

# Using Industrial Robotic Arms in NVIDIA Omniverse Isaac Sim

Dr. rer. nat. Erdal Pekel

*info@erdalpekel.de*

ROS Meetup Stuttgart  
September 18, 2024

# Presentation Overview

## 1 Introduction

- Personal Info
- Omniverse Isaac Sim
- Presentation Scope

## 2 Main

- System Requirements
- User Interface
- Robot Import
- Isaac Sim Extension
- Isaac - ROS connection

## 3 Demo

## 4 Conclusion

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion

# Personal Info

## CV

- Information Systems studies at Technical University of Munich from 2013 to 2019
- Researcher at Technical University of Munich from 2019 to 2023
- PhD from Informatics Faculty, thesis title "X-ray Computed Tomography with a Robotic Sample Holder"
- Currently assisting teams as a consultant for robotic software development and simulation

# Personal Info - Dissertation Project

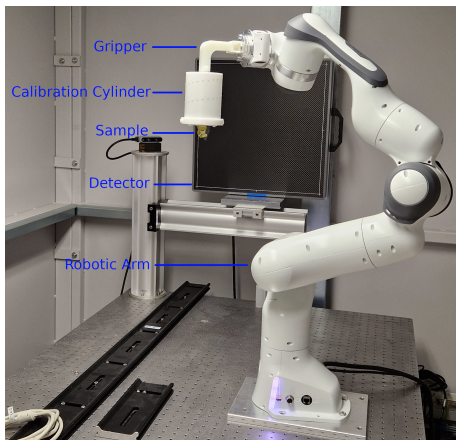


Figure: Dissertation Project Setup.

## Interests

- ROS and all technologies related to the ecosystem
- Simulation technologies such as NVIDIA Omniverse Isaac
- Robotic arms, hardware associated with the operation of robotic systems, e.g. depth cameras
- Latest developments in C++ and Python



# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion

# Omniverse Isaac Sim

# Omniverse Isaac Sim

- Scalable robotics simulation built on omniverse platform
- Python and C++ API
- Operates on OpenUSD standard
- Physics simulation and ray-tracing
- Sensor simulation: RGB, Lidar, Radar, IMU, etc.
- ROS bridge for interaction with ROS
- Large database of existing 3D models

# Omniverse Isaac Sim

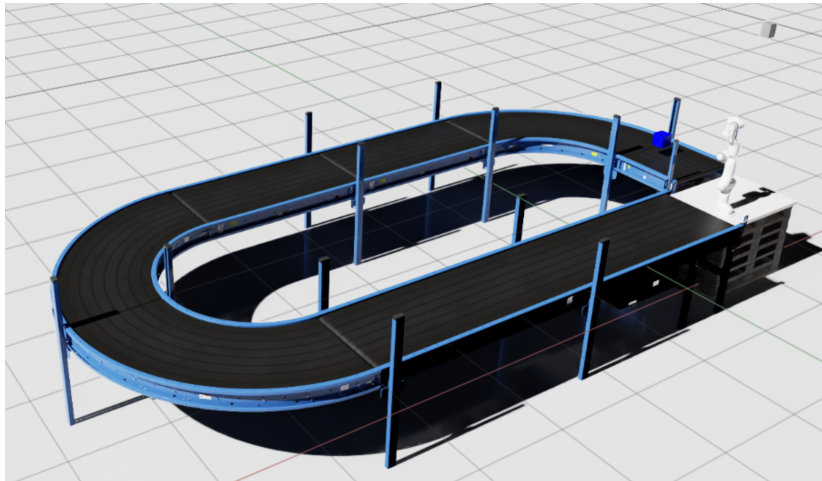


Figure: Sample Scene with Panda Robot in Omniverse Isaac Sim.

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion

# Presentation Scope

# Presentation Scope

- Introduce key aspects of using industrial robotic arms with ROS in Isaac Sim
- Sample project: Control Panda robot with ROS2 / MoveIt using RViz
- Provide reference to GitHub repositories with Code

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion



# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

**System Requirements**

User Interface

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion

# System Requirements

# System Requirements

- NVIDIA RTX ready GPU
- Specific GPU driver version (see docs)
- Ubuntu 22.04
- ROS2 Humble
- Containerization with Docker possible
- Windows is also an option

