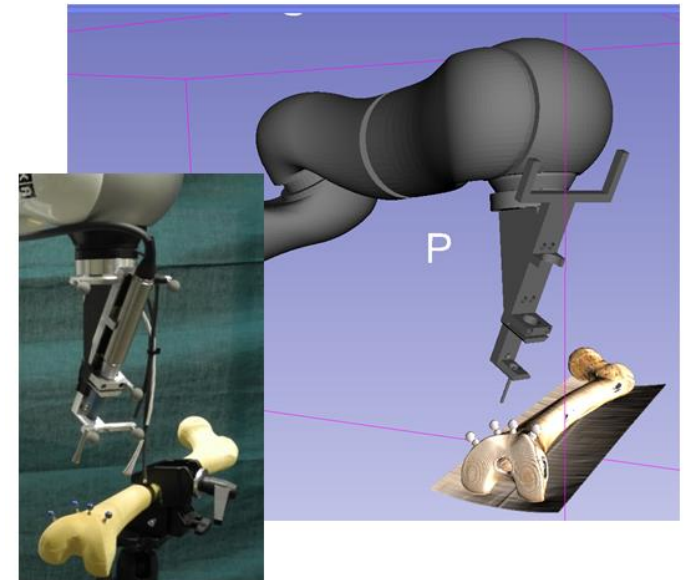
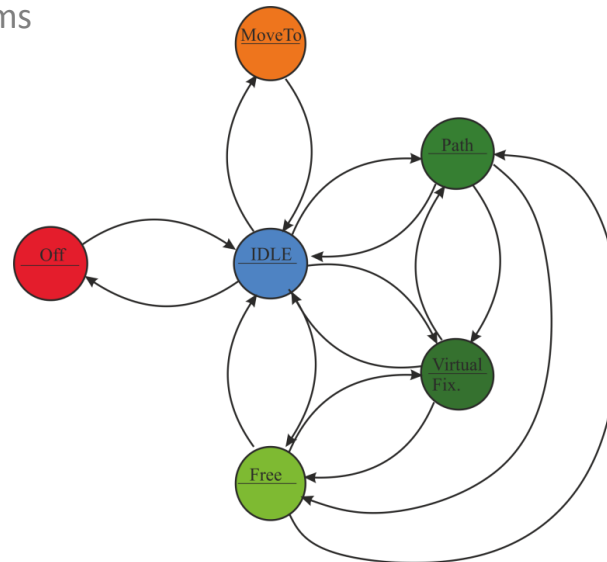


Tutorial

LightWeightRobotIGT – Interface Concept

Sebastian Tauscher
Institute of Mechatronic Systems
Leibniz Universität Hannover



Outline

Introduction

Interface Concept

Visualisation

State control

State machine

Motivation

Integration of robots into IGT systems

- Time consuming process
- Customized solution for each project
- No widely accepted standard

Future vision

- Standardized interface
- ⇒ Easy integration



[Source: openIGTlink.org]



[Source: KUKA]

State of the art

Current research

- IHE & DICOM for surgery
 - OR.NET
 - OpenIGTLink
- ⇒ Growing importance

Unmet needs

- No widely accepted standard for inter device communication

Aim

- Integration concept for a robot into IGT system



[Source: openIGTlink.org]



[Source: KUKA]

Outline

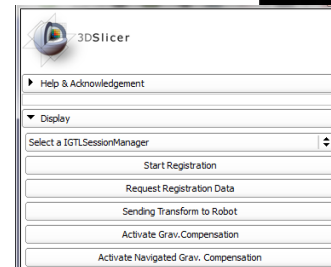
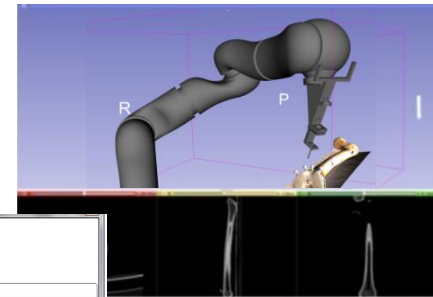
Introduction

