# USARSim/ROS: A Combined Framework for Robotic Control and Simulation

Stephen Balakirsky[1] Zeid Kootbally[2]

[1]Intelligent Systems Division
National Institute of Standards and Technology
Gaithersburg, Maryland 20899

[2]Department of Mechanical Engineering
University of Maryland
College Park, Maryland, 20742

ASME/ISCIE 2012 International Symposium on Flexible Automation, 2012

# Outline

# Advantages of Simulation

(simulation1-1.mov)

(simulation1-3.mov)

1. Experimentation in a safe and low-cost environment
   - Disable risks of harming personnel or equipment
   - Run particular test scenarios repeatedly
   - Provide “apples-to-apples" comparisons

# Advantages of Simulation

(simulation1-1.mov)

(simulation1-3.mov)

1. Experimentation in a safe and low-cost environment
   - Disable risks of harming personnel or equipment
   - Run particular test scenarios repeatedly
   - Provide “apples-to-apples" comparisons
2. Practical feedback when designing real world systems
   - Determine the correctness and efficiency of the design
   - Focus only on new algorithms development and deployment
   - Disable emphasis on the hardware aspects

# Advantages of Simulation

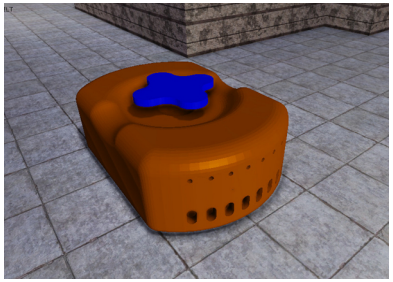
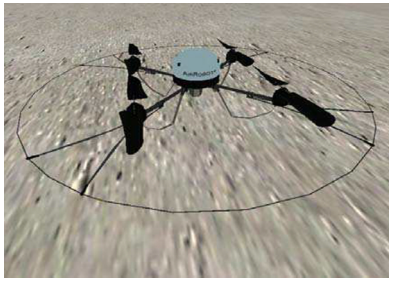
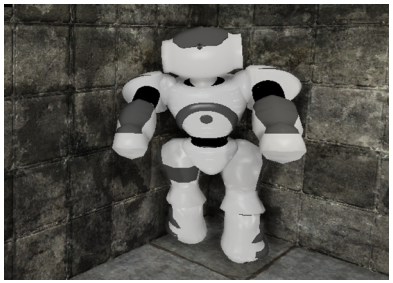
(simulation1-1.mov)

(simulation1-3.mov)

1. Experimentation in a safe and low-cost environment
   - Disable risks of harming personnel or equipment
   - Run particular test scenarios repeatedly
   - Provide “apples-to-apples" comparisons
2. Practical feedback when designing real world systems
   - Determine the correctness and efficiency of the design
   - Focus only on new algorithms development and deployment
   - Disable emphasis on the hardware aspects
3. Study a problem at several level of abstractions
   - Provide extensive testing opportunities
   - Simulate major components of the robotic architecture (e.g., advanced sensors)
   - Allow rapid prototyping

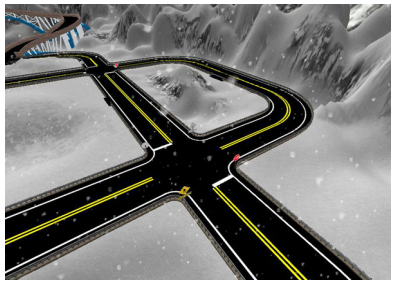
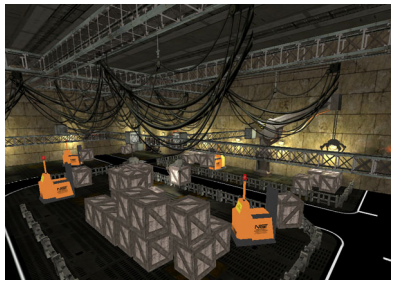
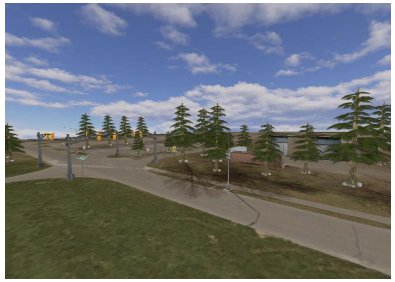
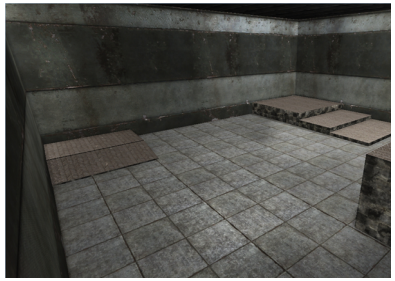
# Outline

