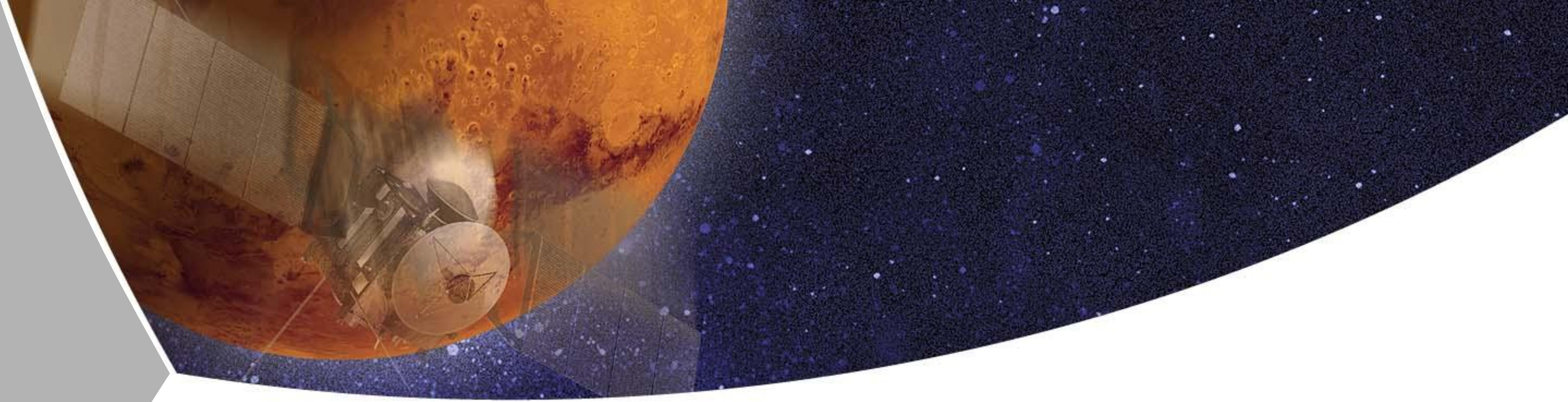


On-ground experimental verification of a torque controlled free-floating robot

Marco De Stefano, Jordi Artigas, Alessandro M. Giordano, Roberto Lampariello and

Alin-Albu Schaeffer

Institute of Robotics and Mechatronics – German Aerospace Center (DLR),

Oberpfaffenhofen, Germany

Deutsches Zentrum für Luft- und Raumfahrt e.V. in der Helmholtz-Gemeinschaft

Outline

1. Introduction

2. On-ground experimental verification of a torque controlled free-floating robot

- Dynamics and kinematics of a space robot
- Design of the torque controllers :
  - (1) Torque controller using the transpose of the generalized Jacobian
  - (2) Torque controller based on the inverse dynamics
- Simulations and Discussion

