

APPLICATION OF THE KUKA KUBE TEST-BED FOR THE HARDWARE-IN-THE-LOOP VALIDATION OF THE SPACE MANIPULATOR CONTROL SYSTEM

Mateusz WOJTUNIK¹, Piotr ŁUCZAK², Tomasz RYBUS¹,
Grzegorz GRANOSIK²

¹ Centrum Badań Kosmicznych Polskiej Akademii Nauk (CBK PAN),
Warsaw, Poland

² Lodz University of Technology, Institute of Automatic Control, Department
of Robot Control, Lodz, Poland

e-mails: mwojtunik@cbk.waw.pl, piotr.luczak.2@p.lodz.pl, trybus@cbk.waw.pl,
grzegorz.granosik@p.lodz.pl

ABSTRACT. The on-ground validation of control systems designed for manipulators working in orbit is very difficult due to the necessity of simulating the microgravity environment on Earth. In this paper, we present the possibilities of utilising the KUKA KUBE test-bed with industrial robots to experimentally verify space systems using hardware-in-the-loop tests. The fixed-base KUKA industrial robot is operated in gravitational environment, while the space system model plant is solved in real time parallel to on-ground experiment. The test-bed measurements are the input of the model plant, and the output of the model is treated as an input for the industrial robot actuation. In the performed experiment, the control system based on the Dynamic Jacobian is validated. The desired point that is reached by the manipulator's end-effector is constant in the simulated environment and moving with respect to the test-bed frame. The position of the space manipulator's end-effector is calculated by evaluating dynamics of the satellite in real-time model. The results show that the control system applied to the KUKA robot works correctly. The measurements from the torque sensors mounted in KUKA robot's joints are in accordance with the simulation results. This fact enhances the possibilities of gravity compensation, thus simulating microgravity environment on the test-bed.

Keywords: space manipulator, hardware-in-the-loop, experimental validation, control systems, orbital robotics, KUKA industrial robot

1. INTRODUCTION

In the near future, unmanned servicing satellites equipped with manipulators will be used for on-orbit servicing (OOS) of malfunctioned satellites (Tatsch et al., 2006) and for active debris removal (ADR) missions (Ellery, 2019). The capture operation, performed with the use of the manipulator, will be the most difficult part of the proposed missions. Control of the manipulator mounted on the free-floating satellite presents unique challenges as the motion of the manipulator affects the position and attitude of the satellite (Dubowsky and Papadopoulos, 1993). Various approaches are proposed for control of the satellite-manipulator system (Flores-Abad et al., 2014). The control law based on the Dynamic Jacobian allows to take into account



the influence of the manipulator motion on the state of the satellite (Yoshida and Umetani, 1990).

On-ground validation of control systems developed for manipulators working in orbit is also challenging because it requires simulation of microgravity conditions (Xu et al., 2011). Several different solutions can be used to simulate these conditions (Menon et al., 2007). Experiments performed during parabolic flights are expensive, and relatively high residual gravity acceleration is present during zero-G parabola (Menon et al., 2005). Drop towers provide very low residual gravity acceleration, but the size of the drop capsule is small and time of free fall is very short (Watanabe et al., 1998). Majority of on-ground experiments in the field of space robotics are performed on the planar air-bearing microgravity simulators (Rybus and Seweryn, 2016). However, the motion on such test-beds is limited to the horizontal plane. There are only few air-bearing test-beds that allow to recreate three-dimensional motion, but such experiments are very complicated (Saulnier et al., 2014). Thus, hardware-in-the-loop experiments are often used for validation of control systems developed for space manipulators (Wilde et al., 2019). In such an approach, the experiments are performed in the Earth's gravitational conditions, while the dynamic behaviour of the satellite-manipulator system is simulated using dynamic equations of motion.

There are two most common approaches for the hardware-in-the-loop experiments in the field of space robotics. In the first approach, the space manipulator is mounted on an industrial manipulator. The industrial manipulator is responsible for simulating the motion of the servicing satellite based on the mathematical model of the system. Such experiments were performed, among other things, on the on-orbit servicing simulator (OOS-SIM) facility operated by the German Aerospace Center (DLR) (Artigas et al., 2015; De Stefano et al., 2021) and on the platform-art facility operated by GMV (Colmenarejo et al., 2019; Seweryn et al., 2017). In the second approach, experiments are performed with one fixed-base manipulator (Li et al., 2016). The changes of the servicing satellite position and attitude are taken into account in simulation, but they are not physically introduced on the test-bed.

The goal of the presented work was to demonstrate the possibility of performing hardware-in-the-loop validation of the space manipulator control system using the KUKA KUBE test-bed operated by the Institute of Automatic Control at Lodz University of Technology (Granosik et al., 2016). This paper presents specific application of this unique test-bed for the purpose of performing experiments in the field of space robotics. In the considered scenario, the end-effector of the manipulator is controlled to reach the desired position and orientation defined in the inertial frame of reference. Dynamic model of the satellite-manipulator system is solved in real time to calculate changes of the satellite position and attitude caused by the manipulator motion. These changes are taken into account by the control system, and the desired position and orientation of the end-effector with respect to the base of the manipulator are updated.

The paper is organised as follows. The dynamic equations of the satellite-manipulator system and the control scheme based on the Dynamic Jacobian are presented in Section 2. KUKA KUBE test-bed is described in Section 3. The obtained results are presented in Section 4 and discussed in Section 5. Conclusions are given in Section 6.

2. DYNAMICS AND CONTROL OF FREE-FLOATING SPACE MANIPULATOR

We consider a manipulator (with n rotational joints) mounted on a free-floating satellite. The schematic view of the system is presented in Figure 1. The non-moving and non-rotating inertial frame is indicated as Π_{ine} .

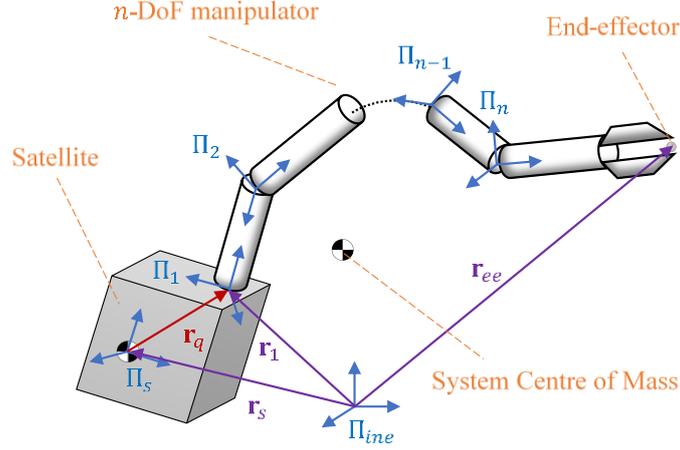


Figure 1. Schematic view of the n -DoF space manipulator

2.1. Dynamics

The dynamic equations of the system are based on the work performed by Seweryn and Banaszkiwicz (2008) and Rybus et al. (2017). The configuration of the system is described by the generalised coordinates vector as follows:

$$\mathbf{q}_p = [\mathbf{r}_s^T \quad \boldsymbol{\theta}_s^T \quad \boldsymbol{\theta}_m^T]^T \quad (1)$$

where $\mathbf{r}_s \in \mathbb{R}^3$ denotes the position vector of the satellite and $\boldsymbol{\theta}_s \in \mathbb{R}^3$ denotes the attitude of the satellite described by Euler angles in ZYX convention, whereas $\boldsymbol{\theta}_m = [\theta_{m_1} \quad \theta_{m_2} \quad \dots \quad \theta_{m_{n-1}} \quad \theta_{m_n}]^T \in \mathbb{R}^n$ is the vector of manipulator joint angular positions.

The vector of generalised velocities of the system is given by

$$\mathbf{q}_v = [\mathbf{v}_s^T \quad \boldsymbol{\omega}_s^T \quad \dot{\boldsymbol{\theta}}_m^T]^T \quad (2)$$

where $\mathbf{v}_s = \dot{\mathbf{r}}_s \in \mathbb{R}^3$ denotes the translational velocity of the satellite centre of mass (CoM), $\boldsymbol{\omega}_s = \mathbf{T}_\theta^{-1} \dot{\boldsymbol{\theta}}_s \in \mathbb{R}^3$ denotes the angular velocity of the satellite ($\mathbf{T}_\theta \in \mathbb{R}^{3 \times 3}$ is the transformation matrix that maps Euler angles time derivatives into the angular velocity vector) and $\dot{\boldsymbol{\theta}}_m \in \mathbb{R}^n$ is the vector of manipulator joint angular velocities.