Interface Concept

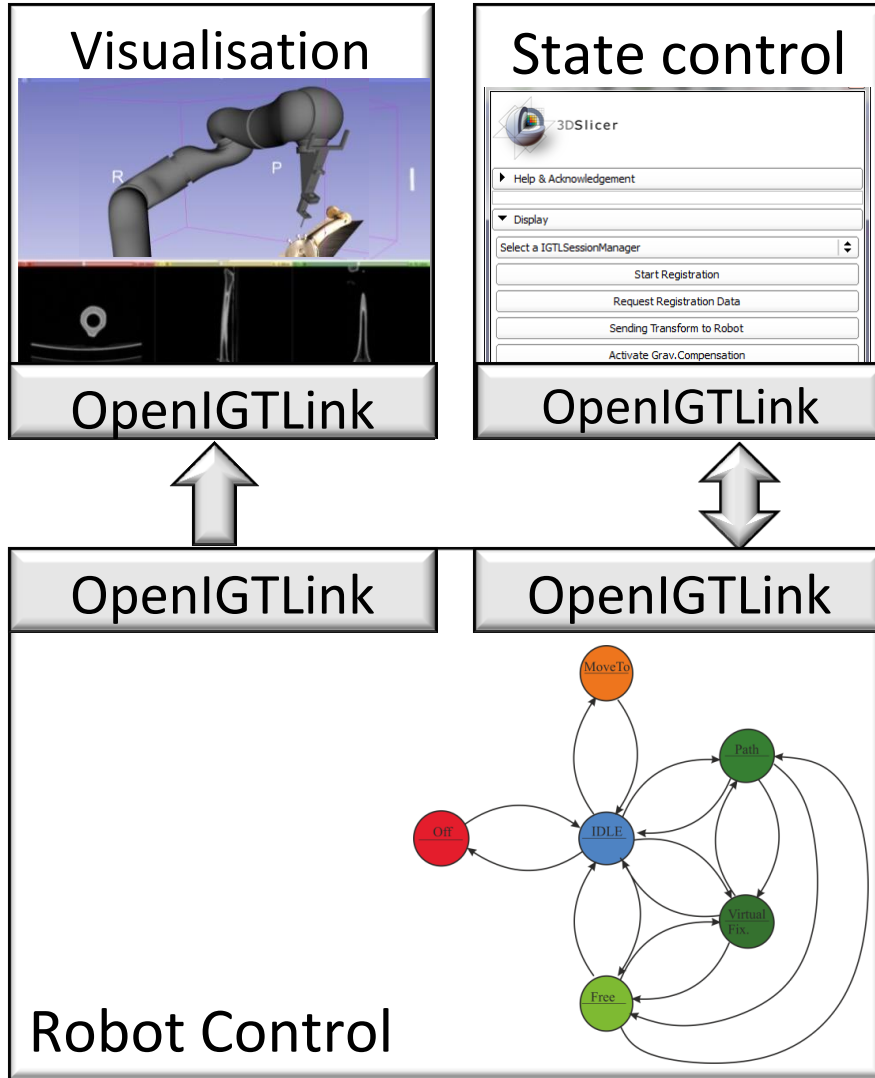
Visualisation

State control

State machine



Interface – Concept



Robot as element of IGT system

Separate visualisation & state control interface

OpenIGTLink based

- Small foot print & widely used
- Open protocol for IGT

State machine for intuitive and direct control

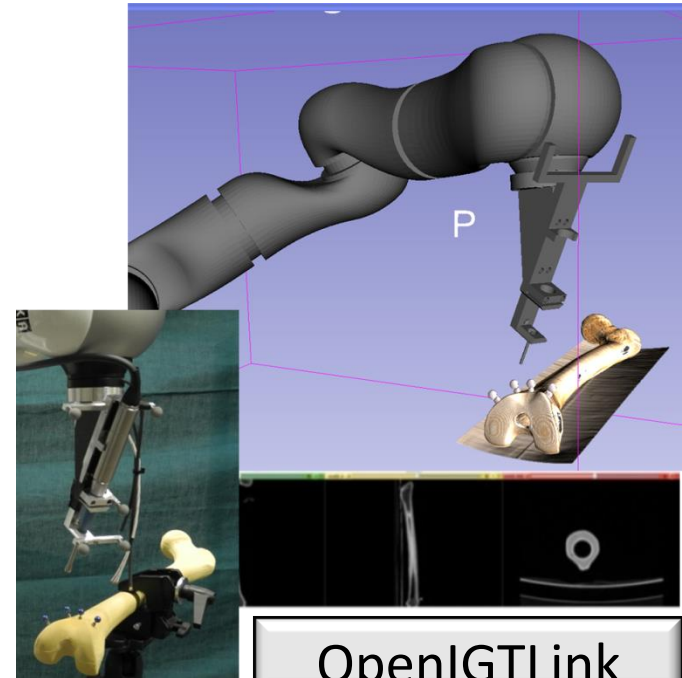
Interface – Visualisation

Requirements

- Real-time ability > 30 Hz
- Latency < 100 ms
- Unidirectional

Data type

- Transformation matrices
 - Image space or robot coordinate system
 - End-effector or joint pose
- ⇒ OpenIGTLink TRANSFORM



