

IsaacSim5G

A Virtual Experimentation Platform for
5G-Connected Robotics

Varun Raghavendra, **Rajeev Gangula** and Tommaso Melodia
Northeastern University

5G for Connected Robotics



URLLC

Ultra-low latency

< 1 ms response

Real-time control, haptic feedback, and instant safety stops.



mMTC

Massive connectivity

1,000s of devices

Thousands of robots and sensors on a single network.



Slicing

Network slicing

Guaranteed QoS

Isolated virtual lanes protect safety-critical operations.



MEC

Mobile edge computing

< 5 ms compute

On-site compute for real-time AI inference.



TSN

Time-sensitive networking

Deterministic

Synchronized timing coordinates multi-robot fleets.

The Experimentation Problem

Even with a private 5G network and robots, real-world experimentation is painful!

Expensive Hardware

Real robots, 5G equipment, and test environments cost

Safety Constraints

Limits what you can test — can't crash a real robot!

Human Hours

Setup, calibration, a single experiment run takes hours of effort

No Repeatability

Environmental conditions, hardware state, and timing are never identical

No Synthetic Data

Can't generate edge cases, rare events, or controlled impairments on demand

Ground Truth

Getting precise position, timing, and channel ground truth in the field is hard

Physics Simulators

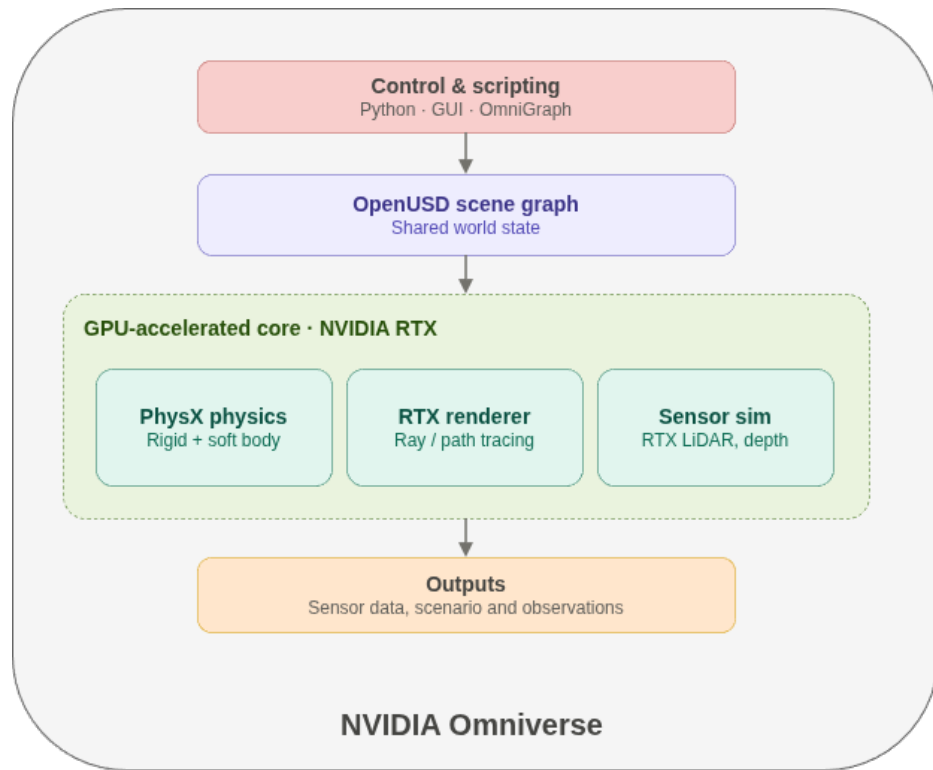
What Do They Provide?