The vector of generalised forces acting on the system is given by

$$\mathbf{Q} = [\mathbf{F}_s^T \quad \boldsymbol{\tau}_s^T \quad \boldsymbol{\tau}_m^T]^T \quad (3)$$

where $\mathbf{F}_s \in \mathbb{R}^3$ denotes the vector of forces acting on the satellite CoM, $\boldsymbol{\tau}_s \in \mathbb{R}^3$ denotes the vector of torques acting on the satellite and $\boldsymbol{\tau}_m \in \mathbb{R}^n$ denotes the joint driving (control) torques vector.

The general form of dynamic equation of motion for the free-floating space manipulator is given by

$$\begin{bmatrix} \mathbf{M}_s & \mathbf{M}_{sm} \\ \mathbf{M}_{sm}^T & \mathbf{M}_m \end{bmatrix} \dot{\mathbf{q}}_v + \begin{bmatrix} \mathbf{C}_s \\ \mathbf{C}_m \end{bmatrix} \mathbf{q}_v + \begin{bmatrix} \mathbf{G}_s \\ \mathbf{G}_m \end{bmatrix} = \mathbf{Q} \quad (4)$$

where $\mathbf{M}_s \in \mathbb{R}^{6 \times 6}$ is the mass matrix of the satellite, $\mathbf{M}_m \in \mathbb{R}^{n \times n}$ is the mass matrix of the manipulator, $\mathbf{M}_{sm} \in \mathbb{R}^{6 \times n}$ is the mass matrix that couples motion of the manipulator with motion of the satellite, $\mathbf{C}_s \in \mathbb{R}^{6 \times (6+n)}$ and $\mathbf{C}_m \in \mathbb{R}^{n \times (6+n)}$ are Coriolis matrices of the satellite and the manipulator, respectively, $\mathbf{G}_s = [\mathbf{F}_{G_s}^T \quad \mathbf{0}_{1 \times 3}]^T \in \mathbb{R}^6$, where $\mathbf{F}_{G_s}^T$ denotes the vector of gravitational force acting on the satellite CoM and $\mathbf{0}_{1 \times 3}$ denotes 1×3 vector filled with zeros, and $\mathbf{G}_m \in \mathbb{R}^n$ denotes the vector of reaction forces and torques acting on the manipulator joints

and induced by the gravitational force. The mass and Coriolis matrices of the system are defined by Rybus et al. (2022).

In case of the free-floating space manipulator, the state vector $\mathbf{x} = [\mathbf{q}_p^T \quad \mathbf{q}_v^T]^T$ has $12 + 2n$ components. The orbital motion of the system is not considered. The gravitational potential energy of the system is usually assumed to be equal to zero. The gravity gradient is neglected due to the relatively small size of the manipulator (Cavenago et al., 2019). In addition, in close proximity of the potential target satellite, the satellite Attitude and Orbit Control System (AOCS) is assumed to be turned off, hence $\mathbf{F}_s = \mathbf{0}_{3 \times 1}$ and $\boldsymbol{\tau}_s = \mathbf{0}_{3 \times 1}$. Such a system has conserved total momentum and angular momentum – the system is nonholonomic (Ratajczak and Tchoń, 2020). In case of the fixed-base manipulator working on Earth, the state vector $\mathbf{x} = [\boldsymbol{\theta}_m^T \quad \dot{\boldsymbol{\theta}}_m^T]^T$ has $2n$ components. The joint reaction torques \mathbf{G}_m caused by the gravitational force have significant influence on the system dynamics. The dynamic equation of motion (4) then takes much simpler form (Featherstone and Orin, 2008).

2.2. Control scheme

In the following section, the control scheme for the space manipulator is presented. The forward kinematics of the system on the velocity level is given by

$$\begin{bmatrix} \mathbf{v}_{ee} \\ \boldsymbol{\omega}_{ee} \end{bmatrix} = \mathbf{J}_s \begin{bmatrix} \mathbf{v}_s \\ \boldsymbol{\omega}_s \end{bmatrix} + \mathbf{J}_m \dot{\boldsymbol{\theta}}_m \quad (5)$$

where $\mathbf{J}_s \in \mathbb{R}^{6 \times 6}$ denotes the Jacobian of the satellite and $\mathbf{J}_m \in \mathbb{R}^{6 \times n}$ denotes standard Jacobian of the manipulator expressed in Π_{ine} (Rybus et al., 2017). In case of fixed-base manipulators, the right-hand side of equation (5) consists only of the second component as the velocity of the base (satellite) equals zero.

Let us assume that the initial total momentum and angular momentum of the system are equal to zero. The nonholonomic constraints are then formed by relations expressing the momentum and angular momentum as follows:

$$\begin{bmatrix} \mathbf{P} \\ \mathbf{L} \end{bmatrix} = \mathbf{H}_2 \begin{bmatrix} \mathbf{v}_s \\ \boldsymbol{\omega}_s \end{bmatrix} + \mathbf{H}_3 \dot{\boldsymbol{\theta}}_m = \mathbf{0}_{6 \times 1} \quad (6)$$

where matrices $\mathbf{H}_2 \in \mathbb{R}^{6 \times 6}$ and $\mathbf{H}_3 \in \mathbb{R}^{6 \times n}$ were presented by Rybus et al. (2017).

Equation (6) can be solved for the translational and angular velocity of the satellite as follows:

$$\begin{bmatrix} \mathbf{v}_s \\ \boldsymbol{\omega}_s \end{bmatrix} = -\mathbf{H}_2^{-1} \mathbf{H}_3 \dot{\boldsymbol{\theta}}_m \quad (7)$$

Substituting (7) into (5) yields

$$\begin{bmatrix} \mathbf{v}_{ee} \\ \boldsymbol{\omega}_{ee} \end{bmatrix} = (\mathbf{J}_m - \mathbf{J}_s \mathbf{H}_2^{-1} \mathbf{H}_3) \dot{\boldsymbol{\theta}}_m = \mathbf{J}_d \dot{\boldsymbol{\theta}}_m \quad (8)$$

where the matrix $\mathbf{J}_d \in \mathbb{R}^{6 \times n}$ is called the Dynamic Jacobian.

The Dynamic Jacobian is dependent on both configuration of the manipulator and the attitude of the satellite. Equation (8) can be solved for joint angular velocities as follows:

$$\dot{\boldsymbol{\theta}}_m = \mathbf{J}_d^\# \begin{bmatrix} \mathbf{v}_{ee} \\ \boldsymbol{\omega}_{ee} \end{bmatrix} \quad (9)$$

where $\#$ denotes the inverse of the square matrix or Moore–Penrose pseudoinverse in case of non-square matrix.

We consider controlling the end-effector (tip of the manipulator) to the desired position and orientation expressed in Π_{ine} . The desired velocity of the end-effector is evaluated by multiplying the control error by constant gain coefficients as follows:

$$\begin{bmatrix} \mathbf{v}_{ee\ des} \\ \boldsymbol{\omega}_{ee\ des} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_v(\mathbf{r}_{ee\ des} - \mathbf{r}_{ee\ act}) \\ \mathbf{K}_\omega \mathbf{T}_\Theta^{-1}(\boldsymbol{\Theta}_{ee\ des} - \boldsymbol{\Theta}_{ee\ act}) \end{bmatrix} \quad (10)$$

where subscript *des* denotes the desired values, subscript *act* denotes the actual, measured values, $\boldsymbol{\Theta}_{ee} \in \mathbb{R}^3$ is the orientation vector of the end-effector described by Euler angles in ZYX convention and $\mathbf{K}_v \in \mathbb{R}^{3 \times 3}$ and $\mathbf{K}_\omega \in \mathbb{R}^{3 \times 3}$ denote the diagonal positive-definite gain matrices for translational and angular velocities, respectively.

Substituting the end-effector velocity in (9) with the desired values expressed by the right-hand side of (10) yields the final form of the control law:

$$\dot{\boldsymbol{\theta}}_{m\ des} = \mathbf{J}_d^\# \begin{bmatrix} \mathbf{K}_v(\mathbf{r}_{ee\ des} - \mathbf{r}_{ee\ act}) \\ \mathbf{K}_\omega \mathbf{T}_\Theta^{-1}(\boldsymbol{\Theta}_{ee\ des} - \boldsymbol{\Theta}_{ee\ act}) \end{bmatrix} \quad (11)$$

The desired manipulator joint angular velocities are then sent to the manipulator Proportional-Integral-Derivative (PID) joint controllers (Seweryn et al., 2017). The presented approach forms a cascade control system, where the outer loop is a P controller of the position and orientation of the end-effector and the inner loop is a PID controller of the manipulator joint angular velocities.