## Platforms

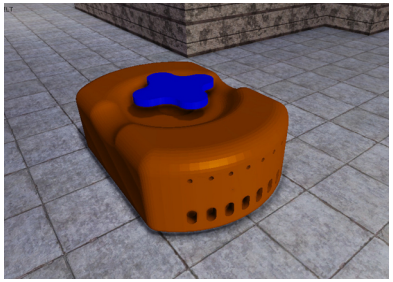
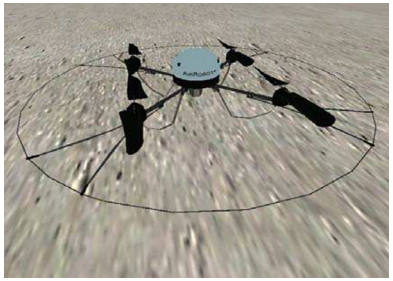
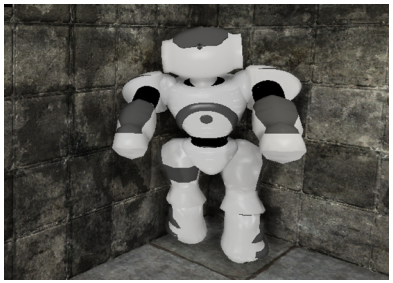





- High-fidelity physics-based simulation system based on the Unreal Development Kit (UDK)

## Environments

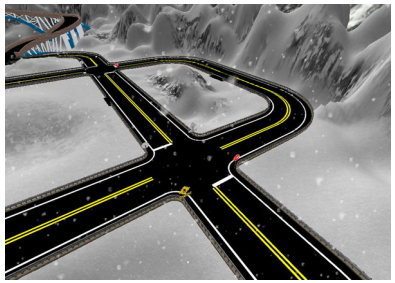
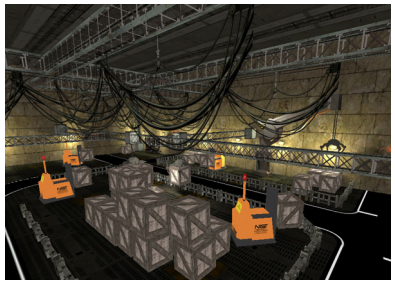
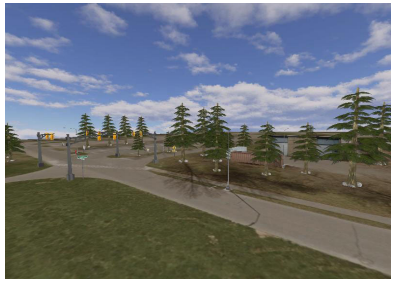
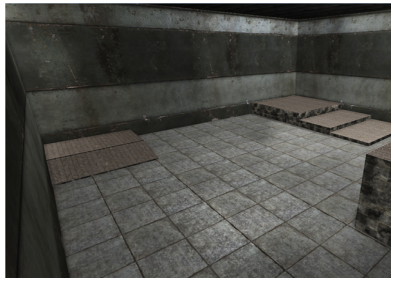





