



HAL
open science

Digital Twin Driven Smart Home: A Feasibility Study

Alireza Asvadi, Andrei Mitriakov, Christophe Lohr, Panagiotis Papadakis

► **To cite this version:**

Alireza Asvadi, Andrei Mitriakov, Christophe Lohr, Panagiotis Papadakis. Digital Twin Driven Smart Home: A Feasibility Study. 9th International Conference On Smart Living and Public Health (ICOST 2022), 2022, Paris, France. hal-03691144

HAL Id: hal-03691144

<https://hal.science/hal-03691144>

Submitted on 8 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Digital Twin Driven Smart Home: A Feasibility Study

Alireza Asvadi, Andrei Mitriakov, Christophe Lohr, and Panagiotis Papadakis

IMT Atlantique, Lab-STICC, UMR CNRS 6285, team RAMBO
F-29238 Brest, France
`{firstname.lastname}@imt-atlantique.fr`

Abstract. We aim to facilitate the daily-life activities of frail or elderly people in collaboration with mobile assistive robots through the means of a digital twin-powered smart home. Being able to quickly and efficiently produce a digital twin of the human user’s environment, can help to further develop personalized assistive solutions. As our first investigation toward this goal, we describe our proof-of-concept “digital twin-driven smart home” implementation. It consists of a virtual representation, robot navigation and environment semantics using open-source software. The initial obtained results on the building process of the digital twin are encouraging and suggest the possibility of integration of digital twin for smart spaces.

Keywords: Ambient assisted living, cyber physical system, living-lab, semantics

1 Introduction

In recent years, the availability of low-cost sensors and open-source middle-ware software have opened up interesting research areas in the robotics field. In particular, the next generation of simulation models called Digital Twin (DT), which represents a continuous virtual replica of a physical system, has gained increasing attention. It can be used to create a simulation of a smart home including assistive/service robots and human users. It has applications in the optimization of robots and smart home settings. For example, finding the optimal number and configuration of sensors especially when new robots or users are introduced. As another examples, monitoring in real-time, further analysis and learning of edge cases and rare situations in DT for the safety of human users, *e.g.*, by pushing those events from simulation to real-world and vice-versa (thanks to DT that enables a bi-directional link between simulation and real-world). In practice, such optimization should be carried out over a variety of houses and users.

In this context, this paper aims to put forward the use of DT technology in the development of a smart home system. The main contributions are:

- . This work looks at the practical aspects of the building process of DT.
- . Provides analysis and insights into the difference in the performance between the actual and simulated environments.

The paper is organized as follows. Section 2 gives a definition and recent related works. Section 3 details the proposed DT development process. Experimental results are described in Section 4, and Section 5 brings some concluding remarks and proposes directions for further research.

2 Background

With the progress of the internet of things and artificial intelligence, several strategies were developed to advance smart technologies, including Industry 4.0 [11] and Society 5.0 [4]. At the core, these technologies are based on a cyber-physical system (CPS) that integrates physical and digital entities. In this context, DT is an approach to implement cyber-physical integration. The DT is a digital representation of a physical system that precisely models and can predict how the actual system will perform. In this respect, it serves as a tool to bridge the gap between physical and digital worlds.

Definition. Some of the commonly accepted concepts in the definition of the DT are the following : it is a digital representation of a physical entity (including geometric, functional, and usage models), being ultra-realistic and fully consistent with the physical entity, continuous and automatic update to be able to adapt to changes, semantic content to support intelligent decision making, and implementation in a decentralized structure to collaborate with other DTs [17].

DT for Robotics. While there is a rich literature on DT concept [5, 8], there have been few reports on the 3D modeling aspect in robotics and smart home applications. Kousi et al. [9] combined multi-sensor data and CAD models to create a virtual representation (including semantic and dynamic update components) of a production environment. Havard et al. [7] proposed a co-simulation approach between DT and virtual reality in a human-robot collaborative workplace. Another notable work is Habitat 2.0 [16] which is a platform for training simulated robots in interactive environments. Phanden et al. [13] recently reviewed existing simulation software for DT development. For robotics, they found Gazebo and V-REP [15] for modeling sensors and image-based systems, ROBOTRAN to model multi-body systems, and ROBCAD for multi-device robotic and automated manufacturing processes.

