Robotics Simulation

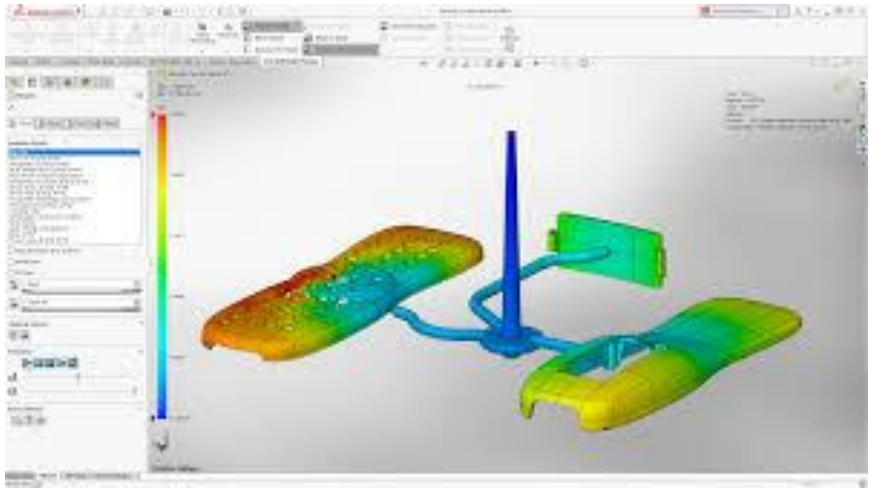
DASE7503 Robotic Systems Integration

Contents

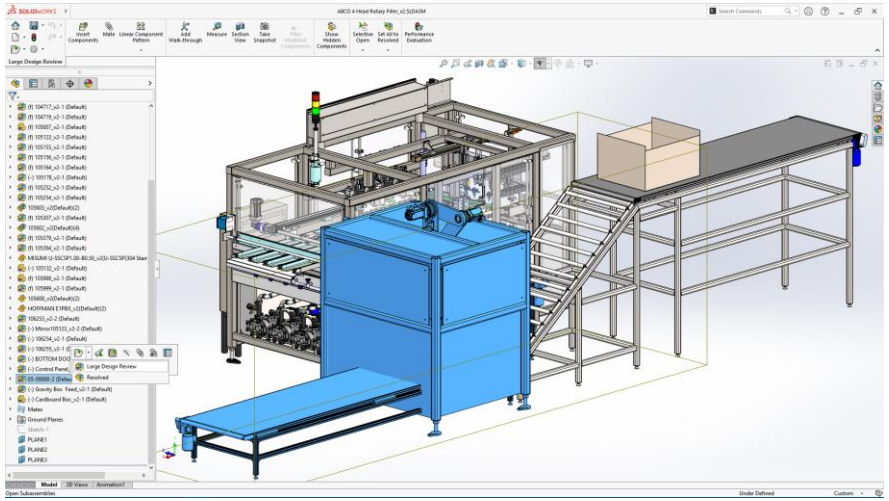
- Introduction
- Popular Robotics Simulation Tools
- Gazebo and Rviz for Robotics Simulation
- Digital Twins for Robotics Simulation

Introduction

- A simulation is an **imitative representation** of a process or system that could exist in the real world.
- In this broad sense, simulation can often be used interchangeably with models.
- Sometimes a clear distinction between the two terms is made, in which simulations require the use of **models**; the model represents the key **characteristics** or **behaviors** of the selected system or process, whereas the simulation represents the evolution of the model over time.
- Another way to distinguish between the terms is to define simulation as **experimentation** with the help of a model. This definition includes time-independent simulations. Often, computers are used to execute the simulation.



Simulation of injection moulding process using SolidWorks Computer-Aided Engineering software



Simulation of production lines using SolidWorks

Source: https://en.wikipedia.org/wiki/Simulation#

Introduction

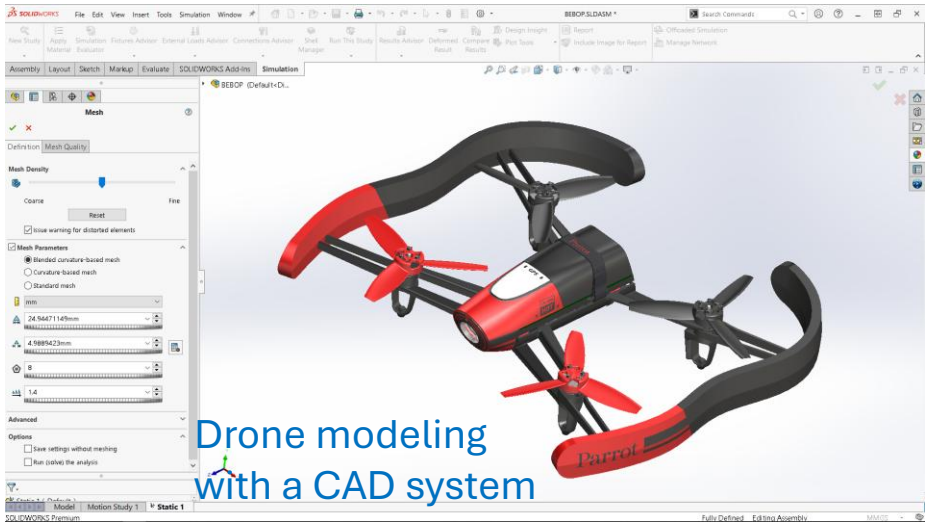
What is Robotics Simulation?

- Involves creating **virtual environments** to develop, test, and validate robotic systems and algorithms **without the need for physical hardware.**
- Enables engineers and researchers to experiment safely, cost-effectively, and rapidly.

