ON-GROUND EXPERIMENTAL VERIFICATION OF A TORQUE CONTROLLED FREE-FLOATING ROBOT

Marco De Stefano, Jordi Artigas, Alessandro Giordano, Roberto Lampariello, and Alin Albu-Schaeffer

Institute of Robotics and Mechatronics - German Aerospace Center (DLR), 82234 Oberpfaffenhofen - Germany,
email: name.lastname@dlr.de

ABSTRACT

In this paper, we present a comparison between two torque-based control algorithms for a free-floating robot: the first uses the generalized Jacobian transpose, while the second is based on a complete feedback linearization of the free floating robot dynamics. The analyzed task is to follow a desired trajectory (e.g. provided by a motion planner) and the controllers must satisfy predefined compliance and impedance conditions for the manipulator end effector. The effectiveness of the algorithms and the performance are demonstrated in simulation and through experiments on a hardware-in-the-loop facility.

Key words: free-floating robot; torque control; robotics facility; OOS; space debris.

1. INTRODUCTION

The number of space debris objects, given by non-functional satellites in low-earth orbits, has been increasing over recent year. Rather than disposing defective satellites, On-Orbit servicing system (OOS) can undertake maintenance task and put them back into operation. To validate the OOS technology, on-ground test facilities are required for testing and verification of planning and control algorithms prior to the launch. Some technologies such as free-fall towers, parabolic flights, air-bearing testbeds, neutral buoyancy, suspended systems and robotic simulators [1], [2] are already available. A first milestone for the robotic technology in space was ROTEX developed by the German Aerospace Center (DLR) [3], where several robotic key technologies were firstly tested on the COLUMBIA Space Shuttle. Further, the Engineering Test Satellite (ETS) VII in 1997 was an important mission where many technologies and control algorithms were tested with the focus on unmanned on-orbit servicing [4]. In [5] the feasibility of grasping, berthing and servicing operations were partly demonstrated in autonomous mode and in teleoperation mode, for a non-cooperative target satellite on an OOS simulation facility on ground (see Fig.1).

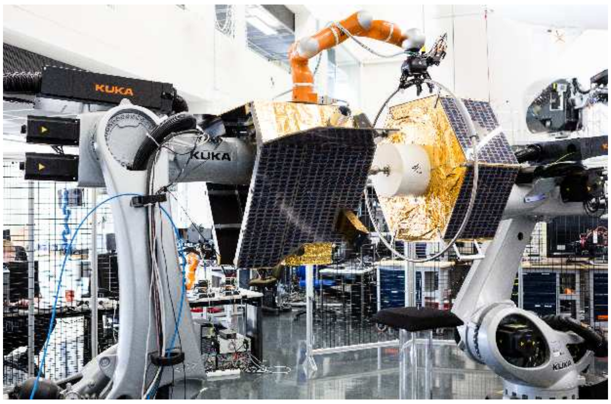

Figure 1: On-ground experimental simulator for a space robot (left) and a target satellite (right)

Space robots can be classified into free-flyer robots, where the satellite base is actuated by thrusters or reaction wheels, and free-floating robots, in which the satellite base is free to float. During the capture phase, the free-floating robot results more energy efficient and this is the case that we consider in our paper. The presence of a floating base induces many challenges from the point of view of robot dynamics, planning and control. Indeed, the robot motion is coupled with the floating motion of the satellite, inducing translations and rotations of the base together with interferences on the desired motion of the end effector. One important consequence of such coupling is the presence of dynamic singularities [6, 7] that are path-dependent, unlike for the fixed-base case. The presence of dynamic singularities further complicates the task of the trajectory planning algorithms [8]. In [9] the kinematic problem for a free floating robot was addressed and the generalized Jacobian was presented. This relates the end effector velocities to the joint velocities, taking into account the conservation of momentum, which characterizes the motion of the free-floating robot in absence of external forces. Based on this concept, a kinematics-based control was then presented in [10]. The dynamics problem was treated in [11], where a generalized transposed Jacobian approach was presented for the end effector regulation control and the stability was proved. In [12] the generalized Jacobian was shown to be the natural extension of the classic Jacobian for free-floating systems, allowing to address most of the control problems using classical approaches for fixed-base manipulators. The effectiveness of the transpose Jacobian control was further investigated also for free-flying systems in [7].

