# VALIDATION OF CONTACT SIMULATION FOR ROBOTIC MANIPULATION IN SPACE

**Arthur Wahl, Georgij Grinshpun, and Juergen Rossmann**

*Institute for Man-Machine Interaction, RWTH Aachen University*
*Ahornstr. 55, D-52074 Aachen, Germany*
{*wahl, grinshpun, rossmann*}*@mmi.rwth-aachen.de*

## ABSTRACT

The detachment as well as reattachment of satellite components during a reconfiguration or repair process represents a challenging task, since the establishment of a stable contact between the connection interfaces of the satellite's components and the robot manipulators of a servicer satellite is required. To asses such scenarios, we propose the application of a simulation driven approach. Our approach builds upon a Virtual Testbed for space environments and presents a novel collision detection algorithm tailored to connection interfaces composed of centering pins and holes. The algorithm performs contact manifold and penetration depth computation in real-time. To validate our approach, docking scenarios between a centering pin and an interface's hole were performed within a dedicated setup in a series of reference experiments with real hardware and robots.

Key words: Collision Detection; Contact Manifold & Penetration Depth Computation; Peg-In-Hole; Hardware Experiments; Virtual Testbeds; Simulation Frameworks.

## 1. INTRODUCTION

Today's satellites are monolithic systems, whose lifetime is often determined by the lifetime of their critical components. Without any feasible way of maintenance or repair, malfunctioning components can lead to an early termination of a satellite's mission period. At the end of its lifetime, a satellite becomes space debris that may endanger other satellites. The concept of modular satellites as presented in the project iBOSS (intelligent Building Blocks for On-Orbit Satellite Servicing, [11]) is aiming to contribute to the reduction of space debris by extending the lifespan of satellites through easier maintainability of their components. A modular design simplifies access to malfunctioning components and enables their replacement through servicer satellites during on-orbit-satellite servicing missions [2], see Figure 1.
Detachment as well as reattachment of components during reconfiguration or repair processes represent a challenging task, since the establishment of a stable contact between the connection interfaces of a satellite's component and the robot manipulators of a servicer satellite is required. Therefore, we propose the application of Virtual Testbeds [7], [9] to assess such scenarios. Virtual

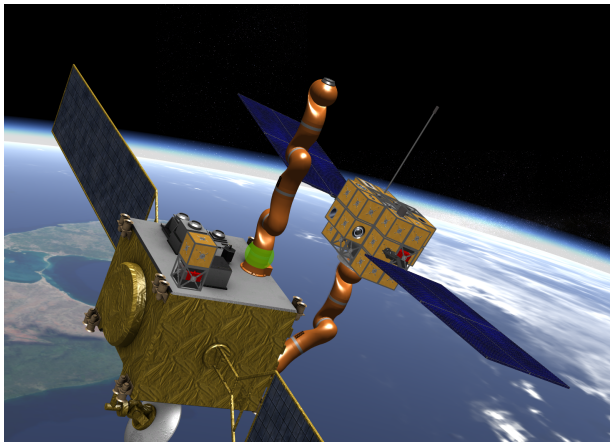

*Figure 1. Simulation of an on-orbit satellite servicing mission in Virtual Robotic Testbed for space environments [7]: A servicer satellite (l.h.s.) is docked to a modular satellite (r.h.s.) and replaces a malfunctioning component (iBOSS concept [11]).*

Testbeds enable analysis, as well as verification of algorithms regarding reconfiguration, handling, and placement of a satellites components by means of simulation.
Establishing a stable contact is essential for the successful completion of any reconfiguration or repair task. The first contact between a servicer's robot manipulator and a target component should establish a stable contact and is usually implemented by multiple centering pins [2]), see Figure 2. These pins enter in contact with the target's mechanical interface and assure the self-centering of the manipulator. A robot manipulator detects occurring impacts between the centering pins and the interface. The manipulator's controller reacts according to the arising forces and allows the centering pins to slide into the interfaces docking holes. In general, the process can be modeled as a Peg-In-Hole situation, which is a well-studied but still challenging task in robotic applications [12], [10], [6].
Therefore we focused on Peg-In-Hole scenarios with respect to the goal of docking a servicer's robot manipulator to a mechanical interface of a modular satellite's component by developing a simulation environment that captures the dynamic behavior between centering pins and docking holes during contact. Our approach builds upon