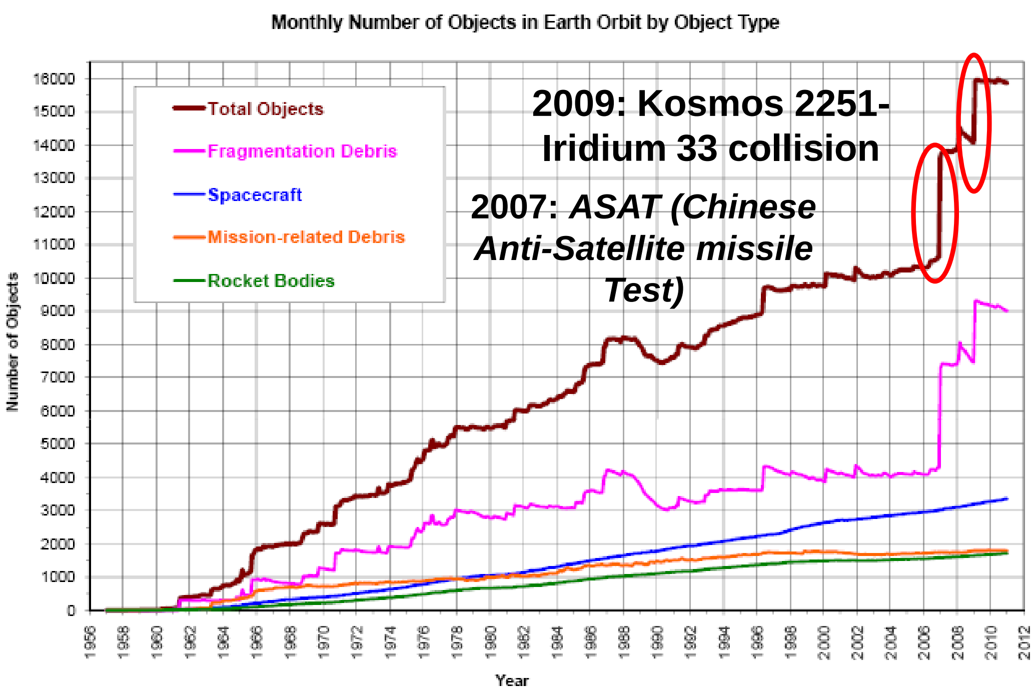
3. Experiments on the DLR-OOS facility

4. Conclusion

1. Introduction

On-Orbit Servicing (OOS) and space debris:

- Human OOS:
  - HST Hubble: Original cost: 500 Million $ → Costs up to 2010: 10 Billion $
  - Most part due to 4 manned OOS missions aboard the Space Shuttle (STS-61, 82, 103 and 109)
  - PALABA B-2 satellite recovery – 1984, Space Shuttle mission STS – 51 A. Five 5 astronauts were involved in the mission
- Space debris hazards



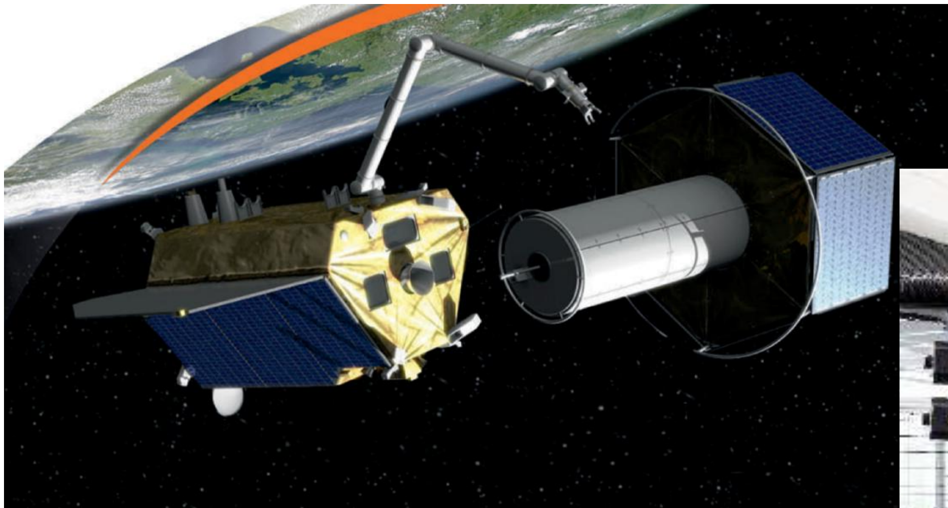
HST Hubble [NASA credit]

Palaba-B2 [NASA credit]

2. On-ground experimental verification of a torque controlled free-floating robot

On-orbit servicing simulator

- Robotic technologies applied to the on-orbit servicing mission
- Validation of control algorithms prior to the launch is required



Space scenario
chaser *(left)* and ***target*** *(right)*

On-ground scenario
two industrial robots and a Light Weight Robot (LWR) –
chaser *(left)* and ***target*** *(right)* – DLR OOS

Dynamics and kinematics of a space robot

- The space robot is a satellite equipped with a manipulator arm
- Space robots are classified into free-flyer and free-floating depending on the presence of actuator at the satellite-base
- The complete free-floating robot dynamics used to design the controller is:

$$\begin{bmatrix} \mathbf{H}_b & \mathbf{H}_{bm} \\ \mathbf{H}_{bm}^T & \mathbf{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}_\mathbf{b} \\ \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{c}_b \\ \mathbf{c}_m \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_b^T \\ \mathbf{J}_m^T \end{bmatrix} \mathbf{F}_e$$

