics, with an intermediate to tight fit between the shaft and the bearing inside the gear (i.e. from 0.002 mm to 0.017 mm, well below the precision and repeatability of usual collaborative robots). This task was slightly modified compared to the competition. As shown in Figure 1, the parts are initially placed randomly in a tray fixed on a table, and, as only one robot is present in our first demonstrator, the base plate is fixed on the table, and the axes are pre-assembled.

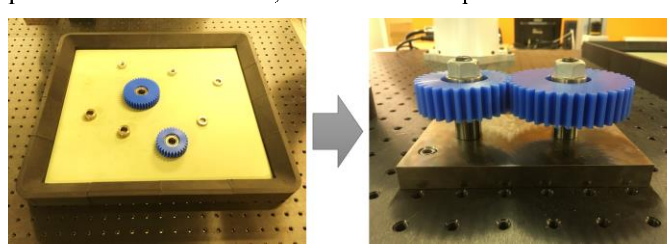

Figure 1. Gear Unit Assembly

A. Hardware Architecture

As shown in Figures 2 and 3, the first Robot Companion demonstrator is composed of a Franka Emika collaborative robot fixed on the table beside the tray and the assembly area, and two vision systems. The first one, used for the recognition and localization of the to-be-assembled objects, is composed of a Logitech C930e 2D webcam allowing for a full HD 1080p capture at 30 fps and a Photoneo Phoxi 3D

Scanner M 3D active camera. Both are fixed to a pole, itself attached to the table, at a position and orientation guarantying a wide coverage of the whole workspace. The second one, intended to monitor and recognize the human operator's activity when he or she intervenes, makes use of a second Logitech C930e webcam, which is fixed on a second pole. The Franka bi-digital gripper is equipped with a custom designed V and circular shaped end-effector allowing to self-center the parts, even in case of an up to few millimeters misalignment due to the vision system, calibration and/or robot inaccuracies. These end-effectors also have shoulders helping parts, and especially gears, to remain in the gripper during the insertions. A digital twin of the whole platform completes the setup.

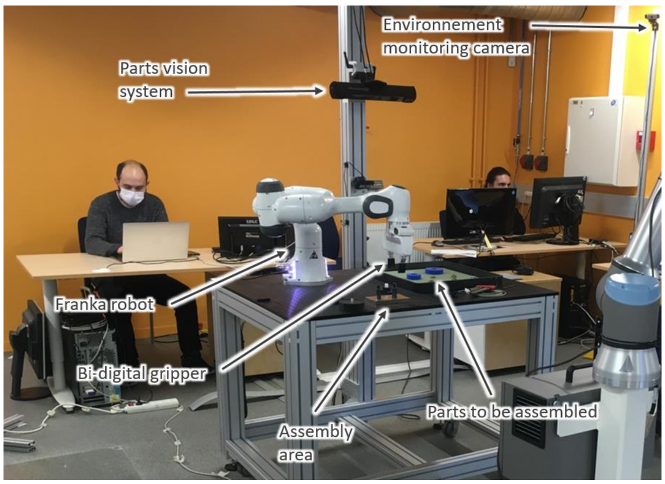

Figure 2. Robot Companion first demonstrator.

The 2D and 3D cameras are connected to a PC also running the orchestrator in charge of planning and monitoring the progress of the task (see Figure 3). A second PC is used to run the digital twin and the action recognition functions. A third PC, linked to the Franka controller, runs a high level robot controller. Finally, a HMI is used to display information and interact with the operator.

Figure 3. Robot Companion first demonstrator hardware architecture.

B. Methodology and Software Architecture

The Robot Companion software architecture, illustrated in Figure 4, is composable and interoperable. It is based on common data and semantic models for describing the task and the environment constraints, enabling connectivity and integration of production resources (i.e. operators, machines, robots). Robot Companion implements an end-to-end digital infrastructure boosting data sharing between engineering, production and management (i.e. horizontal integration), affecting all hierarchy levels of the production (i.e. vertical integration). Separation of concerns and separation of roles[1] are the two key paradigms that define this software architecture. The principle consists in addressing more abstract and general concepts at the higher layers of the architecture and specializing individual and concrete concerns in the lower layers. The definition of appropriate abstractions and interfaces between the various roles in the robotic software development process allows developers for the realization of efficient solutions keeping the focus on their responsibility and expertise alone. The best practices learnt from robotics frameworks definition like HORSE[2] and RobMoSys[3] guided us to adopt a model-based software engineering approach that reduces integration efforts in robotics applications to support the composition of modules from multiple stakeholders and enable the independence from specific execution frameworks.