DT for Smart Home. As to digital models of buildings, To et al. [18] performed drone-based reconstruction for DT augmentation of buildings. Calderita et al. [1] proposed a CPS for ambient assisted living including humans and assistive robots in a care-giving center. A related term to DT is Building Information Modeling (BIM). Czerniawski and Leite [2] reviewed approaches for BIM. They identified the following steps for model creation: collecting sensor data and 3D reconstruction, semantic and geometric modeling for the recognition of semantic classes in sensor data and describing the instances' shapes, and finally the creation of BIM that integrates semantic and geometric components. In comparison with BIM (which is one-way modeling), in a DT there is a bi-directional data flow between the virtual and physical entities.

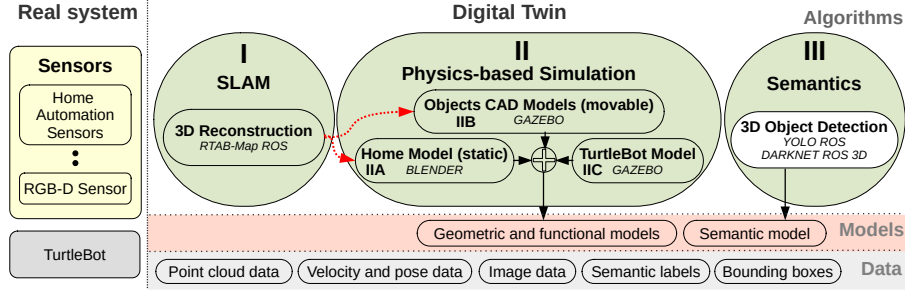


Fig. 1. The conceptual digital twin of the smart home. The dashed-red arrows mark the process that are not implemented in a fully automated way. The white box shows the component that is developed primarily for a feasibility study.

There are a number of other works and projects that discuss related concepts on ambient assisted living and the synergy of ambient intelligence with robots. Examples are GIRAFF+ that consists of a network of smart home sensors and a telepresence robot to monitor elderly people, and STRANDS that explores adaptation to changes over time for mobile robots in 3D dynamic human environments. Previous work of the authors concentrated on the potential services that could be offered from such systems [12].

3 DT-Powered Smart Home

The DT of a smart home is defined as: using a priori information (*e.g.*, architectural plans, input from inhabitants) and given input sensory data (*e.g.*, home automation and robot sensors), build a continuous model including robots and user models. As a first step toward this goal, a prototype DT model of the smart home has been developed. Figure 1 demonstrates an abstraction of our understanding of what a DT workflow should consist of. Ideally, such a workflow has limited dependence on human supervision so as to reduce errors and favor portability and generalization. The steps are defined as follows. The smart home is scanned using a hand-held RGB-D camera and a SLAM algorithm to build a reconstruction of it. A modeling software is used to construct the CAD models of the home and objects inside it from the scan. The CAD models and a robot model are imported into a robotic simulator. A proof-of-concept is developed to enable semantic understanding of objects.

The DT can be conceptualized in **data**, **algorithmic** and **model**. The data layer represents the communicated data including point cloud, navigation (velocity and pose), etc. The algorithmic layer includes software to develop the modules. The model layer comprises the geometric, functional and semantics

including the CAD models, a TurtleBot¹, and a pre-trained object detection model. More information can be found on the project website².

3.1 3D Reconstruction

To have a metric measurement of the actual environment a hand-held scanning is performed. The Orbbec Astra Pro RGB-D camera is used which is shown to be one of the best low-cost sensors for the reconstruction of indoor spaces [3]. The RTAB-Map (Real-Time Appearance-Based Mapping) [10] algorithm is used. It consists of loop closure and proximity detection (using a bag of words), graph optimization (to decrease odometry drift) and global map assembling (using local occupancy grid). It requires an odometry estimation, *i.e.*, motion estimation between consecutive scans. This is performed by computing visual odometry using RTAB-Map and the camera. The process outputs a dense point cloud. The CAD models are created from point cloud data. A modeling software is used to craft the CAD model of the home structure (*e.g.*, walls) from the point cloud, also online public repositories are used to obtain the models of the real home items. It should be noted that an automatic transformation of 3D scans into CAD is an open question (see *e.g.*, Scan2CAD project) that will not be addressed in our work.

3.2 Physics-based Simulation

The Gazebo is used to simulate the dynamics of the system. A digital 3D model is developed to represent the smart home environment and the objects in the simulator. The CAD models are used to define the visual and collision properties of the items in this model. Other physical properties can be defined, for example pose, mass, being static (*e.g.*, walls) and dynamic (*e.g.*, objects). A TurtleBot model is spawned in the simulated system. The use of Gazebo with ROS enables communication between the simulated and the real system.

