DTPAAL: Digital Twinning Pepper and Ambient Assisted Living

Lucia Cascone, Michele Nappi, Fabio Narducci and Ignazio Passero

Published in: IEEE Transactions on Industrial Informatics.

The Version of Record is available online at: doi:10.1109/TII.2021.3090363

***Abstract*—Pepper is a humanoid robot capable of expressing body language, perceiving and interacting with its surrounding environment thanks to a wide set of sensors and actuators and exposing capabilities and high-level interfaces for natural interaction with humans. In this paper, we present the development of VPepper, the Pepper virtual replica, by describing experiences focused on the interaction of the Digital Twin with the replicas of the smart-objects in a smart-home. Pepper robot has been featured with arms and hands, but its motors and actuators cannot support intensive experimental sessions and training procedures to learn how safely touch objects. Here, Digital Twin metaphor plays a crucial role. By a virtual and reliable replica of the robot, machine learning procedures can be seamlessly moved to/from the digital twin with a significant speedup and preventing the physical robot from deterioration. As a practical application, the reported case study is inspired to Ambient Assisted Living in elderly assistance. The experience, as well as the entire design and development process, have revealed VPepper and the smart environment to offer interesting opportunities for the physical accuracy of the simulation and for the availability of Machine Learning instruments that may be converted and adopted for real settings. A final empirical evaluation, performed involving 25 volunteer care givers, confirms the perceived value and the potential usefulness of the system.**

***Index Terms*—Social robot, Digital Twin, Modeling and Simulation, Machine Learning, Unity3D.**

I. INTRODUCTION

SOCIAL robots are systems designed and developed to interact with humans and other robots. Pepper is the first social humanoid robot in the world capable of recognizing basic human faces and emotions. Pepper is nowadays used in private and public contexts [1]. In this work we have chosen to simulate the robot and the surrounding smart devices by modelling their Digital Twins. In recent years, Digital Twin (DT) concept has become a strategic technological trend. The definition of the term was given for the first time in 2001 by Michael Grieves. As a generic definition, reported in [2], we can say that "a DT is a comprehensive software representation of an individual physical object. It includes the properties, conditions, and behavior(s) of the real-life object through models and data. A DT is a set of realistic models that can simulate an object's behavior in the deployed environment. It represents and reflects its physical twin and remains its virtual counterpart across the object's entire lifecycle"[3]. DT integrates the concept of the Internet of Things (IoT), Artificial Intelligence (AI), Augmented Reality (AR) interaction, Machine Learning (ML), and the use of software to create living digital simulation models. These simulated models continuously update, almost in real-time, changing when their physical parts change. The concept of real-time synchronization between the virtual and the real part is a rigorous requirement for the application of the DT. The purpose of this work is to simulate the Pepper robot via its DT, called VPepper. Pepper robot has been featured with hands which are just intended to reproduce gestures and not touching objects. In this work, we exploited the DT metaphor to train the robot to perform this task safely. Training the physical robot is however an intensive and deteriorating task for its motors and actuators. A similar problem has been explored in [4] where the DT approach has been used to virtually train a manufacturing robotic arm. They used reinforcement learning in virtual space and then map this learnt behaviour to a physical robot arm. A similar approach, implemented in [5], proposes virtual simulations as test-bed for numerous possible events and scenarios without physically collecting the data for the experimentation. Also [6] demonstrates how a humanoid robot learns to lift a weight of unknown mass through autonomous trial-and-error search by simulations with its virtual twin. In our work, we implemented the approach of DT by replicating physical humanoid robot by VPepper that enables to perform intensive experimental sessions and learning activities without working on the physical counterpart. This prevents the robot from the excessive deterioration of the mechanical parts as well as it allows to run experimental sessions in parallel. Moreover, experiments on the physical robot are possible at condition that the battery level is high enough to provide electrical power to its motors. Battery duration is limited to about 8 hours in case of intensive use of the robot. This is another advantage of DT since VPepper does not suffer from this limitation. Being able to demonstrate the possibility of teaching Pepper to safely touch objects and people in the environments opens to the use of Pepper robot as personal assistant at home for fragile categories, like elderly of impaired people. The availability of VPepper prevents the physical robot from massive experimentation by implementing

L. Cascone, M. Nappi, F. Narducci and I. Passero are with the Department of Computer Science, University of Salerno, Salerno, Italy, e-mail: (lcascone@unisa.it, fnarducci@unisa.it, mnappi@unisa.it ipassero@unisa.it)

and testing useful techniques that can be virtually trained and "safely" exported to the physical twin when ready. The goal is to combine the native skills of the robot with the learnt ability to soft touching objects and people in the environment. The smart environment, consisting in an interconnection of sensors which share data with the robot, has its DT too so that a seamlessly connection among real and virtual may be achieved. The work is organized as follows. We cover the background for the paper in Section II. In Section III we describe the physical eco-system made by the robot Pepper and the surrounding smart devices and in Section IV we present VPepper and the DTs of the smart-objects. Section V describes a preliminary evaluation of the DT ML controls, and in Section VI we propose a case of study organized for assessing perceived qualities of the system. Section VII concludes the work.