OpenIGTLink



OpenIGTLink

Robot Control

Interface – State control

Requirements

- Reliability
- „Is Alive“ check
- Bidirectional

Data type

- Command string + parameter (p)
 - Unified identifier (UID)
- ⇒ OpenIGTLink STRING

Data format

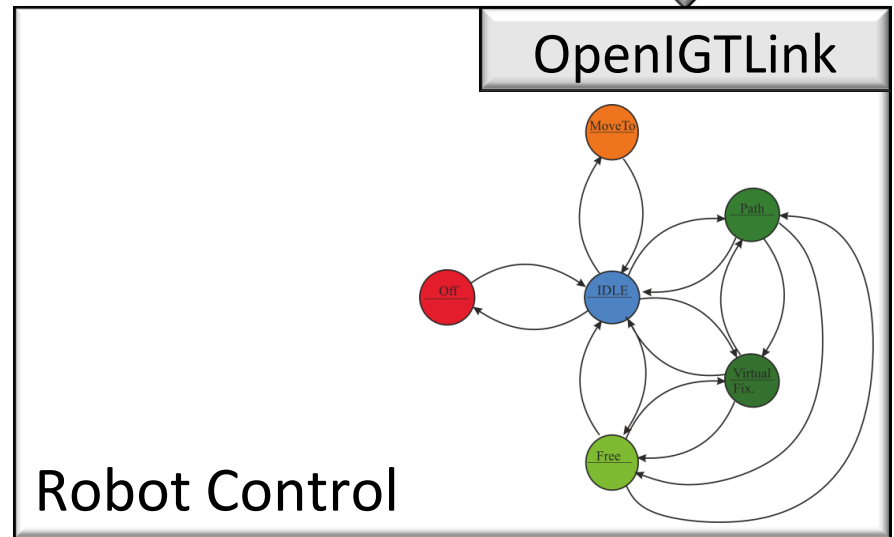
- Device name: “CMD_UID”/”ACK_UID”
- String: “*Commandname;p1;...;pn;*”