## Platforms

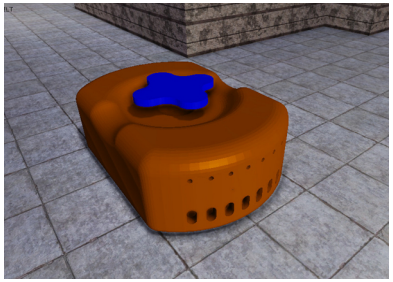
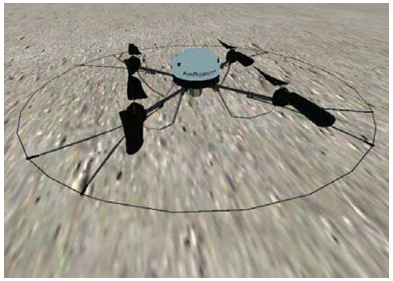
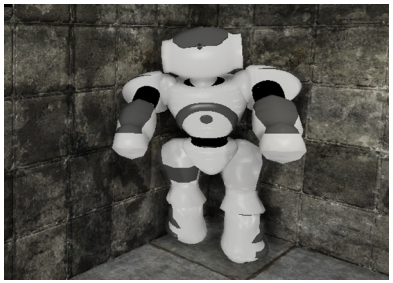





- High-fidelity physics-based simulation system based on the Unreal Development Kit (UDK)
- Through its usage of UDK, USARSim utilizes the physX physics engine and high-quality 3D rendering facilities

## Environments

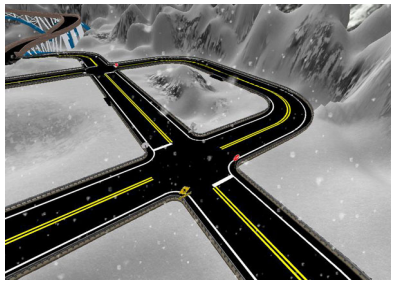
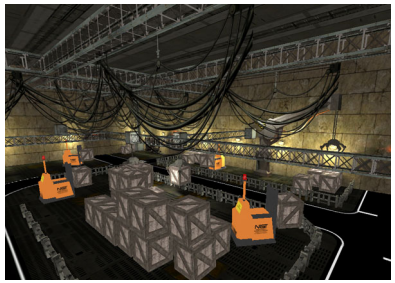
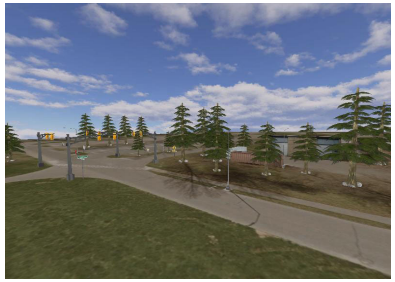
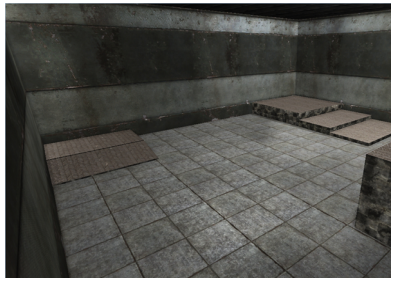





## Platforms

- High-fidelity physics-based simulation system based on the Unreal Development Kit (UDK)
- Through its usage of UDK, USARSim utilizes the physX physics engine and high-quality 3D rendering facilities
- Full user effort can be devoted to the robotics-specific tasks of:
  - modeling platforms
  - modeling environments
  - control systems
  - sensors
  - interface tools

## Environments

# Outline

# The Robot Operating System (ROS)

- Open source framework designed to provide an abstraction layer to complex robotic hardware and software configurations

# The Robot Operating System (ROS)

- Open source framework designed to provide an abstraction layer to complex robotic hardware and software configurations
- Libraries and tools to help software developers create robot applications

# The Robot Operating System (ROS)

- Open source framework designed to provide an abstraction layer to complex robotic hardware and software configurations
- Libraries and tools to help software developers create robot applications

## ROS Concepts

1. Stack: Packages in ROS are organized into ROS stacks which simplifies the process of code sharing
2. Package: A compilation of nodes that can easily be compiled and ported to other computers
3. Node: Executable file within a ROS package
4. Topic: A communication channel between two or more nodes
5. Message: A strictly typed data structure. A node sends a message by publishing it to a given topic

