st-mems-sim2real

Data brief

Sim2Real IMU sensor for NVIDIA Isaac Sim

Product status link
SIM2Real IMU sensor

Product summary	
Software reference	Sim2Real IMU sensor
Package version	1.0.2
Host environment	Ubuntu 22.04, NVIDIA Isaac Sim Full 5.1.0, Python 3.10
Supported models	ASM330LHH
Input data	Clean linear acceleration and angular velocity from the native Isaac IMU on the attached link
Output data	Realistic intertial sensor acceleration and angular velocity stored on the custom sensor prim
Default sampling rate	ASM330LHH: 104 Hz
Configuration method	Per-model JSON file in `data/models/`
Native backend	`sim2real_native_v0_1.so` shared library

Features

Key benefits

- Accelerates IMU simulation setup in NVIDIA Isaac Sim through native `Create → Sensors` menu integration
- Brings ST IMU model behavior into robotic digital twins with configurable sampling rate, full-scale ranges, and vibration enablement
- Supports repeatable verification workflows by comparing realistic inertial Sim2Real output against clean Isaac IMU truth data

Main features

- Native menu integration inside Isaac Sim
- Automatic attachment to the currently selected stage prim
- Visible sensor marker in the 3D viewport
- Physics-step-driven runtime with output data rate scheduling
- JSON-based model configuration for supported IMUs
- Realistic noise-augmented acceleration and gyroscope output generation
- Verification scripts for trajectory logging and plot generation

Typical use cases

- Robot digital twin development
- Synthetic IMU data generation for algorithm evaluation
- Regression testing of estimation, filtering, and sensor-fusion pipelines
- Early evaluation of ST IMU behavior in robotics simulation scenarios
- Demonstrations and internal prototyping workflows in Isaac Sim

Description

Sim2Real IMU sensor provides a practical path to integrate ST IMU behavior into NVIDIA Isaac Sim with minimal setup effort. By combining graphical insertion, model-based configuration, and physics-driven realistic inertial sensor output generation, it helps accelerate simulation-based development and verification workflows for robotics teams using ST inertial sensing concepts in digital environments.

The software creates a custom IMU prim under the selected robot link and marks it visually in the viewport for quick inspection. During simulation, the extension reads clean inertial truth from the corresponding Isaac IMU, applies the Sim2Real native model, and publishes the resulting realistic measurements as custom data on the spawned prim.

This approach is intended to help robotics and perception teams evaluate software stacks against more realistic inertial sensor behavior without requiring custom scripting each time a sensor is added to a scene.

1 Overview

Sim2Real IMU sensor is a software extension for NVIDIA Isaac Sim 5.1.0 that adds STMicroelectronics IMU models directly to the simulation environment. It enables users to place supported IMU instances from the graphical menu, attach them to a selected robot link, and generate physics-driven, noise-augmented and realistic inertial measurements for development, validation, and demonstration workflows.

The current software package supports the `ASM330LHH` sensor model. The model is instantiated from a JSON profile and executed through a runtime scheduler synchronized to the Isaac Sim physics step, allowing the generated IMU stream to follow the configured output data rate independently of the render loop.

1.1 Software architecture

The software package is organized around four main building blocks:

- Extension layer: registers the sensor menu entry and spawns the custom IMU prim
- Configuration layer: loads model-specific JSON parameters from `data/models/`
- Runtime layer: subscribes to Isaac Sim physics events and schedules sensor updates at the configured ODR
- Native backend layer: applies the compiled Sim2Real model to the clean inertial truth stream

1.2 Integration flow

1. Copy the extension into an Isaac Sim extension search path.
2. Point `SIM2REAL_NATIVE_PATH` to the directory containing `sim2real_native_v0_1.so`.
3. Enable the extension from the Isaac Sim Extensions window.
4. Select a robot link in the stage and place an ST IMU from `Create → Sensors → STMicroelectronics IMU`.
5. Start the simulation to begin runtime sensor updates.

1.3 Verification workflow

The package includes a verification flow based on a Franka robot example:

- `verification_script.py` records clean and realistic IMU trajectories
- `plot_verification.py` generates comparison plots for all six inertial axes

This workflow helps users validate that the realistic output remains aligned with the clean motion profile while exhibiting the expected realistic sensor noise characteristics.

1.4 Performance and constraints

- Runtime responsiveness depends on the host workstation, physics configuration, and selected output data rate.
- Fixed RAM and flash metrics are not applicable in the same way as for embedded MCU software, since the package runs as a host-side Isaac Sim extension.
- The native backend must be available as a compiled shared library compatible with Linux x86_64 and Python 3.10.
- Current support is limited to the one included IMU model profile.
- The software is intended for simulation and development workflows, not for replacing full hardware characterization or application-level safety validation.

Revision history

Table 1. **Document revision history**

Date	Version	Changes
01-Apr-2026	1	Initial release

Contents

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

The purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of the purchasers’ products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.