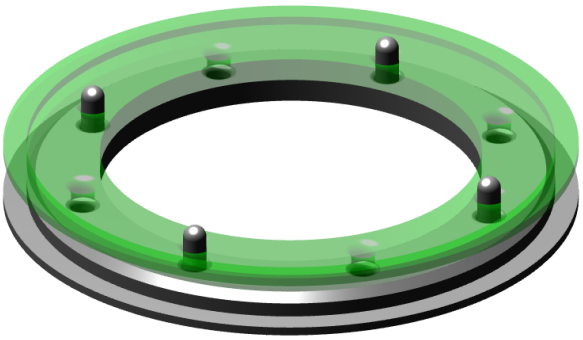

*Figure 2. Centering pins of a component's mechanical interface.*

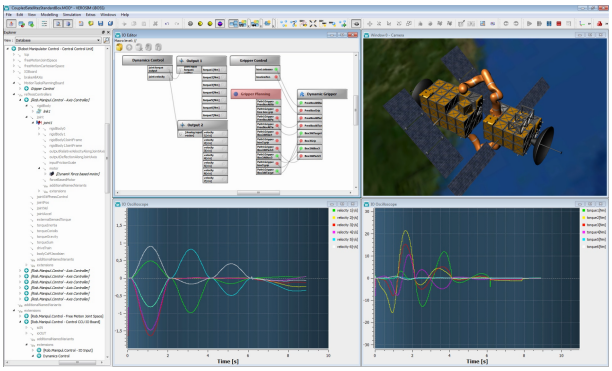

*Figure 3. Virtual Robotic Testbed for space environments [7] (iBOSS concept [11]) (satellite model ©TU Berlin).*

a Virtual Robotic Testbed for space environments [7], [9], see Figure 3, and is based on a multi-body dynamics simulation [4]. One of the main contributions of this work is the development of a discrete collision detection algorithm to simulate interactions between centering pins and docking holes. The algorithm identifies intersections, determines the contact manifold and the corresponding penetration depths for capped cylinders and fitting hole geometries in real-time. By utilizing the Virtual Testbed and integrating the novel collision detection as well by employing simulated robots with endowed compliance control [5], we were able to simulate multiple interface docking scenarios in space environments like e.g. earth's geostationary orbit, as depicted in Figure 1.
The second main contribution of this work is the validation of data generated by our simulation environment during contact-handling. Virtual Testbeds make it possible to validate simulation data by facilitating control options for real robot manipulators as well as access to the data generated by such manipulators [8]. Therefore a physical testbed in form of a robotic cell (initially being used for experimental landing verification [9]) has been extended to perform contact experiments.

## 2. VIRTUAL ROBOTIC TESTBED FOR SPACE ENVIRONMENTS

Virtual Testbeds are advanced 3D simulation environments which model all important aspects of an application and enable a development engineer to systematically examine complex systems, including all relevant components of such systems and their interdependencies. Methods for verification and validation of such systems are provided. For this purpose Virtual Testbeds integrate 3D models, simulation algorithms, and real hardware devices for a holistic view of the dynamic overall system [7].
The Virtual Space Robotics Testbed [9] serves as a decision support system for engineers during the design phase of space mission scenarios like e.g. on-orbit-satellite servicing, rendezvous and docking, planetary landing or exploration and offers a comprehensive set of simulation methods to model robotic applications in space environments e.g. multi-body dynamics, kinematics, orbital mechanics, structural mechanics, pose control and controlling algorithms for robotic manipulation. The modular approach of the Virtual Space Robotics Testbed allows for on-the-fly adaption of various space scenario setups and provides the tools to simulate multiple interface docking scenarios in different space environments by utilizing the aforementioned simulation methods and integrating our novel collision detection algorithm as an extension of the underlying multi-body dynamics simulation.
Furthermore interfaces and control mechanisms for real hardware devices such as robotic manipulators, are an integral part of the Virtual Space Robotics Testbed. They provide the possibility to verify and validate simulated results with real data generated by a physical mockup of the simulated scenario.

## 3. CONTACT SIMULATION

We introduce a novel collision detection algorithm that uses a case by case analysis to determine collisions between an interface's centering pins and the docking hole of an opposed interface, modeled as a Peg-In-Hole scenario. The algorithm uses an analytical approach to compute the resulting contact manifold and penetration depths in real-time. Contact handling is done by a rigid multi-body dynamics simulation [4] that employs generalized tools of contact graph analysis for fast and robust simulations of joint connected multi-body system such as robotic manipulator and provides a collision detection library for multiple primitive geometries like e.g. spheres, boxes or capped cylinders. For fast collision detection and contact manifold determination, further elements of an interface are simulated by combinations of primitive geometries as a simplified physical substitute model of the original geometry.

### 3.1. Specialized Peg-In-Hole Collision Detection

Capsule shaped spheres (CylSpheres) or capped cylinders, depicted on the right side of Figure 4, are basic elements of the underlying dynamics simulation of the Virtual Robotic Testbed and were the most appropriate choice for the physical representation of an interfaces centering pins, regarding the performance of the follow-

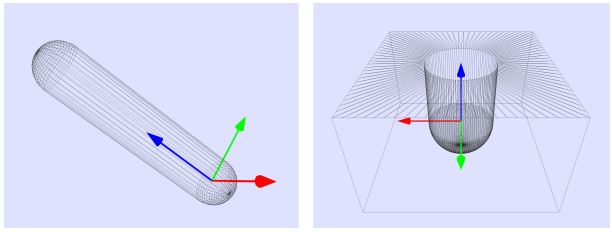

*Figure 4. CylSphere (l.h.s.) and the new geometric type "CylSphereHole" (r.h.s.).*

ing intersection computation and the physical substitute model of a centering pin's geometry. In addition to this, a new geometry had to be designed that models the interfaces docking hole and the corresponding rigid body's physical behavior had to be determined for the dynamic simulation. Therefore the "CylSphereHole" geometry was generated by inserting a negative "CylSphere" with one open end into a box shaped geometry. The open end of the "CylSphere" acts as an opening at the top surface of the box geometry, depicted on the left side of 4.

