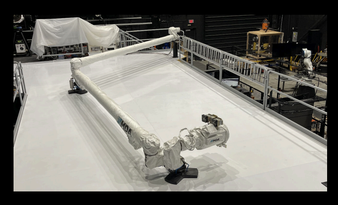
Assembling Large Infrastructures in Space using Intelligent Robots Workshop at the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems

WEST VIRGINIA
ROBOTIC TECHNOLOGY CENTER



Hardware-in-the-loop simulation of modular antenna assembly

A. Gailey, G. Ingram, G. Marani, K. McBride, A. Velasquez

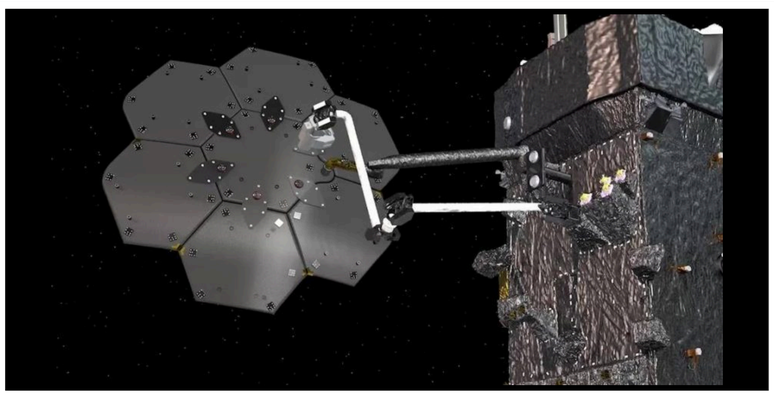
Abstract

Robots are playing an increasing role in space exploration and in-space servicing. Robotic arms are good for performing in-space tasks such as modular assembly. The SPIDER arm (SPace Infrastructure Dexterous Robot) of the OSAM-1 mission represents an example of a system that can perform modular antenna assembly tasks in environmental conditions that would be dangerous for astronauts. One of the WVRTC activities is to perform independent verification and validation of the SPIDER assembly operations.

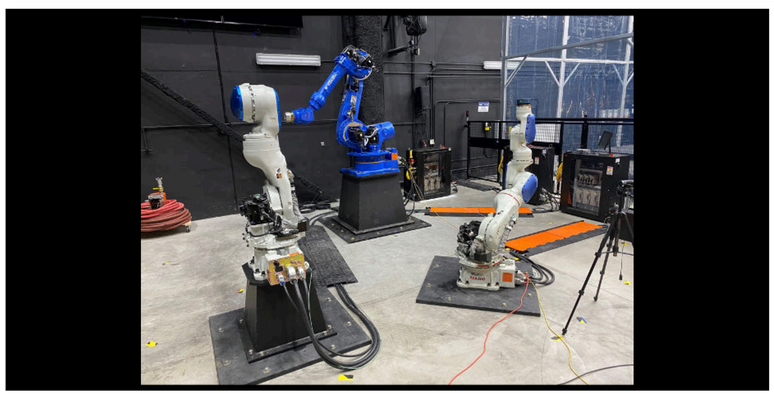
Space robotic systems adopting large manipulators such as the SPIDER arm are complex to test and verify under normal 1-G conditions. Our work involves proof-of-principle testing of a hardware-in-the-loop (HIL) simulator operating in 1-g conditions, which replicates the motion of the space robot end-effector under the same geometric, kinematic and dynamic conditions, and in response to externally applied forces from its space environment.

In particular, our software simulates in real-time the robot dynamics inclusive of flexibility, an important aspect in case of a long robotic arms operating in 0-G conditions. For example, the same oscillatory motions of the space manipulator's end-effector can be mimicked on ground through our industrial manipulator, allowing us to test the feasibility of complex operation such as assembling modular components.

