Proceedings Paper:
Pisanelli, G., Tymczuk, M., Douthwaite, J.A. orcid.org/0000-0002-7149-0372 et al. (2 more authors) (2022) ROSIE: A ROS adapter for a modular digital twinning framework. In: Proceedings of 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN). 2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN), 29 Aug - 02 Sep 2022, Napoli, Italy. Institute of Electrical and Electronics Engineers (IEEE) , pp. 1297-1304. ISBN 9781665406802

https://doi.org/10.1109/ro-man53752.2022.9900842

ROSIE: A ROS Adapter for a Modular Digital Twinning Framework*

Gianmarco Pisanelli[1], Mariusz Tymczuk[1], James A Douthwaite[2], Jonathan M. Aitken[2], James Law[3]

***Abstract*— As robotic systems become more interactive and complex, there is a need to standardise interfaces and simplify development processes. This is particularly pertinent in the field of manufacturing, where human-robot collaboration is on the increase, but where standards and proprietary software are key barriers to deployment and adoption.**

In this article we present the ROSIE Adapter, a general-purpose, modular adapter developed in ROS designed to support the creation and connection of industry-ready digital twins. Together with our previous work on the modular CSI digital-twin framework, we demonstrate how the ROSIE Adapter creates a versatile "plug-and-play" interface that simplifies the development of new robotic processes, and improves accessibility to novice users. Furthermore, the adaptor supports integration of intuitive interface devices, such as speech and augmented reality interfaces, which enable more natural collaboration. We describe the adaptor and its use in two real-world applications, demonstrate the ease of use via a three-day hackathon event, and provide results showing the faithfulness of the arising digital twins to their connected physical systems.

I. INTRODUCTION

Industry 4.0 is causing a gradual shift in the role of humans in the manufacturing industry. Today robots are omnipresent on the manufacturing shopfloor and, together with new smart sensor technologies, allow for intelligent automation processes. Humans are now expected to perform less physical labour and play a role of strategic decision makers and flexible problem-solvers [1]. Digital Twins (DT) can support the realisation of an integrated, flexible and collaborative manufacturing environment to aid the decision making process [2]. Digital data and AI enable users to dematerialise the real factory within a DT, wherein manufacturing processes are virtually simulated, monitored, and controlled [3]. However, this transformation is not always feasible and immediate [4], especially in manufacturing [2]. Heterogeneity of protocols, social acceptance, and need for human functionalities representation in the virtual space are major obstacles to the implementation of DTs in the manufacturing sector [5], [6], [7]. This added complexity mirrors a lack of professional skill sets for an industry that is in a constant state of evolution. In order to simplify the implementation of DTs a strong communication channel needs to be established between the user and the machine. Such communication should closely emulate human-to-human interaction to be the most effective. Humans are primates whose communication evolved through a phylogenetic history millions of years old into a combination of verbal exchanges, iconic gestures and pointing [8]. Visual cues also contribute to aid comprehension and can act as stimulator for emotions [9]. Natural Speech Interfaces (NSI) are capable of translating human language into machine instructions, enhance the control experience by generating verbal conversation with robotic platforms and enabling "hands-free" control of industrial machines. Augmented Reality (AR) interfaces offer an increase in information delivery and assimilation and reduction of the cognitive load by superimposing digital information on the physical world. They also allow the user to interact with the digital environment through gesturing and pointing, thereby bridging the rift between the real and virtual worlds [10]. A User Interface that combines Natural Speech and AR should therefore provide the most familiar interaction experience.

The CSI framework, introduced in our previous works [11], as well as solving the issue of handling heterogeneous data, offers an intuitive modular tool that can be used to assemble safety-critical digital twins with the aids of advanced visuals, basic physics simulation, connections to modern Human-Robot Interfaces (HRIs), remote operation control and training. The focus of this article, the *ROSIE Adapter* (named after the first platform on which it was adopted, see Section V-A), furthers that work by demonstrating how a physical connection with a real-world industrial system can be established taking advantage of both the modularity of the CSI framework and the modular construction of the Robot Operating System Architecture (ROS). In order to illustrate how quickly and easily industrial processes can be plugged into the framework, through the adapter, and integrated with intuitive interfaces (such as AR and NSI), we demonstrate its application on two diverse robotics platforms.

II. DIGITAL TWIN BACKGROUND

A DT consists of a digital informational construct about a physical system created as an entity on its own [12]. Information regarding the state of the physical and digital components can be interrogated simultaneously and compared. The user is presented with a single interface to the cyber-physical system (CPS) that persists throughout its operational lifetime [13]. DTs are crucial for the realisation

*This work was supported by the UKRI via The University of Sheffield's Internal Knowledge Exchange Scheme; Towards Turing 2.0 under the EPSRC Grant EP/W037211/1 and The Alan Turing Institute; the Assuring Autonomy International Programme, a partnership between Lloyd's Register Foundation and the University of York; HVMC Catapult

[1]Advanced Manufacturing Research Centre, The University of Sheffield, Sheffield, UK g.pisanelli@amrc.co.uk, m.tymczuk@amrc.co.uk

[2]Department of Automatic Control and Systems Engineering, The University of Sheffield, Sheffield, UK j.douthwaite@sheffield.ac.uk, jonathan.aitken@sheffield.ac.uk

[3]Advanced Manufacturing Research Centre and Department of Computer Science, The University of Sheffield, Sheffield, UK j.law@sheffield.ac.uk

of smart and interconnected factories which are at the core of Industry 4.0 [14]. Supported by a closed-loop bidirectional communication network, application of DTs promotes asset-twin co-evolution through real-time interaction, control and convergence in three key areas: within the physical space, between the physical and virtual spaces and between historical and real-time data [2]. The historical data collected during operations can also be used to train AI algorithms which can run prognostic and diagnostic analyses and finally optimise the process. Moreover, DTs can be adopted to carry out risk analysis and safety assurance tests, which can comprehensively validate the safety system of hazardous environments [15].