II. Related Work

The DT has multiple uses and it may bring several advantages in disparate industrial sectors. Perception and cognitive capacity towards more autonomous and intelligent robotic systems are examples of such fields where DT can be successful, without a real-time connectivity to the physical system. The DT concept plays an important role in Industry 4.0 [7]. In modern industry, the information has been sufficiently shared among the production equipment, intelligent subsystems and mobile devices via advanced network technology [8]. This condition creates the assumption to make DT approaches implementable in industrial fields. One of the areas in which DT finds greater application is the manufacturing industry, so introducing the definition of smart manufacturing [9]. In the manufacturing domain, it can be used to highlight all the unwanted and unexpected aspects in the life cycle of a product.

In healthcare it can help to define the difference between normality and deviations [10] or cooperate with IoT systems in patient monitoring [11]. Virtual simulations have been extensively adopted in industrial robotics with the goal of improving the performances of industrial processes [12]. In [13] the authors demonstrate the usefulness of DT to support the design, development and management of human-robot production systems. A key problem for human robotic interaction is ensuring that movements preserve the safety of personnel and the robot itself. Dröder et al. [14] develop a part of an experimental simulation platform for human robot interaction with a machine-learning approach for obstacles detection. In recent years, the research areas of the DT have been expanded to include other aspects, such as cybersecurity [15] or the production planning [16]. In [16] the authors explain how to apply DT concept to a body-in-white production system for conceptual and planning projects. The described system uses current information from the cybernetic system to update the planning project. Instead, in [15] the authors aim at integrating DT's concepts of productivity with those of security, improving cybersecurity in production environments. In a virtual environment, cobotic production systems can improve their ability to counteract manipulation attempts to block production or cause damage to other machines or humans. The term cobotics refers to "cooperation" and "robotics" and indicates the collaboration between humans and robots. The purpose of cobots is to automate a wide range of activities and to carry out work in a closer collaboration with people. Cobots have been demonstrated to be crucial in medical fields, where for example surgeons can collaborate closely with a robotic arm (which therefore needs to necessarily is a cobot) [17]. In [17] the authors propose a prototype DT system composed of a robotic arm and HTC Vive virtual reality system connected over a 4G mobile network for an application of remote surgery. As a surgical device was used the Universal Robots UR3 robotic arm. Instead, in [18] the authors describe a Universal Robots UR10, an industrial robot arm whose DT receives continuous and real-time information on the status of robotic arm joints. For the control of humanoid robotic arms, in [19] the authors combine the concept of DT technology with Deep Reinforcement Learning (DRL). A humanoid robot is used with multiple-motion mode and a sophisticated mechanical structure. Their proposal is a robot joint trajectory planning scheme for multitasking-oriented scenarios.

III. The Physical Environment

Pepper is a humanoid robot created and developed by SoftBank Robotics Europe headquartered in Paris. The robot can use the images of this camera to perceive depth and locate itself, using some landmarks in the environment where there are no obstacles nearby. Pepper was developed with the aim of communicating with people in the most natural way possible [20]. To obtain such an interaction, it has been equipped with several sensors that allow it to perceive external stimuli and also communicate with the environment [21].

A. *Pepper Object Touching*

Pepper is not really meant to grab arbitrary objects. This because grabbing an object consists in several preliminary operations involving different sensors, from cameras to pressure sensors. Even if Pepper's hands are such that this task can be partially realised, its hands are rather supposed to enable human-like interactions. However, its actuators and motors can apply forces that might represent a risk when interacting with people and with surrounding objects. Pepper could be trained to softly touch objects so that it can safely touch humans subjects as well as fragile objects. The first task to perform is recognising the object, a three-step process: (i) the training phase, (ii) the creation of the dataset in which sampled images of the object are collected and (iii) the detection of the object. In the storage phase, several images of the objects are collected and stored, using cameras that allow the robot to frame the object in the foreground and from different perspectives. In this way, it is possible to improve the efficiency and quality of the digital archive that is created with new inserted images. Once an object has been recognized, to reach and grasp it the motors that move Pepper have to be activated. The robot can touch the object with both hands or leave one arm motionless and touch the object with the other. In both cases, Pepper must adjust the arm(s) and bend it with the right angulation to touch the object. Pepper has several modules for the implementation

Fig. 1. One of the Movidius smart-cameras.

of these tasks. The most important and used are ALMotion for robot movement, ALVisionRecognition for recognizing different images and ALSensor, raising events corresponding to Pepper's sensors.

B. The smart-object fog

Pepper on-board sensors are integrated with IoT smart-objects, as shown in [21]. We adopt Raspberry boards equipped with cameras and Movidius Neural Compute Sticks (NCS) for obtaining several DL inferences on images of the smart home. The Intel Movidius NCSs accelerate the inferences and enable to process high frame rates. The devices enable the selection of several pre trained architectures, aiming at inferring different classifications: SSD Mobile Net, caffe Mobilenet, Agenet, Gendernet and other custom models [22]. The Raspberry smart-objects expose a MQTT broker: inferred data are organized in Topics and sent via MQTT protocol on the local network.

IV. The Virtual Environment

VirtualPepper (VPepper) is the DT of Pepper and is connected to the fog of smart devices sharing information with the real robot. Starting from a meshed model of the robot, the surfaces have been organized in Blender aiming at obtaining the right hierarchy for the articulations, as prescribed by the adopted joints.