- Replicate real-world physics: gravity, friction, rigid-body dynamics & collision detection
- Photorealistic rendering for digital twins
- Model sensors (LiDAR, cameras, IMU, GPS) with noise and actuator dynamics models
- Robot software integration (ROS 2, PX4)
- Repeatable, controlled experiments
- Support Hardware-in-the-Loop (HIL) and Software-in-the-Loop (SIL) testing pipelines

Domain	Simulator	Highlights
Ground Robots	Gazebo / Ignition	ROS-native, rich plugin ecosystem
Ground Robots	Webots	Open-source, cross-platform
Manipulation	MuJoCo	High-fidelity contact physics, RL
UAV / Drones	AirSim (MS)	Unreal Engine, photorealistic, multi-rotor
UAV / Drones	RotorS (Gazebo)	Multi-MAV, ROS-integrated
Swarms	ARGoS	Lightweight, scalable swarm sim
Autonomous Vehicles	CARLA	Urban driving, OpenDRIVE maps
General / RL	Isaac Sim (NVIDIA)	GPU-accelerated, Omniverse-based

Isaac Sim

- End-to-end GPU-accelerated simulation platform for AI-driven robotics.
- Powered by [NVIDIA Omniverse](#) and OpenUSD
- High-fidelity USD scene graphs, PhysX engine, and RTX rendering
- Robot software integration (ROS, ROS 2)
- Multiple execution modes
 - *Playback time (60 Hz, rendering-oriented)*
 - *Physics step (solver-level, dynamics-oriented)*



Isaac Sim

Simulator	Physics Engine	Ray Tracing	Parallel Rendering	Sensors Stack	RL Integration	SynData Pipeline	ROS1/2 Integration	Digital Twin/Industry	Dynamics
Isaac Sim ¹	PhysX (GPU)	✓	✓	✓	✓	✓	✓	✓	R;D;C;F
Gazebo	ODE / DART / Bullet / Simbody	✗	✗	✓	○	✗	✓	○	R
MuJoCo	MuJoCo	✗	✗	○	✓	✗	○	✗	R;D;C
PyBullet	Bullet	✗	✗	○	✓	✗	○	✗	R;D;C
Webots	ODE	✗	✗	✓	○	✗	✓	○	R;C [○]
CoppeliaSim	Bullet / ODE / Vortex / Newton	○	✗	✓	○	✗	✓	○	R;C [○]
CARLA	PhysX	○	○	✓	○	○	○	✗	R
Habitat	Bullet	✗	○	○	✓	○	✗	✗	R
Unity	PhysX	○	○	○	✓	✓	○	○	R;C

Table 1: Functional comparison of popular simulation platforms, covering rendering, sensor simulation, reinforcement learning (RL) support, synthetic data generation, ROS/ROS 2 integration, digital-twin capabilities, and physical dynamics (R: rigid, D: deformable, C: cloth, F: fluid). ✓ denotes strong native support; ○ indicates partial or plugin-based support; ✗ denotes limited or no support.

[Source] Gao Sicong, Maurice Pagnucco, Tomasz Bednarz, and Yang Song. "NVIDIA Isaac Sim: Enabling Scalable, GPU-Accelerated Simulation for Robotics." *arXiv preprint arXiv:2606.03551* (2026).

IsaacSim5G

- A virtual experimentation platform for 5G-connected robotics
- Coupling **OpenAirInterface (OAI)** and **NVIDIA Isaac sim** with ROS 2 integration