To model the geometry's physical behavior we had to compute its inertia tensor, $\Theta^{\text{csh}}$. This can be done by first computing the inertia tensor of the box shaped geometry, $\Theta^{\text{box}}$, and subtracting the inertia tensor of the hole shaped geometry $\Theta^{\text{hole}}$ from it afterwards. The latter is composed of a cylinder and a cap (half sphere). Our approach starts by computing the "CylSphereHole's" center of mass as well as the mass moment of inertia for the aforementioned shapes $(\text{box}, \text{cyl}, \text{cap})$ with respect to their centers of mass. The orientation of the local coordinate systems of the shapes is identical to the orientation of the "CylSphereHole", as depicted on the left side of Figure 4:

$$\begin{aligned} \Theta_x^{\text{box}} &= \tfrac{1}{12} m^{\text{box}} (b^2 + c^2) \\ \Theta_y^{\text{box}} &= \tfrac{1}{12} m^{\text{box}} (a^2 + c^2) \\ \Theta_z^{\text{box}} &= \tfrac{1}{12} m^{\text{box}} (a^2 + b^2) \end{aligned} \tag{1}$$

$$\begin{aligned} \Theta_{x,y}^{\text{cyl}} &= \tfrac{1}{4} m^{\text{cyl}} \cdot r^2 + \tfrac{1}{12} m^{\text{cyl}} \cdot h^2 \\ \Theta_z^{\text{cyl}} &= \tfrac{1}{2} m^{\text{cyl}} \cdot r^2 \end{aligned} \tag{2}$$

$$\begin{aligned} \Theta_{x,y}^{\text{cap}} &= \left(\tfrac{2}{5} m^{\text{sph}} \cdot r^2\right) * \tfrac{1}{2} - \left(\tfrac{1}{2} \cdot \tfrac{9}{64} m^{\text{sph}} \cdot r^2\right) \\ \Theta_z^{\text{cap}} &= \left(\tfrac{2}{5} m^{\text{sph}} \cdot r^2\right) * \tfrac{1}{2} \end{aligned} \tag{3}$$

where $a$, $b$, $c$ are the side lengths of the box along x, y, and z-axis, $m^{\text{box}}$ is the mass of the box shape, $m^{\text{cyl}}$ is the mass of the cylinder shape, $m^{\text{sph}}$ is the mass of the sphere shape that can be generated by duplicating the cap, $r$ is the radius of the hole and $h$ is the height of the cylinder. The height of the box is chosen as $c = h + r$ to simplify calculations. The resulting mass moments of inertia are expressed with respect to the axes through the center of mass of each shape. These moments of inertia have to be expressed with respect to the "CylSphereHole's" center of mass. This is done by applying the Huygens–Steiner theorem that delivers the mass moment of inertia of the three shapes for axes through the "CylSphereHole's" center of gravity:

$$\begin{aligned} \Theta_x^{\text{box}'} &= \Theta_x^{\text{box}} + m^{\text{box}} {d_{cog}^{\text{box}}}^2 \\ \Theta_y^{\text{box}'} &= \Theta_y^{\text{box}} + m^{\text{box}} {d_{cog}^{\text{box}}}^2 \\ \Theta_z^{\text{box}'} &= \Theta_z^{\text{box}} \end{aligned} \tag{4}$$

$$\begin{aligned} \Theta_{x,y}^{\text{cyl}'} &= \Theta_{x,y}^{\text{cyl}} + m^{\text{cyl}} {d_{cog}^{\text{cyl}}}^2 \\ \Theta_z^{\text{cyl}'} &= \Theta_z^{\text{cyl}} \end{aligned} \tag{5}$$

$$\begin{aligned} \Theta_{x,y}^{\text{cap}'} &= \Theta_{x,y}^{\text{cap}} + m^{\text{cap}} {d_{cog}^{\text{cap}}}^2 \\ \Theta_z^{\text{cap}'} &= \Theta_z^{\text{cap}} \end{aligned} \tag{6}$$

where $d_{cog}^{box} = c_z^{\text{csh}} - c_z^{\text{box}}$, $d_{cog}^{\text{cyl}} = c_z^{\text{csh}} - c_z^{\text{cyl}}$ as well as $d_{\text{cog}}^{\text{cap}} = c_z^{\text{csh}} - c_z^{\text{cap}}$ are the distances of the shape's centers of mass locations ($c^{\text{box,cyl,cap}}$) from the center of mass location of the "CylSphereHole" ($c^{\text{csh}}$) along the z-coordinate (height). Since the orientation of all local coordinate systems is identical to the orientation of the coordinate system of the "CylSphereHole" and all centers of mass lie on the y-axis of that coordinate system, the distance between the centers of mass can be computed by subtracting their z-coordinates (heights). As stated in the beginning of the section, we get the x-, y- and z component of the "CylSphereHole's" inertia tensor, $\Theta_{x,y,z}^{\text{csh}}$, by subtracting the transformed moments of inertia of the cylinder and cap from the moment of inertia of the box.