3.3 Semantics

The detection of the categories of objects of interest (*e.g.*, tables and chairs) and their position can have several benefits in a DT framework. The You Only Look Once (YOLO)-v3 algorithm [14] is used for object detection. It reads camera image data and returns the 2D bounding boxes. The pre-trained model is able to detect 80 classes including chair, sofa, table, bottle, microwave, person, etc. The *darknet_ros_3d* package³ is used to add the 3D bounding boxes of objects to the YOLO. It combines the detected 2D bounding boxes with point cloud data to calculate the 3D bounding boxes.

¹<https://www.turtlebot.com/turtlebot2/>

²<https://sites.google.com/view/heron-project>

³https://github.com/IntelligentRoboticsLabs/gb_visual_detection_3d



Fig. 2. Top-left: Top panoramic view of the smart home. top-right: Illustration of an area in the reconstructed map (from the view shown by a red arrow in the left image). Bottom: the full reconstructed map (from top view).

4 Results and Discussion

Sections 4.1 and 4.2 describe the process results for the creation of the digital 3D model and a comparison of the DT and the actual system. Section 4.3 provides discussion on the semantics aspect. The approach is evaluated in a small experimental flat including a kitchen, a bathroom and a living room with furniture (see Fig. 2-TopLeft).

4.1 Creation of the Digital 3D Model

3D Reconstruction A detailed colored 3D point cloud of the flat is achieved which has about 700k points (see Fig. 2-Bottom). The following are some of the observed limitations.

- . The windows need to be covered in sunlight. The Astra Pro camera works based on infrared technologies. The light coming from the infrared projector can be outshined by the sunlight and no point cloud can be computed.
- . The mirror, window glass, TV surface, and black colored objects degrade the quality of the depth image and the point cloud.

- . The textureless walls, ground and ceiling must be avoided or rectified (*e.g.*, by adding picture frames to plain walls). Because the visual odometry in RTAB-Map is based on extracted features from the RGB-D images. Therefore, in environments without enough features, the odometry cannot be computed.

Digital 3D Model An idealistic 3D model of the flat was made by a team of students. The model was built by first measuring different dimensions of the flat and then recreating it using a modeling software. A quantitative evaluation is performed with a comparison of our CAD model against this idealistic model (see Fig. 3-Top). To perform the evaluation the models are converted to point cloud data using ray tracing operations. The root mean squared error (RMSE) and Hausdorff distance metrics (which are common general-purpose metrics for comparing 3D shapes) are used to estimate the differences (see Table 1). The accuracy of about 9 cm average RMSE is obtained which is encouraging and demonstrates an interesting perspective for an automated reconstruction-based model creation. As it can be seen in Fig. 3-Top, the perpendicularity of the walls in our developed model (shown in red) are preserved well with only a slight slope on the right side. Figure 3-Bottom shows the furnished smart home with a simulated TurtleBot in the Gazebo.

Table 1. A comparison of the quality of the developed smart home’s CAD model against the idealistic model.

Metrics	RMSE	Hausdorff
Error (in m)	0.088	1.354

4.2 Robot Navigation

Experiments were performed to compare the traversed trajectory and the generated 2D maps in the actual and simulated conditions.

Trajectory Evaluation A velocity message with the linear and angular values of +0.25 and -0.1 is published (with the frequency of 10 Hz for 4 seconds) to control the translational and rotational speed of the robot. The objective is to perform the navigation in the DT and reality, and to compare the trajectories.

To obtain the trajectory of the actual Turtlebot, a visual marker was used. We position the marker on the upper part of the Turtlebot in the field of view of a high-resolution camera (see Fig. 4). The robot poses can be retrieved by detecting the marker in the camera image. A toolkit tracks marker poses which enables tracing of the Turtlebot trajectory in the camera coordinate frame. The experiments were performed in 5 iterations. The average and standard deviation of the trajectories in both real and simulated cases were reported in Fig. 5. In addition a simplified analytical solution for the expected trajectory of a

