IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), Daejeon, Korea, October 2016

# Robotic Simulation of On Orbit Servicing Including Hard Impacts

Friedrich Lange, Gerhard Grunwald and Alin Albu-Schäffer

*Abstract*— **Industrial robots are often used for the simulation of satellites during on orbit servicing. In order to cover also the docking phase, both robots are equipped with force-torque sensors, and the measured forces and torques are taken to compute the desired motion of the position controlled robots. Since the system dynamics of robots and of free floating bodies obviously differ, for each robot we distinguish between the really executed and the assumed satellite motion. The difference between the two motions is used to adapt the measured forces in such a way that they correspond to the satellite's trajectory. In this way the docking procedure can be visualized by two robots which closely follow the satellites' trajectories. Stability of the robot control is not compromised even if the dynamics of the satellites and the robots are totally different. Simulation results verify the approach.**

## I. Introduction

In consideration of the huge number of inactive or malfunctioning satellites in space, on orbit servicing is a crucial technique for satellite removal or repair. However, especially the critical phase of approaching and docking has to be simulated before a mission in space is launched. For this purpose, several robotic setups have been built [1], [2], [3], [4], [5], [6], [7], [8], [9], using standard industrial robots with 6 degrees of freedom (dof) for each satellite. This is denoted as *Hardware-in-the-loop* (HIL) simulation.

However, most of the setups exclude the direct contact between the robots since this might cause stability problems. Typically, due to communication delays and to limited deceleration abilities, the impact forces are measured too long and thus the computed accelerations are integrated too long, resulting in a too high reaction speed. A possible solution is a compliant mounting of the docking tool at the servicer satellite, as in the new European Proximity Operations Simulator (EPOS) facility [6]. Besides the advantages for HIL, such a shock attenuation system facilitates the docking [10].

Another idea to solve contact instability is to dismiss the strict relation between the forces that are measured at the robot and the accelerations resulting from them. [11], [12] design a passivity controller which prevents increasing energy during the impact. More precisely, the measured force [11] or the computed velocity [12] is modified in such a way that energy is dissipated and finally stability is preserved.

In this paper we present another method which besides the stabilization property allows to map the assumed contact dynamics of the colliding satellites to the simulation setup. This means that a damping coefficient or a coefficient of restitution can be assigned to the simulation.

The paper is organized as follows: Next, the satellite dynamics are introduced, which have to be simulated by the robotic setup. Then, Sect. III investigates the robot dynamics. Sect. IV finally presents the approach that produces the desired dynamics in spite of the robot characteristics. This method is simulated in Sect. V.

## II. Satellite Dynamics

In this paper we assume that the satellites can be treated as rigid bodies.

### A. Motion of Free Floating Bodies

The well known equations of a free floating body are (see e.g. [10])

$$M\ddot{\mathbf{x}} = \mathbf{f} \tag{1}$$

$$\mathbf{I}\dot{\boldsymbol{\omega}} = \mathbf{m} - \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \tag{2}$$

where $\mathbf{x} \in \mathbb{R}^3$ is the Cartesian position of the body's center[1] in world coordinates. $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity with respect to the center. $\mathbf{f} \in \mathbb{R}^3$ and $\mathbf{m} \in \mathbb{R}^3$ represent the contact forces and torques that are applied to the center of the body. For a free floating body they can be assumed to be zero. $M$ and $\mathbf{I} \in \mathbb{R}^{3\times3}$ are the mass and the moment of inertia of the body with respect to its center.

This means that without contact the center of the body will remain steady, whereas a tumbling rotation may occur, once it is excited.

### B. Contact Forces Between Free Floating Bodies

Impacts are commonly analyzed using equations of the momentum and the kinetic energy. Instead, here we look at the force which can be measured when simulating the satellites' motion in experiments with robots. In this way, an impact is completely defined by (1) and (2), even if it is not purely elastic.

The force $\mathbf{f}$ and the torque $\mathbf{m}$ at the center of gravity are computed from the sensed force $\mathbf{f}_s \in \mathbb{R}^3$ and the sensed torque $\mathbf{m}_s \in \mathbb{R}^3$ by

$$\begin{aligned} \mathbf{f} &= \mathbf{R}_s\mathbf{f}_s - M\mathbf{g} && (3)\\ \mathbf{m} &= \mathbf{R}_s\mathbf{m}_s + (\mathbf{x}_s - \mathbf{x}) \times (\mathbf{R}_s\mathbf{f}_s), && (4)\end{aligned}$$

where $\mathbf{x}_s \in \mathbb{R}^3$ is the position of the force-torque sensor that measures $\mathbf{f}_s$ and $\mathbf{m}_s$. $\mathbf{R}_s \in \mathbb{R}^{3\times3}$ is the rotation matrix that converts the sensor values to the world coordinate system. $\mathbf{g}$ is the gravitational acceleration.

The authors are with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany. Alin Albu-Schäffer is also with the Department of Informatics, Technical University Munich, 85748 Garching, Germany. {friedrich.lange, gerhard.grunwald, alin.albu-schaeffer}@dlr.de

[1] The center of a body is understood as its center of gravity.