$$\Theta_x^{\text{csh}} = \Theta_x^{\text{box}'} - \Theta_x^{\text{cyl}'} - \Theta_x^{\text{cap}'} \tag{7}$$

$$\Theta_y^{\text{csh}} = \Theta_y^{\text{box}'} - \Theta_y^{\text{cyl}'} - \Theta_y^{\text{cap}'} \tag{8}$$

$$\Theta_z^{\text{csh}} = \Theta_z^{\text{box}'} - \Theta_z^{\text{cyl}'} - \Theta_z^{\text{cap}'} \tag{9}$$

This delivers the following inertia tensor for the "CylSphereHole" shape:

$$\Theta^{\text{csh}} = \begin{bmatrix} \Theta_x^{\text{csh}} & 0 & 0 \\ 0 & \Theta_y^{\text{csh}} & 0 \\ 0 & 0 & \Theta_z^{\text{csh}} \end{bmatrix} \tag{10}$$

Our collision detection algorithm subdivides possible collisions between the peg and the hole into several intersection categories and uses a case by case analysis to determine the intersection points as well as the penetration depths. Generally, collisions occur inside the hole, on the rim of the hole or the surfaces of the box.

We use an analytical approach to compute collisions that occur inside the hole. The approach is based on orthogonal projections of the cylindrical body and cap of "CylSphere" onto the major axis, $a^{\text{csh}} = \overrightarrow{P_0^{\text{csh}} P_1^{\text{csh}}}$, of the hole, see Figure 5. We distinguish between collisions of the peg with the spherical bottom cap of the hole and the walls of the cylindrical body of the hole. The algorithm starts by verifying if a "cap contact" has occurred. The endpoints, $P_0^{cs}$, $P_1^{cs}$, of the major axis, $a^{cs}$, of the "CylSphere" are used to determine if the peg has entered that region of the hole. Therefore, the scalar products between

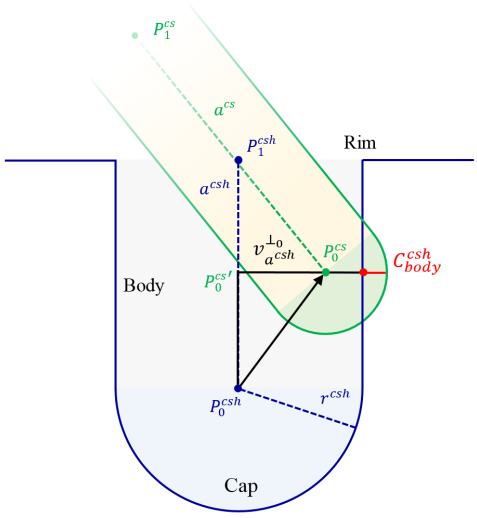


*Figure 5. Intersection point computation on the inside of the hole.*

the endpoints of the peg and the major axis of the hole are computed and the following two conditions are checked:

$$\left\langle \overrightarrow{P_0^{\text{csh}} P_0^{cs}}, a^{\text{csh}} \right\rangle \leq 0 \text{ or } \left\langle \overrightarrow{P_0^{\text{csh}} P_1^{cs}}, a^{\text{csh}} \right\rangle \leq 0 \quad (11)$$

where $P_0^{\text{csh}}$ is the bottom point of the major axis of the hole, see Figure 5. One or both endpoints can be located within the cap of the hole, depending on the size of the "Cylsphere". An intersection has occurred if the distance between the endpoint $P_0^{\text{csh}}$ of the hole and an endpoint of the "CylSphere" is greater than the difference between the radius of the hole $r^{\text{csh}}$ and the radius of the peg $r^{cs}$:

$$\begin{aligned} \left\| \overrightarrow{P_0^{\text{csh}} P_0^{cs}} \right\| &> \left(r^{\text{csh}} - r^{cs}\right) \rightarrow intersection \\ \left\| \overrightarrow{P_0^{\text{csh}} P_1^{cs}} \right\| &> \left(r^{\text{csh}} - r^{cs}\right) \rightarrow intersection \end{aligned} \quad (12)$$

The intersection point $C_{\text{cap}}^{\text{csh}}$ is obtained by:

$$C_{\text{cap}}^{\text{csh}} = P_0^{\text{csh}} + \frac{\overrightarrow{P_0^{\text{csh}} P_{0,1}^{cs}}}{\left\| P_0^{\text{csh}} P_{0,1}^{cs} \right\|} \cdot r^{\text{csh}} \quad (13)$$