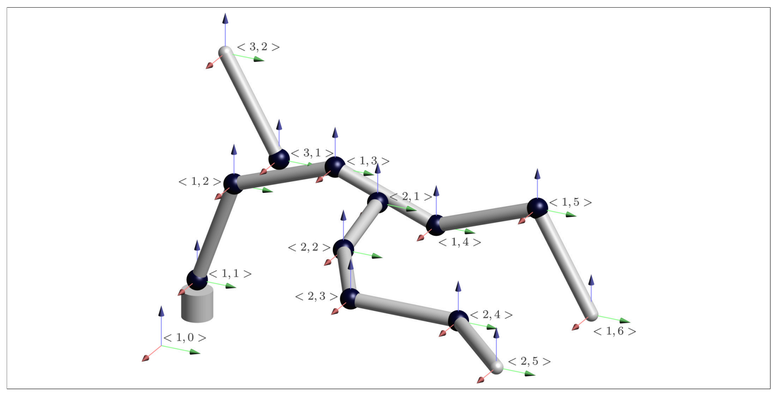
SPIDER Antenna Assembly



Testing Facility



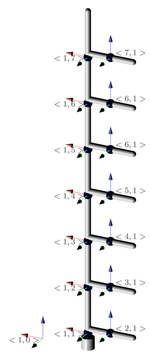
Ramified Kinematic Chain



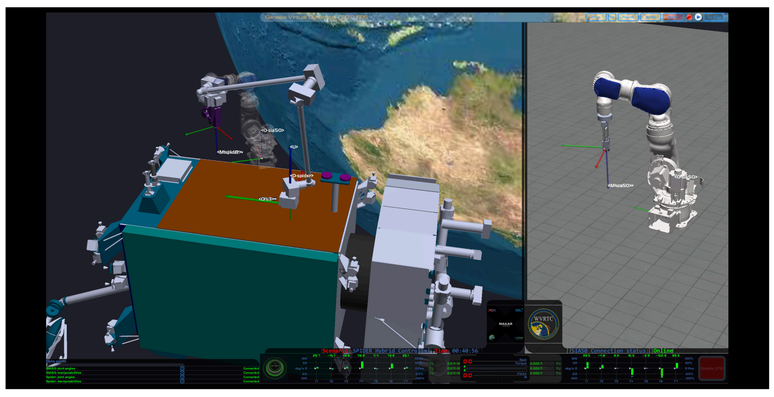
Real-Time Dynamic Simulator

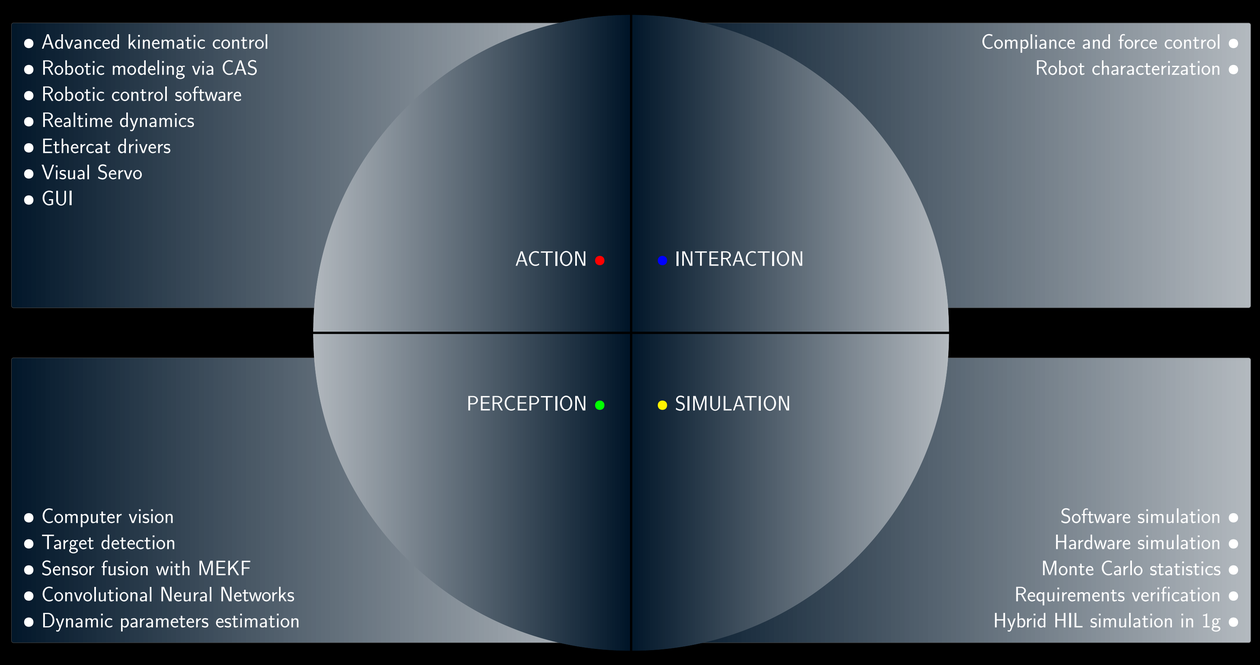
Flexibility with augmented model (+ 1 DoF per joint)