OpenIGTLink



OpenIGTLink



Robot Control

Outline

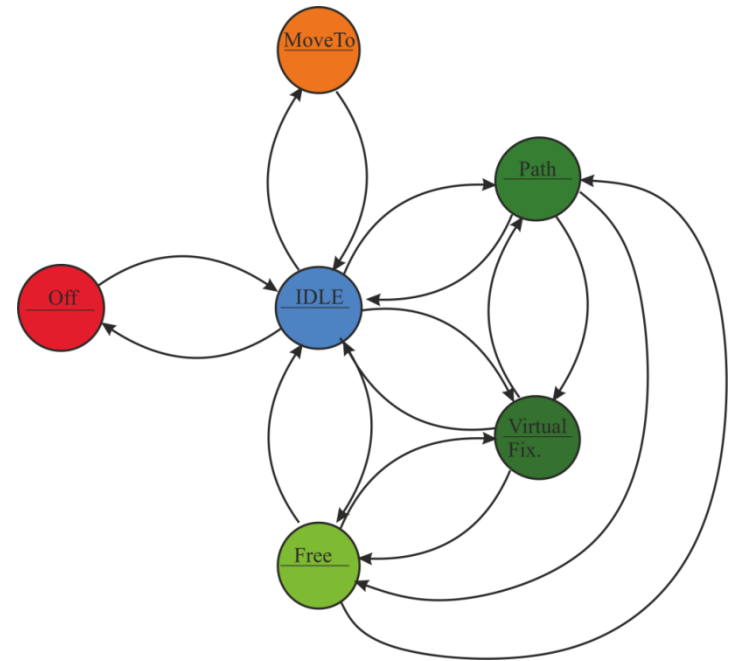
Introduction

Interface Concept

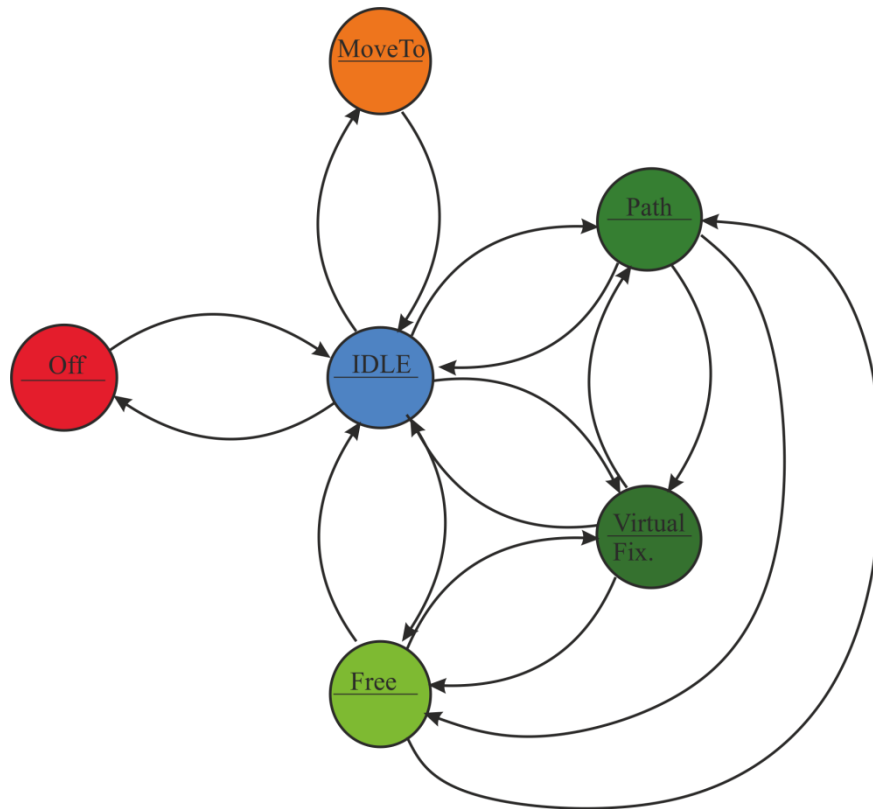
Visualisation

State control

State machine



State machine – Concept



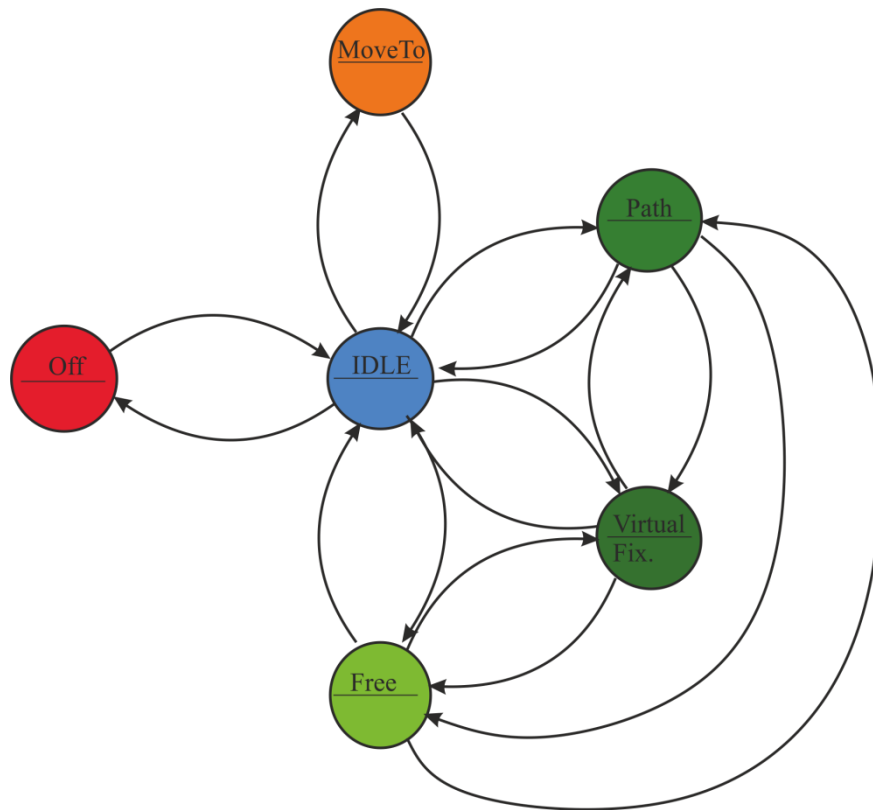
Robotic workflow \neq surgeon's workflow