In this paper, we use hardware-in-the-loop experiments to validate only the outer stage of the cascade controller represented by control law (11). Therefore, the controller equations on the joint level are omitted. This is caused by the fact that the joint controllers used in the experiments are directly applied from the KUKA robot software delivered by the manufacturer. In other words, the desired joint velocities obtained with (11) are directly sent to the KUKA industrial controller.

3. DESCRIPTION OF THE KUKA KUBE TEST-BED

3.1. Hardware description

The KUKA KUBE test-bed is presented in Figure 2. The test-bed is functionally divided into two subsystems: 1) executive part mounted inside cubic 2.5 m × 2.5 m × 2.5 m frame and 2) operator stand for ambidextrous control with both vision and force feedback. Test stand complies with International Standards Organisation/Technical Specification (ISO/TS 15066:2016: Robots and robotic devices – Collaborative robots norm). Executive elements consist of two KUKA LBR iiwa 14 R820 robotic arms equipped with universal tool mounts. Robots operate from side-mounted linear guides. KUKA LBR iiwa 14 R820 is a 7 Degrees of Freedom (DoF) articulated cobot capable of measuring joint torques both with and without gravity compensation. It is also capable of estimating forces acting on its end-effector. Denavit–Hartenberg (DH) parameters of the KUKA robot are presented in Table 1. KUBE test stand and its capabilities are well described by Granosik et al. (2016).

3.2. Software description

The developed software solution consists of two main subsystems: control loop and logging loop. Control loop was implemented in MATLAB using KUKA Sunrise Toolbox (KST) developed by Safeea and Neto (2018). KST consists of two parts: MATLAB client scripts operating on Personal Computer (PC) and MATLAB server java application operating on KUKA controller. Data exchange frequency achieved by Transmission Control Protocol (TCP)

connection provided by KST was sufficient to control the robot, but too slow for logging purposes as it reached only ~ 40 Hz. Logging was performed by using fast research interface (FRI) functionality provided by KUKA. FRI provides User Datagram Protocol (UDP) connection capable of sending full state of the robot with the frequency up to 1 kHz. In addition, another PC with Universal Serial Bus (USB) webcam was used to record the experiments. Both FRI logging and recording were triggered and terminated from the main control loop.

To implement control loop, dynamic parameters of LBR iiwa 14 R820 were needed. Parameters used were taken from the work performed by Stürz et al. (2017). Those parameters were also used by Safeea and Neto (2018) when developing KST.

Due to inability of FRI and KST of simultaneous operation on KUKA controller in as-is state and the low rate of KST connection, it was necessary to modify the controller side of both. In addition, the KST TCP connection speed was increased by concatenating existing KST commands for force, position and torque measurements into one command. Final architecture of connections is presented in Figure 3 (variables presented in the diagram are explained in sections 2 and 4). Logical diagram of the system is presented in Figure 4. The block ‘Perform main control loop iteration’ is discussed in Section 4.1.



Figure 2. KUKA KUBE test-bed

Table 1. KUKA LBR iiwa 14 R820 DH table

i	α_i [deg]	θ_{m_i} [deg]	a_i [m]	d_i [m]
1	90	θ_{m_1}	0	0.360
2	-90	θ_{m_2}	0	0
3	-90	θ_{m_3}	0	0.420
4	90	θ_{m_4}	0	0
5	90	θ_{m_5}	0	0.400
6	-90	θ_{m_6}	0	0
7	0	$\theta_{m_7} + 180$	0	0.126

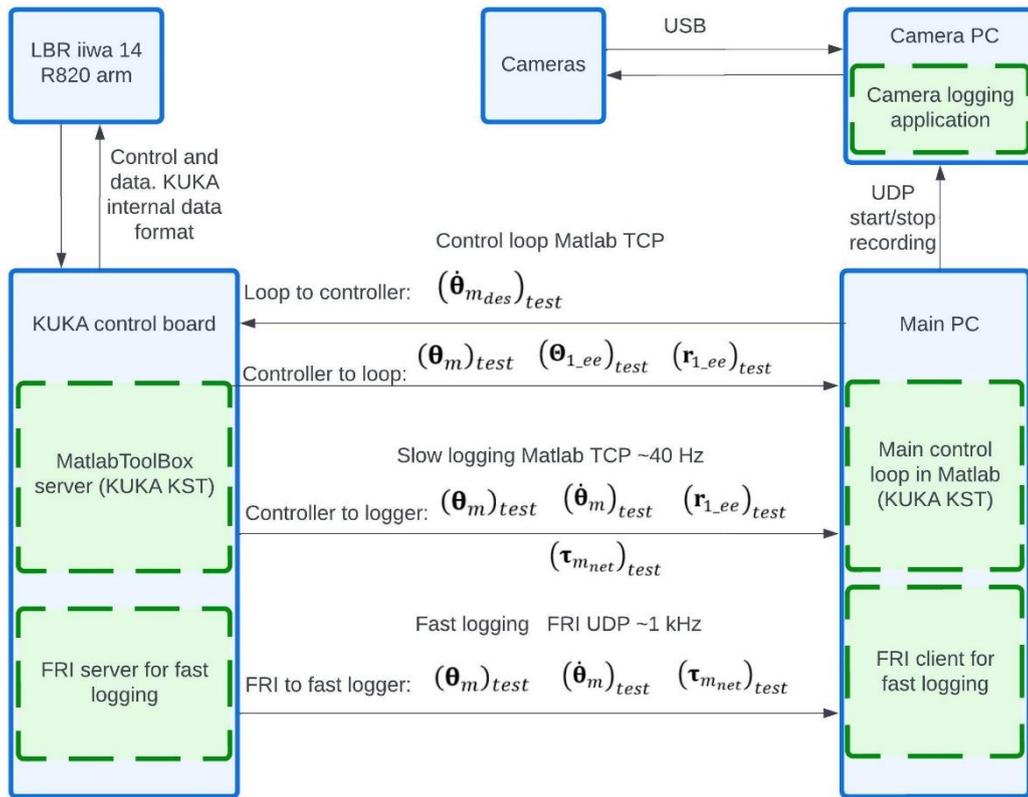


Figure 3. Communication diagram for KUKA KUBE test-bed hardware-in-the-loop experiments. Blue rectangles with continuous borders represent hardware components. Green, inner rectangles with dashed borders represent software components that were created or modified to perform the experiment. Brackets around parameters are added to improve subscript readability

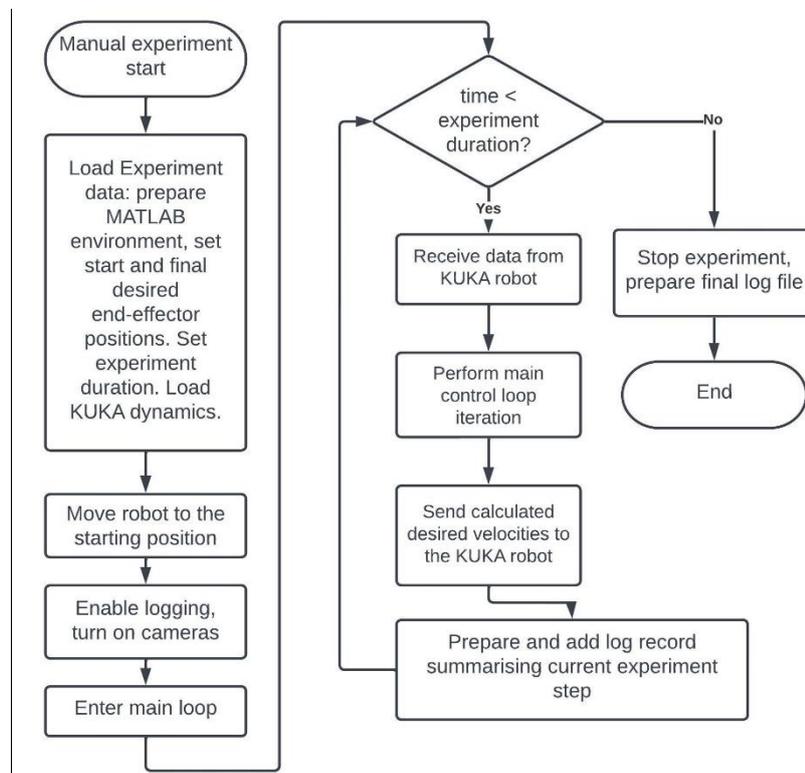


Figure 4. Logical diagram for KUKA KUBE test-bed hardware-in-the-loop experiments

4. RESULTS OF EXPERIMENTS

The KUKA KUBE test-bed was used for hardware-in-the-loop experiment that allows to validate the control system designed for the space manipulator. In addition, the results of the experiments were applied for comparison with the simulation tool, as well as to discuss the advantages and drawbacks of the hardware-in-the-loop test campaign.