The algorithm continues to check for contacts if at least one of the endpoints of the peg is not located within the cap of the hole. The next test case checks for contacts of one of the caps with the cylindrical body of the hole. As before, two conditions are formulated to verify if the peg is located within the body of the hole by computing the corresponding scalar products between the endpoints of the peg and the end point $P_1^{\text{csh}}$ at the top of the major axis of the hole:

$$\left\langle \overrightarrow{P_1^{\text{csh}} P_0^{cs}}, a^{\text{csh}} \right\rangle < 0 \text{ or } \left\langle \overrightarrow{P_1^{\text{csh}} P_1^{cs}} a^{\text{csh}} \right\rangle < 0 \quad (14)$$

The intersection test for this case is done by computing the orthogonal distance between the peg and the major axis of the hole, as depicted in Figure 5. This delivers the minimal distance between both shapes. Comparing the resulting value with the difference between the radii of the hole and the peg, determines if an intersection has occurred. Depending on which of the endpoints is contained within the body of the hole, we project one or both onto the major axis of the hole, in orthogonal direction to that major axis:

$$P_{0,1}^{cs\prime} = \frac{\left\langle \overrightarrow{P_0^{\text{csh}} P_{0,1}^{cs}}, a^{\text{csh}} \right\rangle}{\left\| a^{\text{csh}} \right\|^2} \cdot a^{\text{csh}} \quad (15)$$

Afterwards we check, if the length of the vector $v_{a^{\text{csh}}}^{\perp_{0,1}} = \overrightarrow{P_0^{\text{csh}} P_{0,1}^{cs}} - P_{0,1}^{cs\prime}$ between the projected endpoint, $P_{0,1}^{cs\prime}$, and the original endpoint, $P_{0,1}^{cs}$, is longer than the difference between the radii of the peg and the hole. Hence the following intersection test is executed:

$$\left\| v_{a^{\text{csh}}}^{\perp_{0,1}} \right\| > \left(r^{\text{csh}} - r^{cs}\right) \rightarrow intersection \quad (16)$$

The intersection point $C_{body}^{c??h}$ is obtained by:

$$C_{body}^{\text{csh}} = P_{0,1}^{cs\prime} + \frac{v_{a^{\text{csh}}}^{\perp_{0,1}}}{\left\| v_{a^{\text{csh}}}^{\perp_{0,1}} \right\|} \cdot r^{\text{csh}} \quad (17)$$

The last test case of our algorithm checks for contacts between the peg and the rim of the hole, as depicted in Figure 6. We use a numerical approach that parametrizes the ring shape of the rim and iteratively searches for the minimal distance between the peg and the rim while subdividing the search interval with each iteration as depicted in Figure 7. Collisions between the rim and the peg can occur depending on the size of the "Cylsphere" with one of the caps, the body of the peg and – at the same time – one of the caps and the body or with both caps. Therefore, our algorithm searches for minimal distances between the rim and both caps as well as the body of the peg until a chosen iteration level is reached. Afterwards, the results are used to determine if an intersection has occurred. The algorithm continues to check for such contacts if the former formulated conditions have shown that one or both endpoints of the peg are not located within the body of the hole. The minimal cap distance for one of the caps of the peg is determined by comparing the distances between the corresponding endpoint of the peg and all rim positions on the search interval of an iteration. A search interval consists of eight equidistant positions on the rim. The search interval on the rim is sub-divided in each iteration step in correspondence to the current minima. The rim position $R_j$ that yields the minimal distance $\left\| \overrightarrow{R_j P_{0,1}^{cs}} \right\|$ on an interval is used to choose the next search interval in-between the previous $R_{j-1}$ and the following rim position $R_{j+1}$ on the current interval, as depicted in Figure 7. Afterwards, eight new equidistant positions are computed in-between the next search interval whose distances to the endpoint of the peg have to be compared. The procedure is repeated until the chosen iteration level is reached. An intersection has occurred if the resulting minimum $\left\| \overrightarrow{R_{min} P_{0,1}^{cs}} \right\|$ is smaller

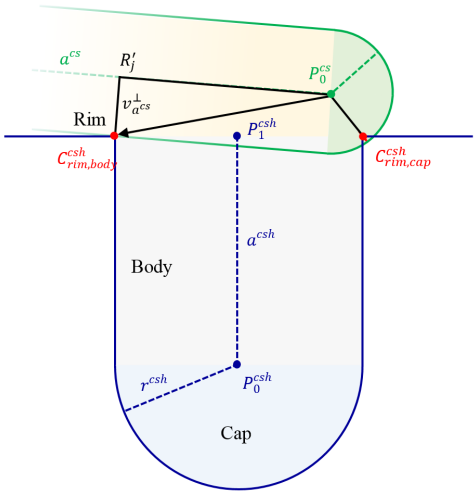


*Figure 6. Intersection point computation for the rim of the hole.*

than the radius of the peg:

$$\left\|\overrightarrow{R_{min}P^{cs}_{0,1}}\right\| < r^{cs} \rightarrow intersection \tag{18}$$