# USARSim & ROS Communication

## USARSim package and RosSim node

- Creation of the USARSim package

# USARSim & ROS Communication

## USARSim package and RosSim node

- Creation of the USARSim package
- The USARSim package contains the RosSim node

# USARSim & ROS Communication

## USARSim package and RosSim node

- Creation of the USARSim package
- The USARSim package contains the RosSim node
  - Relies on several parameters for its configuration

# USARSim & ROS Communication

## USARSim package and RosSim node

- Creation of the USARSim package
- The USARSim package contains the RosSim node
  - Relies on several parameters for its configuration
  - Publishes a ROS transform tree (from the ROS tf package) – Figure 1

## tf Transform tree

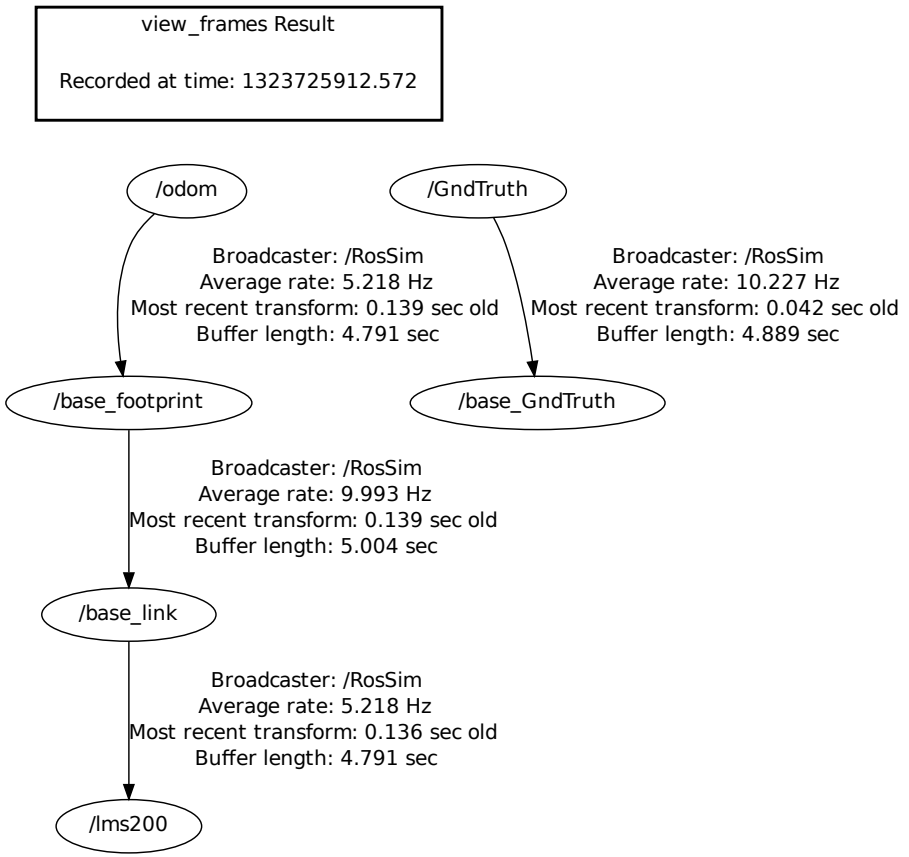



Figure 1: tf Transform tree for P3AT.

# USARSim & ROS Communication

## USARSim package and RosSim node

- Creation of the USARSim package
- The USARSim package contains the RosSim node
  - Relies on several parameters for its configuration
  - Publishes a ROS transform tree (from the ROS tf package) – Figure 1
  - Publishes sensor messages