The major benefits using torque controlled robots is that compliance with the environment can be achieved. A torque controlled robot typically processes the joint sensor torque signal locally at joint level, with lower latencies and high frequency, compared to the position control. Impedance control can therefore be properly implemented in the outer loop controller, where an enhanced compliant behavior at the tool-center-point can be achieved in Cartesian space. We believe that in on-orbit servicing robotics, such a property is valuable in allowing impedance during the grasping.

In this work, a comparison between a controller based on the generalized Jacobian transpose and a torque controller based on an inverse dynamics linearization is performed. The feasibility of their application on a robot with flexible joints is discussed and evaluated experimentally. The tests are performed on a real-time OOS simulation facility on ground [5]. The facility consists of an industrial robot (KUKA) and a 7 degrees of freedom (dof) Light-Weight-Robot (LWR) mounted on its end effector, to simulate the servicer satellite, resulting in a macro-micro configuration with 6+7 dofs (see Fig.1).

2. DYNAMICS AND KINEMATICS OF A SPACE ROBOT

This section introduces the model of a space robot. The general equations of motion for a free-flying robots are [13]:

$$\begin{bmatrix} \mathbf{H}_b & \mathbf{H}_{bm} \\ \mathbf{H}_{bm}^T & \mathbf{H}_m \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}_\mathbf{b} \\ \ddot{\mathbf{q}} \end{bmatrix} + \begin{bmatrix} \mathbf{c}_b \\ \mathbf{c}_m \end{bmatrix} = \begin{bmatrix} \mathbf{F}_b \\ \boldsymbol{\tau} \end{bmatrix} + \begin{bmatrix} \mathbf{J}_b^T \\ \mathbf{J}_m^T \end{bmatrix} \mathbf{F}_e, \tag{1}$$

where $\mathbf{H}_b \in \mathbb{R}^{6\times6}$, $\mathbf{H}_m \in \mathbb{R}^{7\times7}$, $\mathbf{H}_{bm} \in \mathbb{R}^{6\times7}$ are the inertia matrices of the base and manipulator and the coupling inertia matrix between the base and the manipulator, respectively. The vectors $\ddot{\mathbf{x}}_\mathbf{b} \in \mathbb{R}^{6\times1}$ and $\ddot{\mathbf{q}} \in \mathbb{R}^{7\times1}$ are the acceleration of the base and the acceleration of the robot joints; $\mathbf{c}_b \in \mathbb{R}^{6\times1}$ and $\mathbf{c}_m \in \mathbb{R}^{6\times1}$ are the non-linear velocity dependent terms on the base and on the manipulator, respectively. $\mathbf{F}_b \in \mathbb{R}^{6\times1}$ and $\mathbf{F}_e \in \mathbb{R}^{6\times1}$ are the force torque wrenches acting on the center of mass of the base-body or on the end effector, respectively. Moreover, $\boldsymbol{\tau} \in \mathbb{R}^{7\times1}$ is the internal torque vector and $\mathbf{J}_b \in \mathbb{R}^{6\times6}$ and $\mathbf{J}_m \in \mathbb{R}^{6\times7}$ are the Jacobian matrices of the base and manipulator, respectively.

From eq. (1), $\ddot{\mathbf{x}}_\mathbf{b}$ can be eliminated to obtain the following expression:

$$\mathbf{H}^*\ddot{\mathbf{q}} + \mathbf{C}^* = \boldsymbol{\tau} + \mathbf{J}^{*\mathbf{T}}\mathbf{F}_\mathbf{e}. \tag{2}$$

where $\mathbf{H}^* \in \mathbb{R}^{6\times6}$, $\mathbf{C*} \in \mathbb{R}^{6\times1}$, $\mathbf{J*} \in \mathbb{R}^{6\times7}$ are the so-called generalized inertia matrix, generalized Coriolis and centrifugal forces and generalized Jacobian of a space robot, respectively [9]. The matrix $\mathbf{H}^*$ can be mapped onto the Cartesian space using the generalized Jacobian matrix defined as:

$$\mathbf{J}^* = \mathbf{J_m} - \mathbf{J_b}\mathbf{H_b}^{-\mathbf{1}}\mathbf{H_{bm}}. \tag{3}$$

The kinematics between the operational space and the joint space is described as follows:

$$\dot{\mathbf{x}}_\mathbf{e} = \mathbf{J}^*\dot{\mathbf{q}} + \mathbf{J_b}\mathbf{H_b^{-1}} \begin{bmatrix} \mathbf{P} \\ \mathbf{L} \end{bmatrix} \tag{4}$$

where $\dot{\mathbf{x}}_\mathbf{e} \in \mathbb{R}^{6\times1}$ is the end effector velocity vector. $\mathbf{P}$ and $\mathbf{L}$ are the linear and angular momentum with respect to the centre of mass of the system, which, in our case, are constant, since we do not consider any external forces acting.

