Advanced Inspection System: Integrating Robotic Configurations & Controls

Stephan J. Wlodarczyk
NASA Kennedy Space Center
Dual Major: Software Engineering & Information Technology
OSSI Spring Session
Date: 09 APRIL 2014

Advanced Inspection System: Integrating Robotic Configurations & Controls

Stephan J. Wlodarczyk*
Rochester Institute of Technology, Rochester, NY, 14623

The Advanced Inspection System (AIS) promotes the ability to perform field inspections and repairs remotely by operator command. The robotic system is required to be a fully autonomous operation, managed by computer execution of scripts with limited user control. AIS is a complex yet intriguing challenge in which various goals must be reached to achieve complete autonomy. The first steps to building such a system require human computer interactions and computer simulations for testing and verification. Such applications include the use of a wireless video game controller via Bluetooth Technology and the use of Leap Motion, a gesture based motion controller which can be used to manipulate robotic arm movements. Utilizing the Robotic Operating System (ROS) environment, these applications, in accordance to developing a 3D simulation of the system, will provide a foundational test bed for AIS development.

Abbreviations & Acronyms

AIS	=	Advanced Inspection System
BSD	=	Berkeley Software Distribution
COLLADA	=	Collaborative Design Activity
DAE	=	Digital Asset Exchange
DART	=	Dynamic Animation and Robotics Toolkit
DOF	=	Degrees of Freedom
ESC	=	Engineering Services Contract
GPS	=	Global Positioning System
GUI	=	Graphical User Interface
KSC	=	Kennedy Space Center
LIDAR	=	Light Detection and Ranging
NASA	=	National Aeronautics and Space Administration
OCT	=	Office of Chief Technologist
ODE	=	Open Dynamics Engine
ROS	=	Robot Operating System
SBC	=	Single Board Computer
SDF	=	Simulator Description Format
STEP	=	Standard for the Exchange of Product model data
STI	=	Space Technology Initiative
STL	=	Stereo Lithography
UGV	=	Unmanned Ground Vehicle
URDF	=	Unified Robot Description Format

I.Introduction

NASA's Space Technology Initiative (STI), managed by the Office of Chief Technologist (OCT) has established specific goals to develop technologies in support of present and future space exploration activities. The costs to design, develop, implement, test, operate, and maintain the systems must align with NASA's budget and not become a cost burden. To establish a future human presence in space and on planetary or astronomical objects requires systems that operate with minimal human intervention. The improved efficiencies and cost reductions cannot impact system safety and reliability.

The primary objective of the AIS is to develop hardware and software tools that improve inspection, recovery, and maintenance activities for ground and surface systems by using dynamic computer algorithms that analyze physical features of the real-world environment and make decisions and learn from their experience. These features

*NASA Software Engineering Intern, NE-E9, EDL, NASA Kennedy Space Center, FL can be analyzed with specialized sensors and/or visible or spectral band imaging cameras. In addition to utilizing dynamic computer algorithms, simple scripts can effectively and efficiently automate data collection, image processing, and data archiving thus improving inspection and maintenance activities.

The AIS architecture promotes technical diversity, system robustness, process efficiency, and personnel safety. This diversity creates a robust, flexible, and adaptable system capable of initiating, executing, and resolving multiple inspection and remediation scenarios throughout a system's lifecycle. The use of unmanned ground and/or aerial vehicles can improve routine or unplanned equipment inspection efficiency. Robotic manipulators installed on unmanned ground vehicles (UGVs) can improve personnel safety since many of these repairs of faulty equipment occur in hazardous environments. Standardized interfaces effectively and efficiently improve the transfer of information from one subsystem to another. These features form the building blocks of a successful, cost-effective, and sustainable architecture.

II.Background

This report aims to show a segment of the development process thus far in simulating the functionality and services of AIS. There are three critical components involved with producing the simulation. These elements will need to be accurately modeled in order to exhibit system activities. In addition, some models will need to be arranged such that their composed parts can be manipulated through a joystick controller. The demonstration will represent basic autonomous controls and operations of AIS.

A. Terabot-S Robot Manipulator

The Terabot-S robot manipulator arm, manufactured by Oceaneering, is a 5-degrees-of-freedom (DOF) arm system. The arm is mechanically and electrically integrated to the Husky UGV. The manipulator joints have integrated clutches for constant protection against overloads and are fully sealed, making them water and dust tight.

The manipulator uses a quick-release manual tool interface that can include a gripper end-effector. The motor controller provides the capability to control the position, speed, and electrical parameters for the Terabot S manipulator arm servomotors. Commands and status queries are sent and received using the RS-232 serial data interface.