Direct and intuitive control of robot states

States for IGT

- Registration (**Free**)
- Pre-Positioning (**Path**, **VirtualFixtures**, **MoveTo**)
- Targeting (**MoveTo**)
- Save (**IDLE**)

State machine – Concept



Registration (**Free**)

- Robot can be moved freely and manually

Pre-Positioning (**Path**, **VirtualFixtures**, **MoveTo**)

- Robot can be moved freely in restricted volume
- Geometries: plane, cone and path

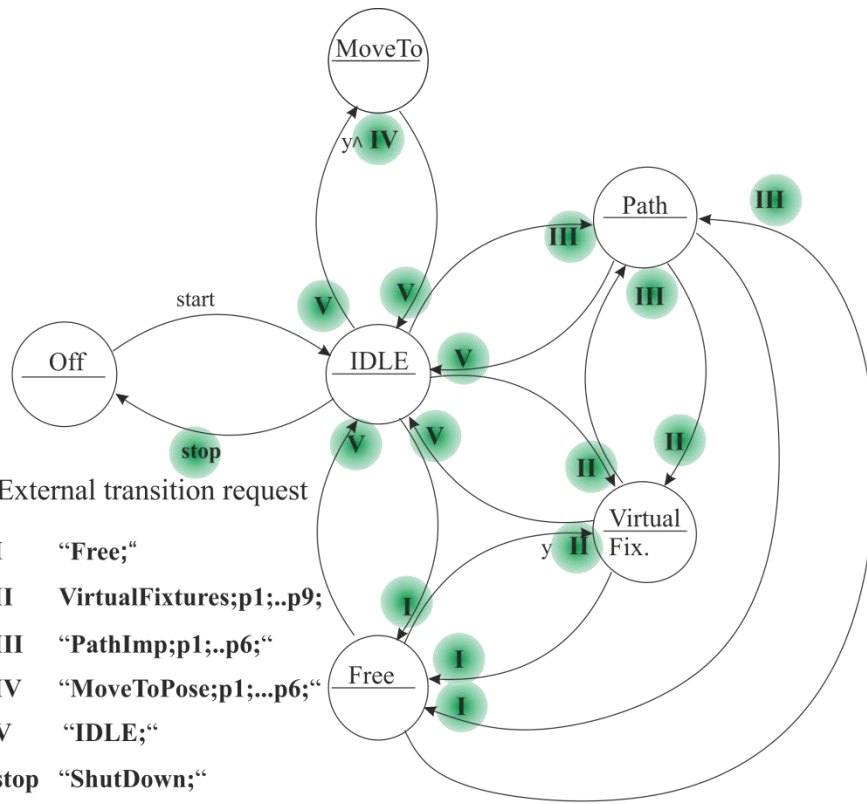
Targeting (**MoveTo**)

- Robot is moving position controlled to target position

Save (**IDLE**)