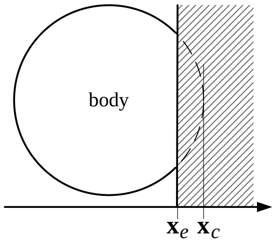
body
$\mathbf{x}_e$ $\mathbf{x}_c$
body 1
body 2
$\mathbf{x}_{c_2}$ $\mathbf{x}_e$ $\mathbf{x}_{c_1}$

Fig. 1. Notation for the contact of a body with a *wall* or another body.

Another fundamental equation is the contact model, e.g.

$$\mathbf{f} = \mathbf{K}_e(\mathbf{x}_e - \mathbf{x}_c) + \mathbf{D}_e(\dot{\mathbf{x}}_e - \dot{\mathbf{x}}_c), \tag{5}$$

which represents the relation between the Cartesian position of the contact point $\mathbf{x}_c \in \mathbb{R}^3$ of a body and the position $\mathbf{x}_e \in \mathbb{R}^3$ of the environment, i.e., another body. In (5) the contact to the environment is modeled as a linear spring-damper system with a stiffness matrix $\mathbf{K}_e \in \mathbb{R}^{3\times3}$ and a damping matrix $\mathbf{D}_e \in \mathbb{R}^{3\times3}$. This model, of course, represents only linear collisions. [13] gives some examples of more general contact models that represent nonlinear partially elastic impacts. Fig. 1 shows the definition of $\mathbf{x}_e$ and $\mathbf{x}_c$ for a body in contact with a *wall* or another body, where *wall* stands for an object of infinite inertia and infinite stiffness. This simpler case is explained first. Otherwise, (5) is valid for each body, possibly with differing parameters.

Concerning the orientation,

$$\mathbf{m} = (\mathbf{x}_c - \mathbf{x}) \times \mathbf{f} \tag{6}$$

gives the input to (2). This assumes implicitly that no torque is exerted at the contact point.

Equations (5) and (6) describe the forces and torques from the intersection $\mathbf{x}_c - \mathbf{x}_e$ of the two bodies. This will be used in Sect. IV.

When two bodies are approaching, (5) and (1) result in an immediate repulsion, once they intersect. Thus the contact will be quite short. After an elastic collision with a *wall*, i.e., $\mathbf{x}_e = const$, the body will retain its original velocity, only with changed direction. For non direct or non central impacts, (6) and (2) will cause a tumbling motion after the collision.

## III. Robot Dynamics

For the simulation of the satellites' dynamics we use two KUKA industrial robots and their industrial controllers KRC4. Thus each robot is position-controlled by setpoints for the motor angles. The actual motor angles $\boldsymbol{\theta} \in \mathbb{R}^6$ are measured, as well as the forces $\mathbf{f}_s$ and torques $\mathbf{m}_s$.

### A. Free Motion

Because of the motor dynamics, the real motor positions $\boldsymbol{\theta}$ are delayed with respect to their setpoints. But this relation is not of interest here, since the motor positions of industrial robots are measured.

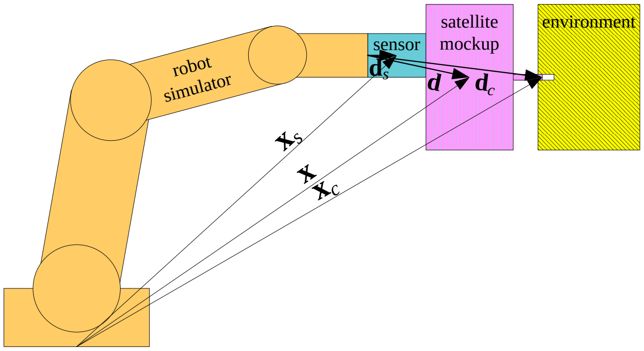

Fig. 2. Model of a satellite which is held by a robot.

However, the position of the robot arm is not completely given by the motor positions. In addition, deflections in the gears result in deviations of the axis positions. In this way the robot axes follow the motor positions delayed. This can be expressed by the dynamic equations of the robot

$$\mathbf{M}_r\ddot{\mathbf{q}} + \mathbf{C}_r\dot{\mathbf{q}} + \mathbf{G}_r = \boldsymbol{\tau} = \mathbf{K}_r(\boldsymbol{\theta} - \mathbf{q}) + \mathbf{D}_r(\dot{\boldsymbol{\theta}} - \dot{\mathbf{q}}). \tag{7}$$

$\boldsymbol{\tau}$ is the vector of the axis torques which result in the motion of the axis positions $\mathbf{q} \in \mathbb{R}^6$, given the dynamical parameters $\mathbf{M}_r \in \mathbb{R}^{6\times6}$, $\mathbf{C}_r \in \mathbb{R}^{6\times6}$, and $\mathbf{G}_r \in \mathbb{R}^6$ accounting for the inertial forces, the centrifugal and Coriolis forces, and gravity acting on the robot arm. This axis torque is identical to the torque in the gear, which is assumed as a spring-damper system for each joint. $\mathbf{K}_r = diag(k_{ri}) \in \mathbb{R}^{6\times6}$ and $\mathbf{D}_r = diag(d_{ri}) \in \mathbb{R}^{6\times6}$ denote the stiffness and the damping coefficient. Thus the axis positions can be computed model-based from the motor positions, but they cannot be directly measured.