Introduction

Main Aspects of Robotics Simulation

- **Physics Modeling:** Accurate representation of real-world physics, including dynamics, kinematics, collision detection, and sensor feedback.
- **Sensor Simulation:** Emulating sensors such as cameras, LIDAR, ultrasonic sensors, and IMUs to test perception algorithms.
- **Robot Modeling:** Creating detailed models of robots, including their joints, actuators, and control systems.
- **Environment Modeling:** Designing complex environments with obstacles, terrains, and objects for navigation and manipulation tasks.
- **Algorithm Testing:** Validating control algorithms, path planning, navigation, and machine learning models in simulation before deployment.



Introduction

Applications of Robotics Simulation

- Algorithm development and testing
- Hardware-in-the-loop testing
- Education and training
- Research in autonomous navigation, manipulation, and human-robot interaction
- Prototyping robotic systems before physical implementation

Popular Robotics Simulation Tools

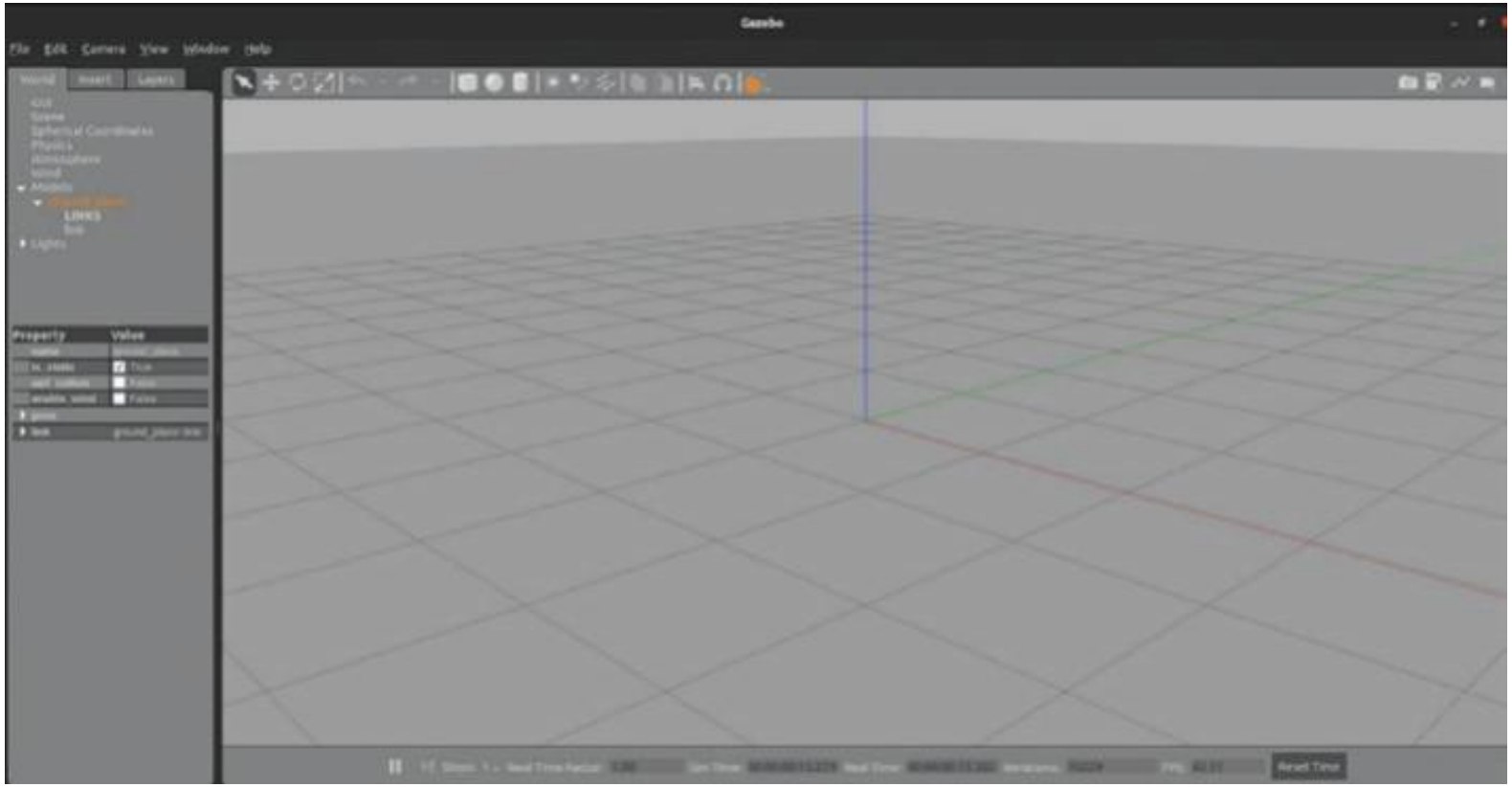
- **Gazebo:** An open-source 3D robotics simulator integrated with ROS (Robot Operating System). It offers realistic physics and sensor simulation.
- **Webots:** A professional robot simulation platform supporting a wide range of robots and environments, suitable for education and research.
- **V-REP / CoppeliaSim:** A versatile simulation environment with extensive API support for robot development and testing.
- **PyBullet:** A Python module for physics simulation, often used for robotics, machine learning, and reinforcement learning.
- **ROS (Robot Operating System):** While primarily a middleware, ROS integrates with various simulators like **Gazebo** to facilitate robot software development.
- **Unity and Unreal Engine:** Game engines used for high-fidelity robotics simulations, especially for perception and AI development.