- A Generalized dynamics can be calculated from the complete free-floating dynamics as:

$$\mathbf{H}^* \ddot{\mathbf{q}} + \mathbf{C}^* = \boldsymbol{\tau} + \mathbf{J}^{*\mathbf{T}} \mathbf{F}_\mathbf{e}$$

- To design the controller we use the generalized Jacobian* **J*** that relates the end-effector velocity to the joints velocity taking into account the conservation of momentum

$$\mathbf{J}^* = \mathbf{J}_\mathbf{m} - \mathbf{J}_\mathbf{b} \mathbf{H}_\mathbf{b}^{-1} \mathbf{H}_\mathbf{bm}$$

* E. Papadopoulos and S. Dubowsky, On the nature of control algorithms for space manipulators In Robotics and Automation, 1990

Controller Design

- **Requirements:** satisfy predefined compliance and impedance
- **Goal:** minimize the error during tracking and at steady state
- **Why torque control?** Compliance with the environment can be achieved

Error design

- A reference trajectory is provided as desired position and orientation
- Position and orientation are defined in the inertial space
- Quaternion representation has been used to define the orientation error

$$\Delta\boldsymbol{\phi} = 2E^T\hat{\boldsymbol{\varepsilon}} \qquad E = I(3,3)\eta - \tilde{\varepsilon}$$

- Position error: $\Delta\mathbf{p} = \mathbf{p_c} - \mathbf{p_t}$

- Vector error as input to the controller: $\Delta\mathbf{x} = [\Delta\mathbf{p}; \Delta\boldsymbol{\phi}]^T$

Torque controller analyzed

(1) Torque controller using the transpose of the generalized Jacobian

(2) Torque controller based on full inverse dynamics linearization

(1) Torque controller using the transpose of the generalized Jacobian

- Compliance behavior between the end-effector and target point
- The LWR robot is a 7 degree of freedoms (dof) and the motion in the null space must be controlled
- The control law takes into account an internal damping torque acting in the null space* of the robot without interacting with the end effector motion

$$\boldsymbol{\tau} = \mathbf{J}^{*\mathbf{T}}\mathbf{F} + (\mathbf{I} - \mathbf{J}^{*\mathbf{T}}\overline{\mathbf{J}}^{*\mathbf{T}})\Gamma + \mathbf{g}.$$

Null Space term

- The vector **g** is here necessary for the application of the controller on the facility on ground.
- The virtual Cartesian forces is modeled like a Proportional Derivative behavior:

$$\mathbf{F} = K_P \Delta\mathbf{x} + K_D \Delta\dot{\mathbf{x}}$$

* J. Russkow and O. Khatib, *A new control structure for free-flying space robot,* In i-SAIRAS, symoposium on Artificial Intelligence Robotics and automation in Space, Toulose 1992

(2) Torque control based on the inverse dynamics

- A strong dependence on the inverse free-floating dynamics is involved in the control law

- Acceleration of the end effector in Cartesian space* is expressed as:

$$\ddot{\mathbf{x}}_{\mathbf{e}} = \mathbf{J}^*\mathbf{H}^{*-1}\boldsymbol{\tau} + (\Lambda^{-1} + \Lambda_b^{-1})\mathbf{F}_{\mathbf{e}} + \boldsymbol{\mu}$$

- The torque law applied takes into the account the null space also:

$$\boldsymbol{\tau} = \mathbf{H}^*\overline{\mathbf{J}}^*\mathbf{u} + \mathbf{H}^*\overline{\mathbf{J}}^*[-\boldsymbol{\mu} - (\Lambda^{-1} + \Lambda_{\mathbf{b}}^{-1})\mathbf{F}_{\mathbf{e}}] + (\mathbf{I} - \mathbf{J}^{*\mathbf{T}}\overline{\mathbf{J}}^{*\mathbf{T}})\Gamma.$$

- It leads to: $\ddot{\mathbf{x}}_{\mathbf{e}} = \mathbf{u} + \mathbf{H}^*\overline{\mathbf{J}}^*(\mathbf{I} - \mathbf{J}^{*\mathbf{T}}\overline{\mathbf{J}}^{*\mathbf{T}})\Gamma$

Null Space term

- Here impedance is applied as:

$$\mathbf{u} = M\ddot{\mathbf{x}}_t - K_p\Delta\mathbf{x} - K_d\Delta\dot{\mathbf{x}}.$$