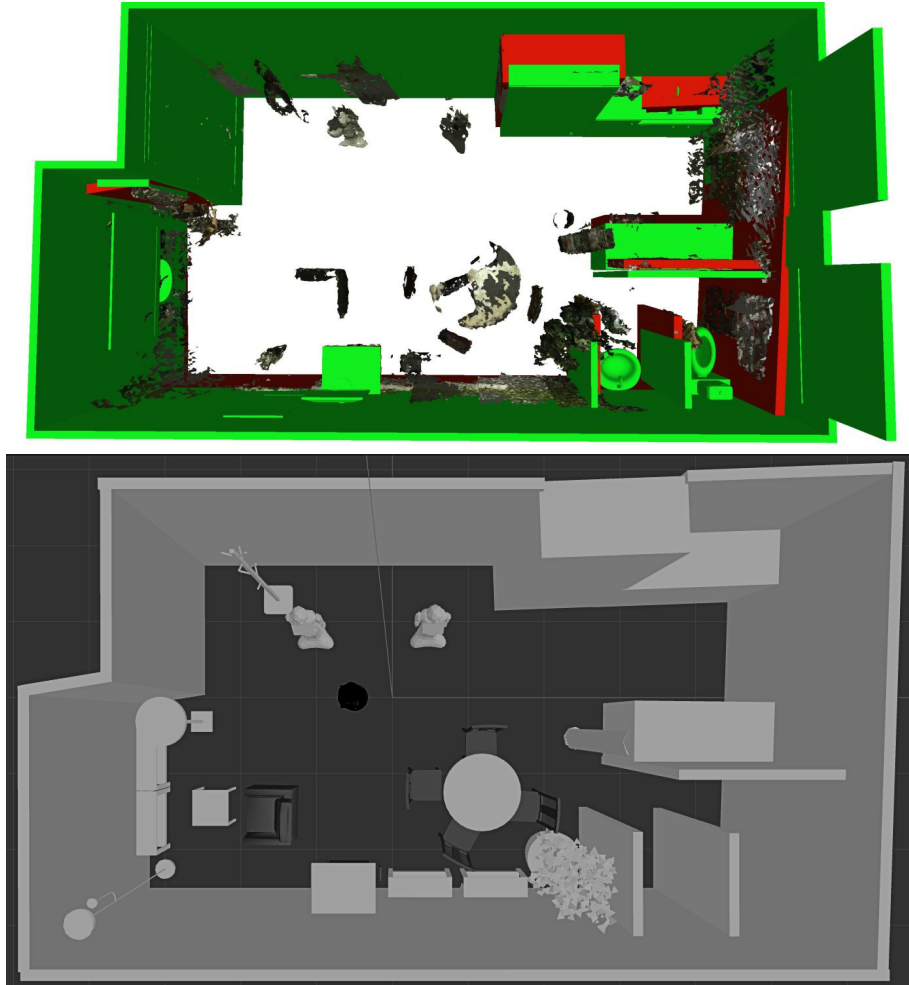


Fig. 3. Top: Illustration of the representations. Superimposed simplified mesh (obtained from the point cloud), the model created from point cloud (red), and the idealistic model (green) for a visual inspection. Bottom: The simulated smart home with a Turtlebot model.

robot performing uniform circular motion is computed. It is equal to traverse of 0.4 radians on the perimeter of a circle with radius of 2.5 m. It should be mentioned that the complete Turtlebot model is a differential drive consisting of two wheels and two caster wheels which was not considered here. It was noticed that the turtlebot in the real condition traversed a shorter distance in the trajectory (about 25 cm difference with the simulated one). This was probably because of the uncertainties on the friction parameters between the robot and



Fig. 4. Left: The trajectory evaluation experiment. Right: Left and right show the Turtlebot at the start and end of the experiment. Top and bottom show the simulated and actual setup.

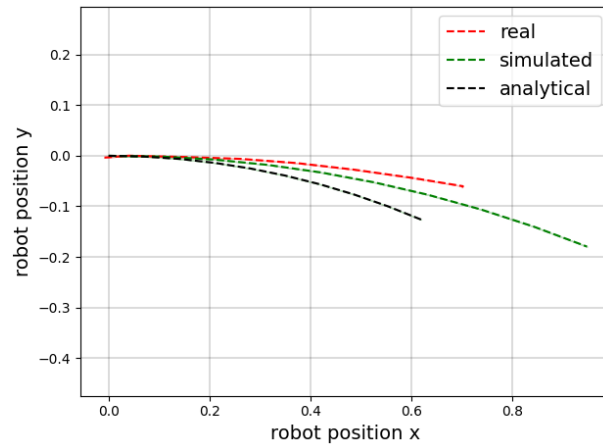


Fig. 5. The average and standard deviation of the obtained trajectories in the real, simulated and analytical situations (the starting point is on the left-side of the curves).

