Autonomous Robotics Research for Ocean Worlds (AISR:ARROW):

Autonomy Application Programming Interface Document

Version 0.2, November 22, 2019

# Background

This Autonomy Interface Document describes and defines the application programming interface (API) to be made available to the candidate autonomy technologies to exercise the functions of the virtual and physical Ocean Worlds lander testbeds. This version of the document lists the current design of the interface. We expect the interface to evolve and be updated in response to the requests of potential users. This document will be updated as the changes are made.

# Operating Environment

## Virtual Testbed

Ubuntu 18.04

ROS Melodic

NVIDIA graphics adapter

## Physical Testbed

Ubuntu 18.04

Two nVidia GeForce RTX 2080 Ti, 11GB available for GPU computation if needed

ROS Melodic

# Command Protocol

The interface will be implemented using ROS messages and actions. Each Command will be available as an action and telemetry is provided as messages.

# Command Dictionary

The following commands will be available. These will be provided as ROS message and action definitions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Module** | **Type** | **Command** | **Description** |
| Arm | Command | CartesianGuardedMove | Guarded end effector move to xyz in the Lander Body Frame frame |
| Arm | Command | JointMove | Move in joint space until move is completed |
| Arm | Command | CartesianMove | Move to specified position in reference frame |
| Arm | Command | ChooseEndEffector | Choose an end effector |
| Arm | Command | Unstow | Unstow arm |
| Arm | Command | Stow | Stow arm |
| Arm | Command | Stop | Stop all motion |
| Arm | Command | BuildCollisionObjects | Set-up/update collision objects for checking collisions |
| Arm | Command | ClearCollisionObjects | Clear collision objects for checking collisions |
| Arm | Command | Initialize | Initialize arm |
| Arm | Command | CheckCollision | Verify the arm motion is collision-free |
| Arm | Telemetry | JointPosition | Joint positions |
| Arm | Telemetry | JointVelocity | Joint velocities |
| Arm | Telemetry | Joint Acceleration | Joint accelerations |
| Arm | Telemetry | EndEffectorPose | End effector pose in arm space |
| Arm | Command | CartesianMoveGuarded | Move to specified position in reference frame until move is completed or until contact |
| Arm | Command | JointMoveGuarded | Move in joint space until move is completed or until contact |
| Arm | Command | JointMoveTime | Move in joint space until move is completed in the specified time |
| Arm | Command | JointMoveOne | Move the specified joint by the specified amount |
| Arm | Command | CartesianMove2TargetGuarded | Move to specified offset from target position in reference frame |
| Arm | Command | JointMove2TargetGuarded | Move to specified offset from target joint pose |
| Arm | Command | SetCollisionIgnore | Sets or unsets a single bit in an IDA collision objects ignore collisions field, allowing a collision to be ignored. |
| Arm | Command | SetCollisionArmIgnore | Enables or disables collision checking between two IDA links bymodifying the link collision matrix. |
| Arm | Command | SetCollisionObject | Sets the parameters of a single IDA collision database objectin the global list of objects. |
| Arm | Command | SetTool | Sets the tool to use for arm operations |
|  |  |  |  |
| Camera | Command | Capture | Take an image |
| Camera | Command | CapturePoint | Point camera at specified location |
| Camera | Command | ExposureType | Exposure type (exposure time specified or automatically determined) |
| Camera | Command | ExposureDuration | Duration of image exposure (not valid for auto-exposure) |
| Camera | Telemetry | DepthFrame | Camera depth frame data |
| Camera | Telemetry | ColorFrame | Camera color frame data |
| Camera | Telemetry | PointCloud | Camera pointlcoud data |
|  |  |  |  |
| CamPanTilt | Telemetry | PanTiltPosition | Pan/Tilt joint angles |
| CamPanTilt | Command | PanTiltJointMove | specify the pan and tilt target angles |
| FltProtectn | Command | ClearAllFaults | Clear all faults |
| FltProtectn | Command | ClearFault | Clear specified fault |
| FltProtectn | Command | SetFault | Set fault for the value passed |
| FltProtectn | Telemetry | Status | System status enum |
| Motion | Command | ScoopOp | Perform arm operations to scoop |
| Motion | Command | DrillOp | Perform arm operations to drill |
| Motion | Command | ConePenetrometerOp | Perform cone penetrometer test  |
| Motion | Command | PrSinkageOp | Preform pressure sinkage test  |
| Motion | Command | BevameterOp | Perform bevameter test |

# Sensor Data - values will be in SI units

System status int32 – enum of system state

Arm joint angle joint\_angle float64[7]

Arm instrument instrument\_id int32

Arm force/torque sensor fts\_data float64[6] – 3D torque vector and a 3D force vector

Instrument type int32 – enum of instrument currently on arm

Instrument pose instrument\_pose float64[7] representing a quaternion and a 3D vector

Perception camera\_image ROS image rgb8 encoding

Perception camera\_image ROS image mono8 encoding

Perception camera\_image ROS image mono16 encoding

Perception point\_cloud ROS pointcloud with rgb and distance channels

Perception depth\_map ROS image 32FC1 encoding

Lander pose float64[7] representing a quaternion and a 3D vector

Instrument-environment surface contact instr\_contact bool

# Fault Conditions

System command timeout

System power failure

System misc\_other failure

Arm joint failure

Arm reach limit

Arm motion failure

Arm collision

Arm misc\_other failure

Instrument pressure sinkage failure

Instrument shear bevameter failure

Instrument cone penetrometer

Instrument scoop failure

Instrument drill failure

Instrument misc\_other failure

Lander misc failure

Perception pointing failure

Perception image capture failure

Perception image processing failure

Perception misc\_other failure