SPART: AN OPEN-SOURCE SIMULATOR FOR SPACECRAFT ROBOTIC ARM DYNAMIC MODELING AND CONTROL

Dr. Josep Virgili Llop, NPS-Mechanical and Aerospace Engineering
CPT. Jerry V. Drew II, NPS-Mechanical and Aerospace Engineering
Prof. Marcello Romano, NPS-Mechanical and Aerospace Engineering

Outline

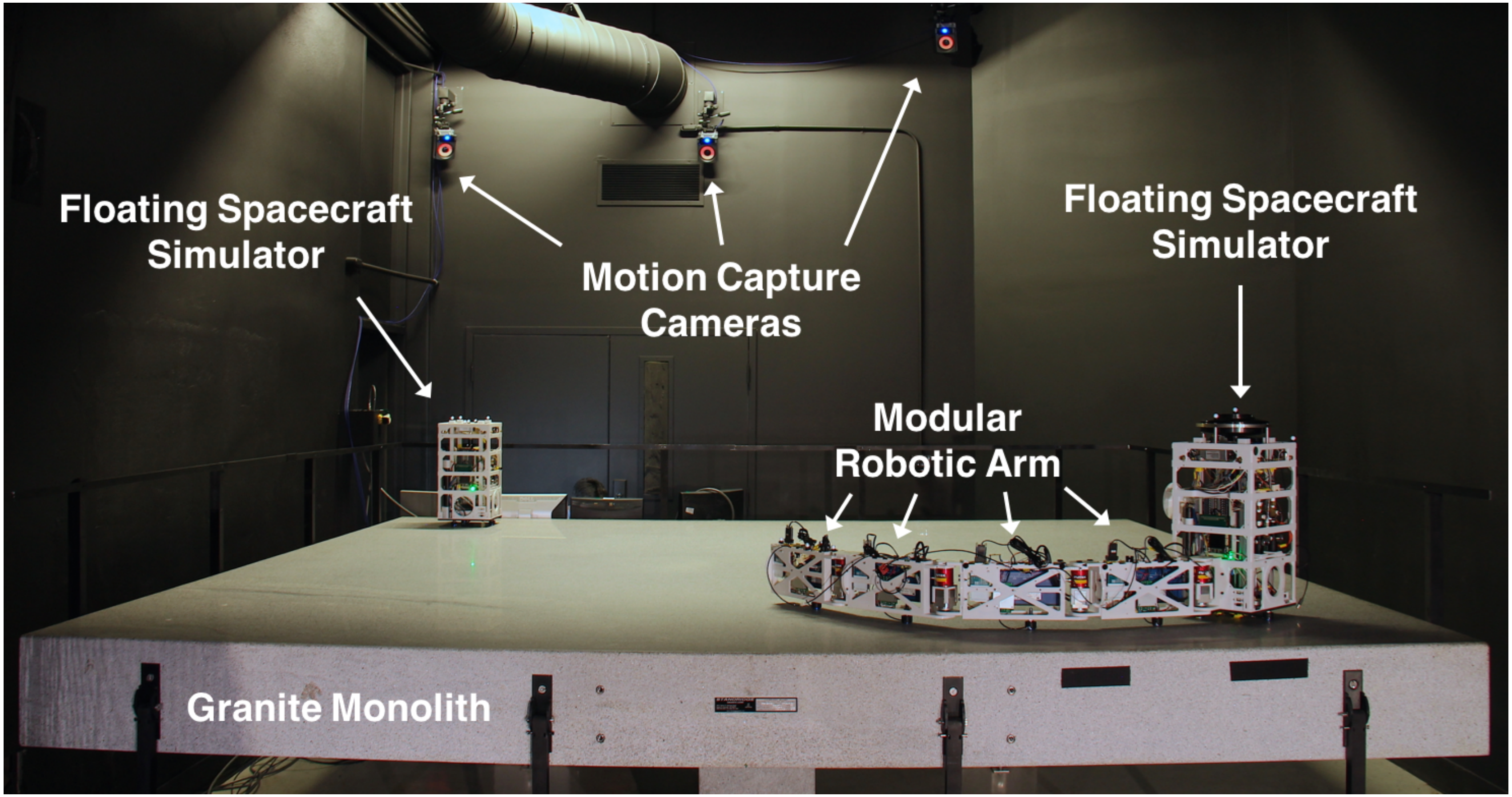
- Motivation
- Other existing tools
- DeNOC kinematic and dynamic modelling approach
- Application examples