In that case, the rim position $R_{min}$ is chosen as an intersection point $C^{\text{csh}}_{rim,\text{cap}}$:

$$C^{\text{csh}}_{rim,\text{cap}} = R_{min} \tag{19}$$

The distance of the body of the peg to the rim is determined by projecting the rim positions of a search interval orthogonal to the major axis of the peg:

$$R'_j = \frac{\left\langle \overrightarrow{P^{cs}_0 R_j}, a^{cs} \right\rangle}{\|a^{cs}\|^2} \cdot a^{cs} v^{\perp_j}_{a^{cs}} = \overrightarrow{P^{cs}_0 R_j} - \overrightarrow{R'_j} \tag{20}$$

where $v^{\perp_j}_{a^{cs}}$ is the vector between the projected rim position, $R'_j$, and the original rim position, $R_j$. As before, the search procedure is repeated until the chosen iteration level is reached and the last minimal distance, $\left\|v^{\perp}_{a^{cs}\,min}\right\|$, between the body of the peg and the rim has been determined. The intersection test is given by:

$$\left\|v^{\perp}_{a^{cs}\,min}\right\| < r^{cs} \rightarrow intersection \tag{21}$$

If an intersection has occurred, the rim position $R_{min}$ that yielded the last minimum is chosen as an intersection point $C^{\text{csh}}_{rim,body}$ for a rim contact.
The set of all intersection points that have been determined throughout the different contact cases are defining the resulting contact manifold between a peg and a hole for all intersections that have occurred during a simulation step. The penetration depths and corresponding penetration directions are used for contact handling. Modeling an interfaces docking holes and centering pins as individual rigid body extensions of the underlying dynamic simulation, facilitates the possibility to influence the contact behavior of both components. Thereby material and

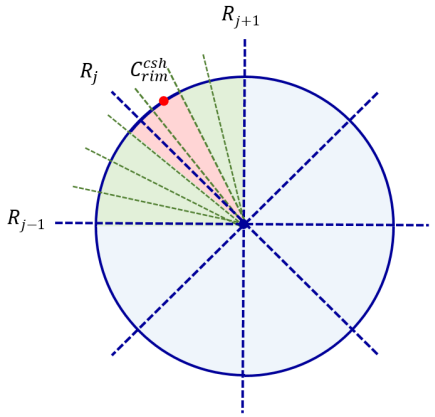


*Figure 7. Sub-division of the search interval along the rim of the hole.*

control parameters in the simulation can be iteratively adjusted during the subsequent validation steps, until the behavior of the virtual interface and robot manipulators is as close as required to their behavior in the physical testbed.

### 3.2. Performance

Opposed to mesh based collision detection approaches, the precision of the contact manifold computation as well as the performance of our approach is not dependent on the complexity of the polygonal models of the interface's components, thereby our algorithm presents a more robust and fast approach for the contact simulation of interface docking scenarios. Real-time rates for configurations of up to 20 centering pins and docking holes could be realized (including contact manifold, penetration depth computation and contact handling) on a 6-core 4.13 GHz Intel i7 with 16 GB of RAM for a simulation time step of 10ms, see Figure 8. The average computation time of the collision detection routine for one simulation step of a peg and a hole took about 0.02ms, so the majority of the time is spent within the contact handling and rendering.

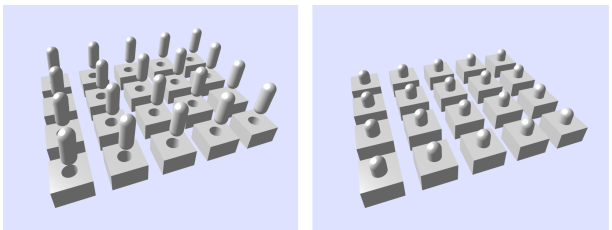

*Figure 8. Specialized Peg-In-Hole Simulation.*

## 4. ROBOTIC VALIDATION

The contact simulation has been validated by conducting reference experiments within a physical testbed setup

composed of a robotic cell and by comparing the simulated results with data generated by the robotic manipulator of that cell, in our case a KUKA-LWR robot. The Virtual Robotic Testbed has been used to simulate different test case contact scenarios between a centering pin and an interfaces docking hole as well as to control the KUKA-LWR robot.

### 4.1. Experimental Setup

The experimental setup should perform reproducible motion sequences in simulation as well reality to obtain comparable results. In addition it should provide the ability to measure the occurring forces, torques, as well as positions of the components during contact of the centering peg and docking hole. Therefore a robotic cell, being initially used for experimental landing verification [9], has been adapted to fit our scenario. The cell includes two KUKA-LWR robots. Both are mounted on a linear axis. Here, the robot manipulators act as measuring instruments. The exact position, as well as exerted forces can be measured with the internal position and torque sensors of the manipulators, while conducting the contact experiments. Subsequently the data is used to identify and quantify the characteristic effects of the contact behavior. Furthermore an experimental table has been installed in range of the robot. The table is attached to the rigid frame structure of the cell and caries the experimental hardware. The setup is considered to be fixed in relation to the robot's base.