3. CONTROLLER DESIGN

One of the most important phases during the capture of a satellite by a space robot is the approach phase. An unexpected contact can lead to losing the target. Usually, torque-based control is preferred in this case since it is based on an impedance behavior between the end point and the target point. The goal of the controller is to minimize the error between the current end effector position and the target position, or otherwise, to follow a reference trajectory to a predefined target grasping point. We assume here that the target information (e.g. position and orientation) is provided by a camera system mounted onto the space robot. We use a quaternions representation in order to define the rotational error. We define $R_c \in \mathbb{R}^{3\times3}$ to be the rotational matrix of the end effector with respect to the inertial frame and $R_t \in \mathbb{R}^{3\times3}$ to be the desired rotational matrix, expressed in the same frame. The error matrix is then defined as $R_\phi = R_t R_c^T$. By using the quaternion representation, a scalar η and a vector $\hat{\boldsymbol{\varepsilon}} \in \mathbb{R}^{3\times1}$ can be defined, such that the orientation error $\Delta\boldsymbol{\phi} \in \mathbb{R}^{3\times1}$ follows as:

$$\Delta\boldsymbol{\phi} = 2E^T\hat{\boldsymbol{\varepsilon}}, \tag{5}$$

where $E \in \mathbb{R}^{3\times3}$ is defined as $E = I(3,3)\eta - \tilde{\varepsilon}$ and where $\tilde{\varepsilon}$ is the known skew-symmetric matrix of the vector $\hat{\boldsymbol{\varepsilon}}$. Analogous for the position error:

$$\Delta\mathbf{p} = \mathbf{p_c} - \mathbf{p_t}, \tag{6}$$

where $\mathbf{p_c}$ is the position of the end effector in the inertial frame and $\mathbf{p_t}$ is the desired position. The error in orientation and position can be expressed as the vector $\Delta\mathbf{x} \in \mathbb{R}^{6\times1}$, given by:

$$\Delta\mathbf{x} = [\Delta\mathbf{p}; \Delta\boldsymbol{\phi}]^T. \tag{7}$$

This representation will be used in the following subsection in order to design the controllers.

3.1. Torque controller using the generalized transposed Jacobian

This controller leads to a compliance behavior between the end-effector and the target point. The control strategy is based on the torque control law (8), where $\boldsymbol{\tau} \in \mathbb{R}^{7\times 1}$ are the input torques to the manipulator:

$$\boldsymbol{\tau} = \mathbf{J}^{*\mathbf{T}}\mathbf{F} + (\mathbf{I} - \mathbf{J}^{*\mathbf{T}}\overline{\mathbf{J}}^{*\mathbf{T}})\Gamma. \tag{8}$$

$\mathbf{J}^{*\mathbf{T}}$ is the generalized Jacobian matrix traspose, while $\mathbf{F} \in \mathbb{R}^{6\times 1}$ is the virtual control forces vector applied at the end-effector. For a redundant robot (such as is the case of the LWR) the null space motion requires particular attention and must be taken into account in the controller law. For this reason, we consider an arbitrary generalized joint torque vector $\Gamma = -D_n\dot{\mathbf{q}}$ that acts as internal damping torque in the null space of the robot, without interfering with the end effector motion. To apply this torque in the null space, we use the $\overline{\mathbf{J}}^* \in \mathbb{R}^{7\times 6}$ [14] that results as the dynamically consistent generalized inverse (note that $\mathbf{J}^*$ is not a square matrix) defined as:

$$\overline{\mathbf{J}}^* = \mathbf{H}^{*-\mathbf{1}}\mathbf{J}^{*\mathbf{T}}\Lambda. \tag{9}$$

where $\Lambda \in \mathbb{R}^{6\times 6}$ is the inertia matrix $\mathbf{H}^*$ in the Cartesian space:

$$\Lambda = \left(\mathbf{J}^*\mathbf{H}^{*-\mathbf{1}}\mathbf{J}^{*\mathbf{T}}\right)^{-1}. \tag{10}$$

Moreover, in eq. (8), the gravitational torque vector $\mathbf{g}$ can be added for the application of the controller on the facility on ground. The virtual Cartesian forces vector $\mathbf{F}$ at the end-effector is modeled like a PD (Proportional Derivative) behavior, defined as:

$$\mathbf{F} = K_P\Delta\mathbf{x} + K_D\Delta\dot{\mathbf{x}}, \tag{11}$$

K_P and K_D are [6x6] positive definite matrices, representing stiffness and damping gains of the controller. Eq. (8) is then computed as internal joint torques to the generalized free floating robot dynamic in eq. (2).