The manipulator arm is also equipped with an AXIS Communications P1214-E miniature IP PoE 720p digital camera capable of H.264 digital video streaming for visual feedback. The camera is connected to and powered by the onboard UGV PoE network switch.[2]

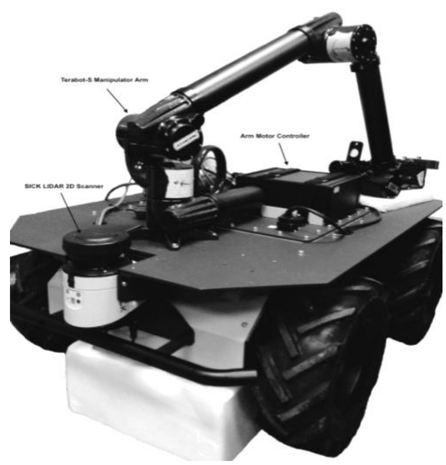

Figure 1. Husky UGV and Terabot-S
The Husky with the Terabot-S manipulator arm, arm motor controller, and LIDAR 2D scanner. The and can be deployed with great efficiency having 5 degrees of freedom, robotic gripper, and articulated joints

B. Husky UGV

The Husky UGV by Clearpath Robotics serves as mobile platform, transporting a multitude of systems to the desired destination of inspection. This compact all-terrain vehicle has a variety of hardware features that support remote inspection and repair activities. These include the Terabot-S manipulator arm as well as a single board computer (SBC), global position system (GPS), a wireless network transceiver, a Power over Ethernet (PoE) network switch, PoE cameras, and other sensors including a Light Detection and Ranging (LIDAR) 2D scanner.[1]

C. Cryogenics Test Laboratory

Located at NASA Kennedy Space Center (KSC), the Cryogenics Test Laboratory is a state of the art research and development facility. The laboratory provides industry and government needs for effectively handling and storing liquid cryogenic propellants both from ground and space perspectives. Thus, the facility area provides cryogen competence, on-site experimental testing, prototyping, problem diagnostics, investigation, and resolution, and a plethora of other system applications.

The Cryogenics Test Bed Facility will be the foundational benchmark for AIS and its future role in supporting hazardous fueling and cryogen unloading operations. The site will be replicated in its entirety through 3D models and be embedded into robotic simulations to prove the usefulness, reliability, and effectiveness of the entire AIS project.[5]

III.Development Setup

The preceding systems will be modeled and visualized utilizing the robot simulation tool, Gazebo, and controlled through ROS. The development environment will require the implementation of a virtual machine running Ubuntu 13.04 (raring) which is a fully supported and tested operating system.

D. Robot Operating System

A flexible framework for writing robot software, ROS is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust behavior across a wide variety of robotic platforms. ROS was built from the ground up to encourage collaborative robotic software development allowing different groups of people to develop and release source code that was specific to a subsystem (i.e. mapping, communications, controls, etc.). ROS has a Berkeley Software Distribution (BSD) license and thus is completely free to use, change, and commercialize.[3]

E. Gazebo

Robot simulation is an essential tool for testing AIS capabilities. Gazebo offers the ability to accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. It provides realistic rendering of environments including high-quality lighting, shadows, and textures. Gazebo also yields a wide array of dynamics and physics engines including Open Dynamics Engine (ODE), Bullet, Simbody, and Dynamic Animation and Robotics Toolkit (DART).

Gazebo also has the capability of generating sensory data with features to incorporate noise and interference from laser range finders, 2D/3D cameras, Kinect motion sensors, and contact sensors. The powerful simulation tool also grants abilities to create custom scripts or plug-ins that replicate control schemes for a specific robot, sensor system, and even environmental conditions such as gravity, atmosphere, and terrain.[4]

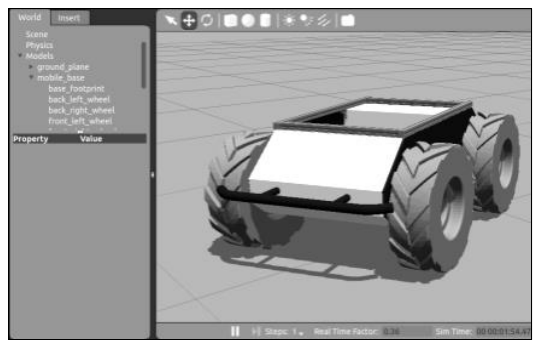

Figure 2. Gazebo Interface
The Gazebo graphical user interface (GUI) with the Husky UGV model imported

F. Virtual Machine

Developing on a virtualization disc image aids in testing software, upgrades, and new configurations. It offers the ability to port disc images to other workstations, back-up critical work and milestones, and grants ROS developers a sandbox to customize to their development strategies and processes.