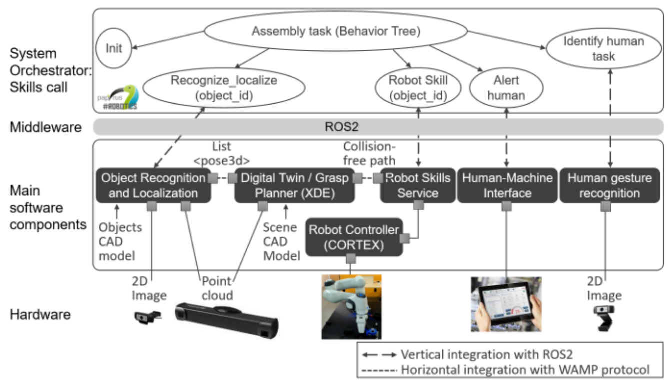

Figure 4. Robot Companion first demonstrator software architecture.

The first layer of this architecture is the system orchestrator. It executes the assembly task, which is programmed as a composition of actions in a behavior tree (BT), and manages the configuration and coordination of the software components in the system. BT actions represent the execution of system functionalities (skills), realized by software components. These skills are implemented inside the Object Recognition and Localization, the Human-Machine Interface and the Robot Skills Service modules. The manner that the skills are implemented is not of interest at this level. The orchestrator only needs to know that the skills are available and that they propose the necessary strategies to achieve the task. When one skill is called, its components are activated and start performing their own computation. The results can then be exploited by other components.

The Robot Companion architecture is also hardware-agnostic. The components are implemented in terms of configurable parameters so that if one decides for example to change the camera, the 3D vision algorithms are reusable. Similarly, the robot skills are parametrized generic strategies, and their reusability from the Franka Emika to a Universal robot UR10e was validated.

[1]https://robmosys.eu/wiki/general_principles:separation_of_levels_and_separation_of_concerns

[2] http://www.horse-project.eu/

[3] https://robmosys.eu/

IV. MAIN COMPONENTS OF THE ROBOT COMPANION

A. Orchestration

The Model-Based Software Engineering platform *Papyrus for Robotics*[4] is used to design and deploy the software system architecture which executes the robot assembly task. Design models include the formal specifications of the software components in Figure 4 that have a link with the uppermost layer (vertical integration) and of the BT models of the different phases of the assembly task. The *skill abstraction* is the formal specification that interfaces the task level and the level in which software components are executed. Skills are not bound to concrete components, but to a coordination interface. Concrete components realize the coordination interface and hence the skill.

The two main phases of the assembly task are *ObjectSelection* and *ObjectProcessing*. The first one selects one object per assembly step from a database of known objects. The second one (Figure 5a) reads the ID of the selected object from the blackboard and executes a sequence of three subtasks with recovery actions that are repeated until they all complete with success. The first subtask (Figure 5b) encodes the logics to recognize and localize the object, the second one to grasp it and the third one to insert or screw it depending on the object type. Condition (green) and action (yellow) BT nodes in Figure 5a and b correspond to the invocations of system skills. Recovery from failure (e.g., object not found) is achieved by alerting the human operator, who can either perform the assembly step for the robot (and the subsequent subtasks are not executed), or ask the robot to retry again the current step. Next, an assembly validation step is performed using the vision skills. The human operator is also alerted in case of unrecoverable errors.

Papyrus for Robotics supports the transition from design-time specifications to deployment and execution through automatic code-generation. The tool processes all the models and generates an implementation compatible with ROS2[5], including the application nodes, XML files and C++ classes compatible with BehaviorTree.CPP[6] and Nav2 [12] to implement the BT logics, as well as all the scripts with build, launch directives and parameters. Figure 5c shows the run-time interaction between the orchestrator (which invokes the skills) and one coordinated component (which exposes them).

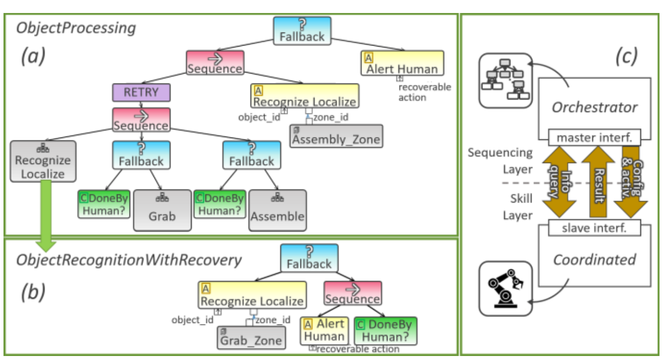

Figure 5. Part of task models (a, b) and coordination interface (c).