A. The simulation engine

VPepper and its smart home environment are modeled in Unity (commonly known as Unity3D) as 3D engine and physics simulation environment. Unity is really popular and adopted for the development of video games and other interactive content, such as architecture visualizations or 3D animations, executed in real-time. Pepper is already supported by several software development kits and emulators[1]. An example is Webots[2] that adopts the Open Dynamics Engine[3] for physics and the OpenGL 3.3 engine for rendering. Webots offers a large asset library made of components ranging from robots, sensors, actuators, objects and basic materials. In our case, the need to connect VPepper to the Raspberry smart-objects and the adequate support and the instruments the

[1]http://wiki.ros.org/Aldebaran
[2]https://www.cyberbotics.com#features
[3]https://bitbucket.org/odedevs/ode/src/master/

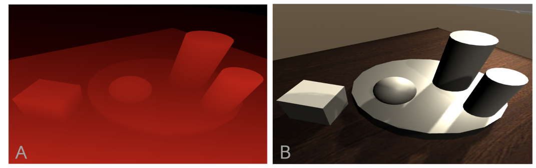

Fig. 2. The Depth Map generated by $DEPTH\_AUTO$ color format (image A) compared with the RGB one (image B).

Unity3D offers respect our simulation aims (better detailed in the following), let us choose this gaming engine.

1) The Physics: Unity allows users to set some of the physical engine parameters so that they can optimize the physical engine according to their needs. As for the proposed DT for the Pepper robot, the Unity environment offers an exceptional set of tools. These range from direct or ray-cast observations, which can easily mimic on-board proximity and distance sensors, to depth cameras and careful physical modeling of joints and motors. The joints adopted for simulating Pepper articulations are the new ArticulationBody. Starting from version 2020.1 beta Unity adopts the new Nvidia PhysX 4.1 engine with excellent performance and realism for modeling kinematic chains, like rag dolls, robotic arms, or complex physical mechanisms. The novelty introduced by the 020.1b version is the "articulation" that organizes a set of segments in a parent-child relationship and permits mutually constrained motion. The ArticulationBody joints enable us to easily model VPepper and to create simulations that accurately reflect the real-world physics.

2) ML-Agents: Another interesting Unity feature is its possibility to be adopted as a Deep Learning training/inference virtual platform. In 2018, Juliani et al. [23] presented a new toolkit for Machine Learning Agents (ML-Agents) in Unity3D. ML-Agents is an open source toolkit that allows simulations to be used as environments for training intelligent agents implementing reinforcement, curriculum and imitation learning methods.

An interesting feature of ML-Agents Unity package is its natural capability to train the VPepper joint controller (see Sec. IV-A1), which, with the capability of performing a parallel training phase, is the main reason to choose Unity for the DT of the Pepper robot.

B. VPepper sensors

Aiming at easily modelling the VPepper and at connecting the DT with the smart-objects and the real robot, we adopted Unity 3D because of the versatility and adaptability to external services, like the one exposed by the smart-objects. The infrared, laser, and sonar sensors that equip the real robot, are emulated by RayPerceptionSensor3D[4]. These components naturally integrate with ML-Agents and can be adopted as observation for training the agents controlling the DT.

The 3D depth sensor is emulated by a specific Unity color format ($DEPTH\_AUTO$ used for emulating the depth

[4]https://docs.unity3d.com/Packages/com.unity.ml-agents@1.0/api/Unity.MLAgents.Sensors.RayPerceptionSensor.html

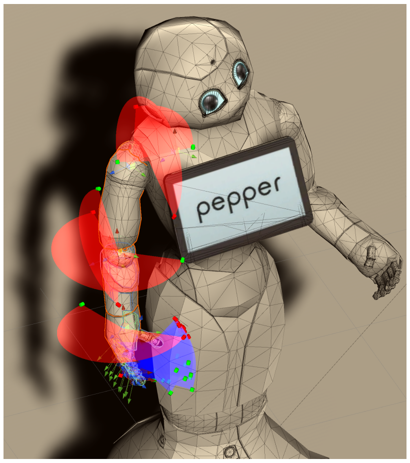

Fig. 3. The ArticulationBody joints adopted for modeling VPepper.

sensor). Figure 2 shows the same image emulated by the depth sensor and a classic RGB Unity camera shot. As it is possible to see in the figure, surface materials and shadow are not considered in depth images: brighter pixels represent closer points and pixels with lower luminosity values depict far objects.

C. VPepper actuators

VPepper joints have been modeled via ArticulationBody with the proper degrees of freedom and angular limits. In particular, the shoulders and elbows have been modeled with pitch and roll degrees of freedom and the wrist only with yaw movement capability. All the angular limits have been set as specified in the relative Aldebaran documentation[5]. Each ArticulationBody joint may be controlled by a linear drive for each free movement direction and towards linear and angular targets.

Figure 3 depicts the articulation hierarchy adopted for modeling VPepper, highlighting in blue the angular limits of finger articulations and in red the ones relative to the other arm articulations.

D. VPepper: object touching

The proposed action consists of bringing the robot hand closer to the target object. It consists in typical Inverse Kinematic (IK) problem since VPepper's arm has to rotate its 6 joints that control the six degrees of freedom of the hand so that this last one can approach the object and touch it.