The kinematic relation between the axis positions and the Cartesian position $\mathbf{x}$ of the body center is represented by the direct kinematic transform $Kin(\cdot)$

$$\mathbf{x} = Kin(\mathbf{q}, \mathbf{d}), \tag{8}$$

where $\mathbf{d} \in \mathbb{R}^3$ is the distance of the body center with respect to the flange of the robot, expressed in the coordinate system of the flange (see Fig. 2)[2].

### B. Contact Scenario

With

$$\mathbf{x}_c = Kin(\mathbf{q}, \mathbf{d}_c) \tag{9}$$

and $\mathbf{d}_c \in \mathbb{R}^3$ from Fig. 2, (5) and (6) hold as well for a robotic contact, assuming that the contact position is known with respect to the robot flange.

In the contact case, the robot dynamics are completed by the external force and torque,

$$\mathbf{M}_r\ddot{\mathbf{q}} + \mathbf{C}_r\dot{\mathbf{q}} + \mathbf{G}_r = \mathbf{K}_r(\boldsymbol{\theta} - \mathbf{q}) + \mathbf{D}_r(\dot{\boldsymbol{\theta}} - \dot{\mathbf{q}}) + \mathbf{J}^T \begin{bmatrix} \mathbf{f} \\ \mathbf{m} \end{bmatrix} \tag{10}$$

where $\mathbf{J}$ is the robot Jacobian, computed for the body center, i.e.,

$$\mathbf{J} = \partial Kin(\mathbf{q}, \mathbf{d}) / \partial \mathbf{q}. \tag{11}$$

[2] Besides $\mathbf{d}$, the kinematic transforms are dependent on the kinematic parameters of the robot.

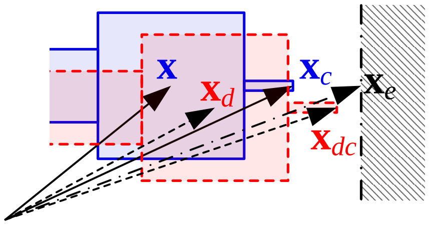

Fig. 3. Position of the real satellite mockup (blue, solid) and the simulated satellite (red, dashed) before a contact. $\mathbf{x}$ and $\mathbf{x}_d$ denote the centers of gravity, whereas $\mathbf{x}_c$ and $\mathbf{x}_{dc}$ are possible contact points with a *wall* (black, dash-dotted) at $\mathbf{x}_e$.

But because of the position control the robot will not give way to the contact force and torque. Therefore, it has to be force-controlled in order to show a satellite-like behavior. However, any reaction will be significantly delayed since in addition to the robot dynamics there are crucial effects of communication delay and control. This means that the contact force is measured longer than it would be the case with free floating bodies. Therefore, whenever the force control would be implemented by accelerating according to (1) and (2), the absolute velocity would be increased after a collision.

## IV. Robotic Simulation of Satellite Dynamics

Sects. II and III demonstrate that the dynamic reaction of robot simulators and real satellites to external forces differ. Therefore, the measured forces are not directly taken to control the robots in our approach. Instead, for each robot, the position of the contacted environment is computed from the measured force and then the robot is moved in order to simulate the repulsion of the satellites.

### A. Main Idea

More precisely, we distinguish between the real robot position $\mathbf{x}$ and a desired robot position $\mathbf{x}_d \in \mathbb{R}^3$, i.e., the position where the robot should be according to the equation of motion of the simulated satellite. Fig. 3 shows the notation. The position $\mathbf{x}_e$ of the environment is computed by (3) and (5) using the contact position $\mathbf{x_c}$ of the real robot. Then, the force $\mathbf{f}_d \in \mathbb{R}^3$ and the torque $\mathbf{m}_d \in \mathbb{R}^3$ of the simulated satellite are computed using its contact position $\mathbf{x}_{dc} \in \mathbb{R}^3$ by

$$\mathbf{f}_d = \mathbf{K}_d(\mathbf{x}_e - \mathbf{x}_{dc}) + \mathbf{D}_d(\dot{\mathbf{x}}_e - \dot{\mathbf{x}}_{dc}) \tag{12}$$

and

$$\mathbf{m}_d = (\mathbf{x}_{dc} - \mathbf{x}_d) \times \mathbf{f}_d, \tag{13}$$

where $\mathbf{K}_d \in \mathbb{R}^{3\times3}$ and $\mathbf{D}_d \in \mathbb{R}^{3\times3}$ are the desired stiffness and damping, i.e., the stiffness and the damping of the real satellite, which might be different to the stiffness $\mathbf{K}_e$ and the damping $\mathbf{D}_e$ of the mockup.