- Elasticity is reduced at the joint level, along the torque transmission path between the motor and the joint
- Dynamic model is augmented by adding extra branches
  - This is equivalent to attaching each motor to the original joint through a spring/mass/damper
  - Total degrees of freedom doubles
- The "spring" acts only on the axis of the motor; every other part of the structure is supposed rigid
- This model assumes the dominant action is the elasticity of the torque transmission between the motor and the joint
- This model requires only **50μs** to perform one update step on a 7-DoF robot

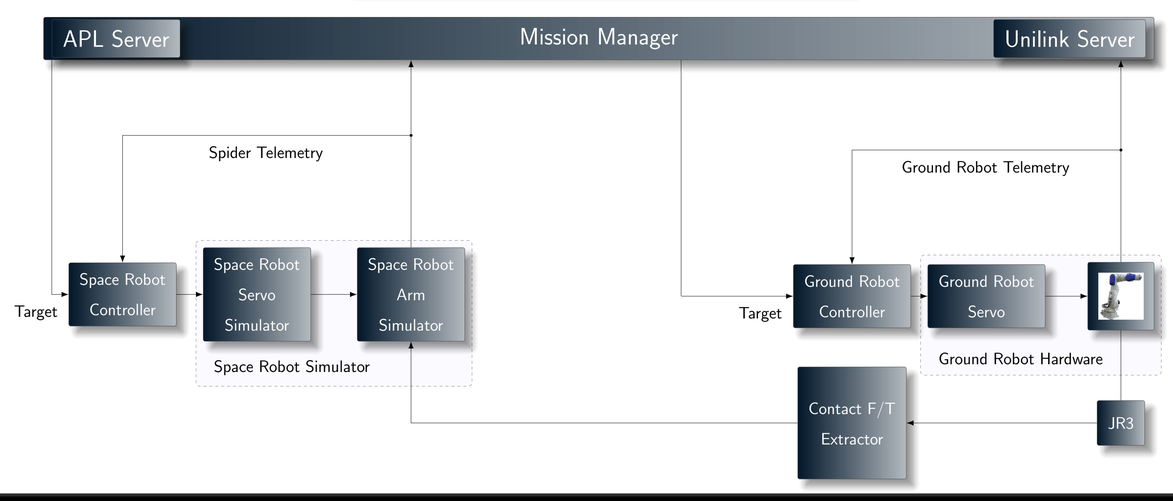


Hybrid Simulation User Interface





Robot Control System



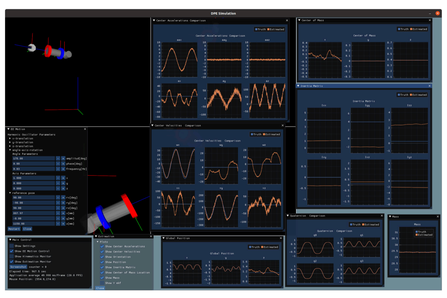
HIL Simulation: Dynamic Parameters Estimation

Dynamic Parameters Estimator:

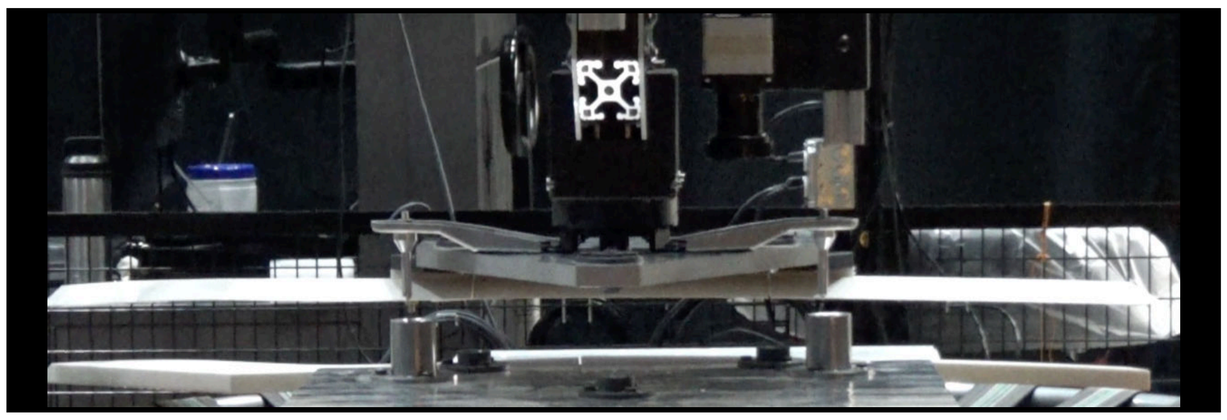
- Identifies dynamic parameters of a robot end-effector
- Estimates velocities and accelerations of the end effector center of mass using an Extended Kalman Filter (EKF)
- Estimates the externally applied forces and moments acting on the end effector using the EKF
- Primary use: **contact forces isolation in a 1-G environment** for the HIL simulator
- A graphical user interface (to the right) shows real-time plots of the estimates for velocities, accelerations, and externally applied forces and torques

How it works

- Inputs: 12-DoF sensor data (linear/angular accelerations, forces and moments)
- Inputs: end effector pose w.r.t the robot base, which is provided by the robot controller
- A mathematical model of the end effector dynamics developed using Maple CAS provides a solution for the F/T sensor inertial forces
- Simulated inertial forces are subtracted from the F/T readings to isolate the contact forces



Assembly Task



HIL Simulation: Real-Time Dynamic Simulator

Dynamic model:

- Lagrange equation for quasi-coordinates[a]:

$$A(q)\,\dot{p} + B(q,p)\,p + V^T C(q) = V^T \mu^e$$

where:

- q is the configuration vector
- p is the quasivelocity vector
- The matrix A is known as the *inertia matrix* of the structure.
- $B(q,p)$ is the Coriolis-centrifugal matrix
- The quantity $V^T C(q)$ represents the projection in the joint velocity space of the gravitational forces
- $V^T \mu^e$ is the projection in the joint velocity space of all the other external forces (for example the actuator forces)

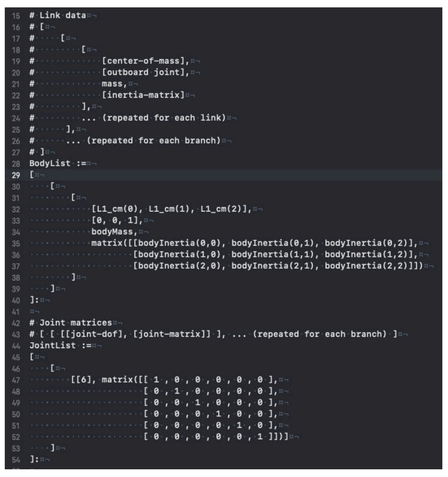
Numeric solver:

- Custom C++ implementation of Runge-Kutta 4

[a] Marani, Yuh: Introduction to Autonomous Manipulation, Springer, 2014, Sec. 2.

HIL Simulation: Code Generation

Main challenge: **real-time dynamics → Automatic Code Generation via Maple**



Several thousands lines of code!!!
The strong iterative automatic optimization makes it possible to run the dynamics at more than **30kHz**