[4] https://www.eclipse.org/papyrus/components/robotics/
[5] https://docs.ros.org/en/foxy/index.html
[6] https://www.behaviortree.dev/

B. Digital Twin

The Robot Companion comes with its Digital Twin which uses Unity3D as graphics engine and XDE (eXtended Dynamics Engine [13]) as physics engine (see Figure 6). XDE is computationally efficient in the physically realistic simulation of the dynamics of rigid multibody mechanical systems, such as robot arms, in interaction with their environment (e.g. gravity, contact, friction forces…). It allows computing the evolution of the system in interactive time, with access to a large range of data relative to the robots (velocities, torques, forces, swept volumes), making it possible to test, refine and validate in advance the functioning of the robot in immersive Virtual Reality (Virtual Commissioning). The models are imported from Solidworks with CAD plugins such as PiXYZ or CAD Exchanger, allowing to use them easily for graphical rendering as well as collision computation. The physics properties of the robot can be extracted from those CAD files, or from URDF (Unified Robot Description Format) files. XDE can also work with point clouds, allowing to update the world model at each step with the data provided by the 3D camera.

The Robot Companion Digital Twin also makes use of some of the technologies presented in the SEEROB framework [14] that aim at evaluating the safety and ergonomics of robotic workstations. Beyond allowing dynamical simulation of the behavior of the robot using numerical models, it makes it possible to directly link the robot model with its real controller. This functionality is used to run concurrently the robot and its model for monitoring purposes.

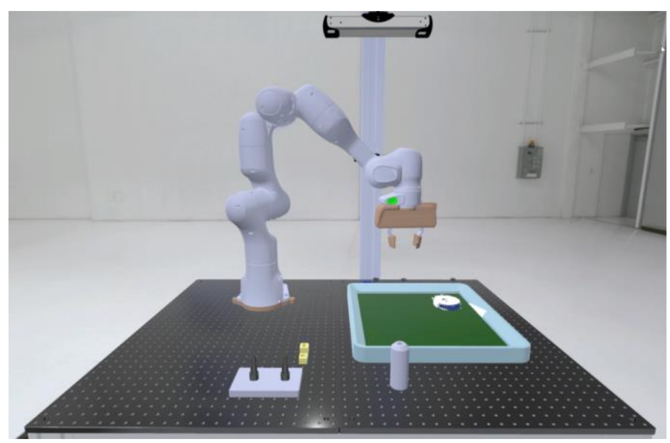

Figure 6. The Robot Companion digital twin.

For the Robot Companion demonstrator, a RRT (i.e. Rapidly-Exploring Random Tree) path planning algorithm is also implemented, with collision avoidance in the simulator, based on the Open Motion Planning Library [15]. The computations are made in the joint space and XDE is used to compute the pose of the robot for each configuration proposed and detect potential interferences with its environment.

C. Parts recognition and localization

The vision module consists in two distinct algorithms working together to generate very precise object poses: an object recognition algorithm, followed by a six degree of freedom (or 6D) registration algorithm. The latter would theoretically be sufficient to recognize the objects on the fly since different objects have different 3D descriptors most of the time. However, some objects in the dataset have very similar 3D shapes (e.g. spacers and washer have similar shapes and dimensions, except different heights) so that they exhibit the same descriptors in practice. While this could be disambiguated through occlusion testing on a render or explicit discrimination based on object height, the recognition problem is tackled here separately. This is in essence similar to “two-stage” object detectors in literature, the object recognition step allowing to make the 6D registration more robust.

The object recognition algorithm is a state of the art EfficientDet D2 object detector [16] trained on 20 000 RGB images generated by automatically mixing annotated objects with backgrounds. This training took four hours on four Nvidia RTX 2080 GPUs from a network pretrained on MS COCO [17]. The output of this algorithm is a set of up to 50 bounding boxes with a class label and a confidence level. Images are captured with an off-the-shelf webcam and the bounding boxes are reprojected (roughly) on the point cloud acquired by the Photoneo sensor for 6D registration.

The 6D registration algorithm is a classical point cloud descriptor-based registration pipeline [18] operating on roughly segmented bounding boxes from the object recognition algorithm. Description uses a custom six dimensional descriptor named hexagon (see Figure 7) whose main purpose is to be fast to compute and match compared to larger descriptors such as FPFH [19]. Since hexagon is so lightweight, it can be computed on every point of the cloud, which greatly helps localizing very small objects. Pose hypotheses are selected in Hough space similarly to [20], refined using point-to-point ICP and scored using the proximity between a render of the registered object and the observed scene point cloud. This whole process is GPU-accelerated and requires less than 0.5s to localize all objects in a given scene point cloud with a submillimetric precision.