The KUKA-LWR offers a control interface called fast research interface (FRI), which enables communication between physical robots and third-party applications. Among other inputs, it accepts target positions in form of joint angles, which is the main input of the robot in our case. Furthermore the FRI provides several control modes e.g. programmable compliance in Cartesian or axis-specific mode [1]. These control modes are suitable to carry out experiments with rigid environments [3].

In accordance with the contact simulation, we chose a capped cylinders (peg) and a round hole formed from the negative of a capped cylinder (both with diameters of 3.5 cm) as experimental shapes.

Due to the flexible testbed structure it was possible to separate the control and evaluation modules from the physical devices. The link to the KUKA-LWR robots can be switched off and replaced on-the-fly by a simulated robot including the rigid body simulation of the Virtual Robotic Testbed, torque controlled motors and the stiffness controller as implemented in the real robot [5]. Hence the controller running the experiment and the evaluation logic was used for both the simulated and the real setup.

### 4.2. Experimental Results

The first experiment was conducted to examine a simple contact of the peg with a flat surface, similar to the docking hole's upper surface. The generated data was used as a reference to verify the data generated by more complex contact scenarios in combination with the hole shape. In that scenario the robot was commanded to move the TCP

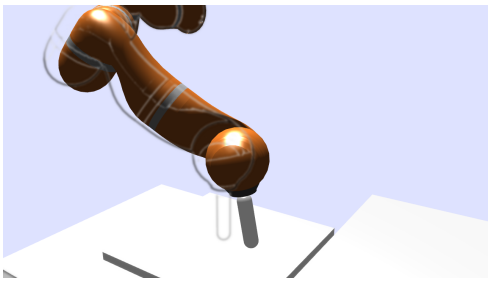

*Figure 9. Robot pushing a peg towards a flat surface.*

downwards while the trajectory was constrained by the surface on the experiment table (target of the trajectory was 5cm below the table). Due to the impact, the robot was not able to follow the command. Figure 9 depicts the resulting configuration of the robot as the target reaches the lowest point. Since the robot's controller was configured in joint stiffness mode the robot slid along the surface on the table instead of stopping at the moment of impact. The gray contour overlay in Figure 9 depicts the robot at the moment of impact.

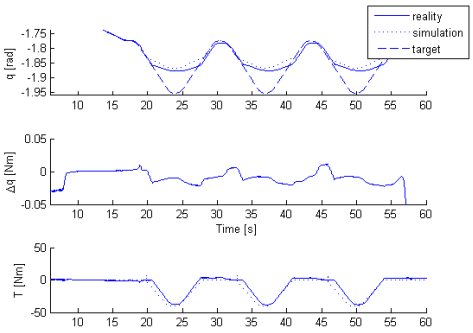


*Figure 10. Measured values of the first joint while contacting the table. Top: The position of the joint, Center: Position difference between simulated and real task, Bottom: the measured joint torque.*

Figure 10 shows the measured data of joint 1 during three repetitions of the aforementioned sliding task. Due to the impact the joint is not able to reach the target position neither in the real nor in the simulated experiment. The deviation between the simulated and real experiment stayed within 0.1 radians for all angles, and the overall standard deviation was measured to be 0.04 radians. Furthermore the shape of the graph of the simulated joint torque matches the shape of the graph of the real joint torque.

A second task was performed to examine the behavior of the peg during contacts with the rim of the hole. The robot conducted two rim-contacts with the peg throughout that task. Therefore the peg was initially positioned vertically above the hole. Furthermore the centers of the peg and the hole were slightly displaced in horizontal direction, so that the planned movement sequence of the robot would cause a rim-contact and the peg would slide over the rim during that movement. The movement se-

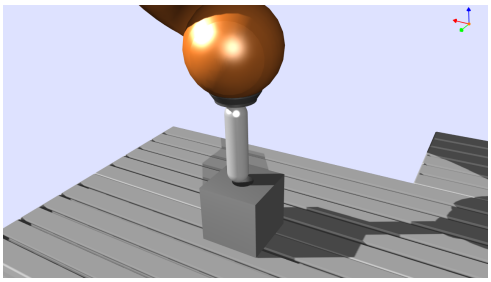

*Figure 11. The peg is sliding over the rim of the hole.*

quence of the robot was composed of the following actions. At the beginning of the sequence the peg was moved downwards to conduct the first contact. Then the peg was moved back to its initial position. Afterwards it was moved to the opposite side of the rim, to conduct the second contact. Figure 11 depicts this scenario. The hereby measured positions and Torques are shown in Figure 12.