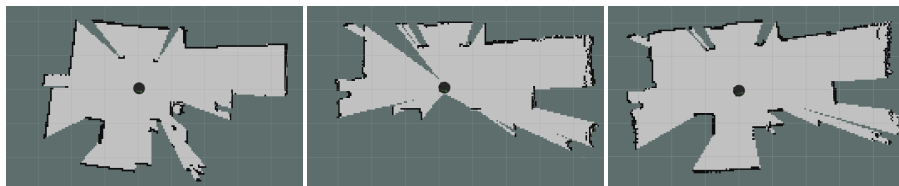


Fig. 6. Illustration of the 2D occupancy grid maps. From left to right: the generated grid map in the simulated environment, the created map in the actual home, the obtained map after the additional rotation in the real case.



Fig. 7. 2D Object detection using YOLO-v3. The images show the detection results in different areas in the smart home.

the ground surface. Due to using a simplified model, there were also differences in the recorded trajectories in comparison with the analytical solution.

Understanding the difference between the virtual model and the real world and adjusting such parameters can improve the robot's behavior in reality. It helps to accurately estimate and plan the robot trajectory and in the application level to provide both reliable and consistent service for elderly users in the smart home.

Mapping Experiment A 2D mapping process is performed and compared in the DT and reality. The turtlebot rotates 360 degrees (with the angular speed of 10 degrees per second) to cover the whole space. The *gmapping*⁴ package from ROS is used [6]. It is a laser-based SLAM. To be able to work with a RGB-D camera, it converts a depth image to a fake laser scan. It uses particle filters (each particle carries an individual map) and several adaptive techniques to learn the occupancy grid map of the environment. Figure 6 shows examples of the generated maps. It was observed that the actual turtlebot could not cover the full profile of the home using the rotation message of 360 degrees. The rotation of 540 degrees was published to the actual turtlebot to fulfill the coverage of the whole area.

4.3 Semantics

Object detection can be used as a coupling mechanism of the real and the digital worlds. As a feasibility study, 2D and 3D versions of YOLO object detection were employed to detect the objects in the smart home.

⁴<http://wiki.ros.org/gmapping>

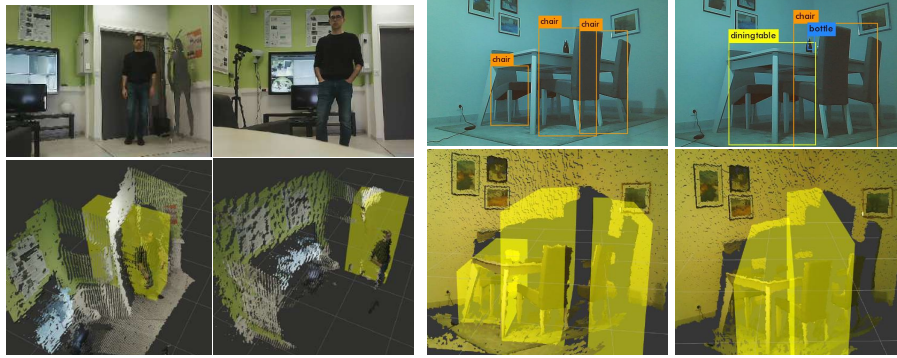


Fig. 8. 3D Object detection using YOLO-v3 and Darknet ROS 3D. Left: shows a person standing and moving in the smart home. Right: shows results in another setting with different furniture.

Quantitative Analysis of 2D Object Detection Figure 7 shows 2D object detection results in different areas of the smart home. An evaluation is performed to quantitatively measure the object detection performance in the smart home. A number of 30 detection frames were selected randomly and the precision, recall, and F1 scores computed for the two most frequent items seen in the smart home: the chairs and TV monitors, with the total number of 50 chairs and 41 monitors observed in all images. As it can be seen in the Table 2, the detection rates were different for each item. For example, the TV monitors were detected with higher precision whereas the detector has a better recall value on chairs.

Table 2. A comparison of 2D object detection in the smart home.

Object/Metrics	Precision	Recall	F-Score
Chair	0.53	0.88	0.66
Monitor	0.97	0.68	0.80

Qualitative Analysis of 3D Object Detection Figure 8 shows an example of the 3D detection results. The top shows the 2D detections. The yellow boxes in the bottom show projected detections into the point cloud in the 3D space.