The desired force and torque are then applied to update the desired position and orientation similarly to (1) and (2).

$$M_d \ddot{\mathbf{x}}_d = \mathbf{f}_d \tag{14}$$

$$\mathbf{I}_d \dot{\boldsymbol{\omega}}_d = \mathbf{m}_d - \boldsymbol{\omega}_d \times \mathbf{I}_d \boldsymbol{\omega}_d \tag{15}$$

The index $d$ denotes that here we can define a desired dynamics that can be different from the inertia of the mockup.

$\mathbf{x}_d$ and the corresponding orientation $\boldsymbol{\phi}_d \in \mathbb{R}^3$ are the simulated pose of the satellite. At the same time, this pose is taken as the desired pose for the robot position control. In this way the robot will approximately execute the same trajectory as the simulated satellite. And since the equations of motion of the satellite, i.e., (12) - (15), do not result in increasing velocity of the satellite, the robot velocity will not increase by the simulated impact as well.

In other words, the simulated satellite's trajectory is computed from its current pose $(\mathbf{x}_d, \boldsymbol{\phi}_d)$ and the position of the environment $\mathbf{x}_e$ that is taken from the measured position $\mathbf{x}$, force $\mathbf{f}$, and torque $\mathbf{m}$ of the robot.

### B. Simplified Procedure

In this paper we discuss a simplified version, in which the robot dynamics and the damping at the contact are neglected, i.e., $\mathbf{q} = \boldsymbol{\theta}$ and $\mathbf{D}_e = \mathbf{0}$. The axis stiffness $\mathbf{K}_r$ may be considered nevertheless by replacing $\mathbf{K}_e$ by $(\mathbf{K}_e^{-1} + \mathbf{J}_c\mathbf{K}_r^{-1}\mathbf{J}_c^T)^{-1}$, where $\mathbf{J}_c$ is the robot Jacobian, computed for the contact point, i.e.,

$$\mathbf{J}_c = \partial \mathit{Kin}(\mathbf{q}, \mathbf{d}_c)/\partial \mathbf{q}. \tag{16}$$

The measured force according to (3) can be computed with $M$ being the mass of the mockup.

Then a normal vector

$$\mathbf{n} = (\mathbf{f}^T\mathbf{f})^{-1/2}\mathbf{f} = \mathbf{f}/||\mathbf{f}|| \tag{17}$$

is computed. The components of $\mathbf{K}_e$ which do not correspond to this normal directions, are zero. Therefore, a scalar equation in normal direction is used instead of (5). For $\mathbf{D}_e = \mathbf{0}$ this results in

$$\mathbf{x}_e = \mathbf{x}_c + (\mathbf{n}^T\mathbf{K}_e\mathbf{n})^{-1}\mathbf{f}, \tag{18}$$

where $\mathbf{n}^T\mathbf{K}_e\mathbf{n} = k_{en}$ is nonzero.

Then (12) and (13) are computed with

$$\mathbf{x}_{dc} = \mathit{Kin}(\mathbf{q}_d, \mathbf{d}_c) \tag{19}$$

where $\mathbf{q}_d \in \mathbb{R}^6$ are the desired axis positions. For $\mathbf{K}_d = \mathbf{K}_e$ and $\mathbf{D}_d = \mathbf{0}$, (12) simplifies to

$$\mathbf{f}_d = \mathbf{f} - \mathbf{n}^T\mathbf{K}_d(\mathbf{x}_{dc} - \mathbf{x}_c)\mathbf{n}. \tag{20}$$

Then, instead of (13),

$$\mathbf{m}_d = \mathbf{m} + (\mathbf{x}_{dc} - \mathbf{x}_d) \times \mathbf{f}_d - (\mathbf{x}_c - \mathbf{x}) \times \mathbf{f} \tag{21}$$

is preferred with $\mathbf{m}$ from (4), since in contrast to the above assumption, within a docking tool torques can be exerted as well.

Then, in addition to the desired dynamics (14)-(15), the inverse kinematic transform $\mathit{InvKin}(\cdot)$

$$\mathbf{q}_d = \mathit{InvKin}(\mathbf{x}_d, \boldsymbol{\phi}_d, \mathbf{d}) \tag{22}$$

is applied, since $\mathbf{q}_d$ is required for the computation of (19).

### C. Full Compensation of the Robot Dynamics

Including a damping $\mathbf{D}_e$, (18) becomes

$$\mathbf{x}_e = \mathbf{x}_c + (x_{en} - x_{cn})\mathbf{n}. \tag{23}$$

with

$$(\mathbf{n}^T\mathbf{K}_e\mathbf{n})(x_{en} - x_{cn}) + (\mathbf{n}^T\mathbf{D}_e\mathbf{n})(\dot{x}_{en} - \dot{x}_{cn}) = \mathbf{n}^T\mathbf{f}. \tag{24}$$

This allows no direct computation of $x_{en}$ which is the component of $\mathbf{x}_e$ in the direction of the normal.

This applies more than ever whenever $\mathbf{x}_c$ differs from the Cartesian position that is computed from the motor positions. Then (10) has to be considered, too.