III. Related Work

This work builds upon established concepts within the DT community, and those of automated testing and industrial automation. While the DT concept has been around since the early 2000s [16], literary examples of DTs are typically specific in application and narrow in focus [17], [18].

Kousi [19] presented a DT with a dynamically updated virtual representation of the shopfloor, combining real time sensor data, resource data, and CAD models. The core model exploits the ROS architecture to achieve a repeatable deployment of software components. However, the model does not present a standardised abstraction level effectively limiting transferability and interoperability between systems. A similar issue is found in [20], [21], and [22]: they provide great examples of interoperability and hardware-software integration agents but their approach is limited to the specific use-case and lack a standardised meta-model. Some have addressed this issue by creating digital twin models based on the OPC UA protocol [23], which is the proposed industrial standard for cross-platform communication and data exchange from sensors to the cloud. However, given the vast variety of the manufacturing industry, the assumption that the sector will adopt a single standard is unlikely.

To summarise, current digital twin technologies lack easy access to complex definitions and behaviours, and a standardisation of modern industry protocols. In [11] we proposed a generalised framework, based in Unity® that uses a versatile and scalable approach. End-user interaction is simplified thanks to a “drop in place” philosophy built on a modular framework. In the following sections, we present our *ROSIE Adapter* which mirrors the same “plug-and-play” approach with physical robotic cells [24]. We will also discuss how such an approach has facilitated the integration of two intuitive interfaces.

IV. The ROSIE Adapter

A. ROS native architecture & ROS Control

The *ROSIE Adapter* uses ROS as an architectural foundation. ROS is the ‘de facto’ standard framework for robotic software development in academia [25]. Thanks to the effort of the ROS-Industrial Consortium, ROS is now making its way into manufacturing industries. As an open-source platform, ROS provides an intuitive middleware layer that supports many hardware and software integration services. It offers a fast track to novel technology, by leveraging common, shared R&D functionalities (motion planning, manipulation, 3D perception, kinematics, control and navigation), allowing the robot user to focus entirely on their own, unique application. The main advantage of using ROS for robotics application is its decentralised architecture. Many robots consist of a subset of networked computer hardware; decentralisation allows for communication with off-board computers for heavy computation commands, effectively enabling edge computing facilitated by a robust and hardware agnostic control mechanism.

The second generation of ROS, ROS2 was launched in December 2017 and has now reached a great level of maturity [26]. ROS2 was redesigned from the ground up to address some of the major vulnerabilities of ROS1. In particular, a lot of work has been done around security. Every message exchanged between ROS2 nodes is encrypted through a DDS layer. Real time functionalities have also been incorporated with the goal of strengthening compatibility with industrial applications. Although the ROSIE Adapter is built on ROS, steps have been taken to guarantee a smoother migration to ROS2 in the near future, including the adaption of the Unity Robotics Hub (URH; see Section IV-C).

The *ROS Control* package is the default API used by ROS to provide simple access to robotic actuators. *ROS Control* provides robotic software tools that directly talk to the robot drives and create intelligent robotic applications. Built on a nested structure of meta-packages, *ROS Control* offers an architecture of abstraction layers which allows to decouple controller code from actuator code. The *Robot Hardware Interface* sits on the first layer and handles the low level communication with the physical hardware (which could be based on EtherCAT, modbus, Arduino or any other protocol). The *Robot Hardware Interface* is the only layer that is robot specific. The second layer is occupied by the *Controller Manager*. This is responsible of launching the right type of controllers for the right type of robot (e.g. manipulator, AGV, or drone), as well as managing the controllers life cycle. The third layer allows the use of third party packages like *Moveit!* and the *navigation_stack*, which enhance robots with off-the-shelf complex motion planning algorithms and 3D visualisation interfaces.

The *ROSIE Adapter*, like *ROS Control* is robot agnostic and builds on the above architecture by adding a fourth layer that allows communication between the third party layer packages and the CSI digital twin framework. This effectively decouples the hardware, their configuration-dependant middleware, drivers and software versions, from their digital representations within our communal DT ecosystem.

B. CSI Framework

In our previous works [11], a modular digital twin framework was proposed for the purpose of Confident Safety Integration (CSI) of CPSs and digital twins. This framework required the development of a tool able to support a wide range of digital twin definitions, interactions and scenarios so that their safety may be explored and evidenced [1].

Within the CSI framework, each DT can be operated in three distinct modes, with varying relationships with the physical hardware, namely:

1) **Model** - Akin to a classical simulation mode, where the digital model has no exchange with the physical component.
2) **Shadow** - Unilateral communication mode, where the state of the physical entity is represented digitally within the framework, but without any feedback to the physical system.
3) **Twin** - A complete, bilateral, communication mode. Both the digital and physical component exchange data and can affect each other's state.