3.2. Torque controller based on the inverse dynamics linearization

This controller is based on a full inverse dynamics of the free floating robot dynamics, in order to apply linear control. The authors in [15] defined the equations of motion in the operational space as a function of the end effector acceleration and joint torques and apply a controller based on feedback linearization for a free-floating robot without flexible joints. The derivative of eq. (4) leads to define the acceleration of the end effector in Cartesian space. Then, by using eq. (2) the equation of motion in Cartesian space can be expressed as:

$$\ddot{\mathbf{x}}_{\mathbf{e}} = \mathbf{J}^*\mathbf{H}^{*-\mathbf{1}}\boldsymbol{\tau} + (\Lambda^{-1} + \Lambda_b^{-1})\mathbf{F}_{\mathbf{e}} + \boldsymbol{\mu}, \tag{12}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{6\times 1}$ is the Coriolis and centrifugal forces vector in Cartesian space and $\Lambda_b = \left(\mathbf{J}^*_{\mathbf{b}}\mathbf{H}^{*-\mathbf{1}}_{\mathbf{b}}\mathbf{J}^{*\mathbf{T}}_{\mathbf{b}}\right)^{-1} \in \mathbb{R}^{6\times 6}$ is the inertia matrix of the base. We want to have a desired behavior of the end effector (later defined) such that $\ddot{\mathbf{x}}_{\mathbf{e}} = \mathbf{u}$ where $\mathbf{u}$ is a desired control. In order to do that we apply a torque defined as:

$$\boldsymbol{\tau} = \mathbf{H}^*\overline{\mathbf{J}}^*\mathbf{u} + \mathbf{H}^*\overline{\mathbf{J}}^*[-\boldsymbol{\mu} - (\Lambda^{-\mathbf{1}} + \Lambda_{\mathbf{b}}^{-\mathbf{1}})\mathbf{F}_{\mathbf{e}}] + \tag{13}$$
$$(\mathbf{I} - \mathbf{J}^{*\mathbf{T}}\overline{\mathbf{J}}^{*\mathbf{T}})\Gamma.$$

In our case, $\mathbf{F}_{\mathbf{e}} = 0$ since we do not consider any external forces acting. Eq. (13) can be substituted in eq. (12) which it leads to a linearized dynamics. The output is the end effector acceleration expressed as:

$$\ddot{\mathbf{x}}_{\mathbf{e}} = \mathbf{u} + \mathbf{H}^*\overline{\mathbf{J}}^*(\mathbf{I} - \mathbf{J}^{*\mathbf{T}}\overline{\mathbf{J}}^{*\mathbf{T}})\Gamma. \tag{14}$$

Here is where the impedance behavior is imposed as:

$$\mathbf{u} = M\ddot{\mathbf{x}}_t - K_p\Delta\mathbf{x} - K_d\Delta\dot{\mathbf{x}}, \tag{15}$$

where $\ddot{\mathbf{x}}_t$ is the desired target acceleration. M,K_P and K_D are [6x6] positive definite matrices representing the mass matrix and the stiffness and damping gains. From the above consideration, asymptotic stability is guaranteed:

$$M\Delta\ddot{\mathbf{x}} + K_d\Delta\dot{\mathbf{x}} + K_p\Delta\mathbf{x} = 0. \tag{16}$$

Note that in the above control, a strong dependence on the dynamics model of the robot results.

4. SIMULATION

The controllers defined above were tested in simulation. The mass and inertia parameters of the simulated servicer satellite can be found in Table 1.

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

Table 1: Mass and inertia properties of the servicer satellite (base of the LWR)

The servicer arm considered is a 7 dof LWR, whose mass and inertia parameter were identified and are reported in Table 2.