$\mathbf{x}_e$ can be estimated from the motor position and its derivatives as well as the sensor values by a linear observer or a Kalman filter.

### D. Simulation of Multiple Satellites

So far, each robot is treated independently of the other, assuming a contact with a *wall*, i.e., a constant environmental position. Since both satellites will be in motion, it is worth to communicate the real and the desired position to the other robot.

Assuming that the forces as well as the position of the environment will be se same for two robots, from (5) and (12) with $\mathbf{D}_e = \mathbf{D}_d = \mathbf{0}$ we get

$$\begin{aligned} \mathbf{f} &= \mathbf{K}_{e_1}(\mathbf{x}_e - \mathbf{x}_{c_1}) && (25)\\ -\mathbf{f} &= \mathbf{K}_{e_2}(\mathbf{x}_e - \mathbf{x}_{c_2}) && (26)\\ \mathbf{f}_d &= \mathbf{K}_{d_1}(\mathbf{x}_{de} - \mathbf{x}_{dc_1}) && (27)\\ -\mathbf{f}_d &= \mathbf{K}_{d_2}(\mathbf{x}_{de} - \mathbf{x}_{dc_2}) && (28) \end{aligned}$$

where the subscripts of the indices $c$ and $d$ denote the robot number and $\mathbf{x}_{de}$ is the modified position of the environment, which corresponds to the desired positions. Then, other than in (18), $\mathbf{x}_e$ and $\mathbf{x}_{de}$ are computed from

$$(\mathbf{K}_{e_1} + \mathbf{K}_{e_2})\mathbf{x}_e = \mathbf{K}_{e_1}\mathbf{x}_{c_1} + \mathbf{K}_{e_2}\mathbf{x}_{c_2} \tag{29}$$

and

$$(\mathbf{K}_{d_1} + \mathbf{K}_{d_2})\mathbf{x}_{de} = \mathbf{K}_{d_1}\mathbf{x}_{dc_1} + \mathbf{K}_{d_2}\mathbf{x}_{dc_2}, \tag{30}$$

similar to Fig. 1.

In the scalar notation this corresponds to

$$(k_{d_1n} + k_{d_2n})x_{den} = k_{d_1n}x_{dc_1n} + k_{d_2n}x_{dc_2n}, \tag{31}$$

where $k_{*n} = \mathbf{n}^T\mathbf{K}_*\mathbf{n}$ and $x_{*n} = \mathbf{n}^T\mathbf{x}_*$ are the components in the direction of the force vector.

However, since absolute positions may be inaccurate, using (25) and (26), (31) is replaced by

$$\begin{aligned} x_{den} = \; & x_{c_1n} + (k_{d_1n} + k_{d_2n})^{-1}k_{d_2n}(k_{e_1n}^{-1} + k_{e_2n}^{-1})||\mathbf{f}|| \\ & + (k_{d_1n} + k_{d_2n})^{-1}k_{d_2n}(x_{dc_2n} - x_{c_2n}) \qquad (32)\\ & + (k_{d_1n} + k_{d_2n})^{-1}k_{d_1n}(x_{dc_1n} - x_{c_1n}) \end{aligned}$$

or

$$\begin{aligned} x_{den} = \; & x_{c_2n} - (k_{d_1n} + k_{d_2n})^{-1}k_{d_1n}(k_{e_1n}^{-1} + k_{e_2n}^{-1})||\mathbf{f}|| \\ & + (k_{d_1n} + k_{d_2n})^{-1}k_{d_2n}(x_{dc_2n} - x_{c_2n}) \qquad (33)\\ & + (k_{d_1n} + k_{d_2n})^{-1}k_{d_1n}(x_{dc_1n} - x_{c_1n}) \end{aligned}$$

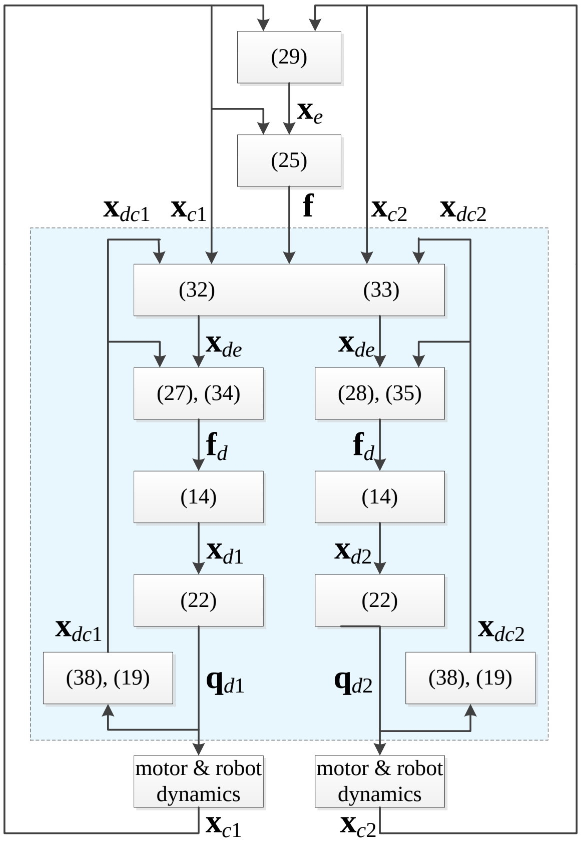

Fig. 4. Simplified flow chart. The colored region represents the procedure.

which use only the force measurement and differences of positions. These equations are used instead of (18).