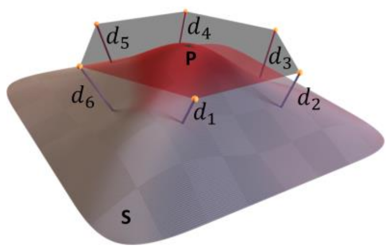

Figure 7. Hexagon is a simple descriptor with straightforward geometric meaning. It is obtained by sampling six signed distances d1-d6 from the tangent plane to the objects' surface.

Grasping configurations are generated automatically by analyzing the objects models offline and determining roughly parallel and planar surfaces opposite to each other where the two fingers of the gripper can make contact. These geometric configurations are pruned a first time by simulating a contact with the actual gripper. Online, configurations which generate collisions with the scene point cloud are removed and the remaining ones are scored in terms of ease of access. Multiple configurations are returned so that trajectory planning can pick whichever one is best for the current task.

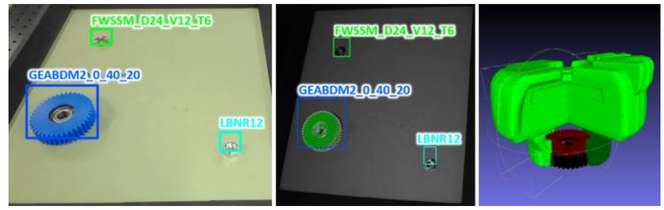

Figure 8. Left: result of the objects recognition algorithm. Center: 3D point cloud of the current object (here the large gear) reprojected on the 2D image. Right: sample of grasps generated for the grasping of the gear. The names of the parts are those defined in the WRS Assembly Challenge.

D. Robot control

The robot controller was programmed using CEA-LIST’s in-house Component-Oriented Execution Engine (CORTEX), which leverages component-oriented programming for control scheme reusability and modularity, in a similar way as OROCOS [21]. This controller communicates with the robot’s low-level API by delivering joint torques commands and gripper instructions (see Figure 9). On the other side, it implements skills that manage mode switches and provide parameters and commands. Implemented control modes include low-level motion planners for trajectories, and hybrid force-position control for assembly steps.

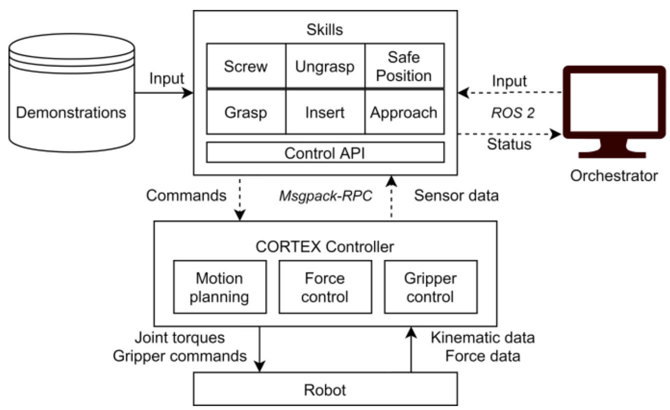

Figure 9. Robot skills implementation.

Skills are high-level reusable robotic functions that implement complex behaviours [22]. They take a minimal parameter set as an input, while providing extensive control on the controller side. Skills either receive input data from past demonstrations, in the case of insertion targets, or from the orchestrator, in the case of grasping targets and trajectories. The insertion skills required a particular focus. Given the limited accuracy of the robot, an hybrid force/position control pattern had to be employed. Stiffness was decoupled between the horizontal plane and the vertical axis, in order to leverage compliance during the assembly. The gears were the most challenging parts, as the insertion tolerance was measured to be less than 10 µm. A 6-dimensionnal Lissajous oscillator was implemented in the controller. Translation oscillations on the horizontal plane enabled the skill to further overcome inaccuracies during the first step of the insertion. Z-axis, high-frequency angular oscillations were an efficient way to prevent the gear from arching against the axis. Finally, low-frequency angular oscillations about the local Z-axis enabled the assembly of the second gear by finding a suitable alignment for both gear’s teeth.

E. Environment and operator monitoring

The Robot Companion nominally works in autonomy. However, an interactive mode in also provided, in case an intervention of a human operator is necessary to solve possible problems. This intervention requires a supervision of the robot's environment and an understanding of the position of the different people in this environment. For this purpose, a classic RGB camera (here a webcam but it could also be a video surveillance camera) is used. The neural network used, PandaNet [23], detects all the people present in the scene and estimates their 3D skeleton from the video signal (see Figure 10). Unlike state-of-the-art approaches such as [24], PandaNet is an approach that guarantees a low and constant processing time no matter how many people are present in the scene. Posture estimation is performed directly from the image signal using a single network, without going through an intermediate step of 2D skeleton estimation using another network, as in [25]. Finally, PandaNet only requires a single image for its estimation, so the latency of the system is zero, unlike approaches such as [26] which require quite large time windows of several seconds to estimate the posture on the central image of this window. This last point is an important feature for a safety element that aims to guarantee the integrity of the operator.