Even if this action involves only the arm articulations, the "object touching" requires to control all the joints that implement the VPepper arm, except the one that is the root of the hierarchy. In virtual simulated environments, the IK solution can be feasibly addressed. In this study, the task is made more complex due to the fact that what VPepper performs in the simulated environment should be seamlessly replicated by the physical counterpart. Consequently, if virtual capabilities of VPepper are limitless, the physical robot has limitations in

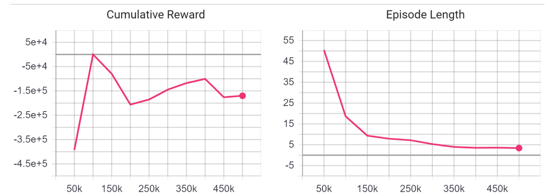

Fig. 4. The TensorBoard report of Cumulative Reward and Episode length for a 500K training session.

the amplitude/velocity of movements. In presence of obstacles or when the object is not ideally in front of the robot, the movements that VPepper has to perform must consider the physical constraints of Pepper robot. We developed a solution based on Reinforcement Learning by implementing the model as suggested by Unity documentation[6] fully exploiting the capabilities of Nvidia PhysX 4.1 engine and ML-Agents package. At the beginning of every training iteration, the robot arm is extended downwards along with the body and the hand of the DT is lower than the table that is modeled solid. In this way, the control should avoid keeping the hand trapped by the table on the shortest path toward the target. A machine learning process based on reinforcement has been set up. The learning technique exploited in this work is the Proximal Policy Optimization (PPO) [24], [25]. In the Reinforcement Learning, the behavior is controlled by an agent that observes some properties in the environment and takes decisions trying to maximize rewards. In the proposed VPepper object touching training, the reward at each step is the negative distance between the DT hand and the target object. In this way, moves that bring the hand far from the target or lock the arm on a critical position (i.e., under the table, or behind the head) are discouraged.

The net which controls the arm has 6 main paths that organize the flow of 69 sources of information collected via observations of joint angular values, spatial orientations, and relative distances. On the base of these observations, the agent should reach the target (pushed by a reward score of 1), or it accumulates negative small penalties proportional to distance and time spent. It is important to point out that the high dimension of the problem is typical in the case of multi-articulated controls, given the high numbers of position and orientation properties to track. However, the search space dimension is reduced because of the limits enforced on the possible movements of the robot's arm.

Figure 4 shows the TensorBoard report describing the training session performed on randomly positioned objects. For every Episode (50.000 actions), the picture reports, on the left side, the Cumulative Reward and, on the right side, the duration in minutes. The entire training lasted about 2 hours on an Alienware 17 3.60GHz, Ram 32 Gb, and Nvidia RTX 2080 8 Gb. As it is possible to notice, the reward is oscillating, but often with a positive trend (it is important to remember the negative penalties that every step accumulates because of the distance from the target). The learning progresses of

[5]http://doc.aldebaran.com/2-0/family/juliette_technical/joints_juliette.html

[6]https://github.com/Unity-Technologies/articulations-robot-demo/tree/mlagents

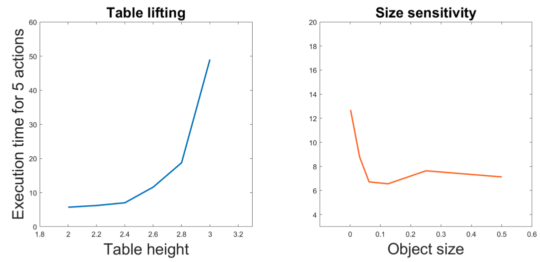

Fig. 5. The performances obtained in the two evaluation tasks, in terms of execution times.

the VPepper controller appears evident by observing the final Epoch duration: the initial phase of 50.000 actions lasted 50 minutes end, after almost all the sessions, the last one requires only a few seconds more than 3 minutes. Observing these timings, extremely accelerated by ML-Agents and by the physical engine, it is possible to appreciate one of the benefits due to the DT in the case of the training of a physical component. Achieving the same goal with a physical robot requires, first, an extremely longer training time due to the limitations of the angular velocities of the joints. Moreover, it may result in an excessive usage of the robot, which significantly deteriorates its mechanical parts.

V. Preliminary Evaluation

We performed a preliminary evaluation of adopted choices about VPepper model. A benchmark was organized aiming at understanding the opportunities offered by the proposed DT, but also the threats to validity of the simulated experiments. After the training phase, we performed two evaluation sessions aiming at assessing how the Reinforcement Learning based control generalizes to conditions different from the training settings. Considering that the spawning of the target happens in a random position, but on the table, we tried to progressively change, during an inference session, the height of the table (the target simply is placed on it). Figure 6, on the left, shows the effects of different table height on the execution time required for a task of 5 object touching inferences. It is important to point out, that inference is performed at the simulation timing, almost equal to the real one and not at the accelerated training one. Another important point is that Unity does not offer an absolute metric system, and all should be kept proportional to the relative scale of important characters of the simulation. As a reference, in the VPepper simulation proposed the DT is 5 units tall. In figure 5, labeled "Table lifting", the VPepper ML control shows good performances. Even if trained at the lowest height (2 units), the VPepper control is able to reach the targets also when the table is higher than the DT elbow. The experiment has been performed progressively increasing table height up to 3 units, and the time for executing the actions has increased up to tenfold. The second experiment has been performed by decreasing the size of the target object. Also in this case, the controller trained for VPepper shows good performances, that, in this situation, significantly outperform what a real robot can obtain. Figure 5, on the right side, reports the execution time performed on targets of different sizes. In the range 0.001-0.03, the time spent by VPepper for performing the actions increases, but for the other values time seems not influenced by object dimensions and the simulation largely outperforms the reality. The discrepancy between reality and simulation may be due to the simplifications adopted for object touching, or to the physical engine characteristics, and should be considered when extending simulation results to the real robot. In general, from this second experience, it appears that VPepper controlling mechanisms are not influenced by strong size variation of the target respect to what adopted for the training phase.