* S.Abiko, R. Lampariello and G. Hirzinger, *Impedance control fora free-floating robot in the grasping of a tumbling target with parameter uncertainty,* In Intelligent Robots and System, 2006 IEEE/RSJ , 2006

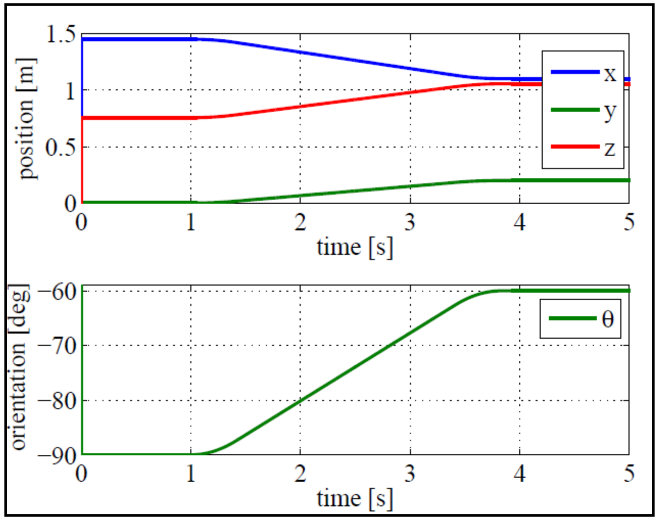
2. Simulations and discussion

Simulation study:

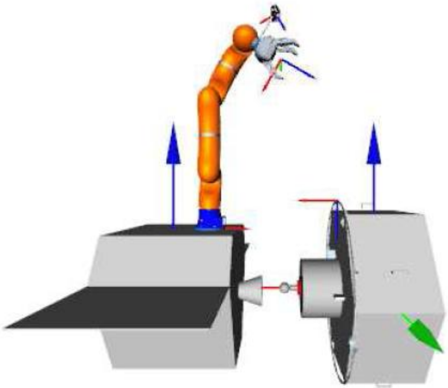
- The considered inertial parameters for the servicer satellite are:

$M_{satellite}[kg]$	$I_x[kgm^2]$	$I_y[kgm^2]$	$I_z[kgm^2]$
150	38	20	23

- A trajectory in the inertial frame is given as input to the torque control interface
- Usually the trajectory is provided by a motion planner to guarantee feasibility with respect to motion constraints
- Two simulations will be shown based on the presented torque control laws (1)-(2) using the same trajectory and initial conditions as input