The estimation of the 3D posture is performed according to a scaling factor, due to monocular processing, and is estimated in a person-related reference. A scaling and change of reference step is performed by extrinsic calibration of the camera in the robot reference. A proximity perimeter around the robot can thus be easily defined and any intrusion into this perimeter signals an intervention in progress. From there, an analysis of the operator's gestures according to the succession of 3D skeletons expressed in the robot's frame of reference can be performed in order to determine the action carried out (picking up a part, adjusting a part, etc). Finally, depending on the actions identified, a message is sent to the orchestrator so that it can update the state of the system and continue processing the robot, once the intervention has been completed and the operator has left the intrusion zone.

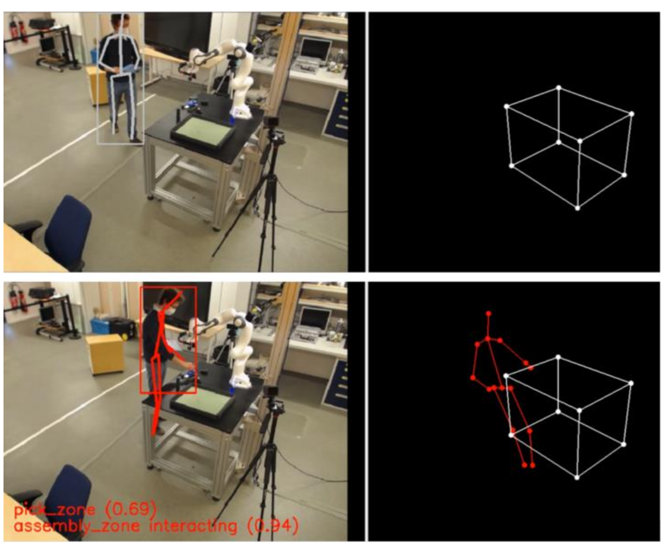

Figure 10. Detection of the presence of an operator and classification of its actions using PandaNet.

F. HMI

The Robot Companion is provided with an adaptive interface displayed on a tablet (see Figure 11) and allowing the operator to follow the robot's progress and to interact with it. In nominal autonomous mode, the interface indicates the task and the part being handled. In interactive mode, it becomes the mean of communication between the operator and the robot.

Figure 11. Main information diplayed on the HMI (left: nominal mode, right: error alert).

When an error is detected, the interface warns the operator in a multimodal and adaptive manner (i.e. visual alert or audio beep or speech message). The modality is chosen according to the operator's preferences and to the ambient situation (the interface can analyze the ambient noise and decide to play an alert rather than an audio message if the background noise is too loud). The alert is contextual (time stamping, type of error, object, zone). For safety reasons the interface prevents the robot from resuming as long as an operator is detected near the robot by the activity detection system to which it is connected through the orchestrator.

The interface also incorporates two input buttons allowing to resume the assembly either at current step (if for example an object was missing and the operator has replaced it in the tray) or at the next step (if the operator has performed the task him or herself). Depending on the activity analysis, the interface will adapt to propose the most appropriate option, the operator remaining responsible for the choice of the recovery mode.

In practice, the interface is in two parts: one part is a shared library called by the ROS application and the other part is an app on a tablet. The communication is done in UDP based on an in-house protocol that guarantees that the messages have been delivered.

V. EVALUATION

The Robot Compagnon was first evaluated in its ability to perform the Gear Assembly in full autonomy. Once the process is launched, the parts are assembled one after another, beginning with one axis (i.e. insertion of the first spacer, tight insertion of the first gear, insertion of the first washer and screwing of the first nut), then the second one (Figure 12 illustrates the assembly of the second spacer). Each part is first recognized and localized using the vision module. Then the digital twin is updated with the object's 3D point cloud and 3D pose, allowing to generate compatible grasps and to compute collision free paths to reach them. These paths are send to the robot controller who is in charge of realizing these trajectories, then to perform the subsequent grasping or assembly skills. During this process, the digital twin runs concurrently with the robot. Also, the different steps are displayed on the HMI.