VI. Case Study

Aiming at operatively demonstrating the usefulness of the proposed DT approach, we move the attention to a case study focused on Ambient-Assisted Living (AAL) domain[7], on elderly people care-giving in the case of not complete self-sufficiency. DT in Healthcare is an emerging engineering paradigm [26]. Creating a live-model for healthcare services, DT introduces new opportunities for patient care including better risk assessment and evaluation without disturbing daily activities [27]. Our case study focuses on Elderly Monitoring by training VPepper to detect simulated anomalous situations tracked by wearable sensors [28] and smart-objects. Wearable sensors can help to monitor patients ascertaining their health condition [29]. In the proposed case, the adoption of a DT, shown in the case of the AAL domain, but easily generalizable, may bring some main advantages thanks to:

- the extension of Pepper with a DT capable of almost the same perception of the environment, but empowered with advanced capabilities;
- the availability of a training environment and a test bench for ML architectures and decision algorithms that can be quickly exported and applied by the real Pepper on real problems;
- the availability of a simulation environment for the training of 'digitally connected' caregivers, generally useful, but indispensable in the cases of emergency situations that happen with a low frequency.

As already discussed in sections above, the benefits of using DT in this context are numerous. Virtually simulated environments as well as robots allow to create all accidental conditions that can be harmful in real environments, unless safety procedures are strictly activated. In turn, this creates the opportunity of achieving a wide collection of data, mandatory for machine learning training processes.

A. The environment

In the proposed case study, VPepper aims to detect possible dangerous conditions for the elderly caregiver. It analyzes information from smart devices and interacts with its physical twin to provide the appropriate service. Figure 6 sketches the data flow and organizes the physical/virtual components involved in the proposed case study, in three Digital Twin couples: (i) DT 1, Pepper and its DT VPepper, (ii) DT 2, the

[7] http://www.aal-europe.eu/

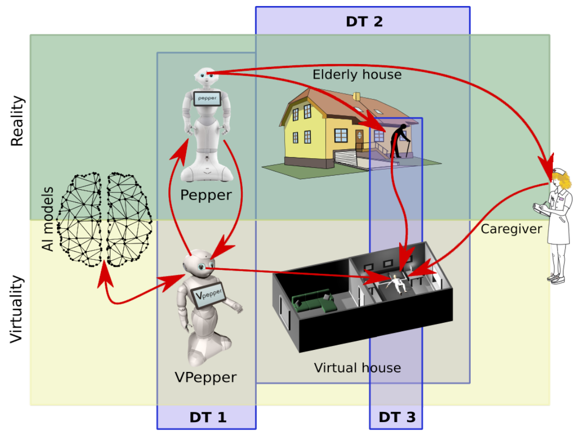

Fig. 6. An operative view of the proposed case study.

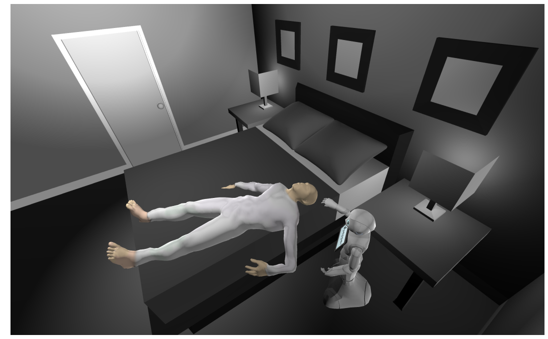

Fig. 7. One of the emergency interventions of VPepper simulated during the case study.

apartment of the elderly and its virtual replica, (iii) DT 3, the elderly person and his/her virtual replica. As it is possible to notice, VPepper is at the center of the communication flow depicted. VPepper continuously updates a 3D model of the house (Virtual house) and can locate and track Pepper and the assisted person in the living environment. Since the elderly person's apartment changes in time, because of moving furniture and other obstacles, Pepper sends to Virtual house continuous updates on the practicability of the routes it can follow. The virtual replica, i.e., the virtual elderly person's apartment in DT 2, exposes all data collected by the sensors and the smart devices in the real apartment. VPepper may be easily cloned for accelerate training, but also for simultaneously performing training and inference. Exploiting this capability, VPepper continuously trains its neural models on the observed data and updates Pepper decision algorithms when the new trained model performs better in terms of missed alerts and false positives. In the elderly person's apartment, for indoor localization of the assisted person, when she/he is out of Pepper field of observation, we adopted a smartwatch connected with Estimote Proximity Beacons[8]. At this aim, we developed also a different solution embedding a Movidius[9] accelerated camera capable of real-time human body recognition. However, we noticed that, in the case of some falls, the patient may be hidden by the furniture. Accidents like a fall or a prolonged stationary position can be inferred and, in these cases, Pepper can reach the elderly to ascertain the severity of the accident. As an additional benefit, the adoption of a smartwatch offers the availability of other information about the user, such as heart rate frequency and number of steps done. On its virtual side, VPepper observes the virtual replica of the real apartment, updated with current elderly position and the data collected by the smartwatch and other sensors.