IV.3D Modeling Configurations

The Terabot-S arm, Husky UGV, and the Cryogenics Test Laboratory are illustrated within Gazebo to provide simulated demos of AIS capabilities and showcase basic scenarios depicting AIS operations. Oceaneering has assembled a model of the Husky UGV and then converted to a file format in which Gazebo can interpret. However, the robot manipulator arm and test laboratory were assembled by the Engineering Services Contract (ESC) and NASA and need to go through the conversion process. The end result of this operation enables developers to easy manipulate individual robotic joints from within Gazebo.

G. Assembly Conversion

All three models were developed using Pro/Engineer, a computer-aided design (CAD) software application, and saved into a specific set of formats, but none of which are relevant to use inside of Gazebo. Thus, in order to proceed, each part of the assembly in the CAD file must be exported as a stereo lithography (STL). These individual parts will be reassembled using Gazebo's simulator description format (SDF).

It is important to note that when bringing these STL model files into Gazebo, they retain the same coordinate positions and pivots points as originally defined within the CAD assembly. These may need to be reconfigured as the parts will be offset and angled incorrectly within the simulated environment.

Thus, another segment of the conversion process entails developers to manually reconfigure the axes alignment and orientation of the specified piece.

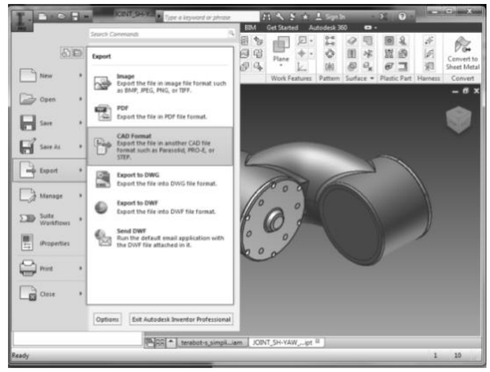

Figure 3. CAD Conversion
The Terabot-S rotating shoulder cuff being exported from Autodesk Inventor to STL file format

H. Modifying Pivot Points

To correct alignment issues, the STL file can be brought into various modeling software programs like 3D Studio Max or Maya and be transposed accordingly. In most cases, the pivot point, the point at which the part's axis is rotated about, needs to be adjusted.

Likewise, the part may need to be rotated or translated within the origin grid of the 3D application environment. Because the part adopts its coordinate position and angle vector from the original CAD file, the coordinates should be set back to 0,0,0 and angles 0,0,1†, (radians) making reassembly a much more efficient and effortless process in Gazebo.

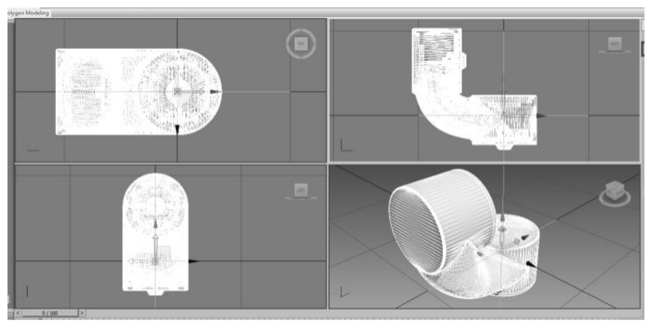

Figure 4. Pivot Point Manipulation in 3D Studio Max
The Terabot-S rotating shoulder cuff's pivot point being relocated to the center origin. The object is also translated and positioned to 0,0,0 with an alignment in the positive z-direction

I. Rigging with SDF

An SDF file compiles all the model components into one dynamic object with instruction sets detailing the movement behavior of each part. Its syntax follows XML standards and the syntax describes all the elements in a Gazebo simulation environment. The Terabot-S manipulator arm has a total of eight parts, seven which can be moved, and one which remains static.

The development of an SDF file can be referred to as a process, known in 3D animation, as rigging. The SDF file is mapping a skeletal structure to the model. The skeletal joints are thus accessible by their defined name within the SDF from applicable programming languages, in the case of ROS, either Python or C++. This link will allow input from a joystick or motion controller to directly change the position of a defined component within the simulation.

```
<!-- Terabot joint 1: PITCH - elbow joint that can translate along the
X/Z-axis plane; joint will thus translate all children connected to it. -->
<link name='terabot_joint1'>
<pose>0.2750 0.0045 0.4450 -1.5700 3.1400 -1.5700</pose>

<inertial>
  <mass>1.000</mass>
  <pose>0.2750 0.0045 0.4450 -1.5700 3.1400 -1.5700</pose>
  <inertia>
    <ixx>0.083</ixx>
    <ixy>0</ixy>
    <ixz>0</ixz>
    <iyy>0.083</iyy>
    <iyz>0</iyz>
    <izz>0.083</izz>
  </inertia>
</inertial>

<!-- Joint 1 will collide with other components of the terabot arm and husky -->
<self_collide>true</self_collide>

<visual name='terabot_joint1_visual'>
  <geometry>
    <mesh>
      <uri>model://terabot/meshes/terabot_joint1_link.DAE</uri>
      <scale> 1 1 1 </scale>
    </mesh>
  </geometry>
  <material>
    <script>
      <name>Gazebo/Red</name>
    </script>
  </material>
</visual>
</link>
```

Figure 5. SDF File Code Snippet
The excerpt details the characteristics of a specific part of the Terabot-S and it includes the weight, position in 3D space, inertial components, type of joint, and the 3D visual model to represent it in Gazebo.