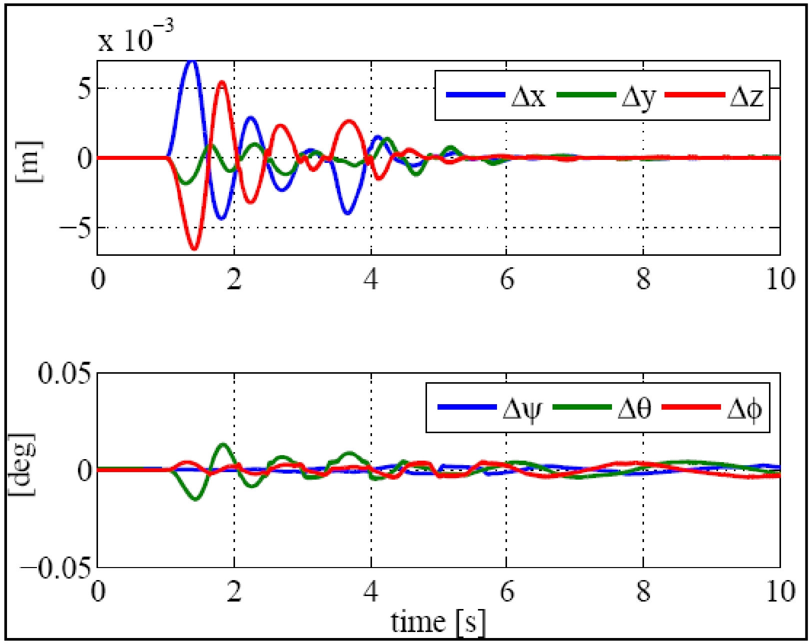


Input trajectory referred in the inertial space

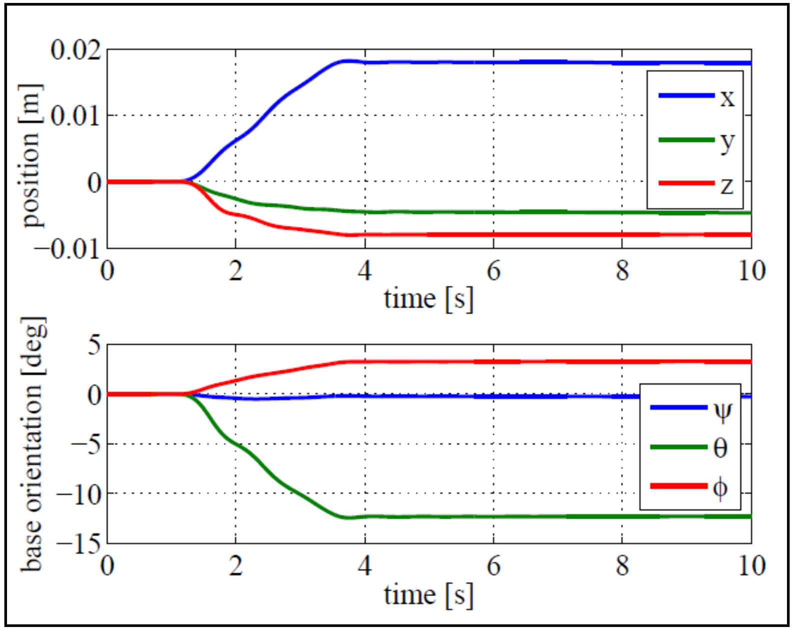

Simulation enviroment

Simulation (1): torque controller using the generalized Jacobian approach

- Tracking error (difference between the input trajectory and the instantaneous end-effector position) detected is 0.007m and 0.8 deg in orientation
- Satellite-base moves dynamically according to the LWR motion



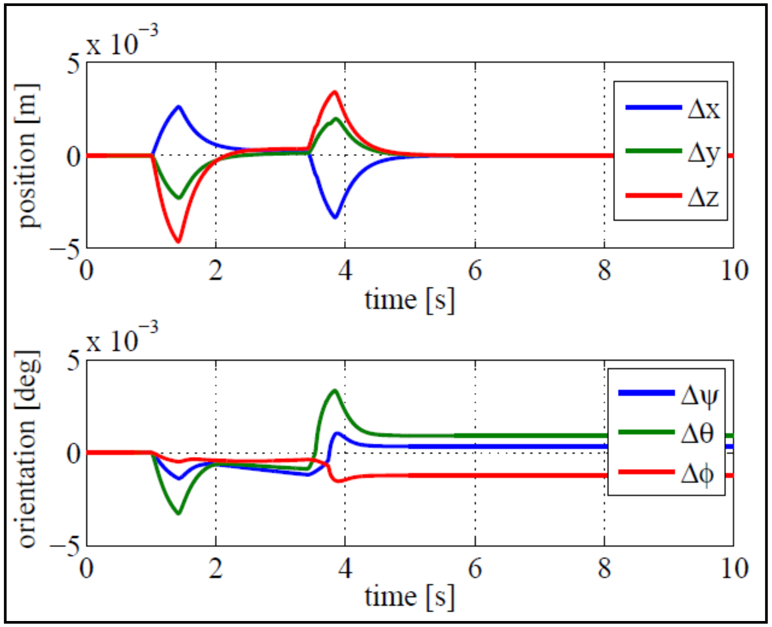
Tracking error in position and orientation