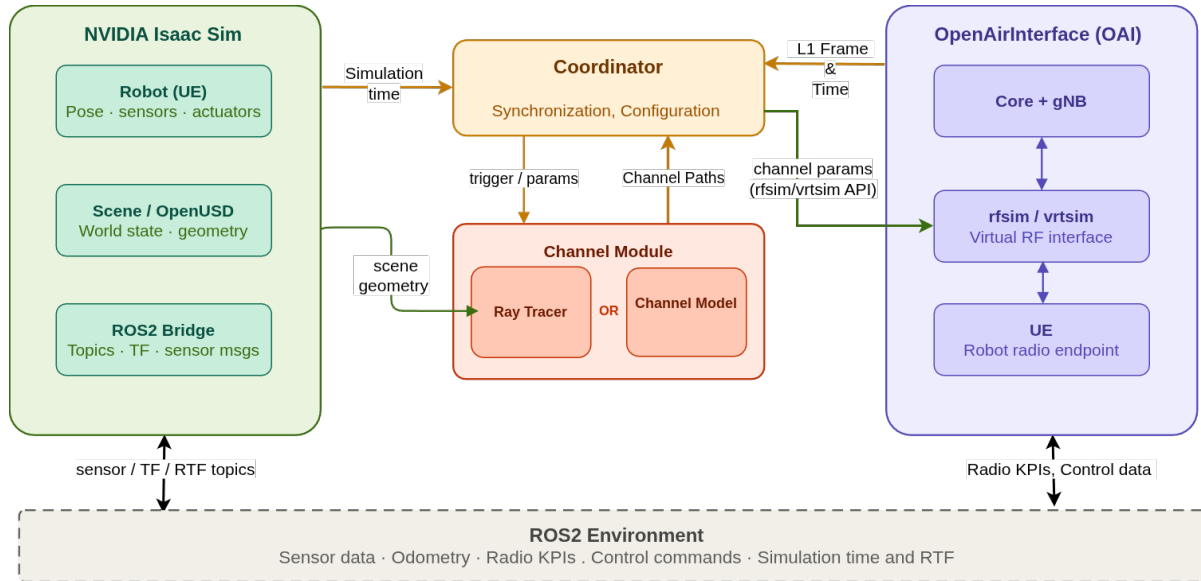
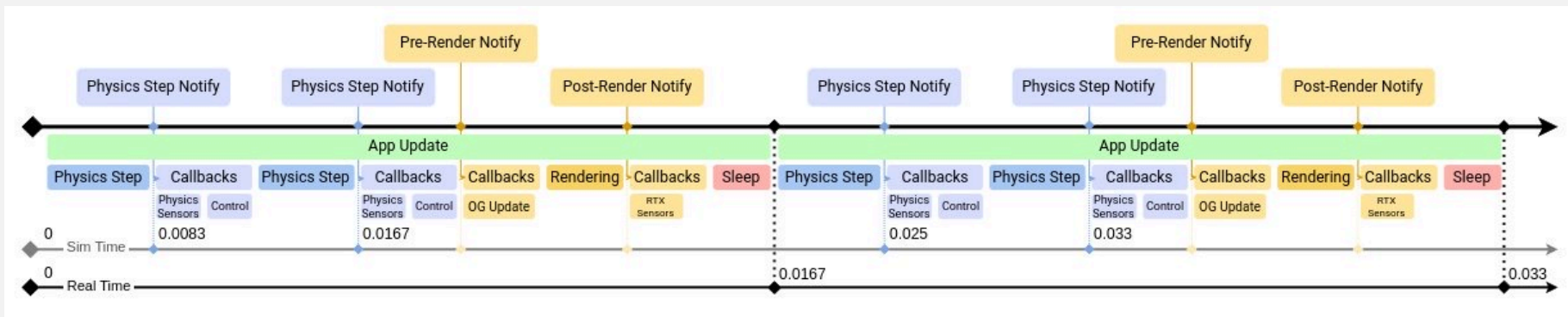


Figure: Isaacsim5G Architecture in single Robot (UE) scenario

Isaac Sim Simulation Timeline



SIM TIME \neq REAL TIME

- Physics can run faster or slower than real-time depending on scene complexity
- Isaac Sim uses a real-time limiter by default
- Multiple physics steps may occur per rendered frame

KEY TIMING PARAMETERS

Parameter	Description
<code>physics_dt</code>	Time per physics step (default 1/60 s)
<code>rendering_dt</code>	Render frame period (default 1/60 s)
<code>render_interval</code>	Physics steps per rendered frame
<code>steps/second</code>	Set in Physics Scene (default 60 Hz)

Synchronization Problem

OAI L1 Time

- Radio frame timing through sample count
- Fixed RTF target

Isaac Sim Time

- Simulation clock runs independently
- RTF can be measured

Wall Clock

- Reference pivot
- OAI and Isaac sim simulation time offsets measured against it
- Drift detection baseline

- Robot has **Physical state** and **Network state**
- `isaac_offset` = `isaac_sim_time` - `wall_time`
- `oai_offset` = `oai_sim_time` - `wall_time`
- Mapping between OAI frame and Isaac time is crucial!