4.1. Experiment description

The schematic representation of the performed experiment is presented in Figure 5. Subscript *sim* denotes variables related to the model plant calculated on the MATLAB level parallel to the experiment, while subscript *test* denotes variables related to the test-bed measurements. No subscript indicates that the variable is related to both model plant and test-bed. The KUKA manipulator is mounted on a fixed base and is operated in gravitational environment. The manipulator joint positions and velocities are measured on the test-bed with encoders. The model plant (marked as black and white system in Figure 5) is represented by the free-floating space manipulator, whose end-effector is controlled by the cascade control system based on the Dynamic Jacobian control law.

The actual current position of the end-effector is equal to the simulated position of the end-effector $(\mathbf{r}_{ee})_{sim} \in \mathbb{R}^3$ and is represented by the sum of the simulated position of the satellite $(\mathbf{r}_s)_{sim} \in \mathbb{R}^3$, the manipulator mounting point vector $\mathbf{r}_q \in \mathbb{R}^3$, which depends on geometrical system parameters, and the measured (on the test-bed) position of the end-effector with respect to the manipulator base $(\mathbf{r}_{1-ee})_{test} \in \mathbb{R}^3$:

$$\mathbf{r}_{ee_{act}} = (\mathbf{r}_{ee})_{sim} = (\mathbf{r}_s)_{sim} + \mathbf{r}_q + (\mathbf{r}_{1-ee})_{test} \quad (12)$$

The current position of the satellite $(\mathbf{r}_s)_{sim}$ is evaluated by calculating and integrating equation (7) in each step of the control system, while the measured position of the end-effector with respect to its base is calculated with forward kinematics based on test-bed joint position measurements $(\boldsymbol{\theta}_m)_{test}$.

The input of the control system is given by the constant desired position of the end-effector with respect to the inertial frame $\mathbf{r}_{ee_{des}}$. It is worth noting that even though these values are constant with respect to the inertial frame, the desired end-effector position with respect to the manipulator base $(\mathbf{r}_{1-ee_{des}})_{test}$ will change with time as a result of the simulated satellite motion $(\mathbf{r}_s)_{sim}$. Similar relations are obtained on the orientation level, which is omitted here.

Equation (12) shows the main idea of the hardware-in-the-loop experiment. We strive to validate the control system of the free-floating manipulator, whereas the on-ground test-beds do not allow us to introduce the motion of the satellite and the free-floating nature of the satellite-manipulator system. Therefore, we introduce a model which accounts for the free-floating nature of the system and we design the experiment so that the model is calculated in real time parallel to the experiment. As a result, the variables of interest for the control system (position and orientation of the end-effector with respect to the inertial frame) depend on both test-bed measurements $(\mathbf{r}_{1-ee})_{test}$ and results of the simulated model $(\mathbf{r}_s)_{sim}$. This concept allows us to combine the robot's hardware components with the free-floating nature of the system, which has high impact on the control system but is very difficult to be emulated on Earth.

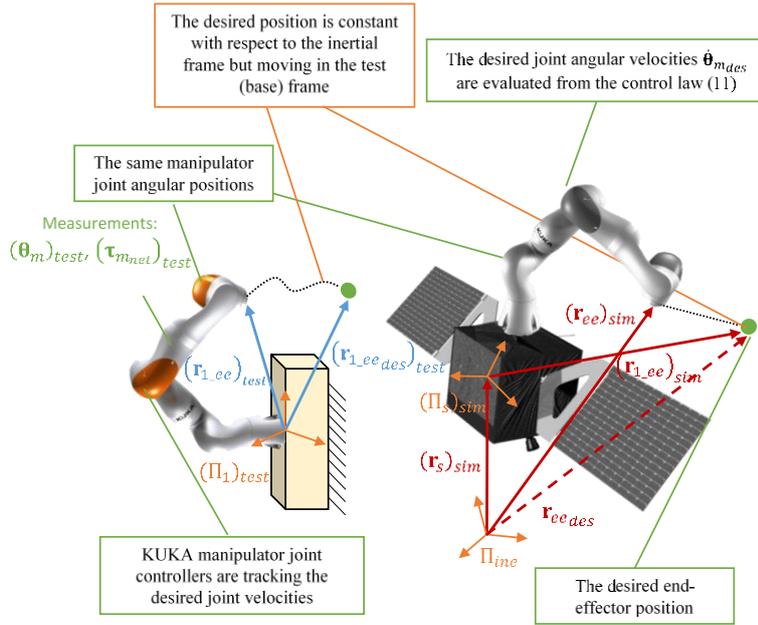


Figure 5. Schematic representation of the performed experiment

The calculation of the position and attitude of the satellite with the use of (7) as well as evaluation of the desired joint angular velocities using control law (11) are performed in the loop programmed with MATLAB KST. The program is running parallel to the experiment and its procedures are called out in each step of the control system. The desired joint velocities are then sent to the KUKA joint controllers that are able to control manipulator joints in the inner loop of the cascade control system.

To summarise, the plant model loop operations are presented as a diagram in Figure 6 and summarised as a list of operations below:

- 1) Receiving the information about the current KUKA manipulator joint angular positions $(\theta_m)_{test}$ as well as the current end-effector position $(\mathbf{r}_{1_ee})_{test}$ and orientation $(\Theta_{1_ee})_{test}$ with respect to the manipulator mounting point
- 2) Calculating $(\dot{\theta}_m)_{test}$ as a numerical derivative of $(\theta_m)_{test}$ and calculating the current satellite velocities $(\mathbf{v}_s)_{sim}$, $(\omega_s)_{sim}$ with equation (7)
- 3) Calculating the current satellite position $(\mathbf{r}_s)_{sim}$ and attitude $(\Theta_s)_{sim}$ by integrating the velocities $(\mathbf{v}_s)_{sim}$, $(\omega_s)_{sim}$ evaluated in step 2
- 4) Calculating the current end-effector position $\mathbf{r}_{ee_{act}}$ and orientation $\Theta_{ee_{act}}$ with respect to Π_{ine} using equation (12)
- 5) Calculating the desired manipulator joint velocities $(\dot{\theta}_{m_{des}})_{test}$ using the control law (11)
- 6) Sending the desired manipulator joint velocities $(\dot{\theta}_{m_{des}})_{test}$ to the KUKA manipulator joint controllers