Motivation

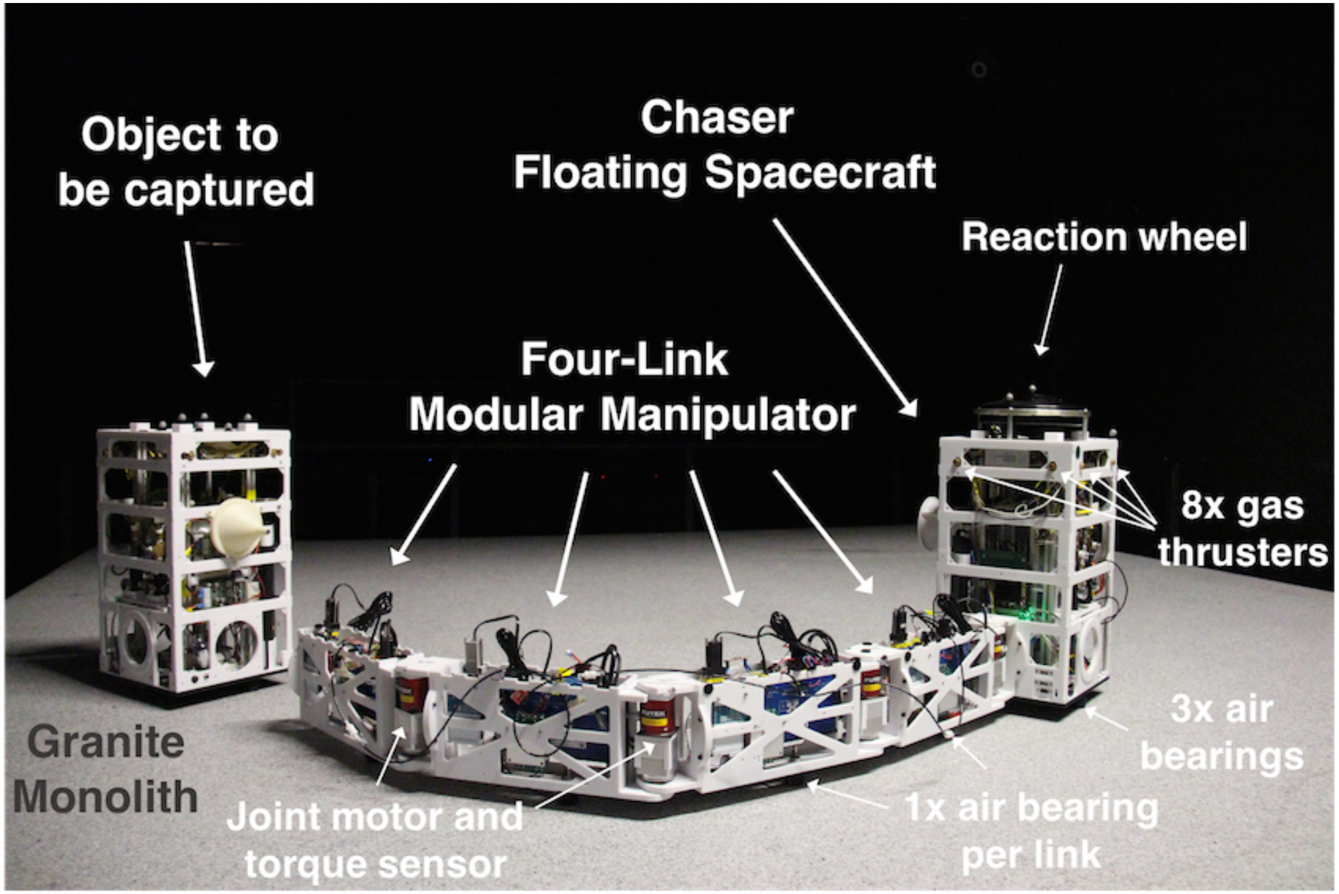
Requirements:

- For teaching and research alike.
- Need to be able to integrate with the existing toolchain for the experimental test bed (Simulink RTW with an RTAI Linux target).
- Students are mainly familiar with Matlab/Simulink.
- Access into the kinematic and dynamic properties is required.
- Collection of function as building blocks for rapid prototyping of guidance and control algorithms (flexible and modular).

Naval Postgraduate School - Floating Spacecraft Simulator



Naval Postgraduate School - Floating Spacecraft Simulator



Other open source tools

Other existing tools:

- ODE
- Bullet
- DART
- Simboby

General physics engines (focused on efficiency), written in C/C++, generally without access to kinematics and dynamic properties and no repository of control algorithms.

Other frameworks

- ROS
- GAZEBO
- MoveIt!