Isaacsim5G

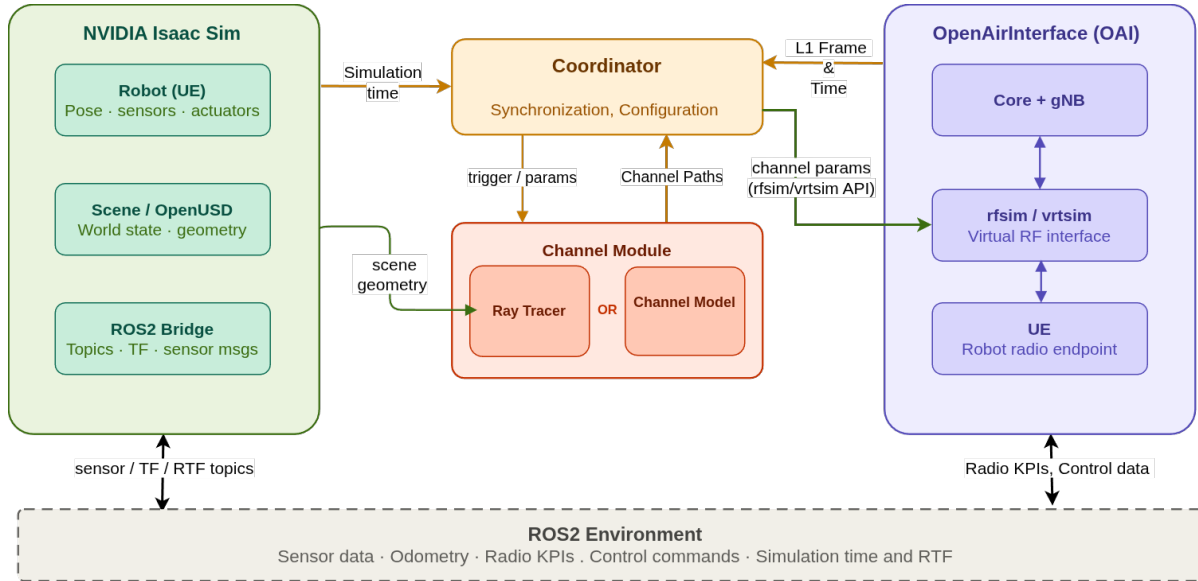


Figure: Isaacsim5G Architecture in single Robot (UE) scenario

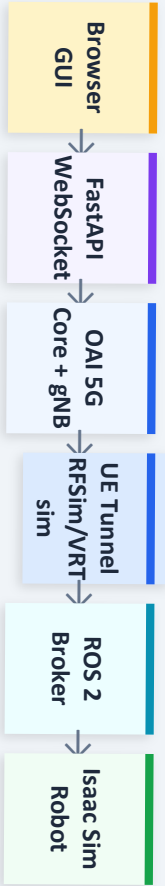
- A virtual experimentation platform for 5G-connected robotics
- Coupling **OpenAirInterface (OAI)** and **NVIDIA Isaac sim**
- ROS2 integration

Enhancements to OAI

- Enforcing real time factor (RTF) in RFSIM
- Inducing channel paths with **{Delay and Doppler}** in RFSIM and VRTSIM