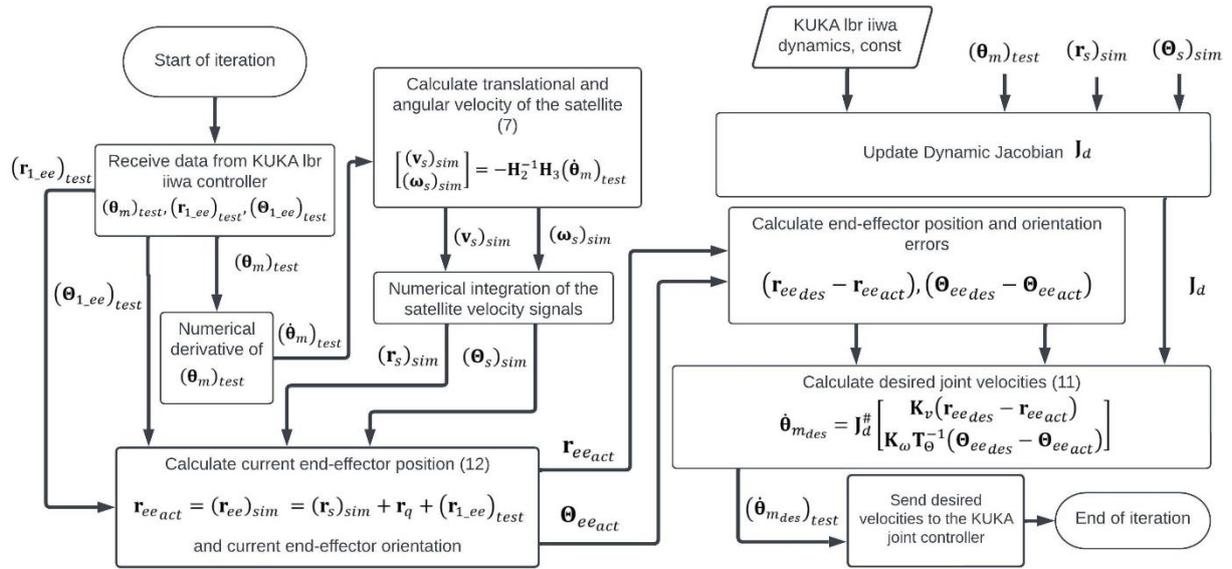


Figure 6. Diagrammatic description of a single control loop iteration

4.2. Control system validation

The parameters of the scenario are described herein. The mass of the satellite is set to 100 kg, whereas the moments of inertia around the main axes of the satellite are 2.83 kgm² in X axis, 6.08 kgm² in Y axis and 7.42 kgm² in Z axis. Products of inertia of the satellite are assumed to be zero. The satellite CoM is initially placed at the origin of Π_{ine} , and the initial attitude of the satellite coincides with the orientation of Π_{ine} . The initial configuration of the KUKA manipulator is $\theta_{m_1} = -26.25$ deg, $\theta_{m_2} = 65.96$ deg, $\theta_{m_3} = 0.72$ deg, $\theta_{m_4} = 67.57$ deg, $\theta_{m_5} = -63.15$ deg, $\theta_{m_6} = -2.56$ deg and $\theta_{m_7} = 140.79$ deg. The initial velocities of the system are set to zero. The initial end-effector position with respect to Π_{ine} is $\mathbf{r}_{ee} = [1.057 \text{ m} \ 0.174 \text{ m} \ 0.328 \text{ m}]^T$, while the initial orientation of the end-effector is $\boldsymbol{\theta}_{ee} = [-92.15 \text{ deg} \ -38.26 \text{ deg} \ -86.22 \text{ deg}]^T$. The desired end-effector position and orientation with respect to Π_{ine} are $\mathbf{r}_{ee_{des}} = [0.948 \text{ m} \ -0.225 \text{ m} \ 0.094 \text{ m}]^T$ and $\boldsymbol{\theta}_{ee_{des}} = [-88.38 \text{ deg} \ -37.27 \text{ deg} \ -56.99 \text{ deg}]^T$. The gain matrices \mathbf{K}_v and \mathbf{K}_ω are set to be identity matrices (each gain equal to 1). The experiment is set to last 20 s. The frames captured by the camera inside the KUKA KUBE test-bed showing the performed experiment are presented in Figure 7.

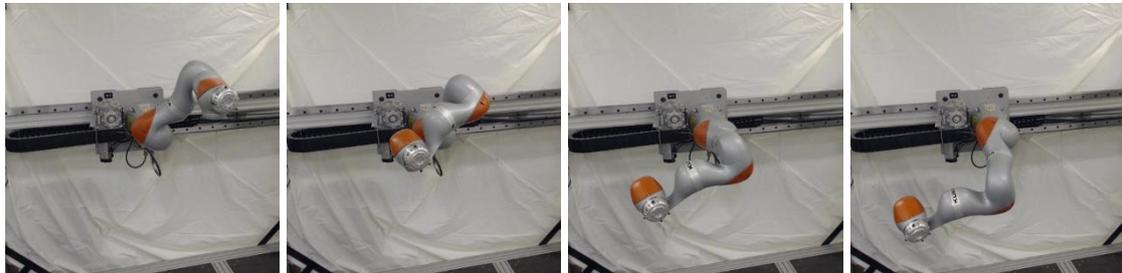


Figure 7. Frames captured by the camera mounted inside the KUKA KUBE test-bed that show the KUKA manipulator during experiments.

From left to right: $t = 0$ (initial configuration), $t = 6.6\text{s}$, $t = 13.3\text{ s}$, $t = 20\text{ s}$ (final configuration)

The results of the experiment are compared with the results of the simulation of the free-floating space manipulator performed after the test. The simulations performed after the experiment are not the same as the model plant used during the hardware-in-the-loop test. These simulations use the dynamic model (4) to simulate the free-floating manipulator motion using the experimental results as an input, whereas the MATLAB control loop used nonholonomic constraints (7) to evaluate the current satellite motion induced by the manipulator. It is also worth noting that the simulations performed to validate the behaviour of the system during the experiment were conducted in the open-loop mode – there was no control system implemented in the simulation. The reason is that the purpose of these simulations is to validate if the free-floating dynamics were correctly included in (12) during experiments.

The simulation is designed using the inverse dynamics approach (Basmadji et al., 2020). The signals of the manipulator joint angular positions measured in the experiment are used to calculate the driving torques using equation (4). Then, the evaluated driving torques are applied to solve the Simulink SimMechanics model of the KUKA manipulator mounted on the free-floating satellite. The simulations were performed with fixed time step equal to 0.005 s. The dynamics equations were integrated numerically using third-order Bogacki–Shampine formula. Due to the fact that calculating numerical derivatives of the measured joint positions is sensitive to encoder noise, the measurements are approximated with the fifth-order polynomials. The measured manipulator joint angles and their approximation used for the simulation are presented in Figure 8, while the approximation error is presented in Figure 9. The manipulator end-effector position and orientation with respect to Π_{ine} evaluated in the test and in the simulation are shown in Figure 10, while the differences between them are shown in Figure 11. Finally, the position and attitude of the satellite integrated in the MATLAB loop are compared with the simulation results in Figure 12, while the differences are presented in Figure 13.

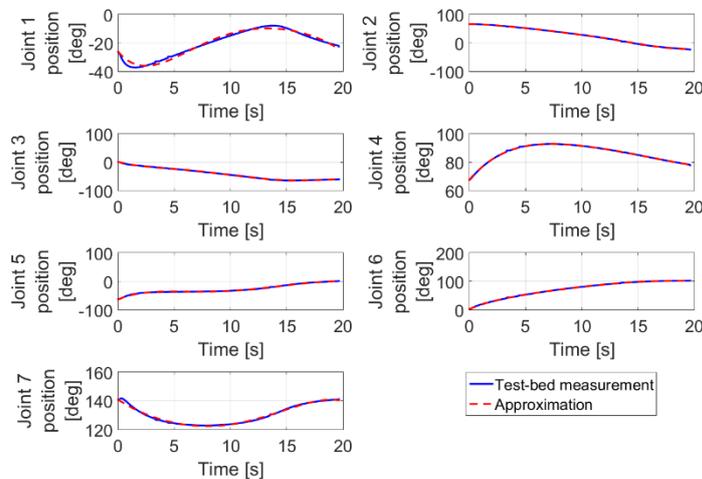


Figure 8. Manipulator joint angular positions measured in the experiment and signal approximation for the comparison simulation

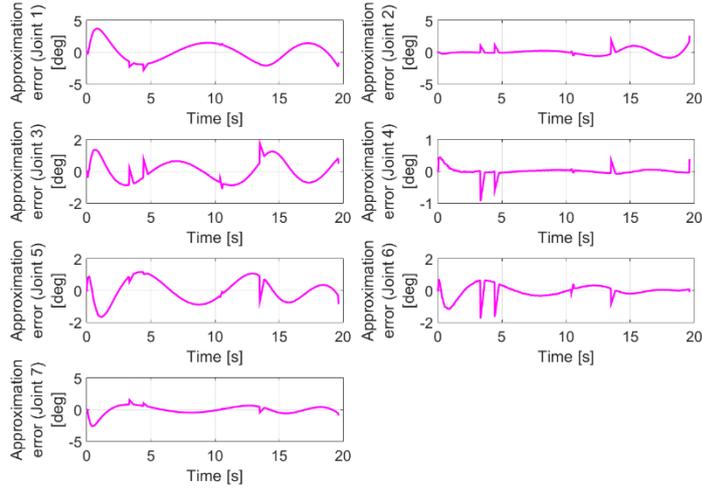


Figure 9. The approximation error for joint angular positions measured in the experiment

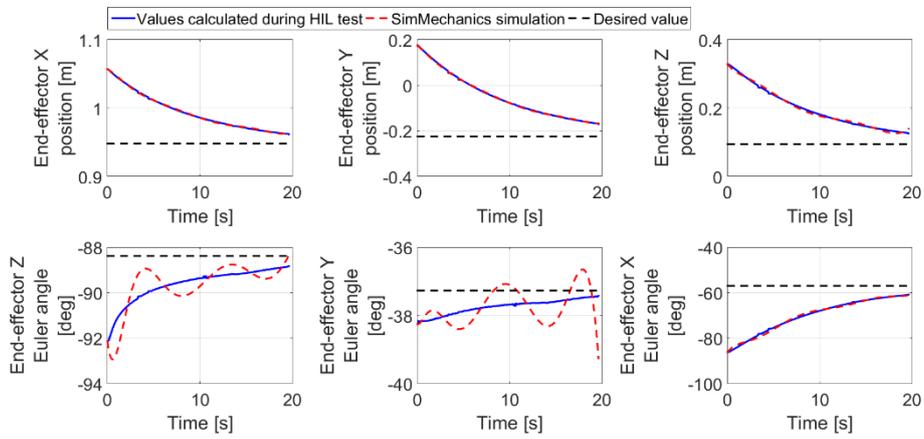


Figure 10. Manipulator end-effector position and orientation with respect to the simulated inertial frame Π_{ine} – comparison with the simulation results