Gazebo

Gazebo is a powerful robot simulation tool that is widely used in the robotics community. Here are some of its key features:

- Provides a highly realistic, 3D physics-based simulation environment that allows users to test and develop robotics algorithms in a safe and controlled virtual environment.
- Allows users to interact with the simulation environment, adding and removing objects (such as walls, partitions, doors, chairs, etc.), changing lighting conditions, adjusting the physics parameters, and so on.
- Supports the simulation of a variety of sensors, such as cameras, stereo cameras, depth cameras, lidars, IMUs, bumpers, and so on.
- Provides support for various drive systems, such as differential drives, skid steer drives, holonomic drives, and so on.

Gazebo

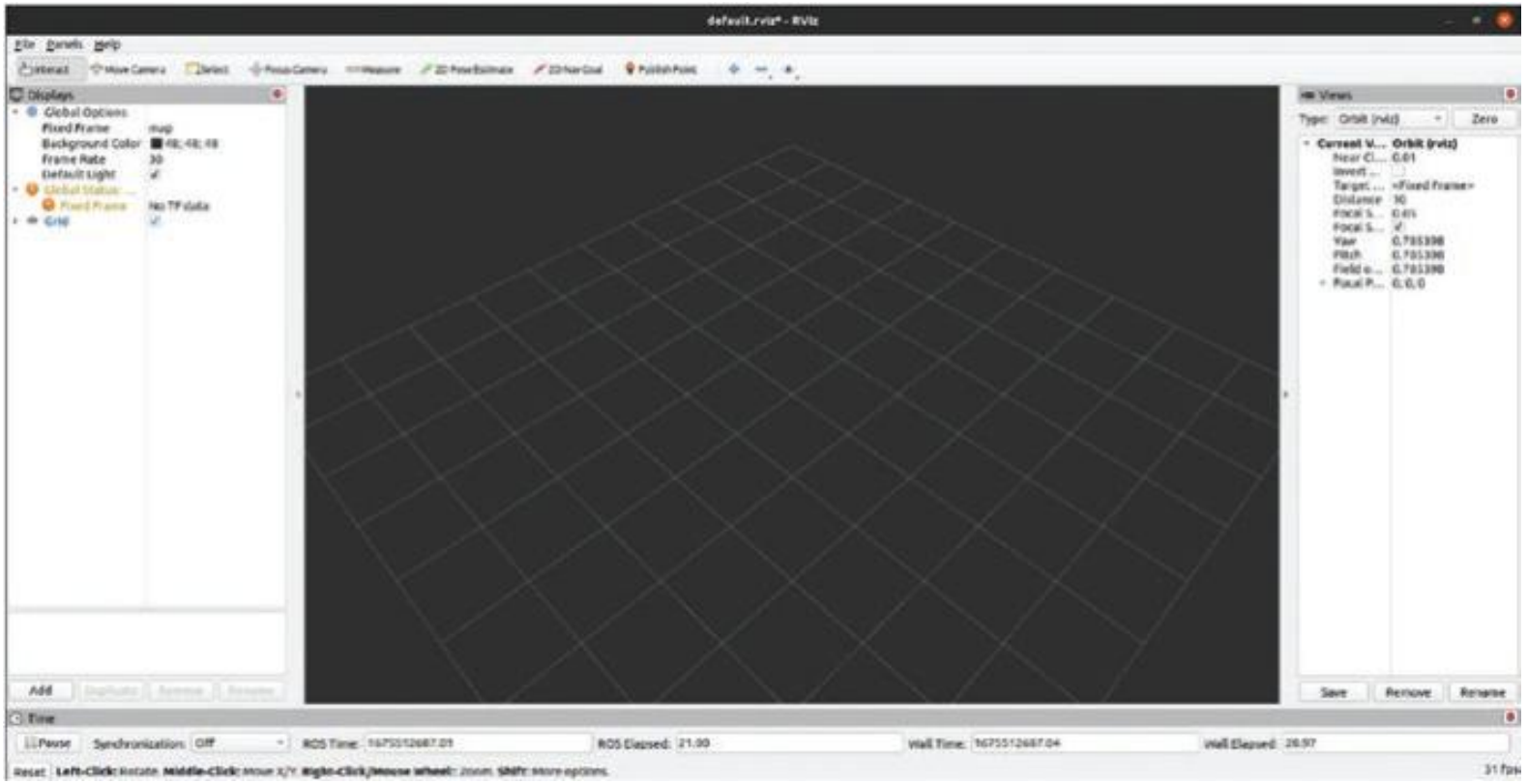


“$ gazebo” command to launch Gazebo default world

RViz

- Stands for “Robot Visualization”.
- Is a tool that visualises several aspects of a robot, such as the sensor readings, movements, joint poses, path generated by the path planner, orientation, estimate of the current pose, and so on.
- Allows you to easily visualize and debug any issues (e.g., you can visualize the path generated by a path planning algorithm in Rviz. This allows you to fine-tune the algorithmic parameters to get a better path according to your requirements).