† The angle of orientation of each part should share a common angle with positive direction. This is to condone consistency between all parts allowing to easily rig the joints when writing a SDF file.

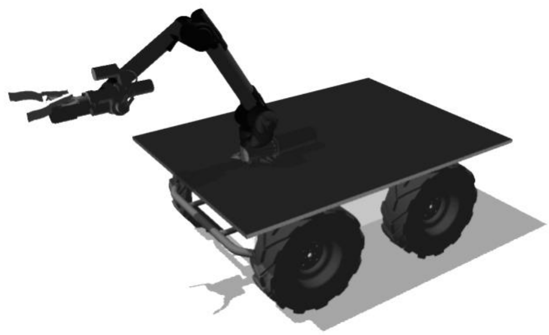

Figure 6. Husky UGV and Terabot-S Gazebo Model
Snapshot of the Husky UGV model with the mated Terabot-S model rendered in the Gazebo simulated environment using SDF

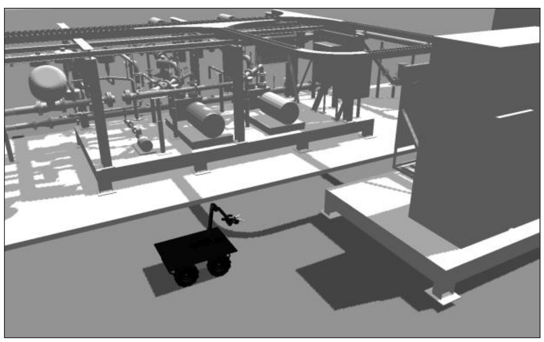

Figure 7. AIS and Cryogenics Test Laboratory
Full simulated mockup of the Cryogenics Test Laboratory at KSC with AIS in Gazebo

V.Joystick Controls

Testing of the robotic joints for the Terabot-S were performed utilizing C++. A PlayStation 3 six-axis motion controller was integrated with the Gazebo environment and is able to simulate teleoperations to both the Husky UGV and Terabot-S models.

The source code and Gazebo mimic the actual messages being sent to and received by the hardware of these two robotic components. Thus, if demonstrations of AIS operations can be performed successfully in simulation, the code can be ported to the AIS with minimal development effort.

VI.Conclusion

The current status for simulation development resides within mapping the joystick to remaining joints of the Terabot-S. The rotating shoulder cuff has been developed, tested, and successfully demonstrated in Gazebo. In addition, all models have been converted and imported into the simulation environment.

ROS and Gazebo provide a foundational test-bed for meeting the requirements and further enhancing AIS capabilities. The simulator is a controlled environment and can be modified to mock various types of conditions, risks, and failure modes that may be probable during an AIS operation. Being able to test, modify, and directly port from simulation to physical entities (i.e. Husky UGV and Terabot-S) yields a new cost effective and reliable development strategy.

Acknowledgments

Stephan Wlodarczyk would like to thank NASA and the internship program for an incredible experience at the Kennedy Space Center. He promotes a special thank you to his mentor, Mark Lewis for his remarkable guidance and superlative knowledge. He also thanks Walter Wehner for his apprenticeship and for being an exceptional co-worker. Lastly, Stephan Wlodarczyk would like to thank the Ground Systems Development and Operations Program and the Advanced Ground Systems Maintenance Element Integration Team for allowing him to live out his dream of working for this extraordinary agency.

References

[1] Oceaneering International, Inc., "Terabot-S", Houston, Texas.
[http://www.oceaneering.com. Accessed 4/1/2014]

[2] Clearpath Robotics, "Husky", Kitchener, Ontario.
[http:// www.clearpathrobotics.com. Accessed 4/1/2014]

[3] Willow Garage, "Robot Operating System", Menlo Park, California.
[http:// www.ros.org. Accessed 4/1/2014]

[4] Open Source Robotics Foundation, "Gazebo", Mountain View, California.
[http:// www.gazebosim.org. Accessed 4/1/2014]

[5] Cryogenic Society of America, "NASA KSC Cryogenics Test Laboratory", Oak Park, Illinois.
[httphttp://www.cryogenicsociety.org. Accessed 4/1/2014]