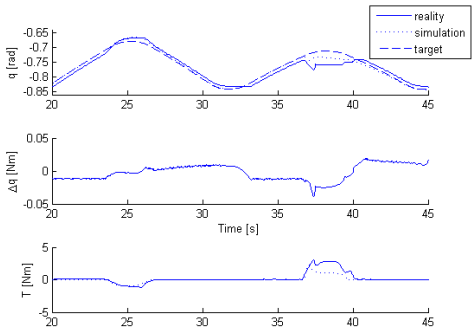


*Figure 12. Measured values of joint 6 during contact of the peg with the rim of the hole. Diagram as in Figure 10.*

The first rim-contact occurred between second 23 and 26 of the experiment, as depicted in the graphs of Figure 12. The second contact occurred between second 36 and 39. The robot configuration was chosen in such a way that the effect of the impact appeared mostly at the joint 6. However the impact is also compensated by the other joints, which leads to small deviations compared to the commanded joint positions. The deviation between the simulated and real experiment was measured to be below 0.05 radians for all joints.
Both test cases indicate that our contact simulation realizes realistic dynamic interactions between the peg and the hole shape.

## 5. CONCLUSION

We introduced a robust and fast collision detection algorithm that uses a case by case analysis to determine collisions between an interfaces centering pins and docking holes, modeled as a Peg-In-Hole scenario. The algorithm uses an analytical approach to compute the contact manifold as well as penetration depths in real-time. Therefore performance is not determined by the discretization level of the underlying polygonal models of the interface's components, as opposed to mesh based collision detection approaches.
We validated the contact simulation using a robot manipulator as measuring instrument. The exact position, as well as exerted torques have been measured with the internal position and torque sensors of the manipulators, while conducting the contact experiments. The data was used to analyze the behavior of the contact simulation in comparison to the behavior of the components in reality. The overall standard deviation of the positions of the robot's joint angles is very small. Furthermore the resemblance of the torques graphs is substantial. Both results indicate, that the behavior of our contact simulations is considerably close to the ideal behavior in reality.
As a next step we will add an additional force-torque sensor to the robot's tool. An additional force-torque sensor will provide the possibility to measure the occurring forces directly at the tool of the robot instead of the joints. Thereby the measurements will not be influenced by the robot's mechanic. Furthermore we are planning to add an automatic tool changer to the setup, that will enable us to perform automated experiment procedures on a larger scale. Based on the generated data we will calibrate the simulation algorithms and identify the hardware behavior in detail.

## ACKNOWLEDGMENTS

Parts of this work were developed in the context of the research project iBOSS-2. Supported by the German Aerospace Center (DLR) with funds of the German Federal Ministry of Economics and Technology (BMWi), support codes 50 RA 1203.

## REFERENCES

[1] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schaeffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, and G. Hirzinger. The kuka-dlr lightweight robot arm - a new reference platform for robotics research and manufacturing. In *Proc. Robotics (ISR), 41st International Symposium on and 6th German Conference on Robotics (ROBOTIK)*, pages 1–8, 2010.

[2] M. Goeller, J. Oberlaender, K. Uhl, A. Roennau, and R. Dillmann. Modular robots for on-orbit satellite servicing. In *Proc. IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2018–2023, 2012.

[3] N. Hogan. Stable execution of contact tasks using impedance control. In *Proc. IEEE International Conference on Robotics and Automation*, pages 1047–1054, 1987.

[4] T. Jung. *Methoden der Mehrkrperdynamiksimulation als Grundlage realittsnaher Virtueller Welten*. PhD thesis, RWTH Aachen University, 2011.

[5] E. G. Kaigom and J. Rossmann. Developing virtual testbeds for intelligent robot manipulators-an

erobotics approach. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1589–1594, 2014.

[6] M. Mandiak and T. Kesavadas. Development of virtual assembly application with haptics, assembly modification and statistical output measures. In *Proc. ASME International Mechanical Engineering Congress and Exposition*, pages 1475–1481, 2005.

[7] J. Rossmann and M. Schluse. Virtual robotic testbeds: A foundation for e-robotics in space, in industry-and in the woods. In *Proc. IEEE Developments in E-systems Engineering (DeSE)*, pages 496–501, 2011.

[8] J. Rossmann, M. Schluse, C. Schlette, and R. Waspe. Control by 3d simulation–a new erobotics approach to control design in automation. In *Intelligent Robotics and Applications*, pages 186–197. Springer, 2012.

[9] J. Rossmann, T. Steil, and M. Springer. Validating the camera and light simulation of a virtual space robotics testbed by means of physical mockup data. In *Proc. International symposium on artificial intelligence, robotics and automation in space (i-SAIRAS)*, pages 1–6, 2012.

[10] T. Tsuruoka, H. Fujioka, T. Moriyama, and H. Mayeda. 3d analysis of contact in peg-hole insertion. In *Proc. IEEE International Symposium on Assembly and Task Planning (ISATP)*, pages 84–89, 1997.

[11] J. Weise, K. Briess, Adomeit, A., H.-G. Reimerdes, M. Gller, and R. Dillmann. An intelligent building blocks concept for on-orbit-satellite servicing. In *Proc. International Symposium on Artificial Intelligence Robotics and Automation in Space (iSAIRAS)*, Turin, Italy, 2012.

[12] J. Zhou, N. Georganas, E. Petriu, X. Shen, and F. Marlic. Modelling contact forces for 3d interactive peg-in-hole virtual reality operations. In *Proc. Instrumentation and Measurement Technology Conference Proceedings (IMTC)*, pages 1397–1402, 2008.