RViz

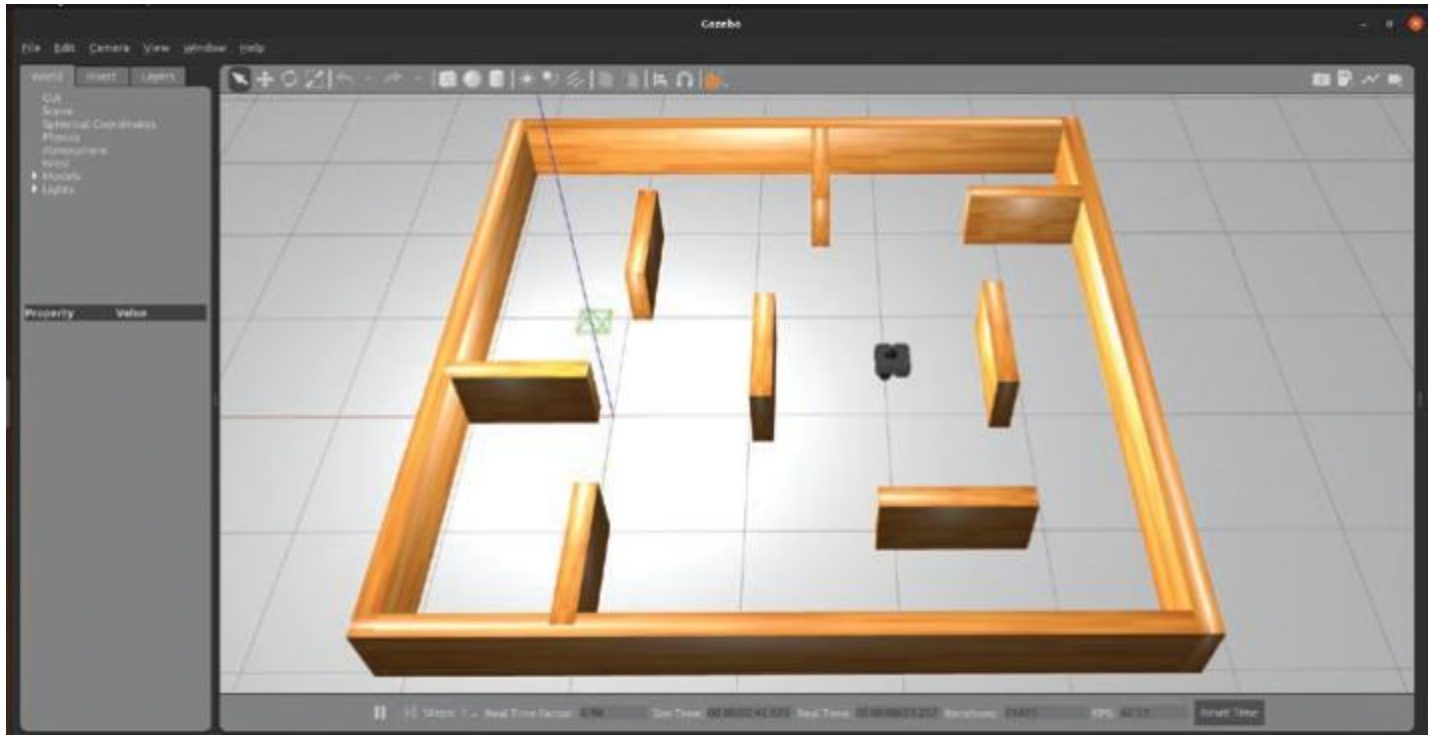


Default Rviz window

- To use the “rviz” command to start RViz

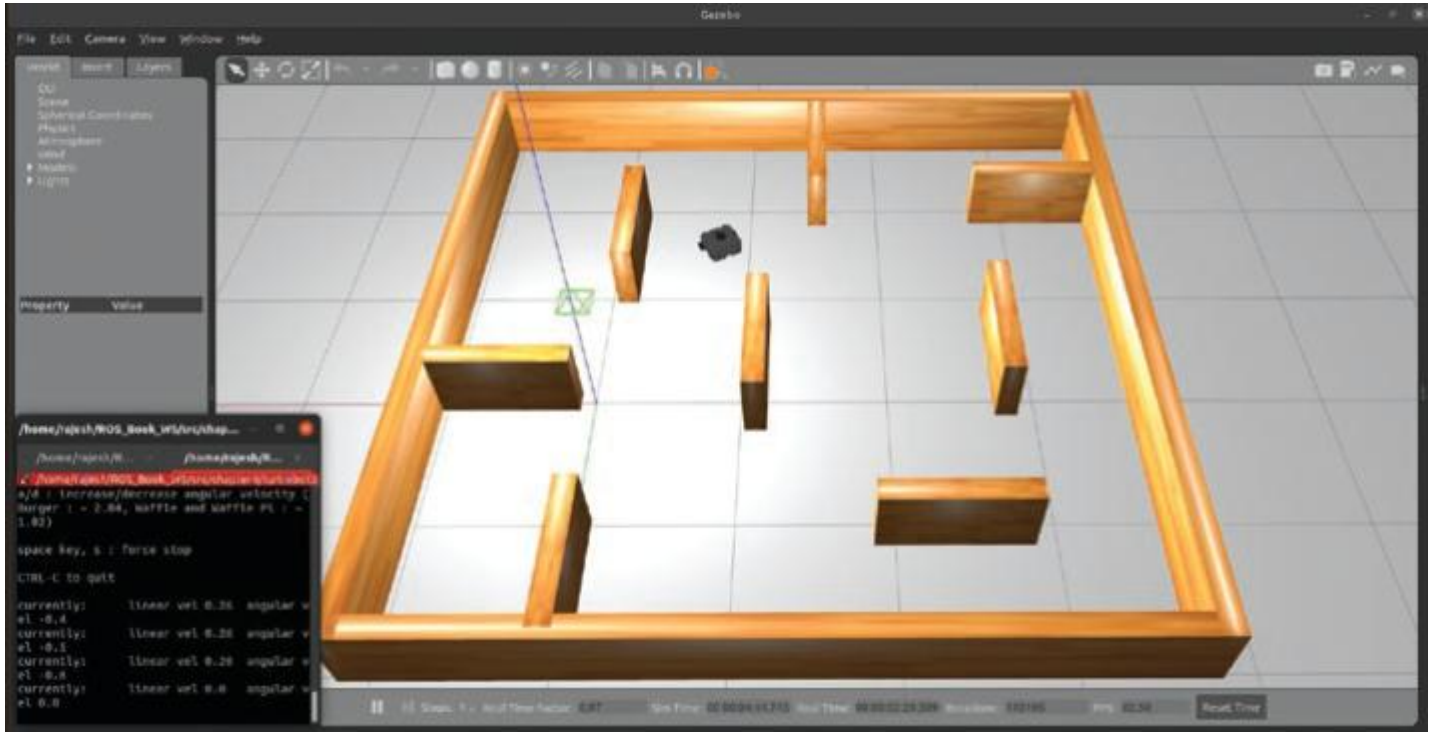
Gazebo Simulation

- A mobile robot in a Gazebo simulation.
- The robot is spawned in the environment with a wooden maze.