The *ROSIE Adapter* is designed to provide an interface to the physical twin to support the CSI framework's definitions of a *Shadow* and *Twin*. This concept is applied at the level of individual systems, and so the modularity of the CSI framework is reflected in the design of the *ROSIE Adapter*.

C. The Adapter architecture

In this section we will illustrate the architecture of the *ROSIE Adapter*. The *ROSIE Adapter* extends the previous ROS service, introduced in [11], by merging the CSI framework with the URH. The URH is a toolbox that builds on basic robotic concepts to assist with robotics simulation. In the URH the communication happens through a direct TCP endpoint, which is much faster and allows for exchange of large image data, which, in turn, allows use of sensors such as cameras. The Hub supports both ROS and ROS2.

The adapter builds on the *ROS Control* package by creating a fourth layer of abstraction, which separates the computation of inverse kinematics and motion planning from Unity® improving the overall DT performance and acting as an edge computing node. This layer consists of a *Motion Request Handler* (see Fig. 1) which has a bilateral communication channel. On one side motion requests coming from the *CSI service packet handler* are processed and delivered to *Moveit!*, which is run from a separate machine. On the other side, state information is collected from *Moveit!* and submitted to the *CSI service packet handler*, which in turn updates the DT definition.

The detailed interaction between the sub-modules is depicted in Fig. 2. The entity can be controlled through the adapter by the *Desired goal constructor*, which constructs the motion request by specifying a goal and the target Physical Twin (PT) that the user wants to drive. The *service request* node converts the motion request into a ROS service request containing the entity's name in a format *Moveit!* understands, (*group_name*), and the desired goal, mapped into its equivalent ROS coordinates (refer to Listing 1). The latter can be either a single point in space that the user wants the robot's end-effector to reach or a full defined trajectory.

[1]All data, samples and presented data can be found on the CSI project repository: *https://github.com/CSI-Cobot/CSI-artefacts*

Listing 1: ROS srv message structure

```
geometry_msgs/Pose[] targets
    std_msgs/String group_name
    std_msgs/String plan_execute
    ---
    std_msgs/String status
```

The goal can also be created visually by the *Waypoints interaction module*. This module simplifies the Human-Robot Collaboration factor thanks to the creation of graspable 3D spheres in the virtual environment generating a more intuitive way of controlling robots. The 3D spheres are translated into waypoints of a trajectory and can be directly dragged around the entity space with the mouse or any other form of graphical user interaction (see Section IV-E on Augmented Reality Interfaces). The service request then calls a *Moveit! request constructor* which directly handles the communication with the robot drives and controllers from the *ROS Control* first and second layer. *Moveit!* then computes the motion planning and inverse kinematics solution, and generates a sequence of joint states that the physical robot executes. The sequence of joint states is published to the *Joint State Publisher* node in real time. The *Joint Subscription* sub-module subscribes to the sequence of joint data and feeds it to DT updating the entity's state.

The *Motion Request Handler* is also capable of generating feedback on the status of the motion planning (e.g."planning computed successfull", "goal out of reach","failed to execute motion", etc.). The *Feedback Publisher* node provides this information in the form of messages, received from *Moveit!*, and submitted as a string of ROS service response messages (refer to Listing 1). The feedback is then visually displayed in the virtual environment. The decoupling between the ROS service response and the *Feedback Publisher*, clearly visible in Fig. 2, was created to allow access to the feedback string to other interfaces not natively embedded in the DT, for example a Natural Speech Interface capable of conveying vocal feedback (see Section IV-D).

A separate module was created to handle communication with mobile robots. This acts in a similar way, by sending velocity command requests to the ROS navigation stack and receiving /robot_pose information which are used to reconstruct the relative motion of mobile platforms with respect to the DT and PT environments. The power of this adapter is that the abstraction layer, as well as enabling the shadow twin mode, also grants the framework the ability of acting as an accurate simulation governed by ROS. In fact, the *Motion Request Handler* not only provides information regarding the real state of the robot, but it can also act as a source for external solvers allowing one to plan and visualise motions of entities in the CSI framework, before deploying them in the real world. This is a key feature that can be used for training purposes even when access to the physical asset is not possible (see Section V-A).

Finally, it is important to note that in the case of the PT being connected, the assumption is that the adapter is built

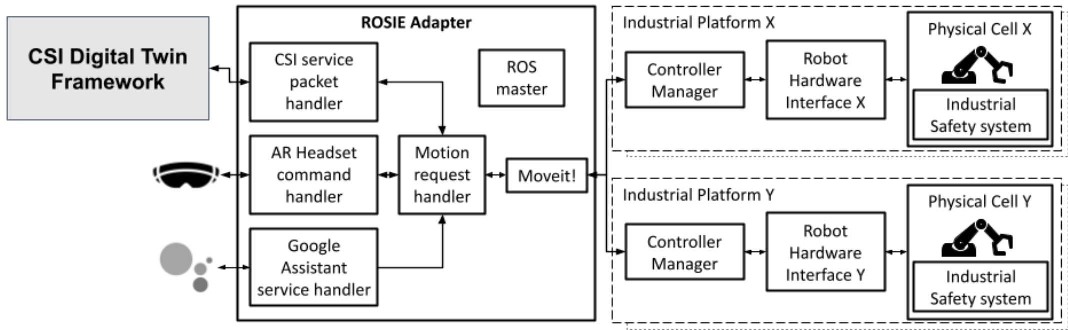

Fig. 1: The *ROSIE Adapter* integrates with the CSI framework at several points, utilising the integrated ROS-connection. The AR interaction and voice assistant are connected as an additional client and modules within ROS to provide connection to and from the devices.