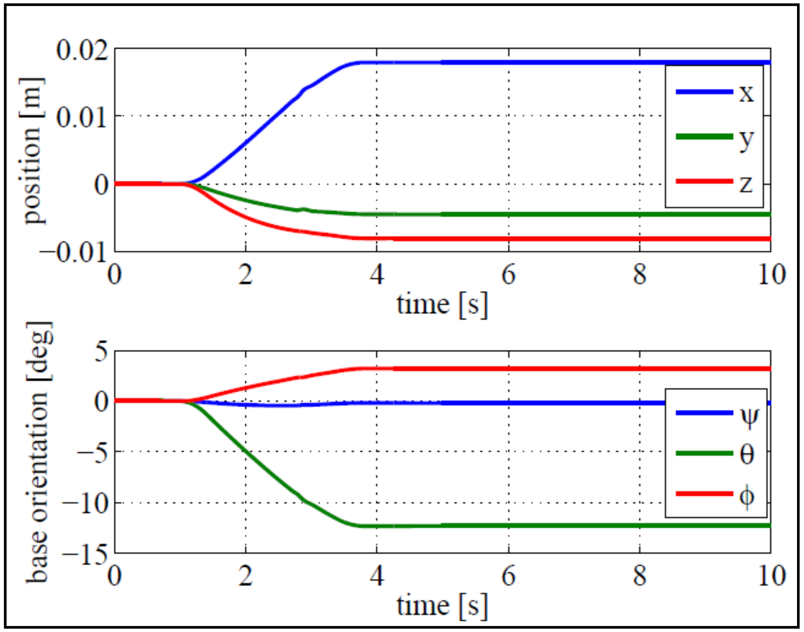
Relative satellite-base motion

Simulation (2): torque control based on the inverse dynamics

- Tracking error detected is 0.004m in position and 0.003 deg in orientation
- Steady state error: 10^-6 m and 0.001 deg
- Satellite-base moves dynamically according to the LWR motion



Tracking error in position and orientation



Relative satellite-base motion

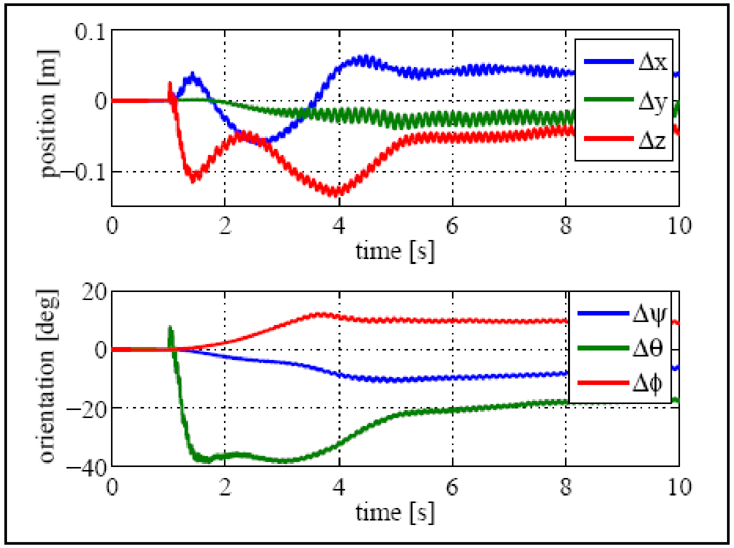
Simulation (2.1) - torque control based on the inverse dynamics: the unmodeled dynamics case

- The simulations show that better performance is achieved using the inverse dynamics control
- This controller proves inefficient to unmodeled dynamics, e.g. flexibility of the joints
- Same simulation for the controller (2) is analyzed considering the flexibility of the joints, modeled as:

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau} + D_m K_m^{-1} \dot{\boldsymbol{\tau}} = \boldsymbol{\tau_m}$$

$$\boldsymbol{\tau} = K_m(\boldsymbol{\theta} - \mathbf{q})$$

- Simulation results show that the error does not converge to zero

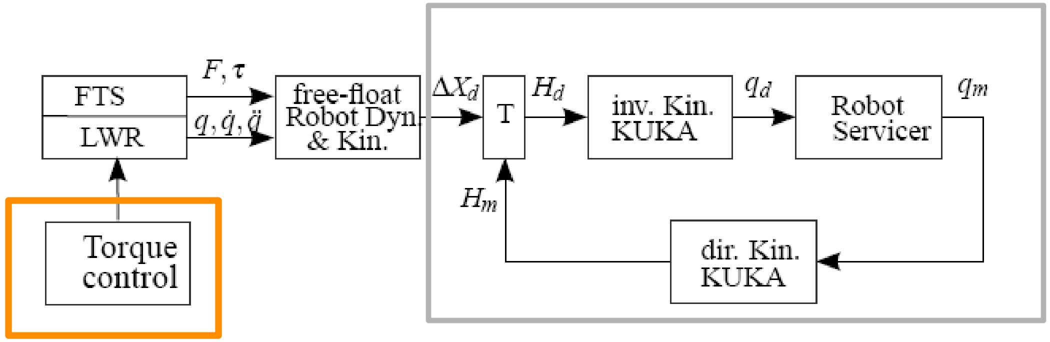


Tracking error considering unmodeled dynamic (e.g. joints flexibility)