$M_{link}[kg]$	$I_x[kgm^2]$	$I_y[kgm^2]$	$I_z[kgm^2]$
2.71	0.023	0.023	0.005
2.71	0.024	0.005	0.024
2.54	0.013	0.013	0.005
2.50	0.023	0.005	0.002
1.30	0.023	0.022	0.003
1.57	0.003	0.003	0.003
4.1	0.024	0.002	0.024

Table 2: Mass and inertia properties of the LWR servicer arm

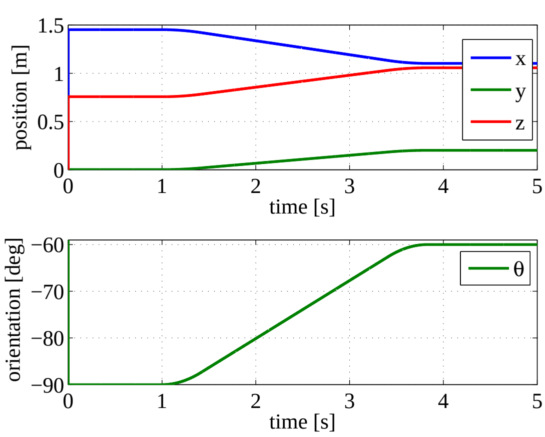

Figure 2: Input trajectory referred to the inertial frame

Furthermore, the simulation was performed on a 2.26 GHz Linux computer in a Simulink/Matlab environment. The error in orientation is show here in plots below with a Euler 321 representation $[\psi,\theta,\phi]$, which is more intuitive than the quaternions representation used by the controller (see Sec.3).

The considered impedance is described by the stiffness and damping gains. For the first Jabobian-based controller we implemented a stiffness of $K_p = diag[1000,1000,1000,200,200,200]$ and a damping of $K_D = diag[25,25,25,4,4,4]$. For the second linearization-based controller, optimal gains were derived from the case of a unitary mass with a damping ratio $\xi = 0.7$ and frequency $f = 10Hz$, resulting in $K_p = diag[40,40,40,40,40,40]$ and damping $K_D = diag[13,13,13,13,13,13]$. Both simulations were performed for the same reference trajectory, shown in Fig. 2. The trajectory along ψ and ϕ is kept to zero, therefore these components are not shown in Fig. 2. Usually, in the autonomous mode, the trajectory is provided by a motion planner in order to guarantee feasibility with respect to motion constraints, such as singularities, collision avoidance and end-effector camera field of view [8].

4.1. Torque controller using the generalized transposed Jacobian

The torque control in eq. (8) is analyzed here in simulation. The error detected during the tracking is shown in Fig. 3. The maximun error detected is 0.007 m in position and 0.01 deg in orientation during the tracking phase. Despite the controller solves only a regulation problem, the tracking performance are satisfactory for the given trajectory. Due to the given LWR motion, the satellite base moves as shown in Fig. 4.

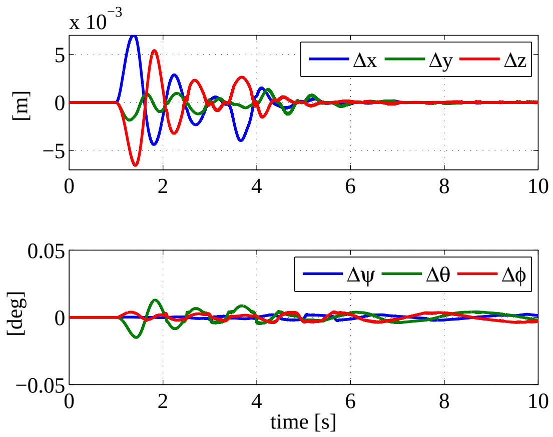

Figure 3: Tracking error with the torque control using the transpose of the generalized Jacobian

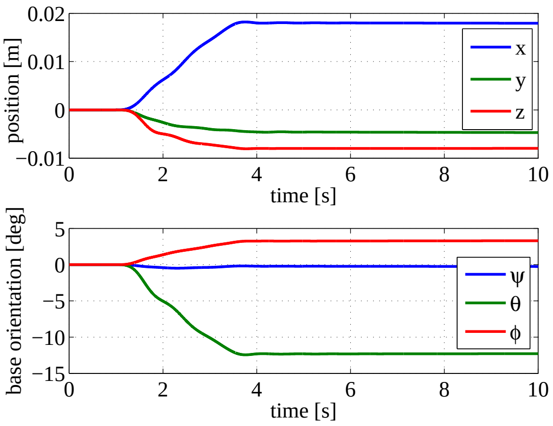

Figure 4: Relative motion of the satellite base

4.2. Torque controller based on the inverse dynamics linearization

The simulation presented here considers a torque input described in eq. (13). The error detected during the tracking is shown in Fig. 5. During the tracking phase the maximum error is found to be 0.004 m for the position and 0.003 deg for the orientation. In the steady state it is 10^{-6} m in position and 0.001 deg in orientation. Due to the given LWR motion, the satellite base moves as shown in Fig. 6.