B. The intervention

In the case of an alert, detected by VPepper via the smart devices, the real Pepper is notified to reach the assisted person and takes a short video of the scene. By safely touching the assisted person, Pepper can capture his/her attention and check physical and mental state with some spoken questions. VPepper analyzes the frames for person and pose detection via RNN [30] and follows the reaction of the patient to the questions by means of Speech-to-Text and natural language processing techniques. As a consequence, VPepper collects information on the scene and realises if the elderly does not respond to stimuli. In such a case, it will activate the emergency protocols for which it is configured. Respect to the gesture of safely touching the assisted person, it is important to point out, that Pepper engines and materials are conform to more stringent safety regulations and safety controls with priority robot motion. Pepper in a domestic environment could be very limited in its movements and the path it chooses could not be safe for its assisted person. For this reason, once an anomalous condition happens, both twins have the ability to process the stream of data from patient's wearable sensors to realise the kind of possible accident occurred. Indeed, the real twin, Pepper has limited processing power while VPepper (running on a regular PC or a workstation, with the services of TensorFlow is not affected by hardware and software limitations. VPepper can be dramatically faster in optimizing the path to follow in a safe manner. Synchronized with its physical twin (fig. 6 DT 1), and the updated state of the elderly person's apartment, it can online direct Pepper in the path to follow. The robot can promptly reach the elderly person, make questions and touch him/her to wait for a response. A case of simulated fall detection is depicted in figure 7. At the same time, Pepper sends an alert with a short video of the elderly fallen to the caregiver that may have a quick look and take the opportune decision independently. The Virtual house, with the tracked or synthetic data, may be also adopted as a simulation environment for training the caregiver. As a future enhancement of environment proposed for the case study, we aim at adding to the caregiver control interface also the ability to send command to Pepper (at the moment available only for VPepper).

C. Evaluation

Aiming at evaluating the perceived qualities of VPepper and of the associated virtual environment, we involved a group of 25 volunteers (composed of 13 males and 12 females, average

[8]https://developer.estimote.com/indoor/map-your-location/
[9]https://movidius.github.io/ncsdk/

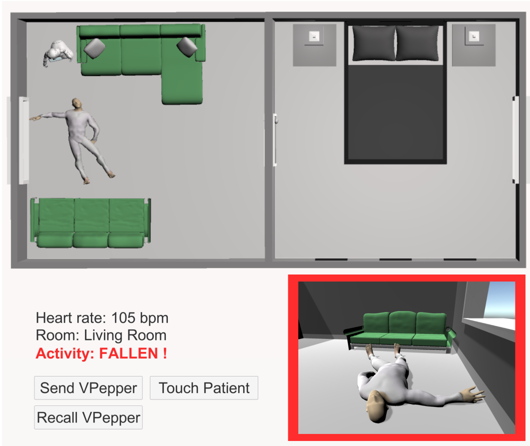

Fig. 8. The interface shown to the caregivers during the simulation.

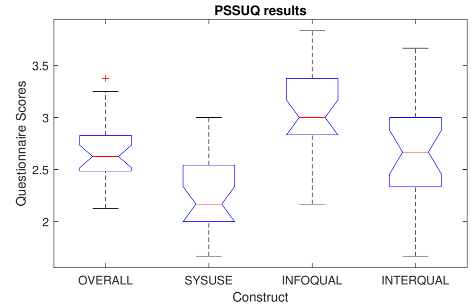

Fig. 9. The user perceptions respect to Usability (OVERALL), System Usefulness (SYSUSE), Interface (INTERQUAL) and Information Quality (INFOQUAL).