## tf Transform tree

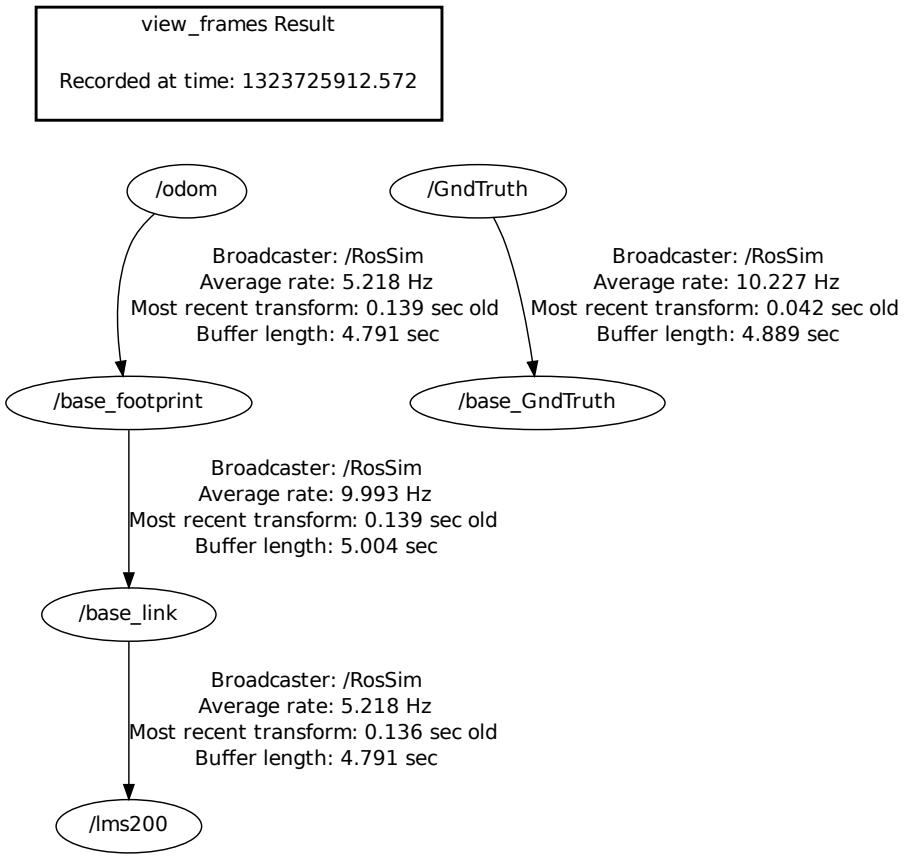



Figure 1: tf Transform tree for P3AT.

# USARSim & ROS Communication

## USARSim package and RosSim node

- Creation of the USARSim package
- The USARSim package contains the RosSim node
  - Relies on several parameters for its configuration
  - Publishes a ROS transform tree (from the ROS tf package) – Figure 1
  - Publishes sensor messages
  - Accepts platform and actuator motion commands