4.3. Torque controller based on the inverse dynamics linearization with flexible joint dynamics

The simulation shows that the controller has good performance by decoupling the complete dynamics (even using lower gains) in case of a rigid joints robot. However, if unmodeled dynamics are not taken into account, the controller behaves differently. For example, let us consider

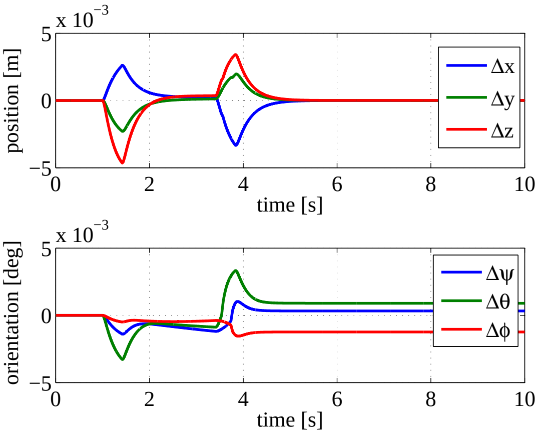

Figure 5: Tracking error with the torque control base on the inverse dynamics linearization

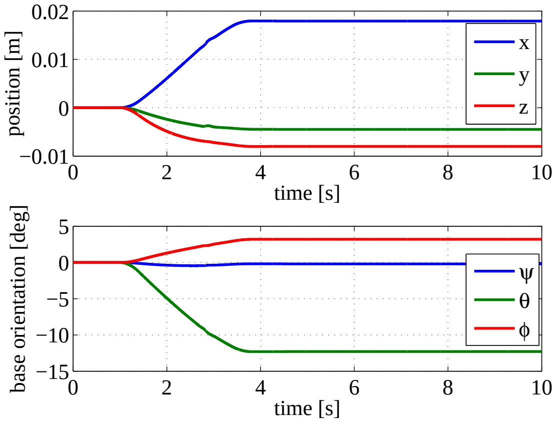

Figure 6: Relative motion of the satellite base

the dynamics of flexible joints [16], given by:

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau} + D_m K_m^{-1}\dot{\boldsymbol{\tau}} = \boldsymbol{\tau_m} \qquad (17)$$

$$\boldsymbol{\tau} = K_m(\boldsymbol{\theta} - \mathbf{q}) \qquad (18)$$

where $\mathbf{B} \in \mathbb{R}^{7\times7}$ is the inertia matrix of the motor, $\boldsymbol{\theta} \in \mathbb{R}^{7\times1}$ is the measured motor position, $K_m, D_m \in \mathbb{R}^{7\times7}$ are diagonal matrices representing the joint stiffness and damping. Considering the flexible joints dynamics and running the same simulation presented above, this shows that the tracking error behaves differently (see Fig. 7). As a matter of fact, the tracking error increases and the robot is not able to follow the reference trajectory.

For this reason, in our experiment we only apply the torque controller based on the generalized Jacobian matrix, since in our facility we use a robot with flexible joints. Moreover, this controller provides higher robustness to the unmodeled robot dynamics. A complete description of this control method is given in [16], however for the case of a fixed-base robot. Furthermore, it is interesting to note that the feedback linearization approach for a robot with flexible joints was found to have a limited performance, due to the necessity of computing higher derivate terms in the resulting control law [16].