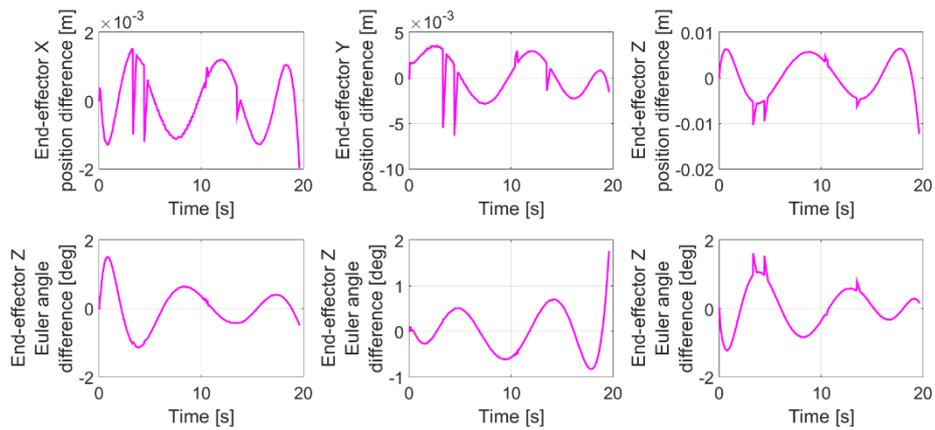


Figure 11. Difference in manipulator end-effector position and orientation with respect to the simulated inertial frame Π_{ine} between the experimental and simulation results

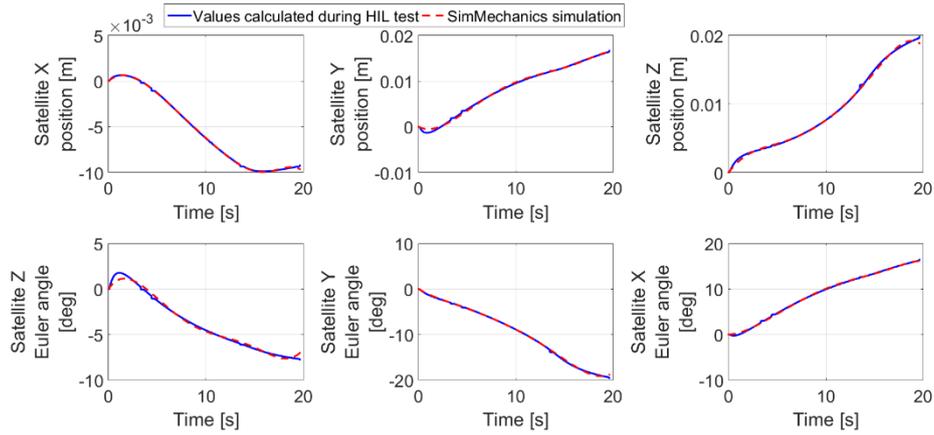


Figure 12. Satellite position and attitude with respect to the simulated inertial frame Π_{ine} – comparison with the simulation results

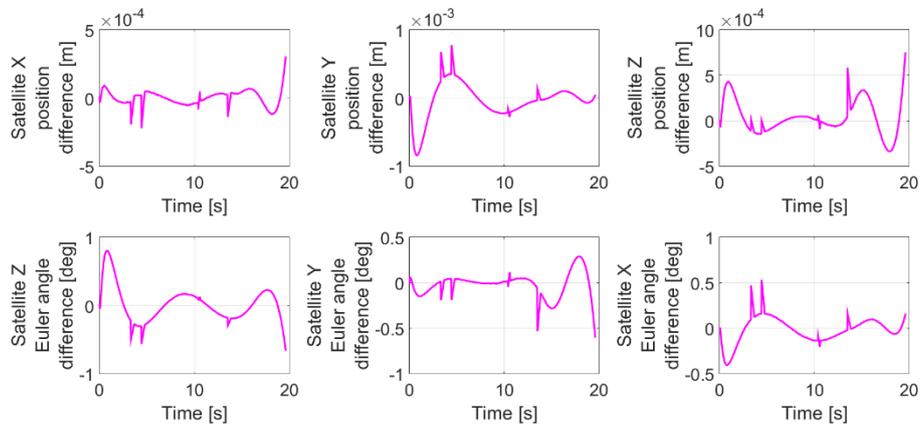


Figure 13. Difference in satellite position and attitude with respect to the simulated inertial frame Π_{ine} between experimental and simulation results

4.3. Joint torque analysis

The KUKA manipulator is equipped with torque sensors mounted in each manipulator joint. These measurements show the net torque $(\tau_{mnet})_{test}$ acting in the axis of rotation for each joint. To validate the correctness of the simulation tool, additional simulation using Simulink SimMechanics model of the fixed-base KUKA manipulator is performed. The time step and the numerical integration method used are the same as in previous simulations. Proper gravitational acceleration vector is set, and joint reaction torques are compared with the test-bed measurements. The time of evaluation for each iteration of the MATLAB loop during the experiment is presented in Figure 14. As an example, we chose to present the torque measured in the second joint compared with simulation results in Figure 15. It is observed that the simulation results are characterised by high torque peaks. The reasoning behind them will be discussed in the next section. To increase the readability of the torque signals, the Y axes limits for joint torque results in Figure 16 are set, so that the simulation torque peaks are not visible. The differences between the net torques measured during the experiments and obtained in the simulations are presented in Figure 17.

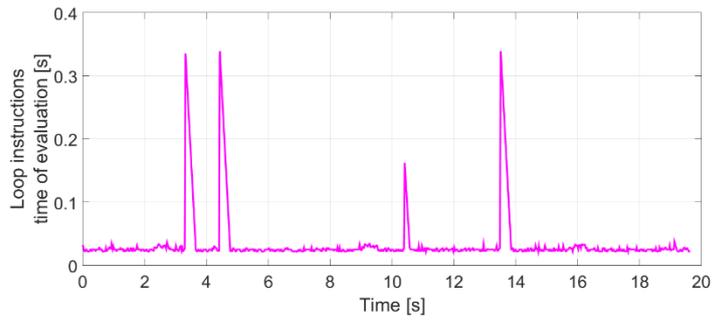


Figure 14. Time of evaluation for each iteration of the MATLAB loop

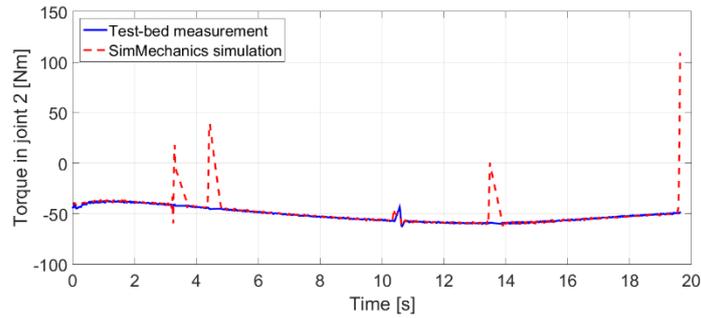


Figure 15. The net torque acting on the second manipulator joint – comparison with the simulation

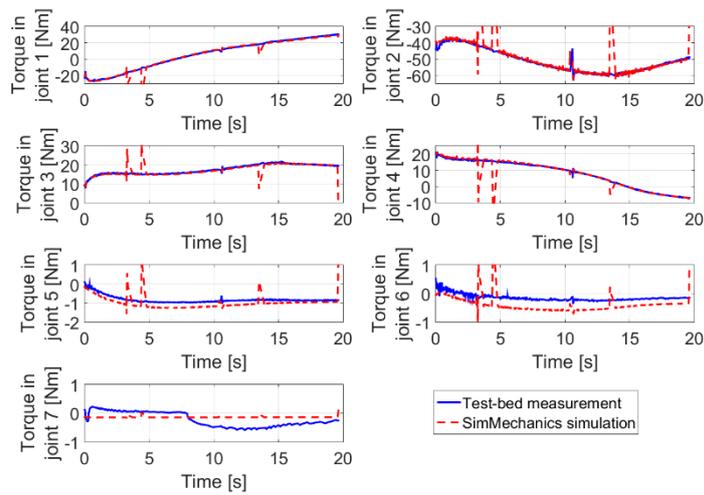


Figure 16. The net torque acting on each manipulator joint – comparison with the simulation