Gazebo Simulation - Teleoperation

- Through “Teleop” control, a mobile robot is drived using the keyboard keys as follows:
  - w – Move forward
  - a – Turn left
  - d – Turn right
  - x – Move backward
  - s – Stop movement



Mapping

- In Gazebo, mapping is the process where a simulated robot uses its onboard sensors, such as LiDAR or depth cameras, to build **a representation of its environment**.
- This results in a map, typically a 2D occupancy grid map, where different pixel values indicate the presence or absence of obstacles, enabling the robot to understand its surroundings and navigate autonomously.

Mapping

How does the Mapping work in Gazebo?

- **Sensor Data Collection**: A simulated robot in Gazebo is equipped with sensors like a LiDAR (Light Detection and Ranging) or a depth camera.
- **Data Processing**: The robot's software, often using frameworks like ROS, processes the raw sensor data to identify obstacles and features in the environment.
- **Occupancy Grid Map Generation**: The processed data is used to create an occupancy grid map.
- **2D Occupancy Maps**: In Gazebo, this is often a 2D image file where white pixels represent free space and black pixels represent occupied areas (obstacles).
- **Data Representation**: Within the map, integer values represent different levels of occupancy, with 0 for completely unoccupied, 100 for fully occupied, and -1 for unknown areas.
- **SLAM Integration**: Mapping is often performed as part of a Simultaneous Localization and Mapping (SLAM) process. SLAM allows the robot to build a map of its environment while simultaneously determining its own position within that map.

Mapping

Purpose of Mapping in Gazebo

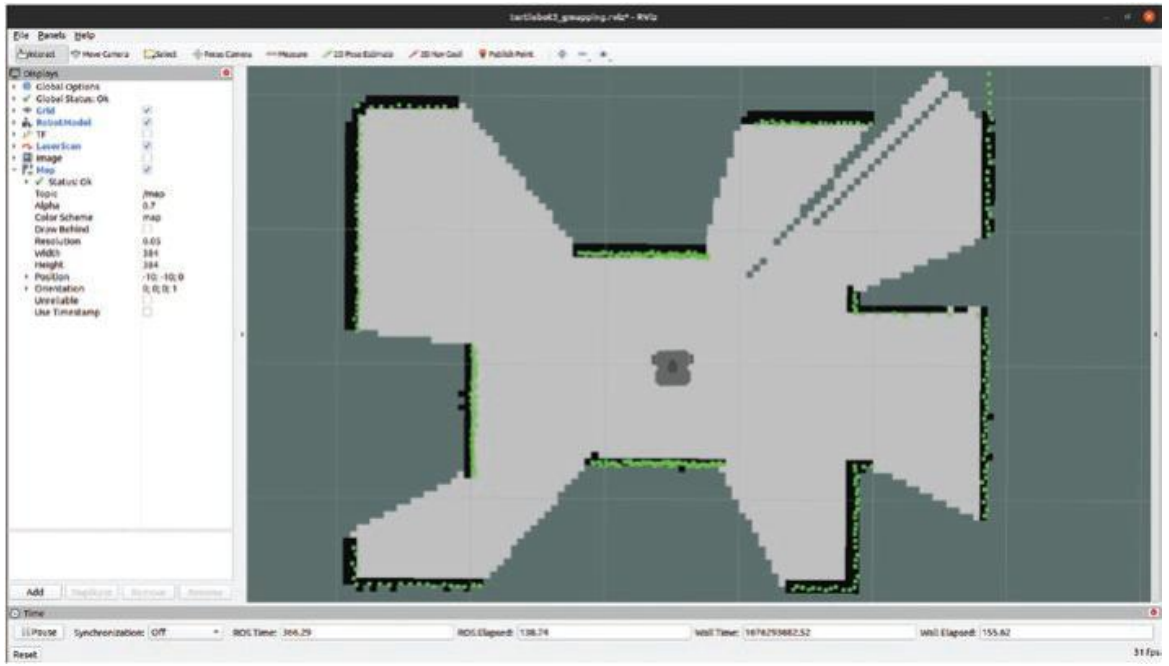
- **Navigation:** The generated map provides the robot with a comprehensive understanding of its surroundings, allowing it to plan efficient paths and avoid obstacles.
- **Localization:** By knowing its position relative to the map, the robot can accurately determine where it is in the simulated world.
- **Environment Perception:** Mapping helps the robot perceive the layout of the environment, whether it's a simple indoor space or a complex, dynamic construction site.

Mapping

- To explore the environment further, move the robot using the teleoperate command. Once the environment is mapped, save the map using this command:
  - rosrun map_server map_saver -f my_map
- Saving the map will generate two files—my_map.pgm and my_map. yaml —where the .pgm file represents the map as **an image**, and the .yaml file contains **metadata** about the map, such as resolution, origin, threshold values, and so on.

Mapping

- To perform mapping, “roslauch” command is used.
- This launches the required nodes for mapping along with the RViz visualizer.
- **Gmapping** is the default mapping algorithm.
- Other mapping algorithms (**cartographer**, **hector slam**, **karto**, and **frontier exploration**) are also available.
- To make the robot apply a mapping algorithm of your choice, add the cartographer, hector, karto, or frontier_exploration value for the slam_methods argument when launching.
- RViz shows the laser scan readings and the map created so far (see the right Figure). The map uses **white** and **black** colors, where white shows free space and black shows occupied space. The laser scan readings are shown in green.