There is a contact with the desired positions whenever

$$\mathbf{n}^T(\mathbf{x}_{de} - \mathbf{x}_{dc_1}) > 0 \tag{34}$$

or

$$\mathbf{n}^T(\mathbf{x}_{de} - \mathbf{x}_{dc_2}) < 0. \tag{35}$$

Otherwise an $\mathbf{x}_{de}$ might be computed but the desired forces will be zero. Then (27) or (28) are not applicable.

The resulting procedure is summarized in Fig. 4.

### E. Determination of the Contact Point

The computation of the kinematic transforms requires the contact point $\mathbf{d}_c$ with respect to the robot flange. This point is not a priori known and may vary during an experiment. Therefore it has to be computed from the forces $\mathbf{f}$ and torques $\mathbf{m}$ at the center of gravity.

Unfortunately, the relation [3]

$$\mathbf{m} = (\mathbf{R}_s(\mathbf{d}_c - \mathbf{d})) \times \mathbf{f} \tag{36}$$

is not sufficient for the determination of $\mathbf{d}_c$. Therefore the component of $\mathbf{d}_c$ in the direction of the force has to be taken from the shape of the satellite.

[3] The orientation of the sensor is equal to the orientation of the robot flange, such that the rotation matrix $\mathbf{R}_s$ can be taken.

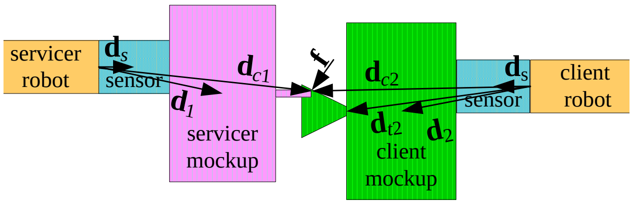

Fig. 5. Possible docking setup for a servicer and a client satellite.

For a setup as in Fig. 5 the contact point of the servicer satellite is obvious. In contrast, the contact point of the client satellite is computed from

$$\mathbf{0} = (\mathbf{R}_s(\mathbf{d}_c - \mathbf{d}_t)) \cdot \mathbf{f} \tag{37}$$

where $\mathbf{d}_t \in \mathbb{R}^3$ is the position of the tool center point, in this case the apex of the cone. Equation (37) means that the force is normal to the surface of the cone.

Equations (36) and (37) are linear in $\mathbf{R}_s(\mathbf{d}_c - \mathbf{d}_t)$ and result in

$$\mathbf{d}_c = \mathbf{d}_t + \mathbf{R}_s^{-1} ||\mathbf{f}||^{-2} (\mathbf{f} \times \mathbf{m}_t) \tag{38}$$

where $\mathbf{m}_t = \mathbf{m} - (\mathbf{R}_s(\mathbf{d}_t - \mathbf{d})) \times \mathbf{f}$ is the torque at the tool center point.

### F. Causality

The computation of $\mathbf{f}_d$ from $\mathbf{x}_{de}$ fails whenever the desired positions are in contact whereas the real positions are not. Because of the delayed robot reaction, this is unlikely when leaving the contact. In contrast, it is likely when approaching since the robot motion is usually delayed with respect to the desired positions.

Therefore, first, the desired trajectories given by (14) and (15) are predicted for several sampling steps in advance, using predictions of the desired force as well. This can be applied in a predictive controller, which in the simplest case at time step $k$ outputs setpoints for $\boldsymbol{\theta}(k)$) which are computed from $\mathbf{x}(k + \Delta t)$ instead of $\mathbf{x}(k)$, where $\Delta t$ is the expected dynamical delay in sampling steps. Thus $\mathbf{x}_d - \mathbf{x}$ is small before the contact.

Second, once a contact has been measured at the real robot positions, it can be checked if a contact had also occurred with the desired positions of the previous sampling steps. This can be tested by (32) to (35) with $x_{dc_*n}$ from a previous time step, whereas $\mathbf{x}_{c_*}$, $x_{c_*n}$, and $||\mathbf{f}||$ are taken from the current time step. If in this way a contact is recognized, the desired forces and torques of this sampling step and the next ones will be recomputed, as well as the desired trajectories. These modified sampling steps of the desired trajectories cannot be used any more for robot control. However they may serve for the computation of an updated simulation.

As mentioned earlier, the robot motion cannot exactly represent the simulated satellite motion, but the differences will be so small that they can hardly be seen. For further computations however, the exact simulated trajectory is important.