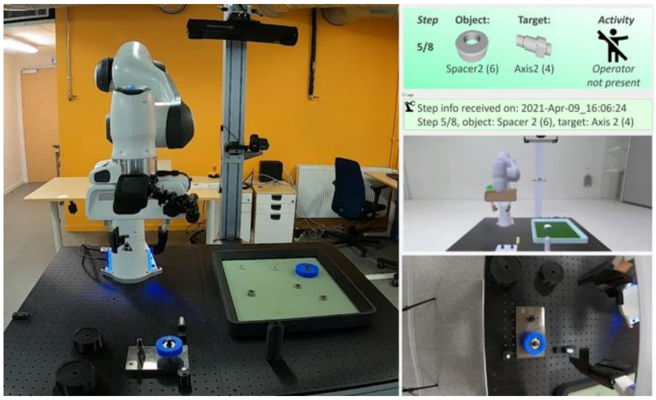

Figure 12. Assembly of the second spacer in autonomous mode. Icons and text are magnified for better readability.

No statistical analysis of the success rate was made to date (this remains a perspective), especially as the system is still subject to improvements and optimization, but it was possible to assemble all parts several times in a row.

Alternatively, if an error is detected, the system switches to interactive mode. A warning is then displayed on the HMI to inform the operator that an action is required. During his or her intervention, the 2D environment camera is used to analyze his or her activity and a message is sent to the orchestrator so that it can update the HMI and resume the assembly properly. The operator can either replace a part that is missing or that fell on the table in the deposit tray or assemble it in place of the robot. The system automatically detects the intervention area (pick zone above the tray or assembly area). This information (zone 1 or zone 2) is displayed on the HMI which suggests resuming either at the current step or directly at the next step (see Figure 13), the final decision and validation being up to the operator.

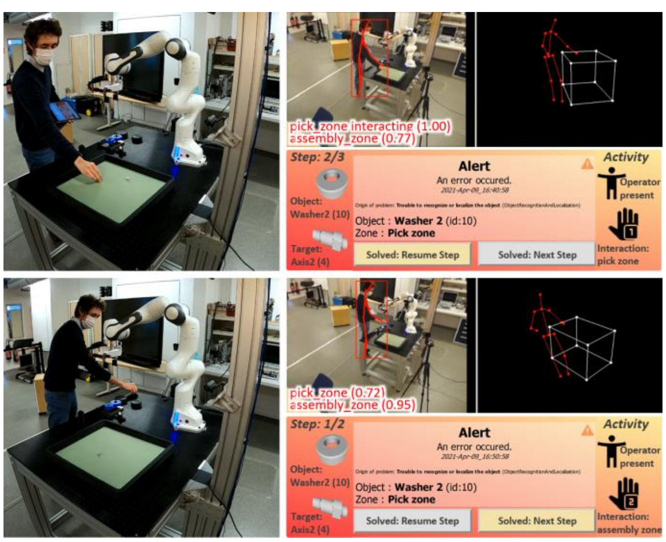

Figure 13. Interactive mode (top: replacement of a part in the tray; bottom : manual assembly in place of the robot). Icons and text are magnified for better readability.

VI. Discussion and Future Work

Despite first successful assemblies, the Robot Companion is currently subject to some limitations. Indeed, the gear assembly remains simple compared to other tasks present for example in the WRS Assembly Challenge [9]. Also, the robot remains significantly slower than a human operator. The ability to work faster and deal with more complex cases calls for continuous improvements, as explained below.

In the short term, we plan to add cables and connectors to the assembly, as in recent assembly challenges. This will require additional skills. Some work will also be done on the vision and assembly sequence, in order to perform tasks in parallel as much as possible, and a second robot will be implemented in order for even more parallelization and reduced assembly time. The default modes library will also be further enriched, with novel situations and an enriched adaptive orchestrator. An embedded camera allowing to get more information on the situation will be implemented and used therefore.

In the longer term, several challenges will be addressed towards the next step of a self-learning robotics. First, situational awareness and dynamic task execution solutions will be studied, allowing the robot to reach operational autonomy. Therefore the robot will have to be able to exploit situation models for planning and monitoring the execution of its tasks. It will also have to detect abnormal situations, and plan and realize adequate remediation actions. Second, the capacity of learning easily and rapidly new tasks is a key factor for the robot flexibility which will be subject to research. For adjusting to situations similar to the one already experienced, the orchestration and skills framework presented above will be further exploited and transfer learning approaches investigated. Then, for handling new tasks in full autonomy, deep reinforcement learning approaches, trained in simulation, will be integrated.