Could provide complement the SPART capabilities (i.e. visualization, navigation and networking) and thus SPART will

Rationale

SPART features:

- Toolkit – Not a generic physics engine
- Based on MATLAB/Simulink for rapid algorithm prototyping and easy integration with NPS-FSS development toolchain.
- Repository of control algorithms.
- Repository of analysis tools.
- Automatic Code Generation is used to generate efficient C/C++ that can run on embedded hardware.

Kinematics

q ← Joint space variables

- q_0 base-spacecraft state
- q_m manipulator joint states

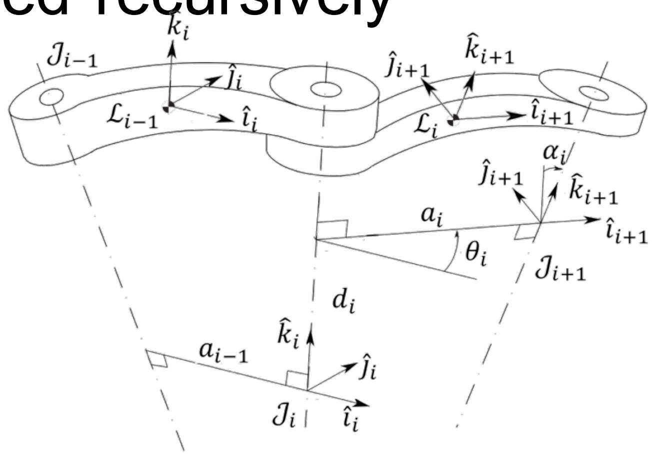
Kinematic transformations map joint space variables into operational space

x= $k(q)$

Coordinate transformations can be obtained recursively

$$ {}^{\mathcal{I}}T_{\mathcal{J}_i} = {}^{\mathcal{I}}T_{\mathcal{L}_0}\,{}^{\mathcal{L}_0}T_{\mathcal{J}_{0+1}} \prod_{j=2}^{i} {}^{\mathcal{J}_{j-1}}T_{\mathcal{J}_j} = {}^{\mathcal{I}}T_{\mathcal{J}_{i-1}}\,{}^{\mathcal{J}_{i-1}}T_{\mathcal{J}_i} $$

System geometry is currently uses a custom data structure. Compatibility with commonly used URDF, SDF, and VSK file formats is on the works.



Denavit-Hartenberg parameters are used to define the relationship between connected joints

Differential Kinematics

Jacobians (natural orthogonal complement) matrices which map joint space velocities into operational space are also provided.

Operational space velocities

$$t_i = J_{0i}\dot{q}_0 + J_{mi}\dot{q}_m$$

Base-spacecraft Jacobian

Manipulator Jacobian

General Equations of motion

Generalized Convective Inertia Matrix (Coriolis and Centrifugal)

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau$$

Generalized Inertia Matrix

Joint space forces

- Recursive $\mathcal{O}(n)$ formulations to obtain the H and C matrices are provided.
- The user has full access to the kinematic and dynamic magnitudes

Floating-Base dynamics

- When the base is left to react to the manipulator motion the base-spceraft motion can be dicounted from the dynamics model

$$H^{\star}\ddot{q}_m + c^{\star} = \tau_m$$

Floating Inertia Matrix ($H^{\star}$) — Velocity dependent terms ($c^{\star}$)

$$t_{x_i} = J^{\star}_{x_i}\dot{q}_m$$

- Floating inertias and Jacobians are also provided and a recursive $\mathcal{O}(n)$ formulations to solve inverse dynamic problem, for a floating base, have also been implemented.