5 Conclusions

In this paper, we presented an approach for the generation of a digital 3D model of a smart home using open-source software and a low-cost sensor. The methodology and initial results (including the obtained models) are promising and appear

to have a potential for the design of DT-driven smart homes. However, further research is needed to explore the integration of the digital profiles of the connected home automation sensors (*e.g.*, thermostat and pressure sensors) into the DT. The next challenge is to extend the developed DT to include other entities: sophisticated robots (than a simple TurtleBot) and human users (*e.g.*, to sense where the user is and what is doing). Another interesting directions are benchmarking, further analysing and quantifying the different steps of DT creation.

Acknowledgments. The authors would like to thank Maroua El Houicha, Zhi Zhang, Jeanne Thévenin, and Damien Bouchabou, students of IMT Atlantique – Brest, for providing the idealistic 3D smart home model. The authors would also like to thank Fabien Dagnat and Jean-Christophe Bach for helpful discussions on the digital twin architecture. The work has been performed in the context of project HERON - Habitat intelligent Et Robotique d’assistance basés sur le jumeau Numérique - financed by the region of Brittany (Région Bretagne) and the department of Finistere (CD29).

References

- [1] Luis V Calderita et al. “Designing a cyber-physical system for ambient assisted living: a use-case analysis for social robot navigation in caregiving centers”. In: *Sensors* 20.14 (2020), p. 4005.
- [2] Thomas Czerniawski and Fernanda Leite. “Automated digital modeling of existing buildings: A review of visual object recognition methods”. In: *Automation in Construction* 113 (2020), p. 103131.
- [3] J. G. da Silva Neto et al. “Comparison of RGB-D sensors for 3D reconstruction”. In: *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*. 2020, pp. 252–261. DOI: 10.1109/SVR51698.2020.00046.
- [4] Atsushi Deguchi. “From Smart City to Society 5.0”. In: *Hitachi-UTokyo Laboratory (H-UTokyo Lab.)* (2020), p. 43.
- [5] Aidan Fuller et al. “Digital twin: Enabling technologies, challenges and open research”. In: *IEEE Access* 8 (2020), pp. 108952–108971.
- [6] Giorgio Grisetti, Cyrill Stachniss, and Wolfram Burgard. “Improved techniques for grid mapping with rao-blackwellized particle filters”. In: *IEEE transactions on Robotics* 23.1 (2007), pp. 34–46.
- [7] Vincent Havard et al. “Digital twin and virtual reality: a co-simulation environment for design and assessment of industrial workstations”. In: *Production & Manufacturing Research* 7.1 (2019), pp. 472–489.
- [8] David Jones et al. “Characterising the Digital Twin: A systematic literature review”. In: *CIRP Journal of Manufacturing Science and Technology* 29 (2020), pp. 36–52.
- [9] Niki Kousi et al. “Digital twin for adaptation of robots’ behavior in flexible robotic assembly lines”. In: *Procedia manufacturing* 28 (2019), pp. 121–126.

- [10] Mathieu Labbé and François Michaud. “RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation”. In: *Journal of Field Robotics* 36.2 (2019), pp. 416–446.
- [11] Heiner Lasi et al. “Industry 4.0”. In: *Business & information systems engineering* 6.4 (2014), pp. 239–242.
- [12] Panagiotis Papadakis et al. “System design for coordinated multi-robot assistance deployment in smart spaces”. In: *2018 Second IEEE International Conference on Robotic Computing (IRC)*. IEEE. 2018, pp. 324–329.
- [13] Rakesh Kumar Phanden, Priavrat Sharma, and Anubhav Dubey. “A review on simulation in digital twin for aerospace, manufacturing and robotics”. In: *Materials Today: Proceedings* (2020).
- [14] Joseph Redmon and Ali Farhadi. “Yolov3: An incremental improvement”. In: *arXiv preprint arXiv:1804.02767* (2018).
- [15] Eric Rohmer, Surya PN Singh, and Marc Freese. “V-REP: A versatile and scalable robot simulation framework”. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2013, pp. 1321–1326.
- [16] Andrew Szot et al. “Habitat 2.0: Training home assistants to rearrange their habitat”. In: *arXiv preprint arXiv:2106.14405* (2021).
- [17] Fei Tao, Meng Zhang, and Andrew Yeh Chris Nee. *Digital twin driven smart manufacturing*. Academic Press, 2019.
- [18] Alex To et al. “Drone-based AI and 3D Reconstruction for Digital Twin Augmentation”. In: *arXiv preprint arXiv:2106.03797* (2021).