## tf Transform tree

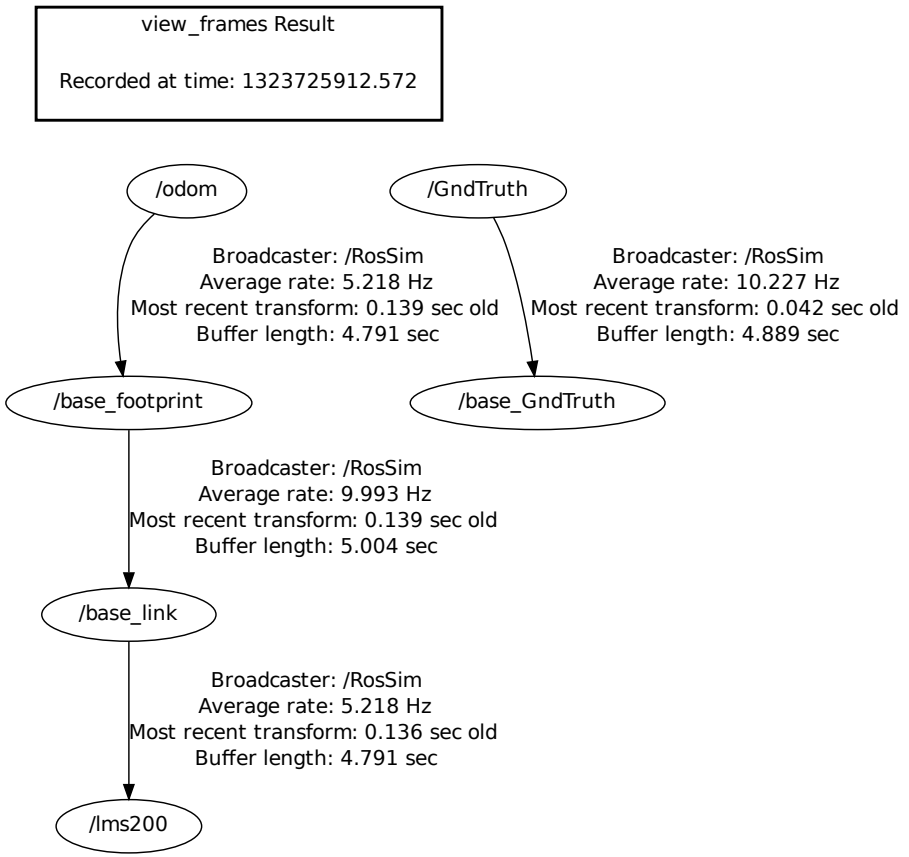



Figure 1: tf Transform tree for P3AT.

# USARSim & ROS Communication

## USARSim package and RosSim node

- Creation of the USARSim package
- The USARSim package contains the RosSim node
  - Relies on several parameters for its configuration
  - Publishes a ROS transform tree (from the ROS tf package) – Figure 1
  - Publishes sensor messages
  - Accepts platform and actuator motion commands
  - Provides a mechanism for
    - Spawning a robot in USARSim
    - Auto-discovering the robot's sensors, actuators, and drive configuration

## tf Transform tree

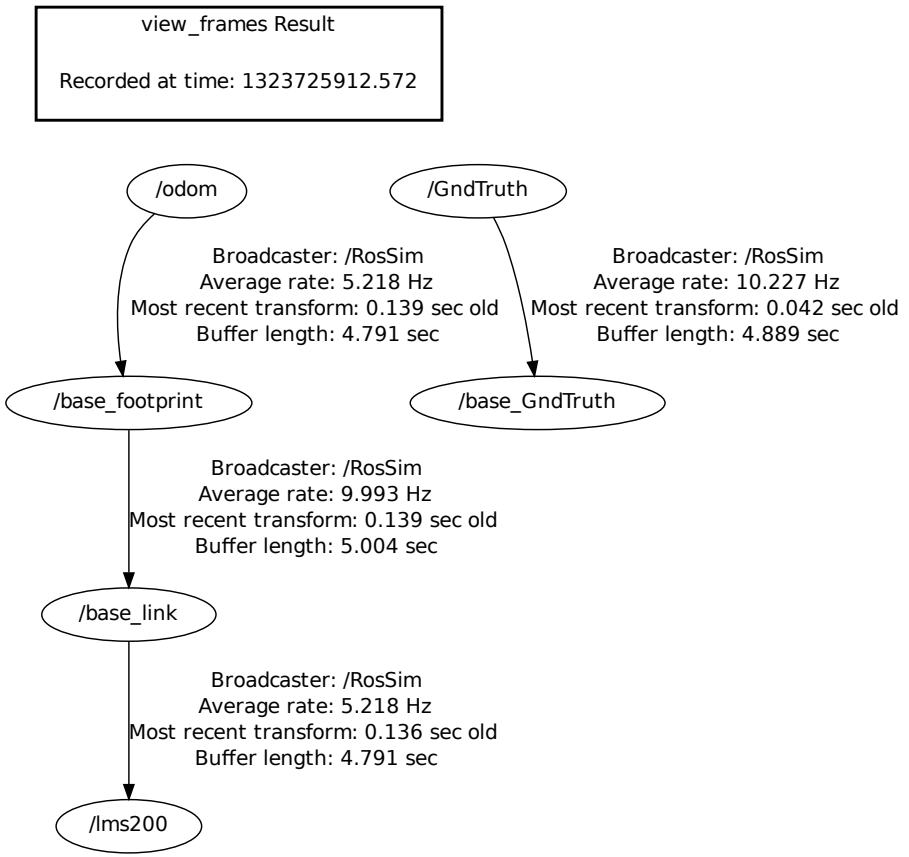



Figure 1: tf Transform tree for P3AT.