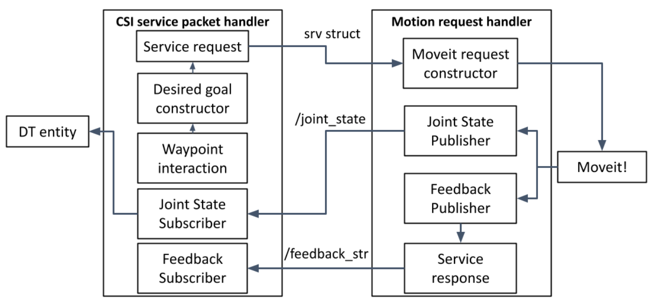

Fig. 2: A system overview describing the communication of state information between the CSI digital twin communication service and the *ROSIE Adapter* Motion Request Handler.

on top of the safety system which has a higher priority. This ensures that safety is always respected and also opens the possibility of running assurance analysis to validate that the physical safety system, based in the real-world, complies to current safety standards.

D. Google Speech Interface

The decoupling achieved by the CSI service handler allowed to separate the motion request from both *Moveit!* and Unity®. The *Motion Request Handler* is capable of accepting motion requests from any source assuming this follows the simple structure shown above. This means that other interfaces (aside from Unity) can be easily integrated to control any robot connected to the ROS network by requesting a pose goal.

As mentioned in Section I, two of the main obstacles to DTs are social acceptance and lack of skills [2]. Natural Speech Interfaces, combined with DT technologies could provide a highly intuitive interface that could help humans overcome distrust of robots, and simplify the Human-Robot Collaboration step by generating verbal conversation with robotic platforms and enabling “hands-free” control.

AI assistants are application programs capable of creating personalised conversations and complete tasks for their users. The Google Assistant (GA) is one of the most popular around the world. Google provides full access to the GA through a Google Assistant Service which exposes a low level API that lets the developer directly manipulate the audio bytes of an Assistant request and response. The GA Service offers minimal client-side processing given that all the natural speech processing is done on the Google Servers. This means that provided that it has access to a microphone and a speaker, virtually any electronic device can have a GA running on-board, without any other hardware requirement.

The GA requires three key elements to work: an utterance, an intent and a response. The utterance is the actual phrase that the user tells the Assistant. The utterance triggers an intent which indicates the real request of the user. The intent is then handled by the fulfillment to generate a response in the form of a JSON payload. The Assistant then converts the payload into rendered speech or multimedia output. Usually the fulfillment is automatically generated by the Google Server, but the GA Service allows generation of custom responses through Dialogflow. This, however, forces the user to stay on their platform, which cannot communicate with local machines. In order to escape the platform we force the Dialogflow fulfillment to trigger a webhook that, in turn, calls a custom local fulfillment.

The interaction process between the user and our GA interface is illustrated in Fig. 4. First, the user sends a vocal command to the Google Server through the API. The Google Server parses the audio into readable text. The text query is then sent to a custom Dialogflow AI model which is able to generate the intent (for example *home position*). The model is usually trained online with a set of possible phrases that the user could ask to generate the specific intent (e.g. *"Can you go to home position please?"*, or *"Drive to home"*). The model classifies the query into an intent and packages it into a fulfilment request, in the form of a JSON payload. The request is then fed to a local ROS node that checks it against a database of preset motion and events. If there is a match, the ROS node constructs the service request structure described in Section IV and sends a motion request to the *Motion Request Handler*. The feedback string that is normally returned by the *Motion Request Handler*, bypasses the Dialogflow fulfillment (which is expected by the GA by default), thanks to our webhook, to directly interact with the custom local fulfilment. Finally, the local fulfillment uses the feedback string to generate a JSON response, which then triggers a GA vocal response.

Although the interface is independent from the CSI framework, it can be run in parallel to establish a bidirectional verbal interaction between the user and the DTs.

E. Hololens 2 Interface

Another interface that the *ROSIE adapter* allowed us to explore is for the Hololens 2. AR is the medium that superimposes digital information on the physical world, thereby merging real and virtual worlds together. When combined with DT technology, detailed robotic processes can be visualised and detailed diagnostic information about the state and health of entities (both PT and DT) can be exposed for a richer and intuitive experience. Moreover, AR opens opportunities in the areas of safety visualisation, operator training and remote operation.

The level of abstraction described in Section IV-C allowed us to seamlessly integrate an AR interface based on the Hololens 2 with the CSI framework. The Mixed Reality Tool Kit was used to directly handle the user interaction [27]. The *AR headset command handler module* (refer to Fig. 1) uses MRTK tools (hand tracking, eye tracking, and UI controls) to automatically interpret user intention. This is then translated into a motion request that is then sent to the Motion Request Handler.