Mapping a visualization in RViz

Navigation

- In Gazebo, robot navigation is the autonomous movement of a robot to a goal destination within a simulated environment while avoiding obstacles.
- This involves using **navigation algorithms** like **SLAM (Simultaneous Localization and Mapping)**, **path planning (e.g., A*)**, and **path tracking**, all configured and controlled within the Gazebo simulation platform to develop and test these systems in a virtual environment before real-world deployment

Navigation

How does it work?

- **Virtual Environment:** Gazebo provides a realistic 3D environment that mimics real-world physics, materials, and sensor data, allowing for the development of robots that behave realistically.
- **Robot Modeling:** A digital model of the mobile robot, including its physical characteristics and simulated sensors (like lidar), is created within Gazebo.
- **Localization & Mapping (SLAM):** The robot uses sensors in the simulation to build a map of its surroundings and determine its precise location within that map (SLAM).
- **Path Planning:** Once a map and the robot's location are known, a path planning algorithm, such as A*, is used to find the most efficient route to a designated goal.
- **Path Tracking & Control:** The robot then follows the planned path, with controllers and sensors in Gazebo providing the necessary feedback for smooth and accurate movement to avoid obstacles and reach the destination.
- **ROS Integration:** The ROS is commonly used in conjunction with Gazebo to manage the navigation system, providing frameworks for sensors, algorithms, and robot control.

Navigation

Advantages of using Gazebo for Navigation

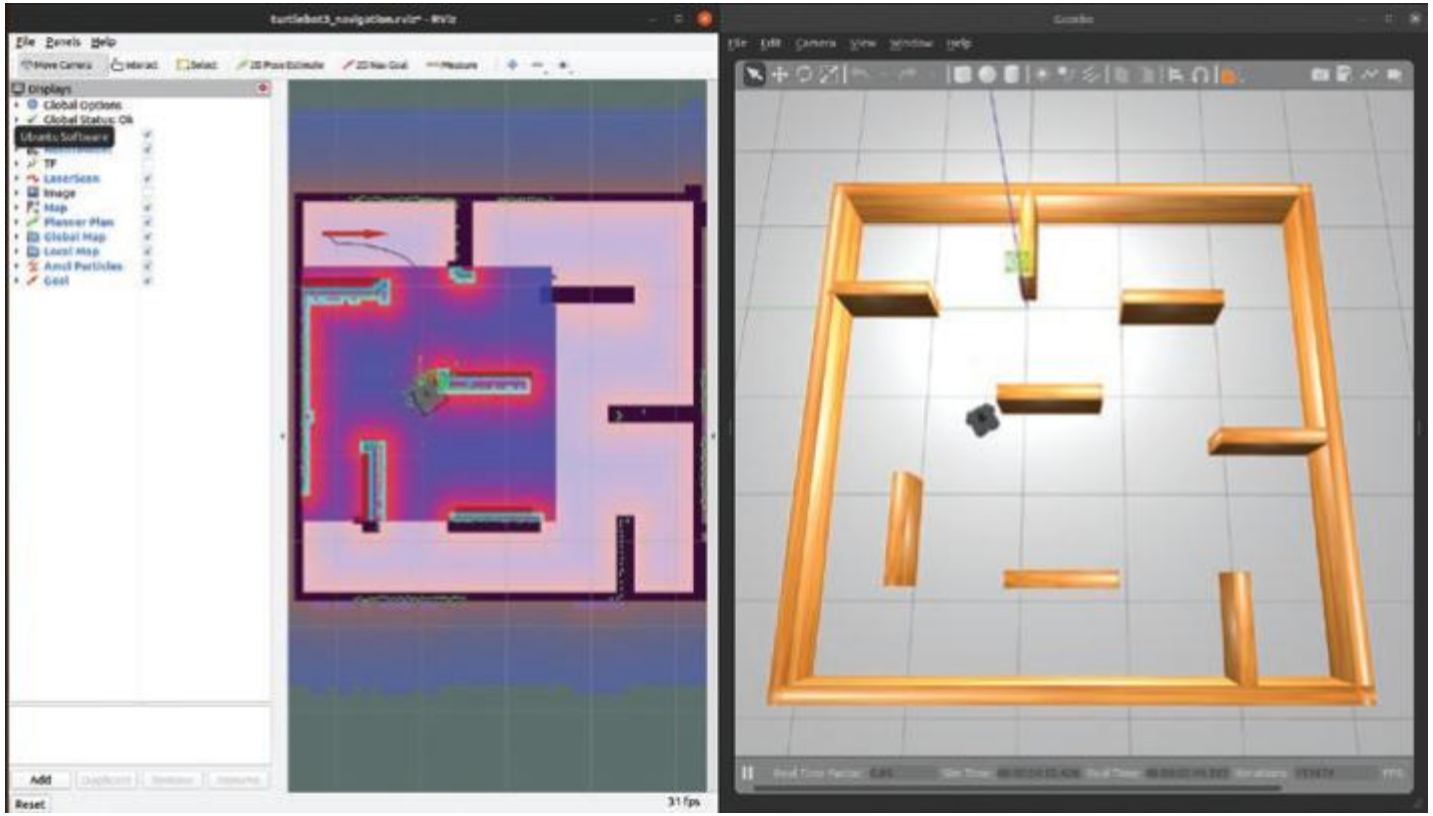
- **Cost-effective**: It's a virtual platform that avoids the costs and risks of physical testing.
- **Accelerated development**: Algorithms can be developed and iterated on quickly in simulation, speeding up the overall development process.
- **Dynamic environments:** Gazebo allows for the creation of complex and dynamic environments to test the robustness of navigation algorithms under various conditions.