age 45) of a non-profit association AVULSS[10]. The involved volunteers are specialized in hospital and home assistance, and have the right skills and experience to serve as caregivers in the proposed case study. Respect to the technology involved in the work, they have different computer skills and very little confidence with 3D environments and video-games (with the exception of 4 young participants). After a short description of the tasks to be performed in the simulation, the volunteers were properly instructed to feel in the shoes of a remote caregiver of a not fully self sufficient elderly person. They interacted with the application by tablet. Five simulations were proposed to all participants: 4 situations in which a caregiver should promptly intervene and one false alert, detected when the elderly person was looking for an object on the floor. Figure 8 shows the GUI adopted for the case study which aggregates a video stream of the entire virtual apartment and, on demand, a direct view on the virtual patient grabbed by the DT. The GUI exposes, with some physical information like heart-rate frequency, user location and activity, three buttons that correspond to the actions a remote caregiver could ask to the real Pepper robot. For evaluation purposes, it is important to point out that the application is fully virtual. The protagonist is only the DT VPepper and the communication with the real twin Pepper is simulated. During the fall simulations, the virtual elderly person suddenly falls down while walking around the apartment, not necessarily in the same room where VPepper is. The remote caregiver should promptly react with a "Send VPepper" command that asks to the DT to go toward the fallen person and to send a video to the caregiver (in fig. 8, in the low rightmost picture red bordered area). The command may be followed by the "Touch Patient" whose action is shown in fig. 8 and is required in the case of a wrong alert. Alternatively, the subject can press the button "Recall VPepper" that brings the DT in its resting location and go on with the next simulation. After the experience with VPepper, all the volunteers were asked to fill the PSSUQ questionnaire (Post-Study System Usability Questionnaire) that collected also comments and suggestions. It is important to point out that, during the whole case study, performed after lock-down period in Italy, the safety of authors and the volunteer have been assured by adopting regulations for SARS-CoV-2 infection prevention. Interpersonal distances were respected, personal protective equipment has been put on, and both the tablet and the environment were disinfected before and after each session. The PSSUQ is derived by the IBM project SUMs [31] and is a 16-items questionnaire that evaluates interface usability on four constructs: (i) overall usability, (ii) perceived System Usefulness, (iii-iv) interface and Information Quality. The questions are evaluated on a Likert scale ranging from 1 (strongly agree) to 7 (strongly disagree). We have chosen PSSUQ because we were interested in evaluating, by VPepper simulation usability, if the caregivers also perceive the proposed environment to be useful for their assistance activities. Figure 9 reports the box-plots describing the result of the subjective evaluation, aggregated respect the four constructs of the PSSQU questionnaire (PSSUQ scale adopts a lower score for a better result). Even if the five simulations were easy to control, and have been performed on a device the volunteers are accustomed to use, the average score for overall usability is 2.67 (between "agree" and "partially agree"). This is a good result in terms of Overall Usability, especially if we consider the low computer literacy of volunteers involved. What appears less performing, even if still above the average, is Information Quality with an average of 3.08, lower respect to the other constructs (still "partially agree"). The construct we were more interested in was System Usefulness that has scored 2.3 ("agree"). Respect to this construct, we have also collected some comments. Many participants have offered their availability for participating at the future experimentation: their assistance activity is actually interrupted because of the COVID-19 pandemic outbreak. Thanks to this system they see a possible way to continue their volunteer activities, even if only virtually and remotely, with hospital patients. Interface quality performs in average with 2.65 score. Participants appreciated both the embedding of two video streams focused on the virtual patient and direct commands to VPepper robot.

VII. Conclusion

The adoption of a DT for a humanoid robot (also in sizes) shows evident advantages in the cases of ML approaches, like the ones on what this work is focused on. The proposed

[10]http://www.avulss.org/, section of Nola, Naples, Italy

simulation is adopted as the environment to train AI/ML architectures (ML-Agents) that can be exported to the real robot after their training and evaluation in virtual. If we consider the need of performing hundreds of thousands of iterations for reaching a good degree of control quality, it is clear that a physical machine is not a good candidate for training.

Another interesting DT opportunity is that VPepper, offering the same aspect, behaviours, and capabilities of Pepper enables the implementation of experiences spanning the entire Reality-Virtuality continuum. The user may be greeted by Pepper (as they actually do in our laboratories) in the real part of the experience, and by VPepper in the virtual settings. In the cases of AAL, for the Pepper-VPepper couple, as shown in the case study, the benefits of a train-update loop, performed in a simulated environment with the DT that is connected to the real environment, appear evident. During these experiences with VPepper, as in the case of the size sensitivity evaluation of our trained model, we had the chance of understanding also some limitations of a simulation environment based on a physical engine due to the traps of floating point arithmetic and vector graphics. An empirical evaluation of the proposed DT, in its surrounding eco-system, has been performed in a case study involving 25 volunteers with experience analogous to the AAL sector. The results confirm a strong interest of users and a good appreciation of the proposed Digital Twinning experience. The results achieved in this study demonstrated the feasible interaction between a physical robot working in a shared environment and its virtual replica, its digital twin. In real application scenarios, the physical robot is demanded to perform all computation and actions to provide a concrete support. In the considered case study the digital twin offers a prompt intervention to remote caregivers in case of critical or accidental conditions of the patients. The cooperation between the digital robot and the physical one, together with the learnt skill of safely touching people and objects (which is offered as an off-the-shelf features of Pepper robot), makes possible to provide an effective remote support, an aspect that has become so prominently important in recent times due to pandemic restrictions caused by COVID-19.

References

[1] P. Barra, C. Bisogni, A. Rapuano, A. F. Abate, and G. Iovane, "HiMessage: An Interactive Voice Mail System with the Humanoid Robot Pepper," in *DASC 2019*. IEEE, 2019, pp. 652–656.

[2] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the iot context: A survey on technical features, scenarios, and architectural models," *Proceedings of the IEEE*, pp. 1–40, 2020.

[3] L. Wright and S. Davidson, "How to tell the difference between a model and a digital twin," *Advanced Modeling and Simulation in Engineering Sciences*, vol. 7, no. 1, pp. 1–13, 2020.

[4] M. Matulis and C. Harvey, "A robot arm digital twin utilising reinforcement learning," *Computers & Graphics*, vol. 95, pp. 106–114, 2021.

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

[6] I. Verner, D. Cuperman, A. Fang, M. Reitman, T. Romm, and G. Balikin, "Robot online learning through digital twin experiments: a weightlifting project," in *Online Engineering & Internet of Things*. Springer, 2018, pp. 307–314.