What the end user sees when putting on the AR headset is a virtual overlay of the robotic cell (see Fig. 3). The user is immediately prompted to position the overlay hologram of the DT on top of the PT using a 3-Point Positioning Method. The method involves the placement of 3 separate points using a snap gesture: The first point is the reference point in which the AR model is placed onto, the second point sets the rotation of the model relative to point 1, and the third point sets the orientation of the model based on the first 2 points. The visual interface is also augmented with diagnostic data (e.g. feedback on the motion planning request) and interactive tools that allow the user to directly interact with the PT. The user can generate an arbitrary number of 3D spheres that the *AR headset command handler module* interprets as waypoints of a desired trajectory (refer to *Waypoint Interaction module* in Section IV-C). The module is also capable of interacting with the robot in the 3 different DT modes simultaneously (see Section IV-B). So, for example the user can request a trajectory at the simulation level, before deploying it on the DT or PT, in order to validate the motion planning generated by the *Moveit!* library.

V. REAL INDUSTRIAL APPLICATIONS

So far we have discussed the architecture and different functionalities of the ROSIE Adapter. We have illustrated how the adapter builds from the modular ROS control API to plug the CSI framework to real world applications. In the following sections we will provide two examples where we have connected the framework to two of our industrial platforms. The examples serve to prove the modularity of the adapter and evaluate the DT performance.

A. Industrial Application I: ROSIE2

ROSIE2 is our ROS demonstrator built at Factory 2050[2]. As part of our ongoing work, ROSIE2 has provided a sophisticated test bench for investigation into safety in collaborative robotic processes. The demonstrator is also facilitating collaboration between academia and industry by providing a platform to demonstrate academic research outputs on industrially-relevant hardware and creating access to industry case-studies. ROSIE2 is also used as a training platform for robotic engineers. In July 2021, the platform was used to run the Manufacturing Robotics Challenge 2021, a hackathon-style event, part of the UK-Robotics and Autonomous Systems (UK-RAS) Network's Summer Showcase. 37 early-career robotics researchers and engineers, across 11 countries were tasked to complete an automatic surface sterilisation procedure using the KUKA LBR iiwa 14, a Robotiq gripper, a sponge, and a sanitising gel (see Fig. 6).

Participants were provided with a full instance of the CSI framework which facilitated communication with the real hardware at Factory 2050 in Sheffield.

The *Moveit! Request constructor* (see Section IV-C) was disabled and participants were tasked to recreate it in order to control the DT. The creation of the module from scratch served as a training exercise to teach participants how to control generic robots in ROS. Each team was also meant to use a Hololens 2 headset to validate their motion planning in simulation and avoid collisions using the *AR Headset command handler* described in Section IV-E. However, due to the lockdown restrictions the event was run virtually and the headsets could not be handed to participants. The abstraction layers created by the *ROSIE Adapter* enabled remote operation of the robotic platform during periods when physical access was limited.

The DT modes described in Section IV-B allowed participants to test their code in simulation first, prudently ensuring none of the safety protocols were violated, before safely moving the physical system, which in some cases was thousands of miles away from the operator. A short film was produced to document the event[3].

As mentioned in Section II, a full digital twin is achieved when the physical and digital systems are two separate entities that coexist and share information so that digital and physical components can be interrogated simultaneously. In order to confirm the full DT level was successfully reached, prior to the event, we instructed the robot to pick up the bottle and recorded its joint data both from the DT and PT. Fig. 5 shows that both the DT and PT use the information coming from the *Motion Request handler* and follow the same motion path. In the graph a small mismatch between the DT and PT joint angles is visible. The Mean Absolute Error (MAE) was calculated for all the seven joints, with the average recorded as 1.42 degrees.

[2]Factory 2050 is a part of the Advanced Manufacturing Research Centre (AMRC) in Sheffield https://www.amrc.co.uk/facilities/factory-2050

[3]The UK-RAS Manufacturing Robotics Challenge 2021 `https://youtu.be/EpjRU0lXBAQ`

Fig. 3: An example of the interface presented to the user while wearing the Hololens 2 headset. The blue panel can be used to create waypoints for a desired trajectory and display feedback on the motion planning operation.

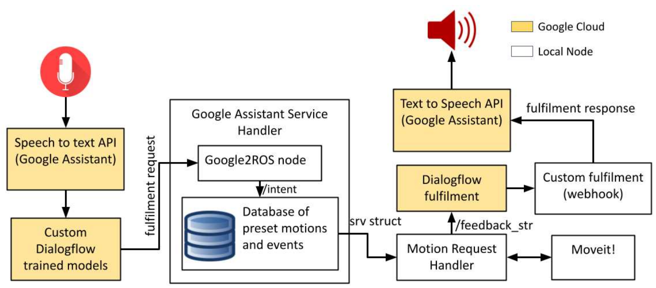

Fig. 4: An overview of the GA interface and interaction with the Google Cloud API. Vocal commands are parsed on the cloud to generate a motion intent. This is then handled locally to originate a motion request. Finally a feedback string is sent back to the API to generate a verbal response.