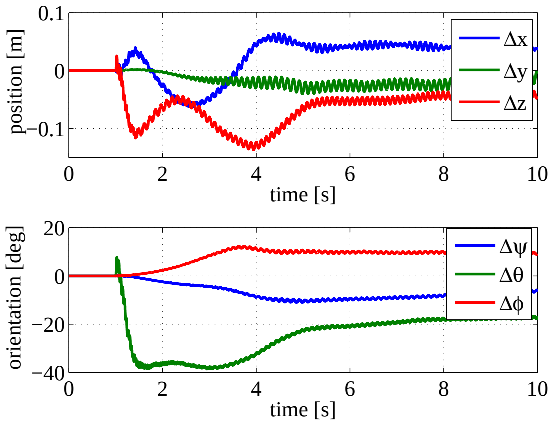

Figure 7: Tracking error considering uncertainties dynamics (e.g. flexible joints) for the controller (13)

5. EXPERIMENTS

The experiments were carried out on the DLR-RMC OOS simulation facility [5]. The data flow of the dynamics computation is shown in Fig. 8. The torque controller

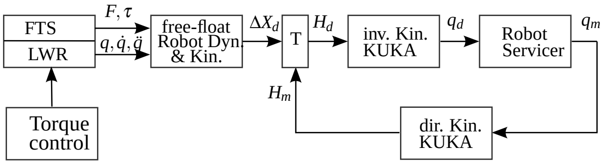

Figure 8: Data flow for the servicer dynamics used for the experiment

runs on a real-time VxVorks computer with a sampling time of 1 ms. It provides the input to the LWR in torque mode defined in Sec. 3. The measured joint positions of the LWR, and their time derivatives, are given as input for the integration of the free-floating base body dynamics (first line of eq. (1)). This provides a ΔX_d as input to the inverse KUKA kinematics, that accordly moves the industrial robot in the Cartesian space.

The experiment presented here consists in following a trajectory (see Fig. 9), similar to the one used in the simulation test (see Sec. 4). The torque control law, used to track this trajectory, is described in eq. (8) (adding the gravity vector $\mathbf{g}$) and the gains K_p and K_D are the same as those uesd in the simulation. The end effector follows the defined trajectory as shown in Fig. 10. The data in Fig. 10 shown a maximum error of 0.008 m in position and 0.8 deg in orientation during the tracking phase. Furthermore, after the maneuver, the maximum error found experimentally is 0.004 m for the position and 0.4 deg for the orientation in the steady state. The difference with the analyzed case in simulation is given by the presence of joints friction in the hardware.