# Outline

# Sensor Interface

- RosSim node automatically matches simulated sensors to the appropriate ROS topic

# Sensor Interface

- RosSim node automatically matches simulated sensors to the appropriate ROS topic
- USARSim's sensors currently supported: Inertial navigation, Ground truth, LADAR

# Sensor Interface

- RosSim node automatically matches simulated sensors to the appropriate ROS topic
- USARSim's sensors currently supported: Inertial navigation, Ground truth, LADAR
- ROS/USARSim interface allows one to utilize known, published algorithms with simulated sensors and environments

# Sensor Interface

- RosSim node automatically matches simulated sensors to the appropriate ROS topic
- USARSim's sensors currently supported: Inertial navigation, Ground truth, LADAR
- ROS/USARSim interface allows one to utilize known, published algorithms with simulated sensors and environments
- USARSim object recognition sensor:
  - Accumulates the number of beam hits that occur on each detected object

# Sensor Interface

- RosSim node automatically matches simulated sensors to the appropriate ROS topic
- USARSim's sensors currently supported: Inertial navigation, Ground truth, LADAR
- ROS/USARSim interface allows one to utilize known, published algorithms with simulated sensors and environments
- USARSim object recognition sensor:
  - Accumulates the number of beam hits that occur on each detected object
  - Future implementation:
    - Use the number of hits and the percentage of the object that is visible to determine the amount of noise to add to the objects position and recognized type

# Sensor Interface

- RosSim node automatically matches simulated sensors to the appropriate ROS topic
- USARSim's sensors currently supported: Inertial navigation, Ground truth, LADAR
- ROS/USARSim interface allows one to utilize known, published algorithms with simulated sensors and environments
- USARSim object recognition sensor:
  - Accumulates the number of beam hits that occur on each detected object
  - Future implementation:
    - Use the number of hits and the percentage of the object that is visible to determine the amount of noise to add to the objects position and recognized type
    - Send information over standard ROS topics

# USARSim Object Recognition Sensor

(ObjectSensor0001.mov)

# Outline

# Robotic Arm Navigation

- Development of an interface that allows the use of the ROS arm_navigation stack
  - Automatic generation of URDF files that describe the kinematics of the arm
  - Joint control through ROS
  - Cartesian control through ROS

# Robotic Arm Navigation

- Development of an interface that allows the use of the ROS arm_navigation stack
  - Automatic generation of URDF files that describe the kinematics of the arm
  - Joint control through ROS
  - Cartesian control through ROS
- In USARSim, robotic arms are composed of individual static meshes attached to one another via links (actuators). Each actuator has its own coordinate frame (Figure 3)

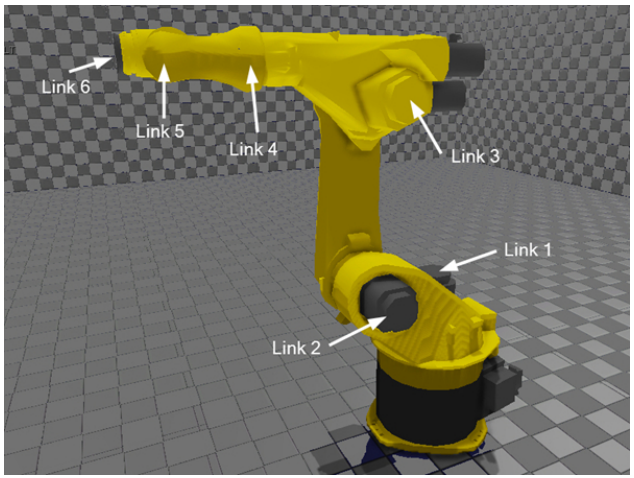



Figure 2: Joint axes in USARSim.