### G. Discussion

Incomplete or inaccurate knowledge of the satellites' dynamics might impair the reliability of the simulation. However, the main response results from the principle of momentum conservation. Therefore, the effect of uncertain $\mathbf{K}_d$ is not significant. If the assumed stiffness $\mathbf{K}_d$ is too high, too high forces will be the consequence and thus a shorter time of contact. The resulting total momentum after an elastic impact will be identical to the total momentum before the impact. But the magnitude of $\mathbf{D}_d$ may be crucial. Therefore a worst case scenario might be simulated, e.g. $\mathbf{D}_d = \mathbf{0}$, in addition to simulations with a damping factor which is assumed from the material properties.

The dynamics of the robot and the mockup can be identified using the existing hardware. But the effect of inaccurate parameters is marginal as well, since only the modification due to $(\mathbf{x}_{dc} - \mathbf{x}_c)$ is affected by the dynamics model.

Immediately after an impact, the desired velocity $\dot{\mathbf{x}}_d(k)$ for the robot at a time step $k$ may largely differ from the previous desired robot velocity $\dot{\mathbf{x}}_d(k-1)$, such that the robot will exceed its acceleration limit. Therefore a trajectory generation method as e.g. [14] or an additional controller have to be used in order to send a feasible desired position to the industrial robot controller.

Stability of robot control is not concerned by this approach because there is no direct feedback of measured forces. Instead, the robot is feedforward controlled to move along a trajectory that is computed from idealized assumptions.

## V. Simulations

The simplified procedure of Sect. IV with $\mathbf{q} = \boldsymbol{\theta}$ and $\mathbf{D}_e = \mathbf{D}_d = \mathbf{0}$ is verified by simulation of a KUKA KR16 robot which is controlled by RSI Ethernet. Position commands $\mathbf{q}_d$ are filtered such that the robot dynamics are not more significant ($\mathbf{q} \approx \boldsymbol{\theta}$). Nevertheless the acceleration constraints of the robot cause a severe delay (see lower part of Fig. 6) between the commanded (solid) and the real (dotted) trajectories. There, the used method from [14] provides the fastest feasible following.

The simulation assumes that the client satellite stands still whereas the servicer satellite approaches with constant velocity. At time 8.04 s a direct central impact occurs. The result of this elastic collision is that the servicer stops down whereas the client accelerates to the original velocity of the servicer (solid lines). This results from equal masses $M_{d_1} = M_{d_2} = 100$ kg. Forces arise between the two intersections of the red and the blue curves, i.e., when the two bodies penetrate. With $k_{e_1} = k_{e_2} = k_{d_1} = k_{d_2} = 100$ N/mm the stiffness is assumed quite low. Nevertheless the measured forces reach 364 N. In contrast, the forces between the satellites will not exceed 157 N. This comes from the relatively small mass. This demonstrates that the difference between the robots' and the satellites' motion significantly depends on the parameters of the two systems.

Fig. 7 shows a further simulation in which the servicer moves towards a docking region, assumed as a cone, see Fig. 5. Now we have several oblique eccentric impacts,

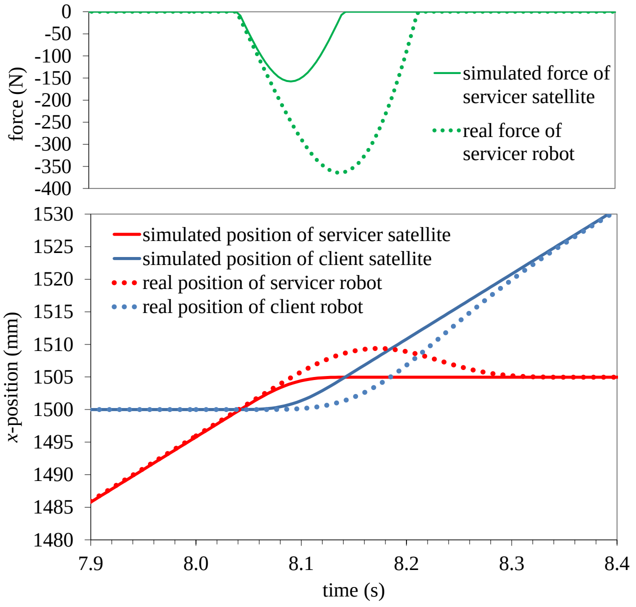

Fig. 6. Simulated forces $\mathbf{f}_d$ and positions $\mathbf{x}_d$, corresponding to the satellites' motions and real (measured / executed) robot forces $\mathbf{f}$ and positions $\mathbf{x}$ for a direct central impact with equally low satellite masses.