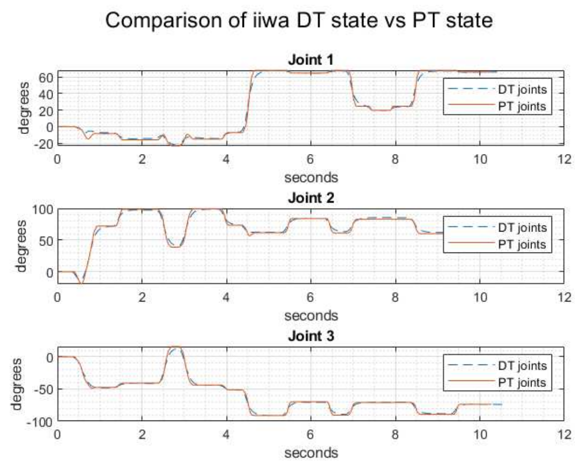

Fig. 5: Before the hackathon event was run, the KUKA iiwa robot was instructed to pick a bottle in order to record and compare DT and PT joint data. Only the first 3 robot joints are shown for brevity.

The error is due to internal Unity physics engine. The engine approximates universal forces in nature such as gravity, acceleration, and friction to recreate a close representation of the real world. In this work the engine properties were left to the default value. In order to exchange the most accurate information between the physical and virtual world, correct the mismatch, and reduce the error, we plan to tune these parameters as part of ongoing work.

At the end of the challenge, we asked the participants to complete a feedback form to relate their experience. Of the 9 responses received, 100% of the participants indicated being "Extremely Satisfied" or "Very Satisfied" with the event, challenges, and platform. Given many participants had only learnt ROS during the event, this suggests the adaptor is sufficiently intuitive for novice users (although dedicated evaluation is required to confirm this).

B. Industrial Application II: Gear Cutting

The second industrial platform that we used as a test bed for the *ROSIE Adapter* involves a mobile manipulator robot. The AMRC gear centre uses the Iconsys iAM-R robot for machine tending and flexible manufacturing of small gear shafts. The platform combines a UR10e collaborative manipulator with a MiR100 AGV for adaptive solutions to manufacturing and logistical processes. The task the iAM-R is used for involves the following steps:

1) A bespoke machine manufactures small gear shafts
2) The shafts are automatically ejected via a conveyor
3) The iAM-R retrieves a shaft from the parts catcher
4) The iAM-R delivers the component to a tumbler

In order to recreate the task inside the framework, we firstly created a digital entity of the robot. The iAM-R does not belong to the pool of numerous commercial industrial robots that support ROS drivers, therefore before the integration with the *ROSIE Adapter* could begin a ROS driver that enables direct communication with the robot actuators was created. The iAM-R ROS driver and controller were constructed by merging the separate UR10e and MiR100 ROS controllers and drivers created by the ROS community. The ROS *Controller Manager* played a key role in ensuring that no conflict occurred during the merge. The *CSI service packet handler* was then expanded to accommodate for velocity command messages which ROS uses to control robotics platforms. A virtual joystick was also included in the CSI framework to allow the user to jog the platform directly from the virtual environment. Once the ROS controller was created, the *ROSIE Adapter* enabled full control of the iAM-R digital

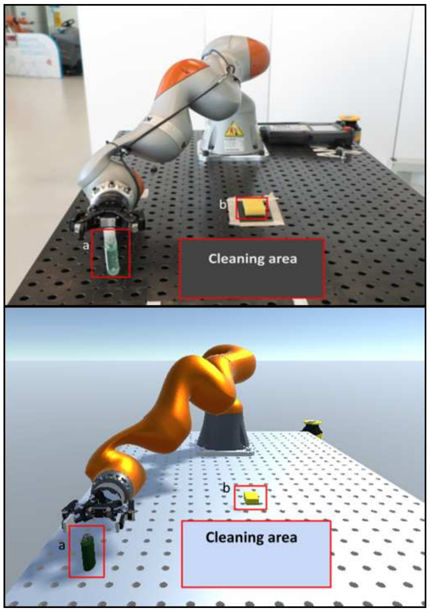

Fig. 6: A comparison of UK-RAS Manufacturing Challenge DT and PT. Participants were asked to pick up the bottle of sanitising gel (a), squeeze the liquid on the table and then place the bottle back. The robot should then pick up the sponge and scrub the cleaning area.

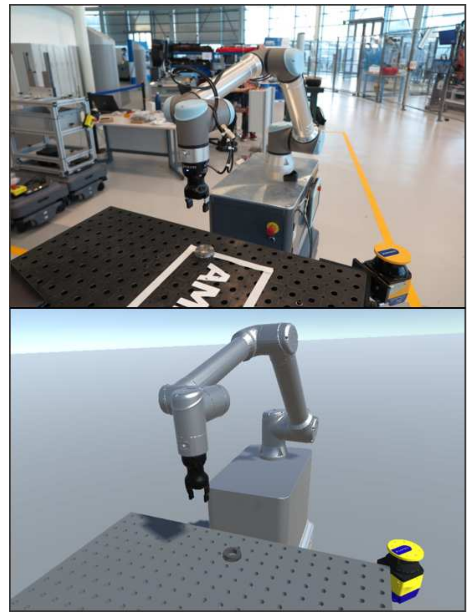

Fig. 7: A comparison of the Gear Cutting DT and PT. (Top) the Iconsys iAM-R mobile robot placing a component on the table, (bottom) the component is placed virtually on the same table.