- Robot holds current position with maximum stiffness

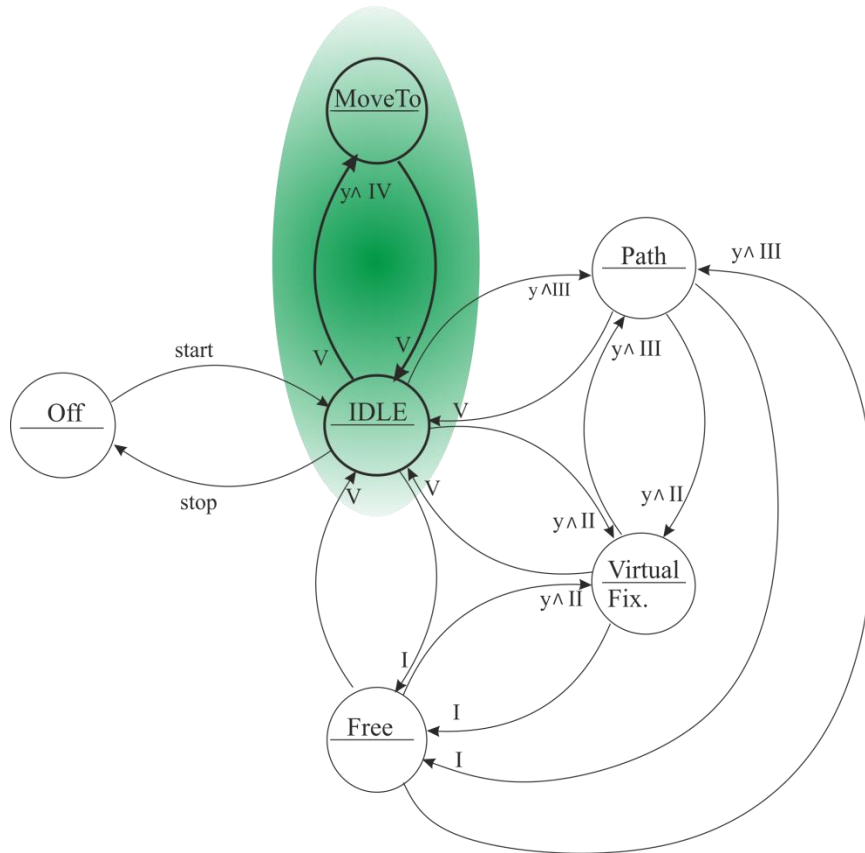
State machine – Transitions



Transitions

- By external request besides in case of error

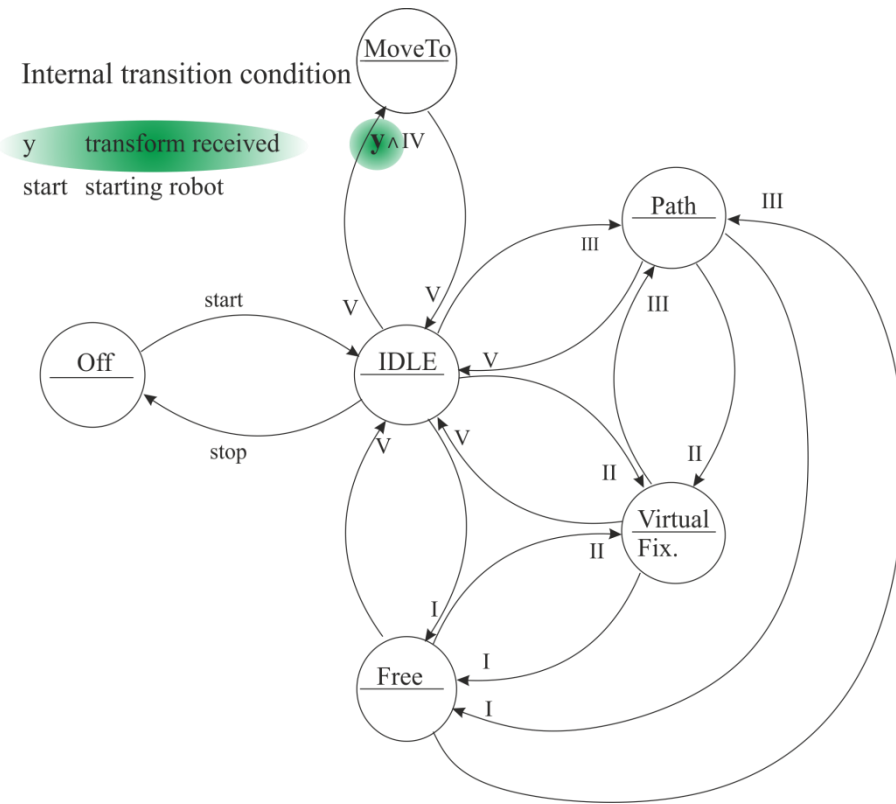
State machine – Transitions



Transitions

- By external request besides in case of error
- From position to impedance controlled states only via IDLE