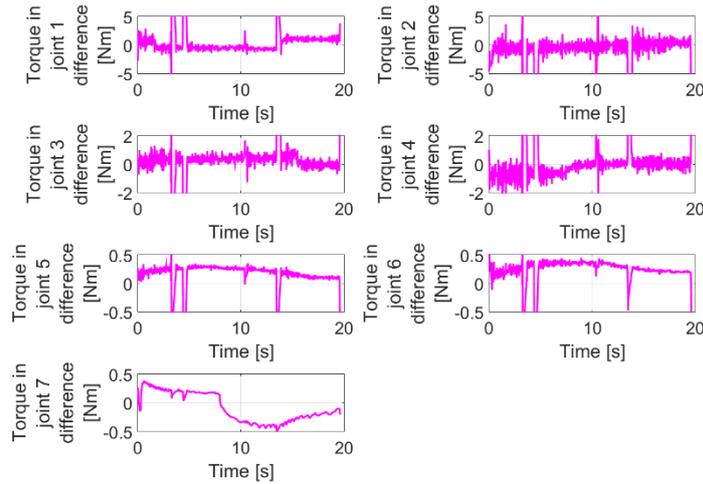


Figure 17. The difference in net torque acting on each manipulator joint between the experimental and simulation results

5. DISCUSSION

The performed experiment validated that the first stage of the cascade control system based on the Dynamic Jacobian control law allows to properly control the manipulator’s end-effector in the inertial frame. The position and orientation of the end-effector with respect to Π_{ine} are converging to the desired values. However, the control error is non-zero at the end of the experiment. The designed algorithm is too slow to achieve the desired values in 20 s. This issue can be solved by increasing the outer loop gain coefficients or by replacing the outer loop P controller with PI or PID controller.

The test results are different than the signals observed in the simulation. The differences between the end-effector position obtained during the experiment and in the simulation performed after the test have the magnitude of several millimetres, while the orientation differences are equal to about 1 deg –2 deg (Figure 11). This is caused by the fact that the manipulator joint position signals (that are the input of the inverse dynamics algorithm) had to be approximated to avoid peaks in numerical derivative of the signals. The differences between the approximated signals and test-bed measurements were equal to up to 3 deg (Figure 9). These differences are influencing especially the end-effector orientation components that did not change significantly during the experiment (Z and Y angles as seen in Figure 10). In addition, differences between the results from the test and the simulation can be caused by time delays in the communication between the MATLAB solving the model plant and KUKA control board responsible for sending test-bed measurements. The phase delay between the model plant results and the test-bed measurements is one of the main challenges in hardware-in-the-loop experiments.

The KUKA joint controllers are significantly faster than the plant model. Therefore, the practical period of the control loop mainly depends on the time of evaluation of the MATLAB model plant. As it was explained in Section 3.2, the achieved frequency was approximately equal to 40 Hz. However, as it can be seen in Figure 14, the time of evaluation for each iteration of the MATLAB loop was changing with time, especially at some time occurrences, the time of evaluation was significantly higher than during the rest of the experiment (this issue is discussed in the next paragraph). Therefore, the practical frequency of the control loop depends on the latency between the model and the real system, which changes with time.

The simulation results for the net torques acting in manipulator joints (for simulation with gravitational environment) are characterised by torque peaks. These are caused by the fact that the joint position signals change drastically at times when the evaluation time of the MATLAB loop is significantly longer (peaks of time of evaluation and torque simulation results occur at the same time stamps). Due to longer time of evaluation of the MATLAB control loop, there was a longer time period between two subsequent control commands, and thus, the KUKA robot moved very quickly with regard to newly sent command. On the other hand, the torque peak was not logged in the test-bed measurements due to the fact that the MATLAB control loop was still performing its calculations. It is worth noting that such peaks might be impacted by the algorithms applied within the MATLAB control loop – the pseudo-inverse of the Dynamic Jacobian as well as other model matrices require relatively long period of time to be calculated. This has large influence on the latency within the experiment.

It can be seen in Figure 14 that there were several occurrences when the time of evaluation of the model plant was longer. It is caused by the fact that at that particular time, the communication between MATLAB and KUKA manipulator controller was longer or the evaluation of the Dynamic Jacobian pseudo-inverse took more time than usual. The simulation performed after the experiments used the measured joint position signals. Fast changes of the joint position occurring at time stamps related to longer latency caused the joint acceleration (calculated numerically) to have high values at these occurrences. As the simulation worked with higher frequency, this is the reason why the torque peaks were obtained in the simulation. As stated before, these peaks were not measured during the experiment due to too high latency caused by longer time evaluation for the MATLAB code. Although the simulation step time influences the accuracy of the results in general, here the reason behind the peaks is directly related to the latency of the MATLAB control loop. Such an effect can be improved in the future by optimising the frequency of the MATLAB control loop or improving the frequency of data logging.

Apart from the events of longer time of evaluation, the average model plant loop time of evaluation was about 0.025 s, which is equivalent to the frequency of 40 Hz stated before as the average practical frequency of the control loop. In addition, the results observed in Figure 16 show that the test-bed measurements are close to simulation results. The differences between the results are equal to several Nm in case of joints 1–4 and tenths of Nm in case of joints 5–7 (Figure 17). The differences might be caused by additional aspects that are not included in the simulation tool, for example, torques induced by friction or gear stiffness.

The test-bed did not work in the real-time regime partially due to the available software tools and non-real-time nature of the preliminary experiments that constituted the software base used for tests described in the paper. Another important reason was achievable control loop frequency including communication with the robot. Initially achieved control loop frequency was very limited. Limitations stemmed mostly from the communication delays and partially from numerical inefficiencies. Because the main purpose of the test-bed was to collect timestamped data and the robot did not use tools and did not perform any specific manipulation task, the decision has been made to sacrifice real-time regime to achieve higher control loop and measurement frequency. Since the measured data are timestamped, we are aware that the system does not operate in the real-time regime.

It is worth noting that the main purpose of the experiment was to prove the concept of hardware-in-the-loop simulations and present that combining the test-bed measurements with simulated satellite position in real time using equation (12) allows to validate the control system with the use of industrial robot. As the main goal of the paper was to examine the potential of the KUKA KUBE test-bed in terms of orbital robotics experiments, a single test performed in the scope of

this study is fully representative. On the other hand, in future work, there must be some discussion concerning performing multiple experiments with various initial conditions and disturbances.

The obtained results indicate that it is possible to manually perform gravity compensation on the test-bed. Thus, it is possible to emulate the KUKA manipulator working under microgravity environment. In the future, there might be a possibility of performing hardware-in-the-loop tests of proper capture manoeuvre of the space manipulator. Comparison between the measured net torque acting on the KUKA manipulator joints and results from the simulation of the fixed-base manipulator working under gravitational environment presents a great potential of the KUKA KUBE test-bed. This is very important as experimental validation of space manipulators' control systems is very difficult due to problems with emulating microgravity environment.

Parameters of the robot used in the implementation were taken from identification done by Stürz et al. (2017). Work performed by Xu et al. (2020) was also focused on deriving dynamical parameters of the LBR iiwa 14 R820 arm, but the values obtained and reported by authors were different. This led to the decision to start deriving own symbolic regressor based on the method presented by Gabicchini et al. (2020). Works are performed in MATLAB Symbolic Toolbox and are still ongoing.

6. CONCLUSIONS

In this paper, we presented the hardware-in-the-loop experiments validating space manipulator control system using KUKA KUBE test-bed. The designed control law is based on the cascade control system. The control law utilises the Dynamic Jacobian to transform the end-effector position and orientation errors to the desired manipulator joint angular velocities. These velocities are then sent to the KUKA joint controllers used in the test-bed.

