The DLR On-Orbit Servicing Testbed


Robotics and Mechatronics Center, DLR

VR-OOS Workshop 2012

DLR, Oberpfaffenhofen
Talk outline

- The DEOS Project
- The DEOS Simulator goals and requirements
- The DEOS Simulator concept
- Conclusion
DEOS Deutsche Orbitale Servicing Mission

Goals:
• develop and verify robotic technology for the capture a tumbling non-cooperative Client satellite with a Servicer spacecraft and
• to de-orbit the coupled configuration in a pre-defined corridor at end of mission
DEOS Deutsche Orbitale Servicing Mission

Motivation:

• Satellite servicing in LEO and GEO
• Active Debris Removal
• Astronaut assistance
• On-orbit assembly
Space Debris Projections

**LEO Environment Projection** (averages of 100 LEGEND MC runs)

- **Reg Launches + 90% PMD**
- **Reg Launches + 90% PMD + ADR2020/02**
- **Reg Launches + 90% PMD + ADR2020/05**

Effective Number of Objects ($>10$ cm)

Year


- 2 objects per year removed
- 5 objects per year removed

PMD – post-mission disposal    ADR – active debris removal

DEOS Deutsche Orbitale Servicing Mission

System description:
• **Servicer** satellite
• 7 DOF Manipulator
• **Client** satellite with no appendages
DEOS Deutsche Orbitale Servicing Mission

Operational conditions:

• Servicer attitude control inactive
• Client motion constrained to limited range of free tumbling states
DEOS Deutsche Orbitale Servicing Mission

Robot control modes for the grasping task:

• Tele-presence
• Semi-autonomy
DEOS Communication to ground
The DEOS Simulator

Goal

• Develop a Servicer-Client test-bed for testing and validating grasping and stabilization strategies

• Identify the best possible strategies
Requirements (1)

- Must simulate free floating dynamics for both Servicer and Client as realistic as possible
- Must integrate a robotic arm with similar characteristics to the DEOS arm
- Must simulate all possible contact dynamics between all the elements
- Must have a large enough workspace
- Must be up-scalable
Requirements (2)

- Must be a development platform
- Must help to identify critical points
- Must integrate similar sensors
- Must use identical software and operative system
The **Core** is the software which will be developed in the Simulator and used in the Mission. It includes:

- A Communication framework
- A robot control framework
- Video processing (?)
- Autonomy features (FDIR)

The **Frame** is the environment where the Core runs. Possible Frames are:

- Mission (Space)
- Virtual (model based)
- Laboratory (robots based)

The **Interface** adapts the Frame such that the Core remains identical.

Premise: **The Core is identical for all possible Frames.**
**Autonomous Mode**: Decomposition of complex task in a set of elementary actions.

**Controller**: Compliant trajectories. Tracking. Stabilization methods.

**FDIR**: Collision avoidance

**HIROSCO**: Software and communication framework for space applications.

**Real Time OS**: VxWorks
The **CORE**: Telepresence Mode

**TOP** (Task Oriented Programming): Decomposition of complex task in a set of elementary actions.

**Controller**: Compliant trajectories. Tracking. Stabilization methods.

**FDIR**: Collision avoidance

**HIROSCO**: Software and communication framework for space applications.

**Real Time OS**: VxWorks
Frames

- **Environment:** Space
- **Carrier:** Spacecraft
- **Arm:** Deos Arm

- **Environment:** Laboratory
- **Carrier:** Industrial robot
- **Arm:** Leight Weight Robot
Frames

- **Environment:** Laboratory
- **Carrier:** Light Weight Robot (LWR)
- **Arm:** Light Weight Robot

- **Environment:** Virtual
- **Carrier:** Light Weight Robot
- **Arm:** Light Weight Robot
Frames

- Environment: Laboratory
- Master: DLR - HMI
- Slave: Space Justin
## Frames Mapping

<table>
<thead>
<tr>
<th>Frame</th>
<th>Mission</th>
<th>DeosSim</th>
<th>LWRs</th>
<th>Virtual</th>
</tr>
</thead>
<tbody>
<tr>
<td>Servicer</td>
<td>LEO Satellite 1000 Kg</td>
<td>KR120 (1500Kg) Simulated: 300Kg</td>
<td>KR4 (17Kg) Simulated: 100Kg</td>
<td>Virtual Satellite</td>
</tr>
<tr>
<td>Client</td>
<td>LEO Satellite 350 Kg</td>
<td>KR120 (1500Kg) Simulated: 140Kg</td>
<td>KR4 (17Kg) Simulated: 45Kg</td>
<td>Virtual Satellite</td>
</tr>
<tr>
<td>Arm</td>
<td>DEOS Arm</td>
<td>KR4+ (17Kg)</td>
<td>KR4 (17Kg)</td>
<td>Virtual DEOS Arm</td>
</tr>
<tr>
<td>Gripper</td>
<td>DEOS Gripper</td>
<td>Schunk PG70</td>
<td>Schunk PG70</td>
<td>DEOS Gripper</td>
</tr>
<tr>
<td>Camera</td>
<td>DEOS Cam</td>
<td>Prosilica GigE</td>
<td>- (Camera Simulation)</td>
<td></td>
</tr>
<tr>
<td>Stars Sensor</td>
<td>DEOS Sensor</td>
<td>IMU</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Force Sensor</td>
<td>Joint Trq Sensor</td>
<td>DLR FTS</td>
<td>DLR FTS</td>
<td>FTS Simulation</td>
</tr>
</tbody>
</table>
The Mission Frame

- DEOS Arm
- DEOS Gripper
- Client Satellite
- Grasping points
- Servicer Satellite
- Camera
- Stars Sensor, Magnetometer
- Antenna
Free-floating robot control modes

- **Two philosophies:**
  - **Fixed base**
    - Simple: control = grounded robot arm.
    - But: highly energy consuming for the base.
  
  - **Frees base**
    - Complex: robot control must take into account free body dynamics.
    - But: energy efficient, since the base cooperates with the task.
The Light-Weight-Robot Frame

Servicer (LWR)

Force-Torque Sensor

Client (LWR)

Force-Torque Sensor

Satellite Mockup

Industrial Gripper
Simulating free-floating dynamics with two fixed-base robots

Base motion of the Servicer is projected onto the Client
The DeosSim Frame

- Force-Trq Sensor
- IMU (Stars Sensor)
- Servicer Mockup
- Arm (KR4+)
- Gripper
- Force-Trq Sensor
- Client Mockup

Servicer (KR120)  Client (KR120)
Conclusions

- Different Simulation Frames have been presented
- Each Frame is used for testing and verifying different elements of the real mission
- The Core remains identical in each frame
- The simulator represents the design platform for the real mission
- Other uses for the simulator are planned, e.g. satellites with solar panels or robot manipulators mounted on helicopters
Thanks!