[7] 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.

[8] L. Ren, Z. Meng, X. Wang, L. Zhang, and L. T. Yang, "A data-driven approach of product quality prediction for complex production systems," *IEEE Transactions on Industrial Informatics*, 2020.

[9] Y. Fang, C. Peng, P. Lou, Z. Zhou, J. Hu, and J. Yan, "Digital-twin-based job shop scheduling toward smart manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 12, pp. 6425–6435, 2019.

[10] G. Ahmadi-Assalemi, H. Al-Khateeb, C. Maple, G. Epiphaniou, Z. Alhaboby, S. Alkaabi, and D. Alhaboby, "Digital twins for precision healthcare," *Advanced Sciences and Technologies for Security Applications*, pp. 133–158, 2020.

[11] M. M. Rathore, A. Ahmad, A. Paul, J. Wan, and D. Zhang, "Real-time medical emergency response system: exploiting iot and big data for public health," *Journal of medical systems*, vol. 40, no. 12, p. 283, 2016.

[12] M. Schluse, M. Priggemeyer, L. Atorf, and J. Rossmann, "Experimentable digital twins—streamlining simulation-based systems engineering for industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1722–1731, 2018.

[13] 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).

[14] K. Dröder, P. Bobka, T. Germann, F. Gabriel, and F. Dietrich, "A machine learning-enhanced digital twin approach for human-robot-collaboration," *Procedia Cirp*, vol. 76, pp. 187–192, 2018.

[15] A. Bécue, Y. Fourastier, I. Praça, A. Savarit, C. Baron, B. Gradussofs, E. Pouille, and C. Thomas, "Cyberfactory#1 — securing the industry 4.0 with cyber-ranges and digital twins," in *2018 14th IEEE International Workshop on Factory Communication Systems (WFCS)*, 2018, pp. 1–4.

[16] F. Biesinger, D. Meike, B. Kraß, and M. Weyrich, "A case study for a digital twin of body-in-white production systems general concept for automated updating of planning projects in the digital factory," in *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2018, pp. 19–26.

[17] H. Laaki, Y. Miche, and K. Tammi, "Prototyping a digital twin for real time remote control over mobile networks: Application of remote surgery," *IEEE Access*, vol. 7, pp. 20 325–20 336, 2019.

[18] A. Ardanza, A. Moreno, Álvaro Segura, M. de la Cruz, and D. Aguinaga, "Sustainable and flexible industrial human machine interfaces to support adaptable applications in the industry 4.0 paradigm," *International Journal of Production Research*, vol. 57, no. 12, pp. 4045–4059, 2019.

[19] M. Lacomblez, B. Jeanne, V. Havard, and D. Baudry, "Co-simulation architecture between a digital twin and a virtual reality environment in an industrial context," vol. 8, 2018, pp. 429–434.

[20] A. F. Abate, C. Bisogni, L. Cascone, A. Castiglione, G. Costabile, and I. Mercuri, "Social Robot Interactions for Social Engineering: Opportunities and Open Issues," in *DASC 2020*. IEEE, 2020.

[21] A. F. Abate, P. Barra, C. Bisogni, L. Cascone, and I. Passero, "Contextual trust model with a humanoid robot defense for attacks to smart eco-systems," *IEEE Access*, vol. 8, pp. 207 404–207 414, 2020.

[22] P. Barra, C. Bisogni, M. Nappi, D. Freire-Obregón, and M. Castrillón-Santana, "Gender classification on 2d human skeleton," in *2019 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART)*, 2019, pp. 1–4.

[23] A. Juliani, V.-P. Berges, E. Vckay, Y. Gao, H. Henry, M. Mattar, and D. Lange, "Unity: A general platform for intelligent agents," *ArXiv*, vol. abs/1809.02627, 2018.

[24] "Proximal policy optimization," https://openai.com/blog/openai-baselines-ppo/, accessed: 2020-12-10.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[26] K. Bruynseels, F. Santoni de Sio, and J. van den Hoven, "Digital twins in health care: ethical implications of an emerging engineering paradigm," *Frontiers in genetics*, vol. 9, p. 31, 2018.

[27] G. Ahmadi-Assalemi, H. Al-Khateeb, C. Maple, G. Epiphaniou, Z. A. Alhaboby, S. Alkaabi, and D. Alhaboby, "Digital twins for precision healthcare," in *Cyber Defence in the Age of AI, Smart Societies and Augmented Humanity*. Springer, 2020, pp. 133–158.

[28] S. Din and A. Paul, "Smart health monitoring and management system: Toward autonomous wearable sensing for internet of things using big data analytics," *Future Generation Computer Systems*, vol. 91, pp. 611 – 619, 2019.

[29] A. Paul, H. Pinjari, W.-H. Hong, H. Seo, and S. Rho, “Fog computing-based iot for health monitoring system,” *Journal of Sensors*, vol. 2018, 2018.

[30] D. Freire-Obregón, M. Castrillón Santana, P. Barra, C. Bisogni, and M. Nappi, “An attention recurrent model for human cooperation detection,” *Computer Vision and Image Understanding*, vol. 197-198, p. 102991, 06 2020.

[31] J. R. Lewis, “Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use,” *International Journal of Human-Computer Interaction*, vol. 7, no. 1, pp. 57–78, 1995.