Navigation

- To provide a goal pose for robot navigation, click the **2D Nav Goal** button in RViz.
- Click the 2D Nav Goal button and then click and drag it to set the required goal pose on the map.
- The robot autonomously generates a **global path**, avoiding all the **static obstacles** on the map that block the goal.
- It creates a local path that follows the global path and avoids any dynamic obstacles along the way.

Navigation

- The RViz visualizer is shown on the left. You can see the laser scan data, map, costmaps, local/global paths, goal pose, localization estimate, and so on.
- The Gazebo simulation is shown on the right and it depicts the actual behavior of the robot.



Left: RViz visualization

Right: Gazebo simulation

Digital Twins for Robotics Simulation

What is a digital Twin (DT)?

- A **Digital Twin (DT)** creates **high-fidelity virtual models** of physical objects in virtual space in order to simulate behaviors in the real world and provide feedback.
- A DT reflects a **bi-directional dynamic mapping process**; it breaks the barriers in the product lifecycle and provides a **complete digital footprint of products**.
- As a result, DTs enable companies to predict and detect physical issues sooner and more accurately, **optimise manufacturing processes**, and **produce better products**.





Digital Twins for Robotics Simulation

What is a digital Twin (DT)?

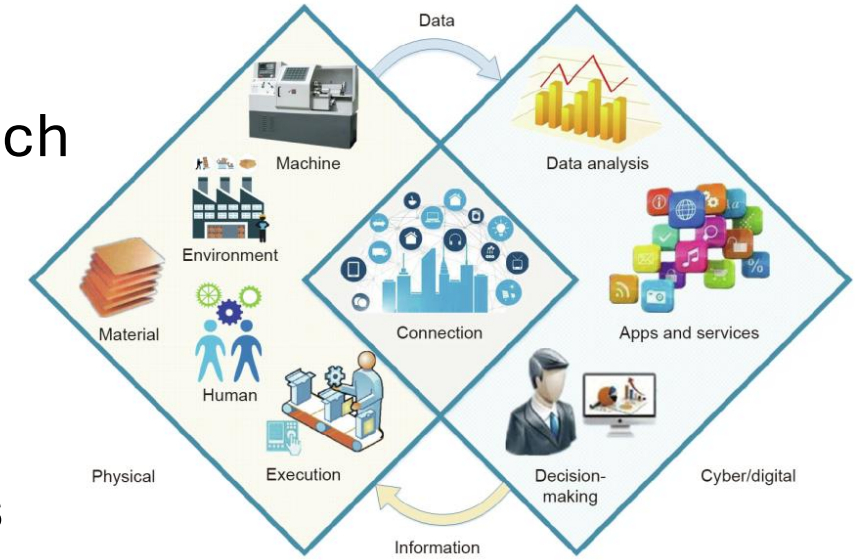
- A digital twin can also be considered a **digital replica** of a living or non-living physical entity.
- It refers to a digital replica of potential and actual physical assets (physical twin), processes, people, places, systems and devices that can be used for various purposes.
- The digital representation provides both the elements and the dynamics of how an Internet of things device operates and lives throughout its life cycle.

"Digital Twin". https://en.wikipedia.org/wiki/Digital_twin, Wikipedia. Retrieved October 2025.

Digital Twins for Robotics Simulation

Digital-Physical Mapping in Digital Twins (DTs)

- In manufacturing, both Cyber-Physical Systems (CPS) and DTs include two parts: the physical part and the cyber/digital part.
- The physical part consists of various manufacturing resources like Robotic Arms and Mobile Robots, which can be summarised as “human/machine/material/environment
- The cyber/digital part, which has various ubiquitous apps and services, incorporates smart data management, analytics, and computing capabilities



Digital Twins for Robotics Simulation

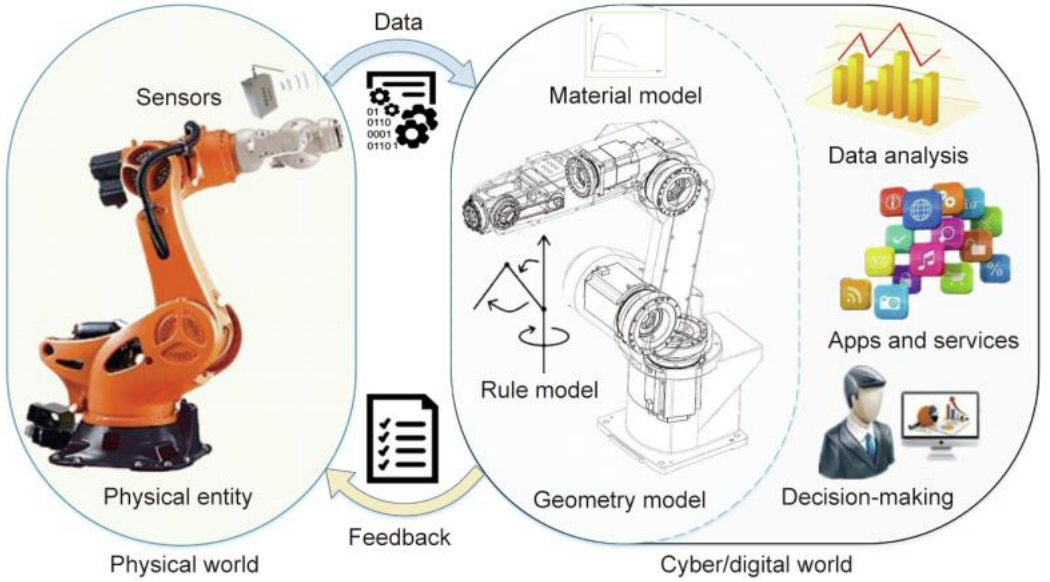
Digital-Physical Mapping in Digital Twins (DTs)