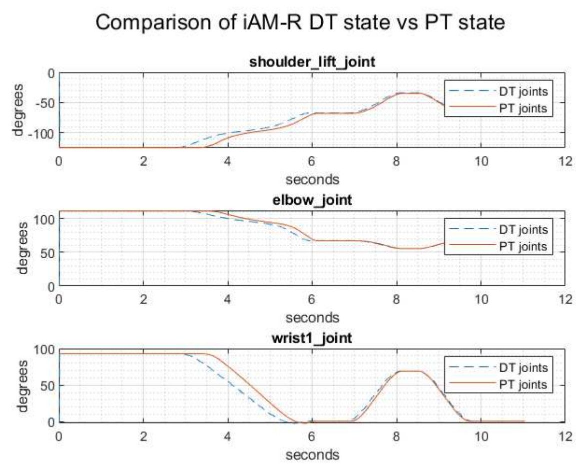

Fig. 8: DT and PT joint data recorded while the iAM-R platform places the gear on the table. Only 3 joints are shown for brevity.

twin from the CSI framework. Very little configuration was required to connect the adapter to the new robot confirming the real "plug-and-play" nature of the adapter. Moreover, the integration with the adapter allowed us to enhance the iAM-R into a more collaborative and autonomous platform. After a scan of the factory floor the robot was taught the positions of key areas in the factory (e.g. parts catcher and tumbler). The *Google Speech Interface* then enabled the robot to understand verbal requests. Therefore, rather then manually programming the robot, once the manufacturing of the component is completed, the human operator can simply verbally request the robot to pick up the object and move it to a desired location, greatly simplifying the human-robot interaction experience. Once again, in order to validate that the full digital twin mode was reached, the DT and PT data was recorded during a place operation. Fig. 8 shows that both the DT and PT executed the same motion during the place task. However, this time a noticeable offset was observed in the first 6 seconds of the process. The MAE was calculated for this operation, with the average joint error between the physical iAM-R platform and its Digital Twin measured as 1.71 degrees. After an extensive investigation we concluded that the offset was caused by the inertial data that was generated during the creation of the ROS controllers. The virtual model underestimates the inertial loads of the robot joints causing the DT to reach the desired position a few milliseconds before the PT. However, the graph also shows that the DT uses the PT information, mainly the joint state data, to correct the trajectory after the initial error. The correction is achieved thanks to the *Joint State Subscriber*, described in Section IV, which ensures that a closed loop is established between a DT entity request and the PT live state.

VI. CONCLUSIONS

In this paper we presented a ROS adapter that provides a versatile "plug-and-play" interface to simplify the development and deployment of novel robotic manufacturing processes, and the integration of intuitive interfaces for human-robot collaboration. Paired with our digital twinning framework, which standardises communication and physical system representation across different hardware platforms and provides validation for safety controllers, the two lower the barriers to the development and control of human-robot collaborative processes.

The "plug-and-play" nature and versatility of the adapter has been demonstrated in two applications involving two different robotic platforms and processes: in the first, the ease of use and accessibility was demonstrated through its use in an international robotics hackathon, enabling novice users to rapidly and remotely develop, deploy, and test a controller for a collaborative robot; in the second, the adapter was used to develop a mobile collaborative robot process, and integrate a speech interface to enable intuitive human-robot interaction.

For both studies we provided results to compare the motion of the physical robot with data within the digital twin. The data confirms the similarities between the two representations, but also highlights the need for careful tuning of parameters to ensure best fit to the physical system properties. A benefit of the adapter and digital twin approach is that the data from the physical system enables the connected digital model to converge with the physical system over time, even when parameters are not tuned correctly (compared to segregated simulation models which provide less accurate system information).

Whilst we have demonstrated the capabilities of the adapter in reducing development barriers for human-robot collaboration, we plan to formally evaluate the intuitiveness of the interfaces against a benchmark in the near future. With the platforms and adapter in place, we will next turn our attention to their use in assuring the safety of human-robot collaboration, to further evaluating user trust, and also migrate the Adapter to ROS2.

ACKNOWLEDGMENT

We would like to thank the (AMRC) IMG Digital Team for their work on the 3 Point Positioning Method and Ben Cook (AMRC Gear Centre) for providing access to the iAM-R platform.

REFERENCES

[1] A. K. Inkulu, M. V. A. R. Bahubalendruni, A. Dara, and S. K, "Challenges and opportunities in human robot collaboration context of industry 4.0-a state of the art review," *Industrial robot*, vol. 49, no. 2, pp. 226–239, 2022.

[2] I. Onaji, D. Tiwari, P. Soulatiantork, B. Song, and A. Tiwari, "Digital twin in manufacturing: conceptual framework and case studies," *International journal of computer integrated manufacturing*, 2022.

[3] H. Ahuett-Garza and T. Kurfess, "A brief discussion on the trends of habilitating technologies for industry 4.0 and smart manufacturing," *Manufacturing letters*, vol. 15, pp. 60–63, 2018.

[4] L. Pérez, S. Rodríguez-Jiménez, N. Rodríguez, R. Usamentiaga, and D. F. García, "Digital twin and virtual reality based methodology for multi-robot manufacturing cell commissioning," *Applied sciences*, vol. 10, no. 10, p. 3633, 2020.