The pursuit of a preliminary work initiated along the development of the Robot Companion first demonstrator, but still not implemented in the platform to date, will also be considered. As shown previously, behavior trees is a promising mechanism for modeling the system orchestrator. However, modeling tools, such as Papyrus for Robotics, are meant to be used mainly by system architects who are generally familiar with the modeling environment. In order to make the scenario description in the reach of mere users, a proof of concept using Natural Language Programming (NLP) for mapping a textual description of the scenario (in French) to a behavior tree was developed. This work relies on Machine Learning algorithms that extract from various textual descriptions key words corresponding to the skills called by the orchestrator. The behavior tree model including its leaves and the connections between them can then be constructed automatically. This work is a first step that should be consolidated in future work.

Finally, one key issue that we plan to study in parallel with all these developments is how to assess Robot Companion benefits by measuring qualitative and quantitative Key Performance Indicators (KPIs). While qualitative KPIs are easy to define at this stage, e.g. the system ability to recognize and to manipulate objects, to interact with humans and to recover from failures, quantitative KPIs remain challenging to measure, as it requires to define under which conditions (light in the environment, object brightness, object visibility, etc.) this value is being measured and to compare that value with other systems tested in the same conditions. Another KPI example could be the reduction of the programming effort of the system thanks to the Robot Companion toolchain involving Model-Based Design tool (i.e. Papyrus for Robotics), the robot skills library and the Digital Twin, instead of writing code from scratch. In order to have significant results, it would be interesting to make this evaluation with a statistically relevant number of users having different profiles: system architects, roboticists, vision experts, developers, operators, etc., allowing to deduce whether the system is easy to program and for whom. In practice, the KPIs measurement will be conducted in a separate work in parallel to the system improvement.

VII. Conclusion

In this article, a demonstrator of intelligent interactive robot, called Robot Companion, is presented. This device proved able to assemble a gear unit either in autonomy or with the help of an operator. Contrary to industrial robots, it integrates collaborative robotics functionalities and multimodal HMIs for an efficient collaboration with humans, with the promise to combine human creativity with the precision, speed and reliability of robots. Yet those functions are embodied in a simple design, much more compact and reliable than state of the art humanoid robots usually used to demonstrate such abilities, giving hope to reach real industrial usages in a shorter term. Thanks to the already existing results and future anticipated work, it is expected to reach and even surpass the speed of human operators, making the Robot Companion a real option for robot aided manufacturing.

Acknowledgment

The authors would like to thank the following colleagues at CEA LIST who contributed to the reflections and discussions on the Robot Companion and to some work not implemented on the first demonstrator and/or not emphasized in this paper: M. Anastassova, C. Andriot, R. Besançon, M. Boc, M. Boukallel, T. Boulmier, M. Darouich, G. De Chalendar, B. Gradoussof, M. Grossard, H. Le Borgne, A. Radermacher.

References

[1] SWMAS, 'Manufacturing barometer - Robot adoption: The SME challenge', *National report autumn 2019–20*, 2019, 24 pages.

[2] T. Asfour, M. Wächter, L. Kaul, S. Rader, P. Weiner, S. Ottenhaus, R. Grimm, Y. Zhou, M. Grotz, F. Paus, 'ARMAR-6 - A High-Performance Humanoid for Human–Robot Collaboration in Real-World Scenarios', *IEEE Rob. and Autom. Mag., 2019*, pp. 108-121.

[3] A. Kheddar, S. Caron, P. Gergondet, A. Comport, A. Tanguy, C. Ott, B. Henze, G. Mesesan, J. Englsberger, M.A. Roa, P.B. Wieber, F. Chaumette, F. Spindler, G. Oriolo, L. Lanari, A. Escande, K. Chappellet, F. Kanehiro, P. Rabaté, 'Robots in Arcraft Manufacturing, The Airbus Use Cases', *IEEE Rob. and Autom. Mag.*, 26(4), 2019, pp. 30-45.

[4] M. Bajracharya, J. Borders, D. Helmick, T. Kollar, M Laskey, J. Leichty, J. Ma, U. Nagarajan, A. Ochiai, J. Petersen, K. Shankar, K. Stone, Y. Takaoka, 'A Mobile Manipulation System for One-Shot Teaching of Complex Tasks in Homes', *Proc. IEEE Int. Conf. on Robotics and Automation*, 2020, pp. 11039-11045.

[5] X.V. Wang, A. Mohammed, L. Wang, 'Cloud-based Robotic System: Architecture Framework and Deployment Models', *Proc. Int. Conf. on Flexible Automation and Intelligent Manufacturing*, 2015, 8 pages.

[6] S. Profanter, A. Perzylo, M. Rickert, A. Knoll, 'A Generic Plug & Produce System Composed of Semantic OPC UA Skills', *IEEE Open Journal of the Industrial Electronics Society*, 2, 2021, pp. 128-141.