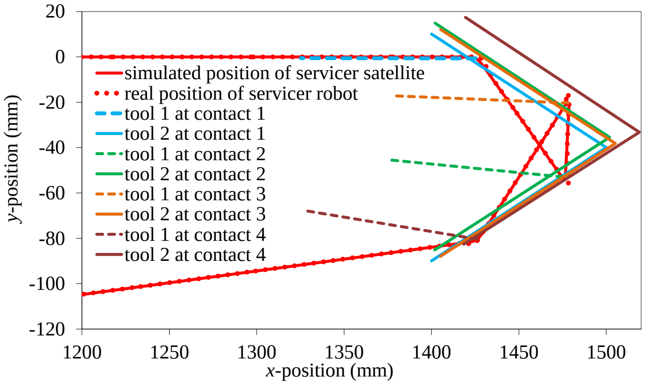

Fig. 7. Real and simulated positions corresponding to the motion of two satellites with masses $M_{d_1} = 100$ kg (servicer) and $M_{d_2} = 1000$ kg (client) and tools according to Fig. 5.

resulting in small rotations as well. In Fig. 7 the bar of the servicer bumps four times (contact 1 - contact 4) against the cone of the client before both satellites diverge. Differences between the satellites' and the robots' motion are visible only close to the contacts. In addition to Fig. 6 now the bar of the sevicer (tool 1) and the cone of the client (tool 2) are marked for each contact in order to visualize the orientation. In this simulation equal moments of inertia $\mathbf{I}_1 = (diag)10$ kgm$^2$ (servicer) and $\mathbf{I}_2 = (diag)300$ kgm$^2$ (client) are assumed such that no tumbling is excited.

## VI. Conclusion

The paper shows that for robotic simulation of the collisions between two satellites it is indispensable to compute an idealized satellite motion while controlling the robots. The simulations show that robots are able to perform a motion which is close to the computed satellites' motion, although their dynamics may be totally different. Limits are reached when the robots are still in contact when the satellites would already collide at another point or when the resulting desired contact time is in the order of magnitude of the sampling time.

Future work will include the robot dynamics and a contact damping as outlined in Sect. IV-C. Then it is possible to simulate partially elastic impacts, i.e., collisions with a coefficient of restitution of less than one. In addition a more realistic simulation can be reached by a more refined contact model, e.g. that from [15], and by identification of the dynamic parameters of the robot and the mockup. Finally the procedure will be executed with real robots, exhibiting real sensors and docking mechanisms.

## References

[1] S. K. Agrawal, G. Hirzinger, K. Landzettel, and R. Schwertassek. A new laboratory simulator for study of motion of free-floating robots relative to space targets. *IEEE Trans. on Robotics and Automation*, 12:627–633, 1996.

[2] R. Krenn and B. Schäfer. Limitations of hardware-in-the-loop simulations of space robotics dynamics using industrial robots. In *Proc. 5th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Noordwijk, The Netherlands, 1999.

[3] Z. Milenkovic, C. D'Souza, Z. Wilson, D. Huish, J. Bendle, and A. Kibler. The space operations simulation center (SOSC) and closed-loop hardware testing for orion rendevous system design. ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20120012941.pdf, 2012.

[4] C. G. Henshaw. The DARPA phoenox spacecraft servicing program: Overview and plans for risk reduction. In *Proc. 12th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Montreal, Canada, 2014.

[5] W. Xu, B. Liang, Y. Xu, C. Li, and W. Qiang. A ground experiment system of free-floating robot for capturing space target. *Journal of Intelligent Robot Systems*, 48:187–208, 2007.

[6] O. Ma, A. Flores-Abad, and T. Boge. Use of industrial robots for hardware-in-the-loop simulation of satellite rendevouz and docking. *Acta Astronautica*, 81:335–347, 2012.

[7] T. Boge, T. Wimmer, O. Ma, and T. Tzschichholz. EPOS - using robotics for RvD simulation of on-orbit servicing missions. In *Proc. AIAA Modeling and Simulation Technologies Conference*, Toronto, Ontario, Canada, 2-5 August 2010.

[8] M. Zebenay, R. Lampariello, T. Boge, and D. Choukroun. A new contact dynamics model tool for hardware-in-the-loop docking simulation. In *Proc. 11th Int. Symp. on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Turin, Italy, 2012.

[9] J. Artigas, M. De Stefano, W. Rackl, R. Lampariello, B. Brunner, W. Bertleff, R. Burger, O. Porges, A. Giordano, C. Borst, and A. Albu-Schäffer. The OOS-SIM: An on-ground simulation facility for on-orbit servicing robotic operations. In *Proc. 2015 IEEE Int. Conf. on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015.

[10] W. Fehse. *Automated Rendevouz and Docking of Spacecraft*. Cambridge Aerospace Series. Cambridge University Press, 2003.

[11] R. Krenn, T. Boge, and M. Zebenay. Passivity control for robotic satellite docking simulators. In *Proc. 9th Int. ESA Conference on Guidance, Navigation & Control Systems (GNC2014)*, Porto, Portugal, June 2014.

[12] M. De Stefano, J. Artigas, W. Rackl, and A. Albu-Schäffer. Passivity of virtual free-floating dynamics rendered on robotic facilities. In *Proc. 2015 IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 781–788, Seattle, WA, USA, May 2015.

[13] R. Brach. *Mechanical Impact Dynamics*. John Wiley, 1991.

[14] F. Lange and A. Albu-Schäffer. Path-accurate online trajectory generation for jerk-limited industrial robots. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):82–89, 2016. Presented at the *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016.

[15] K. H. Hunt and F. R. E. Crossley. Coefficient of restitution interpreted as damping in vibroimpact. *Trans ASME, J. Appl. Mech.*, 42:440–445, 2 1975.