# Robotic Arm Navigation

- Development of an interface that allows the use of the ROS arm_navigation stack
  - Automatic generation of URDF files that describe the kinematics of the arm
  - Joint control through ROS
  - Cartesian control through ROS
- In USARSim, robotic arms are composed of individual static meshes attached to one another via links (actuators). Each actuator has its own coordinate frame (Figure 3)
- RosSim node automatically builds the transform tree (Figure 4) for the various actuator coordinate frames by reading the USARSim Conf and Geo messages

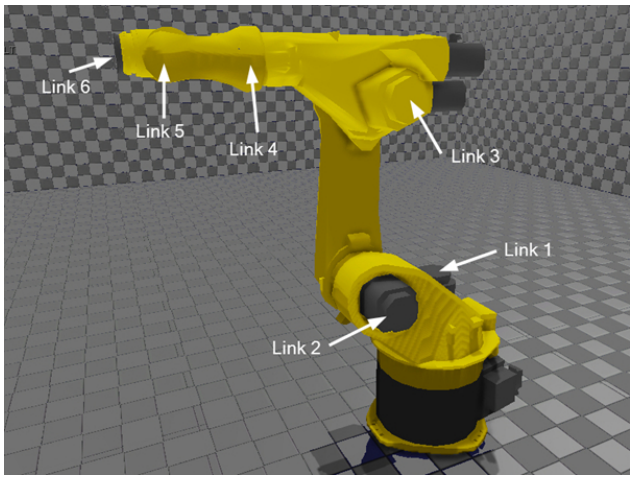



Figure 2: Joint axes in USARSim.

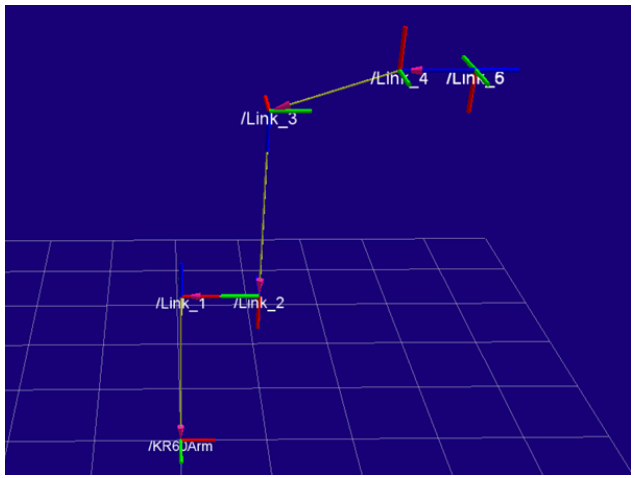



Figure 3: Joint axes from ROS tf topic.

# Robotic Arm in Action

## Tile Grasping

(vacuum.mov)

## Tool Changing

(toolchanger.mov)

# Conclusion and Future Work

- New ROS package (USARSim) and node (RosSim) that allow for the seamless interface of USARSim with ROS

# Conclusion and Future Work

- New ROS package (USARSim) and node (RosSim) that allow for the seamless interface of USARSim with ROS
- The package provides for auto-discovery of robots and sensors, and produces the standard ROS topics that one would expect from a physical platform

# Conclusion and Future Work

- New ROS package (USARSim) and node (RosSim) that allow for the seamless interface of USARSim with ROS
- The package provides for auto-discovery of robots and sensors, and produces the standard ROS topics that one would expect from a physical platform

- Future Work
  - Additional sensors interfaces in USARSim to be supported in the ROS environment
  - Source code for public domain (Sourceforge) (July 2012)
  - Implementation of tooltip and global coordinates for robotic arms (July 2012)

# For Further Reading I



Willow Garage
ROS Wiki
*http://www.ros.org/wiki*, 2012.


USARSim
USARSim Web
*http://www.usarsim.sourceforge.net*, 2012.


Epic Games
Unreal Development Kit
*http://udk.com*, 2012.

THANK YOU!