[5] F. Tao, H. Zhang, A. Liu, and A. Y. C. Nee, "Digital twin in industry: State-of-the-art," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2405–2415, 2019.

[6] M. Blum. and G. Schuh., "Towards a data-oriented optimization of manufacturing processes - a real-time architecture for the order processing as a basis for data analytics methods," in *Proceedings of the 19th International Conference on Enterprise Information Systems - Volume 1: ICEIS,*, INSTICC. SciTePress, 2017, pp. 257–264.

[7] Y. Cheng, Y. Zhang, P. Ji, W. Xu, Z. Zhou, and F. Tao, "Cyber-physical integration for moving digital factories forward towards smart manufacturing: a survey," *International journal of advanced manufacturing technology*, vol. 97, no. 1-4, pp. 1209–1221, 2018.

[8] K. Zuberbühler, "Referential signalling," in *Reference Module in Life Sciences*. Elsevier, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978012809633801116X

[9] A. Tchamova, J. Dezert, N. Bocheva, P. Konstantinova, B. Genova, and M. Stefanova, "A study on human learning ability during classification of motion and colour visual cues and their combination," *Cybernetics and information technologies : CIT*, vol. 21, no. 1, pp. 73–86, 2021.

[10] C. K. Sahu, C. Young, and R. Rai, "Artificial intelligence (ai) in augmented reality (ar)-assisted manufacturing applications: a review," *International journal of production research*, vol. 59, no. 16, pp. 4903–4959, 2021.

[11] J. Douthwaite, B. Lesage, M. Gleirscher, R. Calinescu, J. M. Aitken, R. Alexander, and J. Law, "A modular digital twinning framework for safety assurance of collaborative robotics," *Frontiers in Robotics and AI*, vol. 8, 2021.

[12] M. Grieves and J. Vickers, *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems*. Springer International Publishing, 2017, pp. 85–113.

[13] W. Kritzinger, M. Karner, G. Traar, J. Henjes, and W. Sihn, "Digital twin in manufacturing: A categorical literature review and classification," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018.

[14] Y. Bai, J.-B. You, and I.-K. Lee, "Design and optimization of smart factory control system based on digital twin system model," *Mathematical problems in engineering*, vol. 2021, pp. 1–16, 2021.

[15] S. Kaewunruen, J. Sresakoolchai, W. Ma, and O. Phil-Ebosie, "Digital twin aided vulnerability assessment and risk-based maintenance planning of bridge infrastructures exposed to extreme conditions," *Sustainability (Basel, Switzerland)*, vol. 13, no. 4, pp. 1–19, 2021.

[16] E. H. Glaessgen and D. S. Stargel, "The digital twin paradigm for future NASA and U.S. Air force vehicles," in *Paper for the 53rd Structures, Structural Dynamics and Materials Conference; Special Session on the Digital Twin*. Honolulu, HI: AIAA, 2012, pp. 1–14.

[17] V. Damjanovic-Behrendt and W. Behrendt, "An open source approach to the design and implementation of digital twins for smart manufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 32, no. 4-5, pp. 366–384, 2019.

[18] G. P. Agnusdei, V. Elia, and M. G. Gnoni, "Is digital twin technology supporting safety management? a bibliometric and systematic review," *Applied Sciences*, vol. 11, no. 6, 2021.

[19] N. Kousi, C. Gkournelos, S. Aivaliotis, C. Giannoulis, G. Michalos, and S. Makris, "Digital twin for adaptation of robots' behavior in flexible robotic assembly lines," *Procedia Manufacturing*, vol. 28, pp. 121–126, 2019, 7th International conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2018).

[20] J. A. Erkoyuncu, I. F. del Amo, D. Ariansyah, D. Bulka, R. Vrabič, and R. Roy, "A design framework for adaptive digital twins," *CIRP Annals*, vol. 69, no. 1, pp. 145–148, 2020.

[21] A. AboElHassan, A. Sakr, and S. Yacout, "A framework for digital twin deployment in production systems," in *Advances in Automotive Production Technology – Theory and Application*, P. Weißgraeber, F. Heieck, and C. Ackermann, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 145–152.

[22] C. Liu, P. Jiang, and W. Jiang, "Web-based digital twin modeling and remote control of cyber-physical production systems," *Robotics and Computer-Integrated Manufacturing*, vol. 64, p. 101956, 2020.

[23] H. Park, A. Easwaran, and S. Andalam, "Tila: Twin-in-the-loop architecture for cyber-physical production systems," in *2019 IEEE 37th International Conference on Computer Design (ICCD)*, 2019, pp. 82–90.

[24] J. M. Aitken, S. M. Veres, and M. Judge, "Adaptation of system configuration under the robot operating system," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 4484–4492, 2014.

[25] R. Sheh, H. Komsuoglu, and A. Jacoff, "The open academic robot kit: Lowering the barrier of entry for research into response robotics," in *2014 IEEE International Symposium on Safety, Security, and Rescue Robotics (2014)*, 2014, pp. 1–6.

[26] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.abm6074

[27] Microsoft. (2022) What is the mixed reality toolkit. [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/mrtk-unity/?view=mrtkunity-2021-05