3. Experiments on the DLR – OOS facility

The DLR-OOS facility

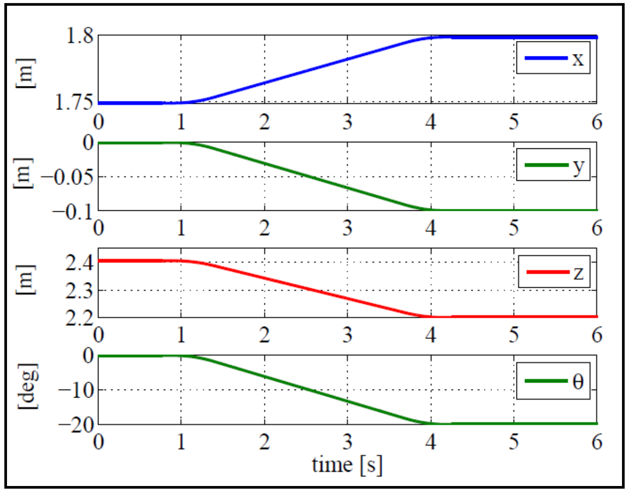
- The torque controller runs on a Real-time operating system
- The integration of the satellite-base dynamics is performed relative to the states of the LWR (e.g. velocities, positions and accelerations of the joints)
- The integrated data ΔX_d is the input to the KUKA kinematics that moves the robot in Cartesian space



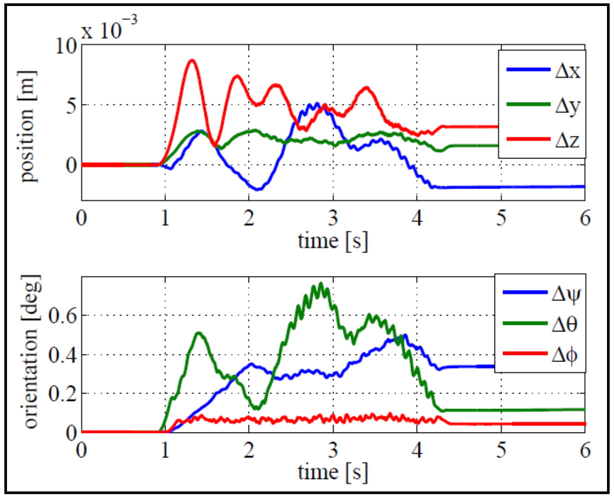
Data flow of the servicer dynamics used for the experiment

Experimental results

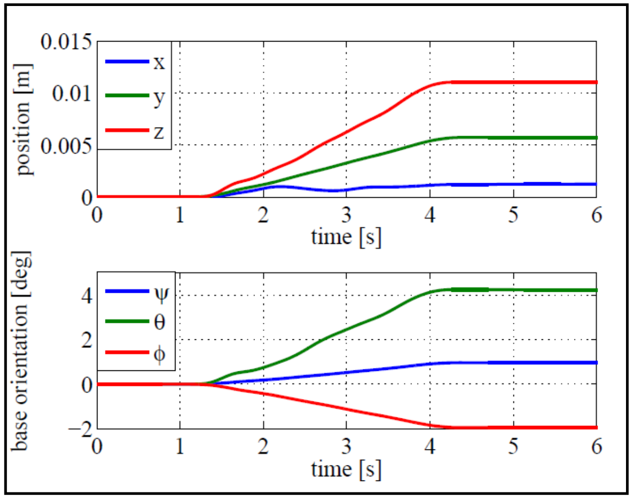
- The considered inertial parameters of the space robot are the same as in the simulation study
- The torque controller based on the generalized Jacobian were tested on the facility
- A trajectory in the inertial frame is given as input to the torque control interface