Desired Reaction Maneuver

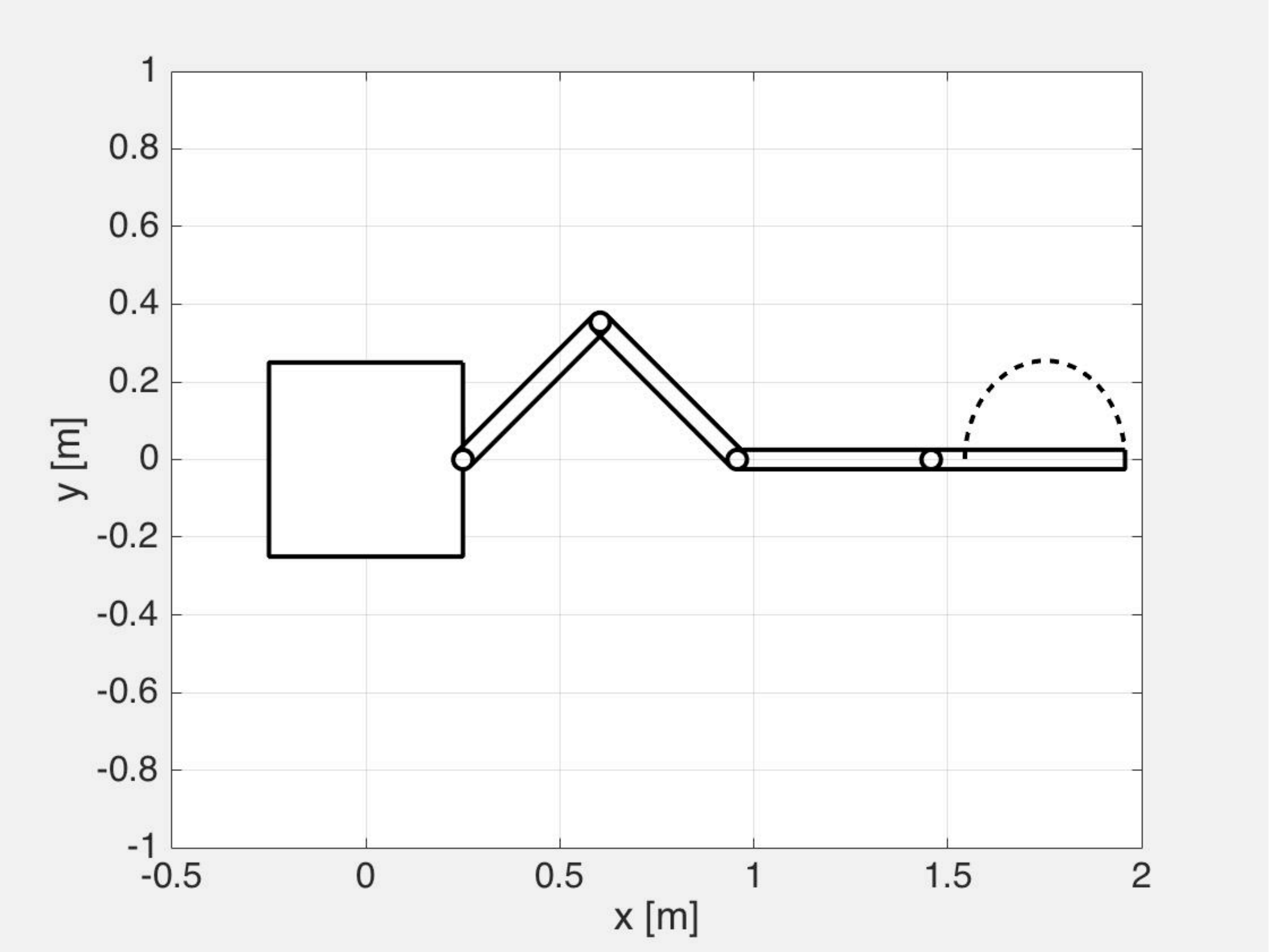
These functions toolkit allows us to create plants (solving the dynamic problems) and controllers.

$$\dot{q}_m = \begin{bmatrix} -H_0^{-1} H_{0m} \\ J^{\star}_{x_i} \end{bmatrix}^{-1} \left(\begin{bmatrix} \dot{q}_0 \\ t_{x_i} \end{bmatrix} - \begin{bmatrix} H_0^{-1} \\ J_{0x_i} H_0^{-1} \end{bmatrix} \mathcal{M}' \right)$$

Initial angular momentum

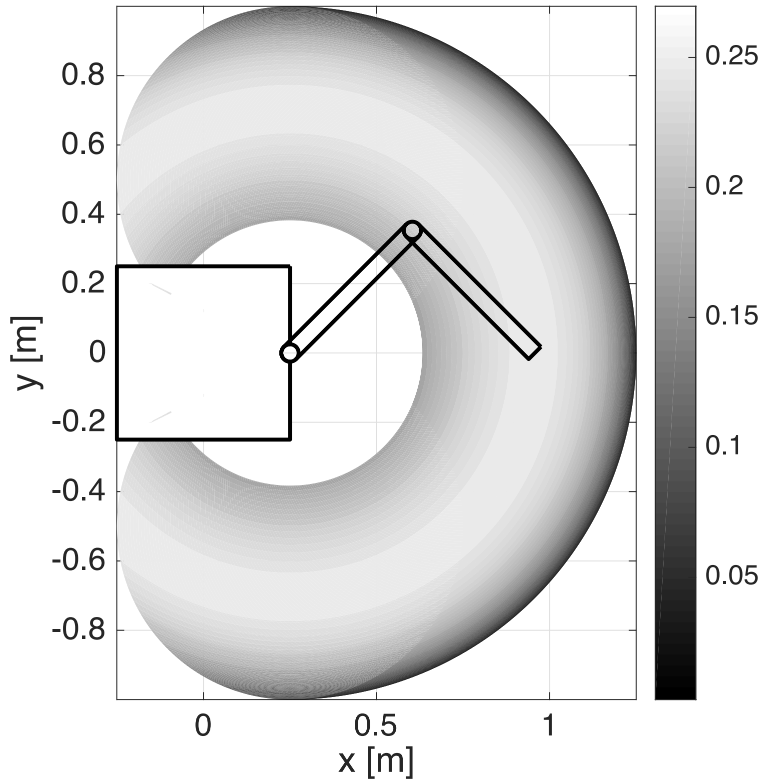
The DRM maneuver uses kinematic redundancy to imposed a desired base-spacecraft reaction.