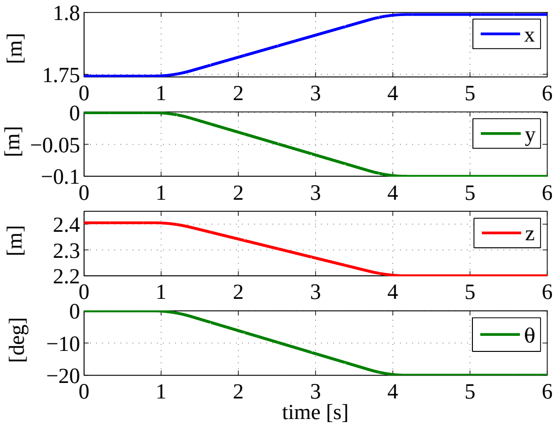

Figure 9: Input trajectory referred to the inertial frame

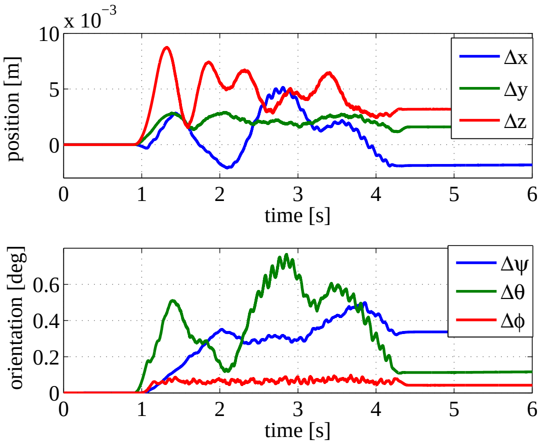

Figure 10: Experimental results: Tracking error in position and orientation

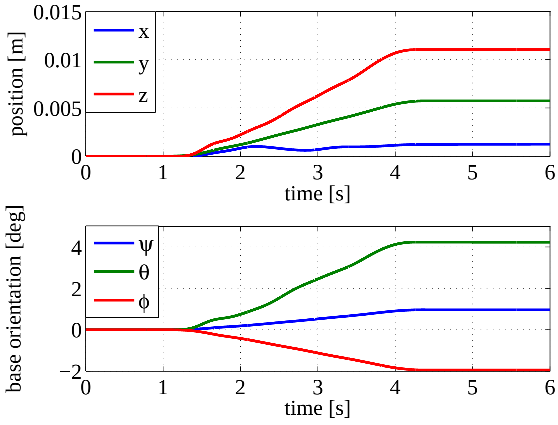

Figure 11: Experimental results: Measured relative motion of the satellite base

Moreover, the measured free floating robot motion, as reaction to the LWR motion, is shown in Fig. 11. This test is an experimental proof that using the generalized Jacobian matrix the end effector position tracks the reference trajectory to a satisfactory degree. Since the robot is controlled in torque mode, some uncertainties given by the gravity compensation and by the joints friction are intrinsic in the system. Nevertheless, the controller is able to deal with them adequately.

6. CONCLUSION

In a free floating robot the dynamic coupling between the manipulator and its base leads to a different end effector position, compared to that of a fixed base robot. For this reason, a controller for a fixed base robot results inefficient for a planned task (e.g. tracking or approaching a target). The controller presented in this paper is applied to solve a regulation control problem using the generalized Jacobian matrix, therefore intrinsically accounting for the free-floating dynamics. Experimental results show that the controller has a good performance during the tracking maneuver. A compararison is made to a controller based on the full inverse dynamics linearization, showing that only the first can deal with unmodelled dynamics resulting from flexibility in the robot joints.

REFERENCES

[1] Jana L. Schwartz, Mason A. Peck, and Christopher D. Hall. Historical review of air-bearing spacecraft simulators. In *Journal of Guidance, Control and Dynamics*, pages AAS 03–125, 2003.

[2] Florian Sellmaier, Toralf Boge, Jrn Spurmann, Sylvain Gully, Thomas Rupp, and Felix Huber. On-orbit servicing missions: Challenges and solutions for spacecraft operations. In *American Institute of Aeronautics and Astronautics, Inc.. SpaceOps 2010 Conference, 25,Huntsville, Alabama, USA*, 2010.

[3] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl. Sensor-based space robotics-rotex and its telerobotic features. *Robotics and Automation, IEEE Transactions on*, 9(5):649–663, Oct 1993.

[4] Kazuya Yoshida. Engineering test satellite vii flight experiments for space robot dynamics and control: Theories on laboratory test beds ten years ago, now in orbit. *The International Journal of Robotics Research*, 22(5):321–335, 2003.

[5] J. Artigas, M. De Stefano, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O. Porges, A. Giordano, C. Borst, and A. Albu-Schaeffer. 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*, 2015.

[6] E. Papadopoulos and D. Dubowsky. Dynamic singularities in free-floating space manipulators. In *Journal of Dynamic Systems, Measurement, and Control*, pages 115(1):44–52., 1993.

[7] S. Dubowsky and E. Papadopoulos. The kinematics, dynamics, and control of free-flying and free-floating space robotic systems. *Robotics and Automation, IEEE Transactions on*, 9(5):531–543, Oct 1993.

[8] R. Lampariello and G. Hirzinger. Generating feasible trajectories for autonomous on-orbit grasping of spinning debris in a useful time. In *EEE/RSJ International Conference on Intelligent Robots and Systems 2013 (IROS 13)*. Tokyo, Japan, 2013.

[9] Yoji Umetani and Kazuya Yoshida. Continuous path control of space manipulators mounted on {OMV}. *Acta Astronautica*, 15(12):981 – 986, 1987.

[10] Y. Umetani and K. Yoshida. Resolved motion rate control of space manipulators with generalized jacobian matrix. *Robotics and Automation, IEEE Transactions on*, 5(3):303–314, Jun 1989.

[11] Y. Masutani, F. Miyazaki, and S. Arimoto. Sensory feedback control for space manipulators. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 1346–1351 vol.3, May 1989.

[12] E. Papadopoulos and S. Dubowsky. On the nature of control algorithms for space manipulators. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 1102–1108 vol.2, May 1990.

[13] Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[14] Jeff Russkow and Oussama Khatib. A new control structure for free-flying space robots. In *i-SAIRAS symposium on Artificial Intelligence, Robotics and Automation in Space*, Toulouse 1992.

[15] S. Abiko, R. Lampariello, and G. Hirzinger. Impedance control for a free-floating robot in the grasping of a tumbling target with parameter uncertainty. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 1020–1025, Oct 2006.

[16] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *Int. J. Rob. Res.*, 26(1):23–39, January 2007.