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

**User Interface**

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion

# User Interface

# User Interface

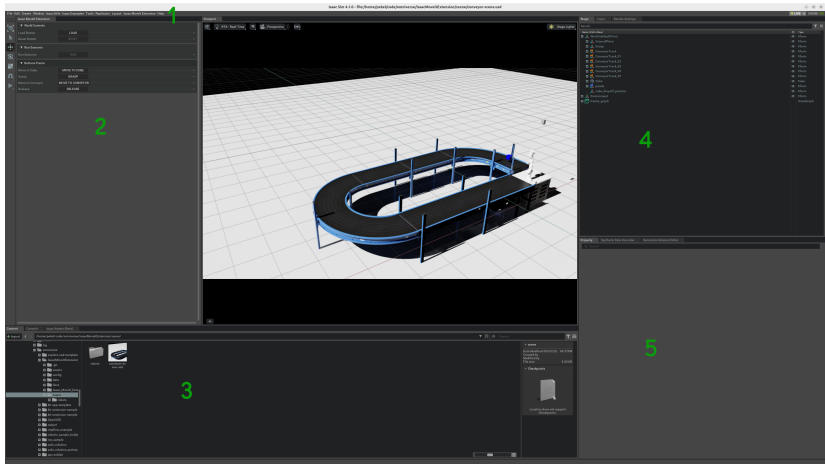


Figure: Omniverse Isaac Sim User Interface.

- 1 Menu bar: App settings and loaded extensions
- 2 Sample extension's window with controls
- 3 Content explorer for drag & drop model insertion
- 4 Scene structure explorer
- 5 Scene element properties

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

**Robot Import**

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion



# Robot Import

# Robot Import

## URDF

- 1 Source ROS2 workspace containing robot mesh files in `.bashrc`
- 2 Include inertial and mass values in robot definition
- 3 Use URDF importer to convert robot model into USD format

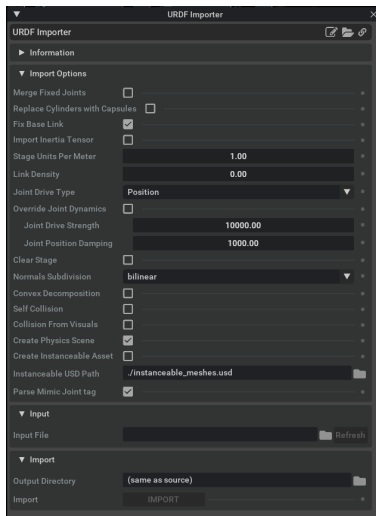


Figure: Isaac Sim URDF-importer.

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

Robot Import

**Isaac Sim Extension**

Isaac - ROS connection

## 3 Demo

## 4 Conclusion

# Isaac Sim Extension

# Isaac Sim Extension

## Scene

- 1 Create empty scene
- 2 Add robotic arm's USD to scene
- 3 Save scene

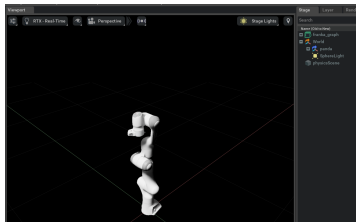


Figure: Add robot to scene.

## Extension

- 1 Create extension with Isaac Sim extension template generator
- 2 Load scene with robotic arm at extension startup



Figure: Isaac Sim Extension Generator.

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion

# Isaac - ROS connection

# Isaac - ROS connection: Overview

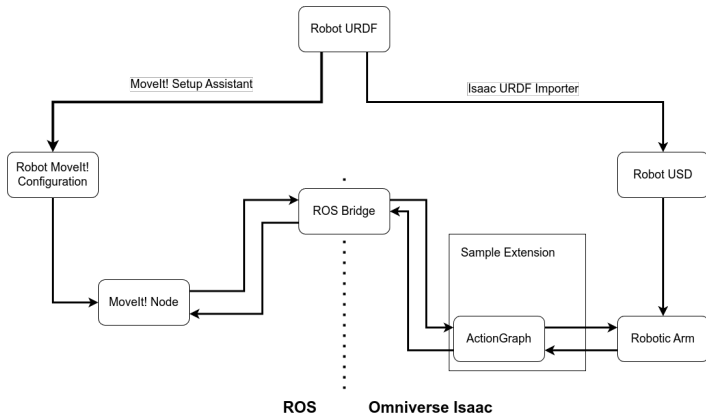


Figure: Isaac Sim - ROS Overview.



- Modeling actions that are executed with every simulation cycle
- In our case, we
  - read ROS simulation time,
  - publish TF tree of simulation objects,
  - publish robot joint state and
  - read desired robot joint states.