Desired Reaction Maneuver - Video

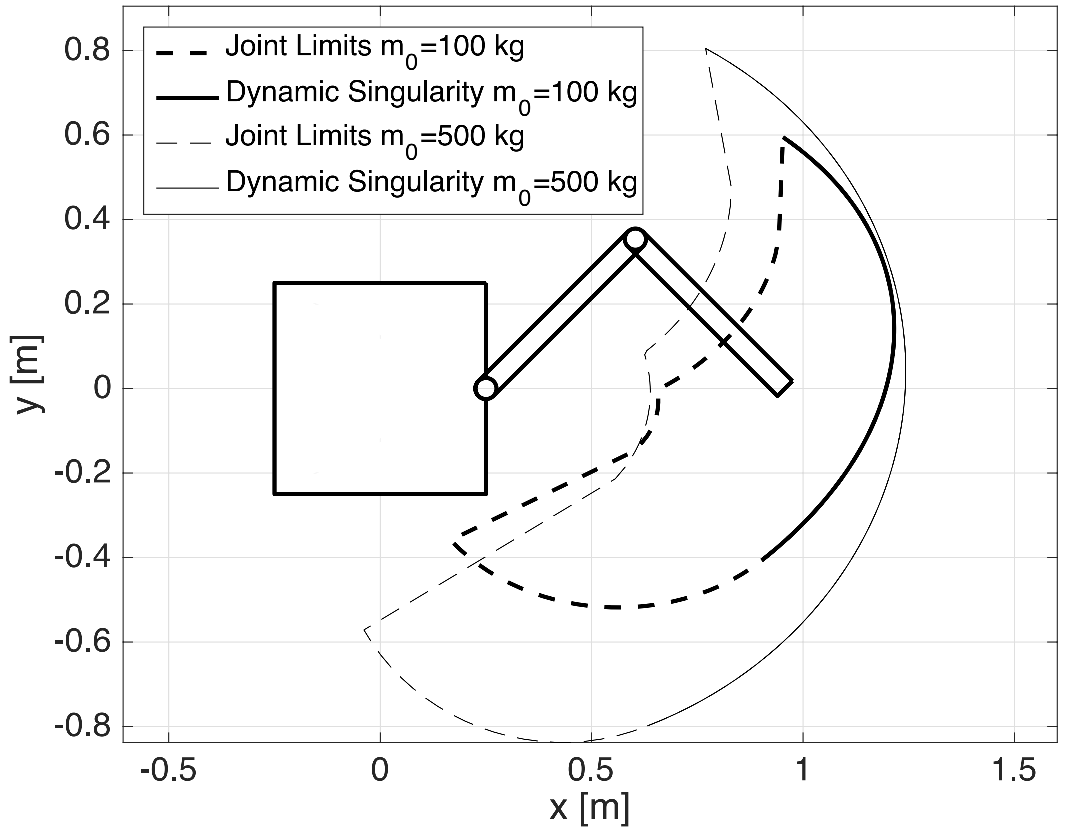


Examples

Fixed-base workspace



Straight Path floating workspace



Release notes

Source code and documentation available:

https://github.com/NPS-SRL/SPART

Release under a GNU GPL v3 license.

Thank you for your attention.

Questions?

SPART: an open-source simulator for spacecraft robotic arm dynamic modeling and control

Dr. Josep Virgili Llop, NPS-Mechanical and Aerospace Engineering
CPT Jerry V. Drew II, NPS-Mechanical and Aerospace Engineering
Prof. Marcello Romano, NPS-Mechanical and Aerospace Engineering