The validation experiment is based on the hardware-in-the-loop test. Fixed-base KUKA industrial manipulator working under gravitational environment controls the desired manipulator joint angular velocities and provides measurements of the joint angular positions, end-effector position and orientation with respect to base as well as net torques acting in the axis of rotation of each joint (measured by torque sensors). The desired manipulator joint velocities are evaluated in real time in the MATLAB loop that solves the model plant of the free-floating space manipulator. Nonholonomic constraints equations are used to evaluate the actual position and attitude of the satellite. As a result, the desired position and orientation of the end-effector are constant in the simulated inertial frame, but time variant in the base frame related to the test-bed. The results present great potential of the KUKA KUBE test-bed in terms of hardware-in-the-loop experiments for space systems validation. It is very difficult to emulate the microgravity environment on Earth. Thus, hardware-in-the-loop tests are a great opportunity to overcome this problem. The results show that there is a compatibility between the test-bed measurements in the performed experiment and simulation results from the SimMechanics model of the free-floating space manipulator. In addition, simulations of the KUKA manipulator working under gravitational environment are compatible with torque sensor measurements, which indicates that precise gravity compensation is possible. In the future, there is a possibility to perform hardware-in-the-loop experiments of the capture manoeuvre with the use of the KUKA KUBE test-bed and space manipulator model plant solved in real time parallel to the motion of the robot.

Acknowledgements. This paper was partially supported by the Polish National Centre for Research and Development project no. LIDER/19/0117/L-10/18/NCBR/2019.

REFERENCES

- Artigas, J., De Stefano, M., Rackl, W., Lampariello, R., Brunner, B., Bertleff, W., Burger, R., Porges, O., Giordano, A., Borst, C., Albu-Schaeffer, A. (2015) The OOS-SIM: An on-ground simulation facility for on-orbit servicing robotic operations, *Proc. IEEE International Conference on Robotics and Automation (ICRA'2015)*, Seattle, WA, USA.
- Basmadji, F.L., Seweryn, K., Sasiadek, J.Z. (2020) Space robot motion planning in the presence of linear and angular momenta, *Multibody System Dynamics*, Vol. 50, 71-96.
- Cavenago, F., Giordano, A.M., Massari, M. (2019) Contact force observer for space robots, *Proc. 58th IEEE Conference on Decision and Control (CDC'2019)*, Nice, France.
- Colmenarejo, P., Graziano, M., Novelli, G., Mora, D., Serra, P., Tomassini, A., Seweryn, K., Prisco, G., Gil Fernandez, J. (2019) On ground validation of debris removal technologies, *Acta Astronautica*, Vol. 158, 206-219.
- De Stefano, M., Mishra, H., Giordano, A.M., Lampariello, R., Ott, C. (2021) A Relative Dynamics Formulation for Hardware-in-the-Loop Simulation of On-Orbit Robotic Missions, *IEEE Robotics and Automation Letters*, Vol. 6, No. 2, 3569-3576.
- Dubowsky, S., Papadopoulos, E. (1993) The kinematics, dynamics, and control of free-flying and free-floating space robotic systems, *IEEE Transactions on robotics and automation*, Vol. 9, No. 5, 531-543.
- Ellery, A. (2019) Tutorial review on space manipulators for space debris mitigation, *Robotics*, Vol. 8, No. 2, 34.
- Featherstone, R., Orin, D. E. (2008) Dynamics, In: Siciliano, B., Khatib, O. (eds.), *Handbook of Robotics*, Chapter 2, Springer-Verlag, Berlin Heidelberg, 35-66.
- Flores-Abad, A., Ma, O., Pham, K., Ulrich, S. (2014) A review of space robotics technologies for on-orbit servicing, *Progress in aerospace sciences*, Vol. 68, 1-26.
- Gabiccini, M., Bracci, A., De Carli, D., Fredianelli, M., Bicchi, A. (2009) Explicit Lagrangian Formulation of the Dynamic Regressors for Serial Manipulators, *Proc. XIX Aimeta Congress*, Ancona, Italy.
- Granosik, G., Zubrycki, I., Soghatyan, T., Zarychta, D., Gawryszewski, M. (2016) KUBE - platforma robotyczna dla badań naukowych i prac wdrożeniowych [in polish: KUBE - Robotic platform for research and implementation], In: Tchoń K. and Zieliński C. (eds.), *Prace Naukowe Politechniki Warszawskiej: Elektronika*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, Vol. 195, No. 1, 223-234.
- Li, Z., Wang, B., Liu, H. (2016) Target capturing control for space robots with unknown mass properties: a self-tuning method based on gyros and cameras, *Sensors*, Vol. 16, No. 9, 1383.
- Menon, C., Aboudan, A., Cocuzza, S., Bulgarelli, A., Angrilli, F. (2005) Free-flying robot tested on parabolic flights: Kinematic control, *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 623-630.
- Menon, C., Busolo, S., Cocuzza, S., Aboudan, A., Bulgarelli, A., Bettanini, C., Marchesi, M., Angrilli, F. (2007) Issues and solutions for testing free-flying robots, *Acta Astronautica*, Vol. 60, No. 12, 957-965.
- Ratajczak, J., Tchoń, K. (2020) Normal forms and singularities of non-holonomic robotic systems: a study of free-floating space robots, *Systems & Control Letters*, Vol. 138, 104661.

- Rybus, T., Seweryn, K. (2016) Planar air-bearing microgravity simulators: review of applications, existing solutions and design parameters, *Acta Astronautica*, Vol. 120, 239-259.
- Rybus, T., Seweryn, K., Szaśiadek, J.Z. (2017) Control system for the free-floating space manipulator based on Nonlinear Model Predictive Control (NMPC), *Journal of Intelligent & Robotic Systems*, Vol. 85, No. 3, 491-509.
- Rybus, T., Wojtunik, M., Basmadji, F.L. (2022) Optimal collision-free path planning of a free-floating space robot using spline-based trajectories, *Acta Astronautica*, Vol. 190, 395-408.
- Safeea, M., Neto, P. (2018) KUKA Sunrise Toolbox: Interfacing Collaborative Robots With MATLAB, *IEEE Robotics & Automation Magazine*, Vol. 26, No. 1, 91-96.
- Saulnier, K., Pérez, D., Huang, R.C., Gallardo, D., Tilton, G., Bevilacqua, R. (2014) A six-degree-of-freedom hardware-in-the-loop simulator for small spacecraft, *Acta Astronautica*, Vol. 105, No. 2, 444-462.
- Seweryn, K., Banaszkiwicz, M. (2008) Optimization of the trajectory of a general free-flying manipulator during the rendezvous maneuver, *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit (AIAA-GNC'2008)*, Honolulu, HI, USA.
- Seweryn, K., Baran, J., Barciński, T., Colmenarejo, P., Łoś, A., Kowalski, T., Mollinedo, L., Mora, D., Musiał, J., Novelli, G., Oleś, J., Ososińska, K., Paśko, P., Prisco, G., Rybus, T., Serra, P., Wawrzaszek, R., Gil Fernandes, J. (2017) The prototype of space manipulator WMS LEMUR dedicated to capture tumbling satellites in on-orbit environment, *Proc. 11th International Workshop on Robot Motion and Control (RoMoCo'2007)*, Wąsowo, Poland.
- Stürz, Y.R., Affolter, L.M., Smith, R.S. (2017) Parameter Identification of the KUKA LBR iiwa Robot Including Constraints on Physical Feasibility, *IFAC-PapersOnLine*, Vol. 50, No. 1, 6863-6868.
- Tatsch, A., Fitz-Coy, N., Gladun, S. (2006) On-orbit servicing: A brief survey, *Proc. Performance Metrics for Intelligent Systems Workshop (PerMIS'2006)*, Gaithersburg, MD, USA.
- Watanabe, Y., Araki, K., Nakamura, Y. (1998) Microgravity experiments for a visual feedback control of a space robot capturing a target, *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, Canada, 1998.
- Wilde, M., Clark, C., Romano, M. (2019) Historical survey of kinematic and dynamic spacecraft simulators for laboratory experimentation of on-orbit proximity maneuvers, *Progress in Aerospace Sciences*, Vol. 110, 100552.
- Xu, T., Fan J., Chen, Y., Ng, X., Ang, M. H., Fang, Q., Zhu, Y., Zhao, J. (2020) Dynamic Identification of the KUKA LBR iiwa Robot With Retrieval of Physical Parameters Using Global Optimization, *IEEE Access*, Vol. 8, 108018-108031.
- Xu, W., Liang, B., Xu, Y. (2011) Survey of modeling, planning, and ground verification of space robotic systems, *Acta Astronautica*, Vol. 68, No. 11-12, 1629-1649.
- Yoshida, K., Umetani, Y. (1990) Control of space free-flying robot, *Proc. 29th IEEE Conference on Decision and Control (CDC'1990)*, Honolulu, HI, USA.

Received: 2022-12-14

Reviewed: 2023-04-01 (undisclosed name); 2023-04-23 (T. Uhl)

Accepted: 2023-08-17