# Isaac - ROS connection: Action Graph

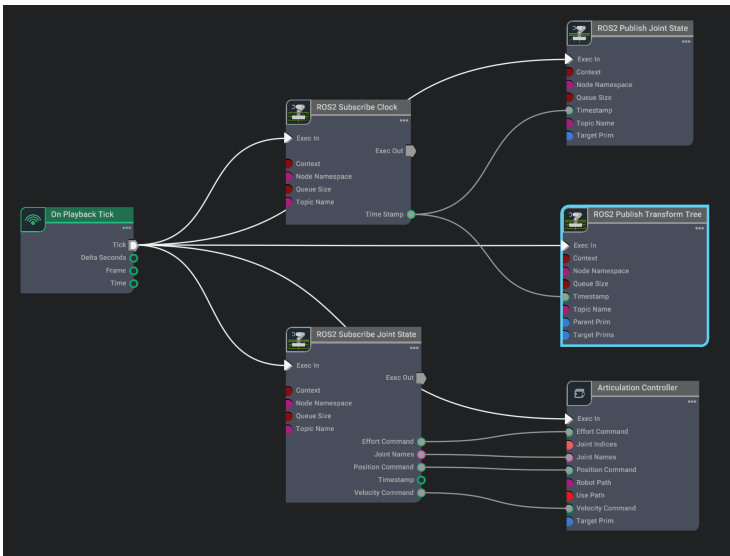


Figure: Isaac Sim Action Graph.

# Isaac - ROS connection: Clock Synchronization

- Clock synchronization between Isaac Sim and ROS time necessary for key components
- E.g., timestamps of TF transforms or other ROS messages
- One option: Broadcast simulation time with custom ROS node to Isaac Sim
- Ingest ROS simulation time to Isaac Sim via Action Graph

# Isaac - ROS connection: Topic-based control

- The robotic arm runs inside Isaac Sim
- *ros\_control* has no direct access to robot's joints
- *topic\_based\_control* interface allows joint control via ROS topics
- ROS topics 'joint\_states' and 'joint\_commands' are used for control
- Isaac Sim knows about these topics with Action Graph initialization

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion

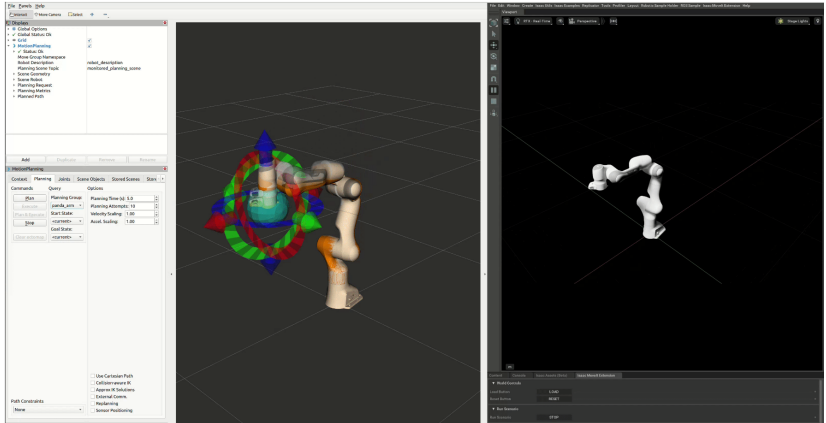


Figure: RViz Isaac Sim Demo.

## Important Links:

- Personal blog <https://erdalpekel.de/>
- GitHub repositories:
  - Isaac Sim Extension: <https://github.com/erdalpekel/IsaacMoveitExtension>
  - ROS2 simulation package: [https://github.com/erdalpekel/isaac\\_moveit\\_sample](https://github.com/erdalpekel/isaac_moveit_sample)
  - ROS2 simulation msgs package: [https://github.com/erdalpekel/isaac\\_moveit\\_msgs](https://github.com/erdalpekel/isaac_moveit_msgs)
  - ROS2 Panda description package: [https://github.com/erdalpekel/franka\\_description](https://github.com/erdalpekel/franka_description)

# Table of Contents

## 1 Introduction

Personal Info

Omniverse Isaac Sim

Presentation Scope

## 2 Main

System Requirements

User Interface

Robot Import

Isaac Sim Extension

Isaac - ROS connection

## 3 Demo

## 4 Conclusion



## Key Takeaways

- Isaac Sim is compatible with ROS2
- Existing robot models can be imported and reused
- Python API and extension templates reduce entry barrier

## Let's stay in touch!

- E-Mail: [info@erdalpekel.de](mailto:info@erdalpekel.de)
- GitHub: <https://github.com/erdalpekel>
- LinkedIn: <https://www.linkedin.com/in/erdalpekel/>