[7] P. Zimmermann, E. Axmann, B. Brandenbourger, K. Dorofeev, A. Mankowski, P. Zanini, 'Skill-based Engineering and Control on Field-Device-Level with OPC UA', *Proc. IEEE Int. Conf. on Emerging Technologies and Factory Automation*, 2019, pp. 1101-1108.

[8] N. Correll, K.E. Bekris, D. Berenson, O. Brock, A.Causo, K. Hauser, K. Okada, A. Rodriguez, J.M. Romano, P.R. Wurman, 'Analysis and Observations From the First Amazon Picking Challenge', *IEEE Trans. on Automation Science and Engineering*, 15(1), 2018, pp. 172-188.

[9] Y. Yokokohji, Y. Kawai, M. Shibata, Y. Aiyama, S. Kotosaka, W. Uemura, A. Noda, H. Dobashi, T. Sakaguchi, K. Yokoi, 'Assembly Challenge: a robot competition of the Industrial Robotics Category, World Robot Summit – summary of the pre-competition in 2018', *Advanced Robotics*, 33(17), 2019, pp. 876-899.

[10] E. Matheson, R. Minto, E.G.G. Zampieri, M. Faccio, G. Rosati, 'Human–Robot Collaboration in Manufacturing Applications: A Review', *Robotics*, 8(4), 2019, 25 pages.

[11] T. Kawasaki, 'These are the droïds you're looking for', *Highlighting Japan*, 95, 2016, pp. 8-9.

[12] S. Macenski, F. Martín, R. White, J. Clavero, 'The Marathon 2: A Navigation System', *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 2718-2725.

[13] X. Merlhiot, J. Le Garrec, G. Saupin, C. Andriot, 'The XDE mechanical kernel: Efficient and robust simulation of multibody dynamics with intermittent nonsmooth contacts', *Proc. Joint Int. Conf. on Multibody System Dynamics*, 2012.

[14] V. Weistroffer, F. Keith, A. Bisiaux, C. Andriot, Claude, A. Lasnier, 'Using Physics-Based Digital Twins and Extended Reality for the Safety and Ergonomics Evaluation of Cobotic Workstations', *Frontiers in Virtual Reality*, 3, 2022, 18 pages.

[15] I.A. Sucan, M. Moll, L.E. Kavraki, 'The Open Motion Planning Library', *IEEE Rob. and Autom. Mag.*, 19(4), 2012, pp. 72-82.

[16] M. Tan, R. Pang, Q.V. Le, 'EfficientDet: Scalable and Efficient Object Detection', *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 10778-10787.

[17] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, C.L. Zitnick, 'Microsoft COCO: Common Objects in Context', *Proc. Europ. Conf. on Computer Vision*, 2014, pp. 740-755.

[18] Y. Guo, M. Bennamoun, F. Sohel, J. Wan, M. Lu, '3D free form object recognition using rotational projection statistics', *Proc. IEEE Workshop on Applications of Computer Vision*, 2013, pp. 1-8.

[19] R.B. Rusu, N. Blodow, M. Beetz, 'Fast Point Feature Histograms (FPFH) for 3D registration', *Proc. IEEE Int. Conf. on Robotics and Automation*, 2009, pp. 3212-3217.

[20] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, J. Wan, 'Rotational Projection Statistics for 3D Local Surface Description and Object Recognition', *Int. Journal of Computer Vision*, 105(1), 2013, pp. 63-86.

[21] H. Bruyninckx, 'Open Robot Control Software: the OROCOS project', *Proc. IEEE Int Conf. on Robotics and Automation*, 2001, pp. 2523-2528.

[22] M.R. Pedersen, L. Nalpantidis, R.S. Andersen, C. Schou, S. Bogh, V. Krüger, O. Madsen, 'Robot skills for manufacturing: From concept to industrial deployment', *Robotics and Computer-Integrated Manufacturing*, 37, 2016, pp. 282-291.

[23] A. Benzine, F. Chabot, B. Luvison, Q.C. Pham, C. Achard; 'PandaNet: Anchor-Based Single-Shot Multi-Person 3D Pose Estimation', *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2020, pp. 6856-6865.

[24] G. Moon, J.Y. Chang, K.M. Lee; 'Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image', *Proc. Int. Conf. on Computer Vision*, 2019.

[25] C. Li, G.H. Lee; 'Generating Multiple Hypotheses for 3D Human Pose Estimation with Mixture Density Network', P*roc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 9887-9895.

[26] R. Liu, J. Shen, H. Wang, C. Chen, S.C. Cheung, V. Asari; 'Attention Mechanism Exploits Temporal Contexts: Real-Time 3D Human Pose Reconstruction', *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2020.