- The vision for DTs is to provide a **comprehensive physical and functional description** of a component, product, or system.
- The first and most important step is to create **high-fidelity virtual models** to realistically reproduce the geometries, physical properties, behaviours, and rules of the physical world.
- The virtual models and physical entities have a similar appearance, like twins, and the same behaviours, like a mirror image.
- The virtual models in a digital environment can directly optimise the operations and adjust physical process through feedback.
- Using **bi-directional dynamic mapping**, the physical entities and virtual models **co-evolve**.
- The mapping relationship between the physical and digital worlds of a DT provides a **one-to-one correspondence**.
- A virtual model, which integrates geometry, structure, behaviour, rules, and functional properties, represents a **specific physical object**.

Digital Twins for Robotics Simulation

Digital-Physical Mapping in DTs

- **Virtual models** can be used as a supplement to enrich the composition and functions of CPS.
- DT technology can be considered a necessary foundation for building CPS and for opening the way to the realisation of CPS
- The combination of CPS and DTs would help manufacturers achieve more precise, better, and more efficient management



Digital Twins for Robotics Simulation

Digital-Physical Mapping in DTs – Controls in CPS and DTS

- **Control** is aimed at maintaining a system at **an acceptable level of operational normalcy in response to disturbances**.
- Control is the **core function** of CPS and DTs.
- The virtual models and physical processes of a DT co-evolve over the lifecycle of the product or system.
- The control in CPS and DTs includes two parts:
  - The physical assets or processes affecting cyber/digital presentation, and
  - The cyber/digital representation control of physical assets or processes.
- For consistency, real-time data from the physical world are collected using sensors, and are communicated to the cyber/digital world in order to **drive the cyber/digital elements in synchronisation with the physical world**.
- Some disturbances, including threats of an unexpected and malicious nature, can be eliminated through cyber-physical interactive control.

Digital Twins for Robotics Simulation

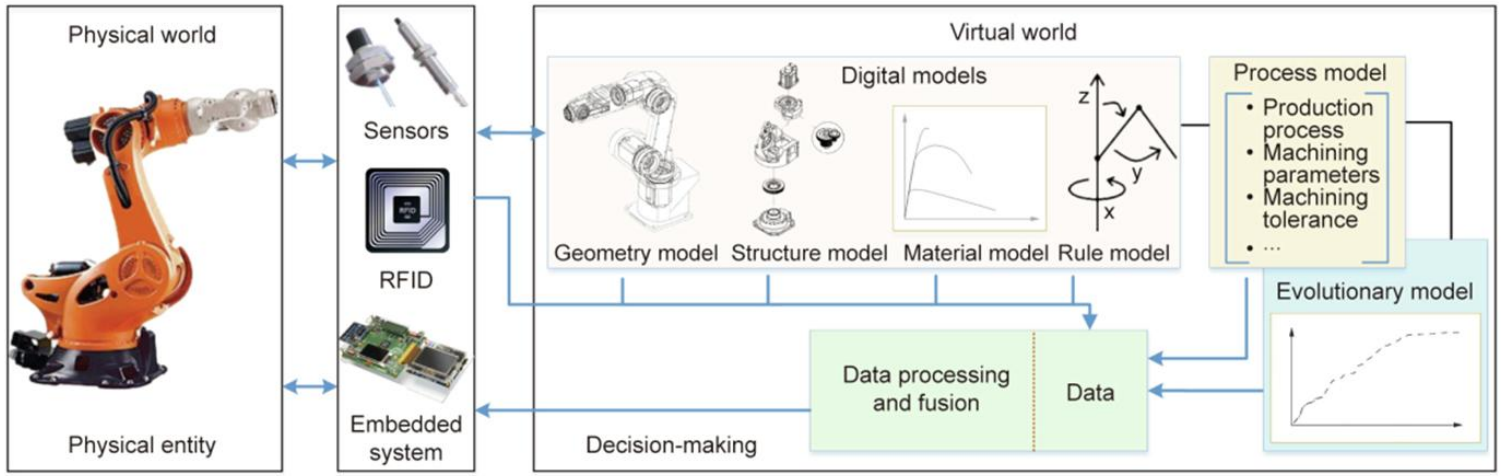
Function Implementation of DTs

- The main idea of a DT is to create a **digital copy** (i.e., virtual models) for physical entities in order to
  - simulate and reflect their state and behaviours through modelling and simulation analysis, and
  - predict and control their future states and behaviours through feedback.
- The DT integrates whole elements, the entire business, and the process data to ensure consistency.
- The virtual models integrating the geometry, structure, material properties, rules, process, and so forth enable **the digitisation and visualisation of the production system and process**.

Digital Twins for Robotics Simulation

Function Implementation of DTs

- The virtual models serve as a **communication and recording mechanism** to help interpret the behaviours of machines or systems and to predict their future state based on real-time data, historical data, experience, and knowledge, and on data from the models.
- The **virtual models** and **data** can be considered as the core elements of a DT.
- Advances in **new IT** have had a profound impact on DTs.



Digital Twins for Robotics Simulation

Function Implementation of DTs

- The Internet of Things (IoT), Cloud Computing (CC), Big Data Analytics (BDA), and AI the core element of the new IT.
- In manufacturing, due to digitalisation, manufacturing resources that use industrial technology to generate a large volume of various kind of data.
- Such data can be collected in real time to be stored and computed because of IoT.
- With computing and storage resources, CC can effectively meet the needs of data computing and storage, while big data technology can effectively mine hidden useful information knowledge, thus improving the intelligence and better satisfying the dynamic service demand.