Application 1: Teleoperation over 5G

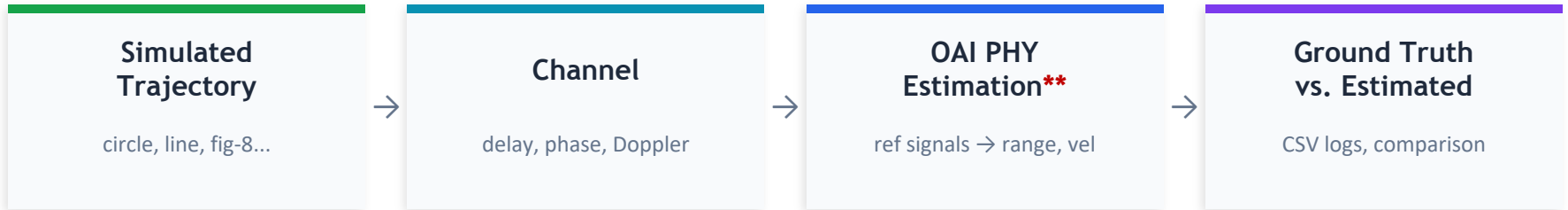
The screenshot displays a multi-window environment for teleoperating a robot over 5G. On the left, a terminal window shows network performance metrics and logs for a UE (User Equipment) connected to a 5G network. The main area is a 3D simulation of a warehouse environment, featuring a yellow forklift and a white mobile robot. On the right, a web browser window displays the 'IsaacSim5G Teleop' interface, which includes a live video feed of the simulation, a control panel with directional buttons (W, A, S, D) and a 'STOP' button, and various speed and rate sliders (Linear speed, Angular speed, Send rate). The interface also shows a 'Drive' section with a 'W/S forward-reverse' indicator and a 'UE Tunnel sim' section. At the bottom, an 'ACTIVITY LOG' window shows configuration details for the platform and camera setup.



Application 2: Localization and Tracking

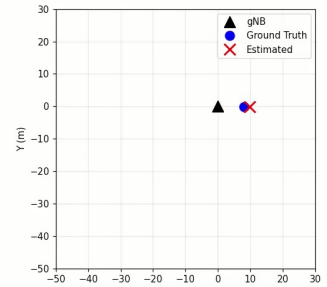
The ground truth problem: In real localization experiments, precise ground truth requires expensive motion capture systems, GPS is unreliable indoors, and repeating the exact same trajectory with identical conditions is expensive.

Solution: Ground truth comes from the physics simulator — exact position, velocity, heading at every timestep. Using 5G reference signals in OAI, we estimate range and Doppler shift. Compare estimation vs. ground truth perfectly, sweep parameters, repeat endlessly.

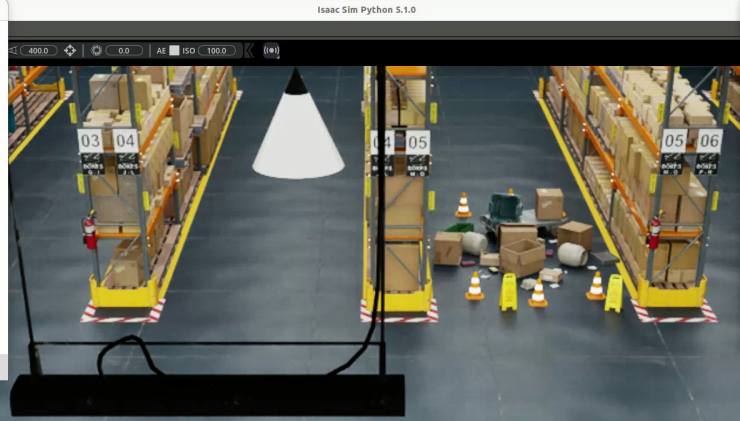


**Rajeev Gangula, Tommaso Melodia, Rakesh Mundlamuri, and Florian Kaltenberger, "Round trip time estimation utilizing cyclic shift of uplink reference signal." In *IEEE International Conference on Communications (ICC)*, 2025.

Figure 1
5G Robot Localization: Ground Truth vs Estimated



GT points: 500 | Est points: 500 | gNB(m): (0,0) z=3.0m



http://localhost:8080 133% fps

topIC: /Isaac/ViewPort2

Drive

W/S forward-reverse - A/D strafe left-right - Q/E rotate left-right - Space E-stop

↑ W

← A STOP → D

↓ S

Linear speed (m/s) 1.50

Limitations & Future Directions

- **Resource-intensive** — heavy multi-process stack (Isaac Sim + OAI + ROS 2), high-fidelity rendering + full 5G stack demands significant compute
- **Adaptive brokering** — learned delay distributions and adapting the RTF
- Multi-robot orchestration and coordination
- **ISAC** — Integrated sensing and communication with Digital Twins