1. Input trajectory



2. Tracking error in position and orientation



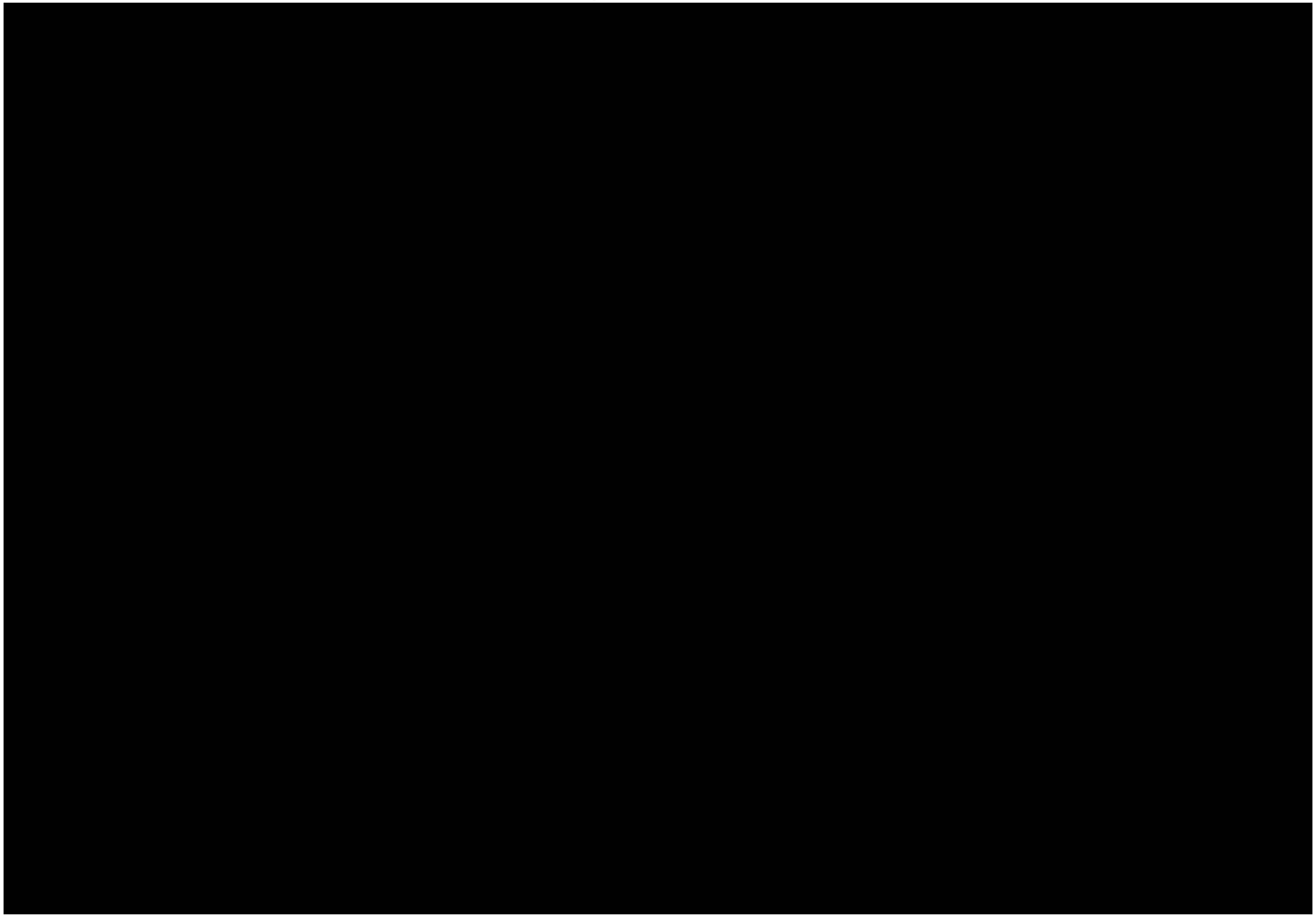
3. Relative motion of the satellite-base

4. Conclusion

4. Conclusion

- A comparison between two torque controller methods were performed in a simulation study
- The controller using the complete inverse dynamics loses performance due to unmodeled dynamics (e.g. flexible joints)
- The controller based on the transpose of the generalized Jacobian was performed in an experimental study on the DLR-OOS facility showing good performance

The DLR-On Orbit Servicing Simulator (to be presented at ICRA 2015)*



* J. Artigas, M. De Stefano, W. Rackl, R. Lampariello, et al. *The oos-sim: an on-gound simulation facility for on orbit servicing robotic operations*
In *Proceedings to Robotics and Automation (ICRA), 2015 IEEE International Conference, Seattle, May 2015.*