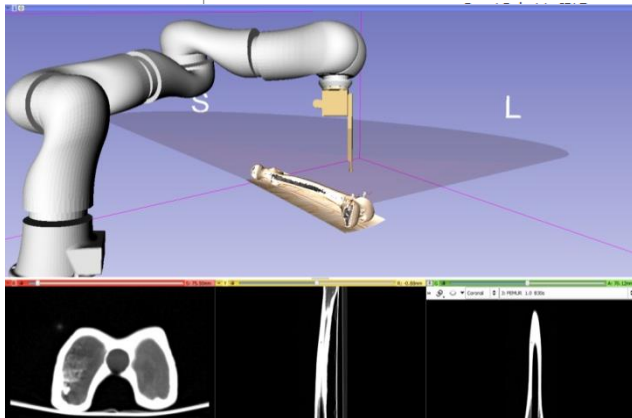
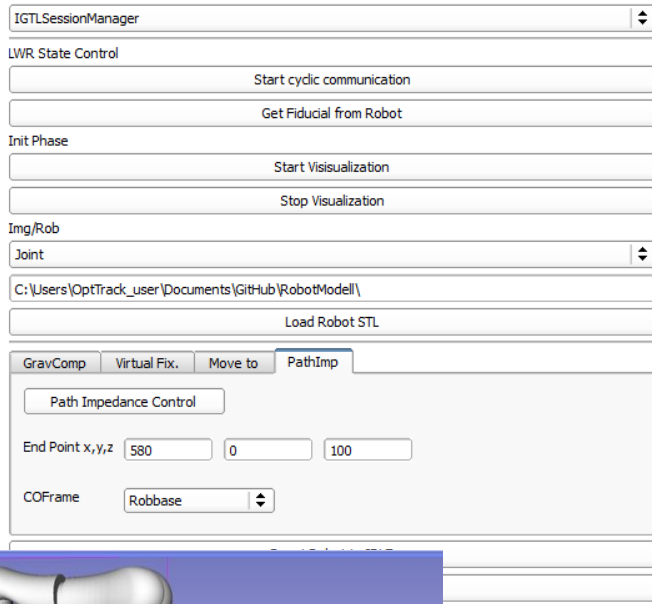
State machine – Transitions



Transitions

- By external request besides in case of error
- From position to impedance controlled states only via IDLE
- To targeting state only after registration

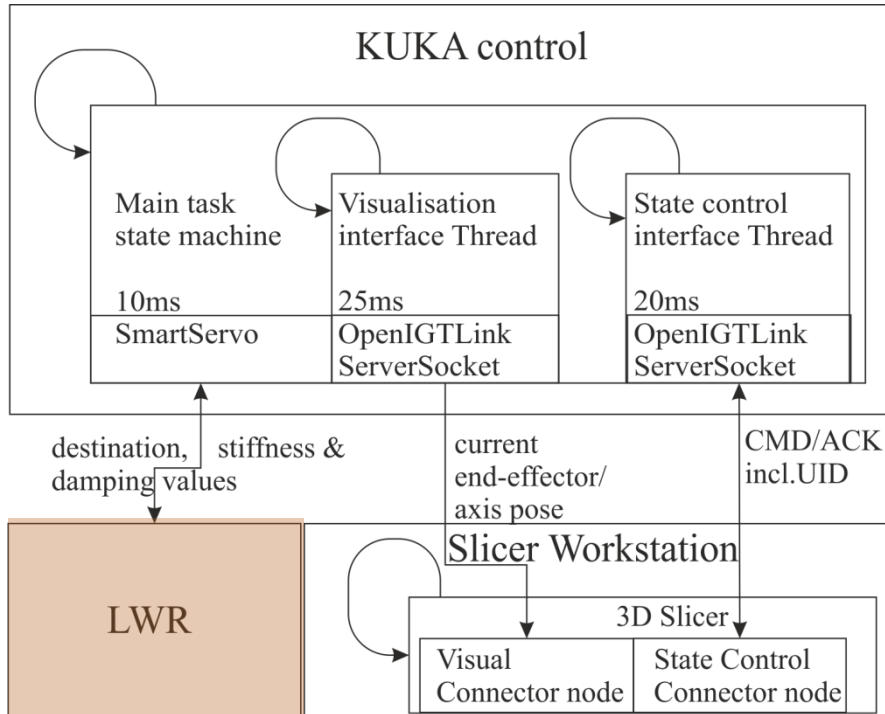
State machine – Control



3D Slicer module *LightWeightRobotIGT*

- Sending command strings
- Editing parameters for states
- Acquiring points for point based registration
- Visualisation of robot model, active virtual fixtures, force vector at TCP, and current robot state
- Cyclic communication

Test setup



Slicer workstation

- 3D Slicer module
LightWeightRobotIGT as state control

KUKA control

- Java robot application
 - Visualisation & state control interface thread
 - State machine thread

KUKA Light weight robot (LWR)

Tutorial

LightWeightRobotIGT – Interface Concept

Sebastian Tauscher
Institute of Mechatronic Systems
Leibniz Universität Hannover

