Tutorial

LightWeightRobotIGT – Getting Started

Sebastian Tauscher
Institute of Mechatronic Systems
Leibniz Universität Hannover
Sebastian.Tauscher@imes.uni-hannover.de
Outline

System Overview

Requirements

Set up KUKA Sunrise control

Install Example

Install LightWeightRobotIGT

Run Example
System Overview - 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**
System Overview

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)
System Requirements

Requirements

- Robotic system
  - KUKA sunrise control & LWR iiwa
  - KUKA Sunrise.Connectivity Smart Servo Motion Extension
  - Notebook/Desktop PC with Sunrise.Workbench 1.0 or higher
  - LWROpenIGTIF package including exemplary state machine

- Slicer Workstation
  - 3D Slicer 4.4 64 Bit
  - See http://www.slicer.org
  - Point-to-point ethernet connection to Sunrise control (use the same notebook/desktop PC for the Sunrise Workbench and 3D Slicer)
Outline

System Overview

Requirements

Set up KUKA Sunrise control

Install Example

Install LightWeightRobotIGT

Run Example
Installing LWROpenIGTIF

- Download LWROpenIGTIF at GitHub including STL-files of the LWR (RobotModel)
  - [https://github.com/tauscherSw/LWROpenIGTIF.git](https://github.com/tauscherSw/LWROpenIGTIF.git)
- Add the LWROpenIGTIF and the OpenIGTLink folder to your sunrise project source folder
  - Example: C:\Devel\KUKA\YourSunriseProject\src
- Add the OpenIGTLinkLib folder to the project folder
  - Example: C:\Devel\KUKA\YourSunriseProject
- Add the SimpleStateExample.java file to the application source folder
  - Example: C:\Devel\KUKA\YourSunriseProject\src\application
- Install your sunrise project
- Synchronize the sunrise workbench with the sunrise control
Installing LWROpenIGTIF

- Set path to the SWIGigtutil.dll directory in the LWRVisualizationIF class (line 60)
  - `System.load("PATH/SWIGigtutil.dll");`
  - Path is: 
    `C:\KRC\ApplicationServer\Git\YourSunriseProjectName\OpenIGTLinkLib`

- Copy the STL-folder somewhere on your Slicer Workstation
  - Example: `C:\Program Files\Slicer 4.4.0-2014-11-24\RobotModel\`

- Software documentation of the LWROpenIGTIF classes can be found here:
  - [https://github.com/tauscherSw/LWROpenIGTIF.git](https://github.com/tauscherSw/LWROpenIGTIF.git)
Before running the example you should

- Change the current tool data in the SimpleStateExample.java according to the load and geometry of your tool
- Check if the default start position $q = \{ 0.0, 30.0, 0.0, -60.0, 0.0, 90.0, 0.0 \}$ is safe. **WARNING: There is no safety check!**
- Have experience with the robot and the robot control

Before starting the example you can

- Adjust the cycle times of the different threads according to your needs
Outline

System Overview

Requirements

Set up KUKA Sunrise control

Install Example

Install LightWeightRobotIGT

Run Example
Installing LightWeightRobotIGT

- Download 3D Slicer 4.4 64-Bit Version
- Install 3D Slicer
- Install LightWeightRobotIGT Extension using the Extension Manager
Outline

- System Overview
- Requirements
- Set up KUKA Sunrise control
- Install Example
- Install LightWeightRobotIGT
- Run Example
LightWeightRobotIGT – Run the Example

**Start Connection**

Open 3D Slicer on the Slicer Workstation
Open LightWeightRobotIGT module
- Modules->IGT->LightWeightRobotIGT
- Set path to the folder containing the STL-files
- Check if the IP-address and ports of the robot control is set correctly; default is 192.168.42.2 (Modules->IGT->OpenIGTIF)

Start *YourProject* on the robot control
- Check if two interfaces and the state machine were successfully started on the SmartPad (Control Panel of the robot control)
Start Visualization of Robot and Force Vector

Click Start cyclic communication (yellow)
- Cyclic communication with the state control is now active

Click Load Robot STL (blue)
- Now the robot should be visualized in the 3D view
- Robot colour changes due to the current state (colour coding see state machine description in Tutorial: LightWeightRobotIGT-Introduction)

Click Show TCP Force (green)
- A 3D arrow is now shown at the tool center point
LightWeightRobotIGT – Run the Example

Switching between states

Click on different states as you like

- **WARNING:** Check if the default position of the virtual fixtures is safe and that the robot is not in the locked zone of the virtual fixtures

Now you can:

- Change to Free, Idle, Path and VirtualFixtures
- Visualise active Virtual Fixtures in Slicer (see figures)
- **NOT** set the robot to MoveTo state; Therefore, you need to register/send the T_CT_Base transform to the state control (see next page)
Example of Point based Registration

- Load Dicom data/STL file of your target object
- Define fiducials in a fiducial list using Create-Fiducial
- Set robot to Free mode (click Free in registration tab)
- Move robot to fiducial of physical object
- Click Get Fiducial from robot (in registration tab)
- Repeat this step for all fiducials

Info: For Registration you need a target object and CT-Data of this objects! Furthermore landmarks are needed to obtain the transformation by a point based registration
Registration – Workflow II

- All points are saved in the *Fid_list* annotation node
- Use *Fiducial Registration* module (see figure)
  - Fixed landmarks: *F*
  - Moving landmarks: *Fid_List*
  - Save transform => *T_CT_Base*
- The Matrix *T_CT_Base* is automatically send to the robot control when its value is changed
- Check if the registration was successful
  - Robot is now visualized in relation to target
  - MoveTo enabled (carefully use this state!!)

---

**Info:** For Registration you need a target object and CT-Data of this objects! Furthermore landmarks are needed to obtain the transformation by a point based registration
Visualisation concept - Force

Force representation with similar transform

- Force COF
  - z-axis in force direction
  - at tool centre point
- Transformation from robot base to force coordinate frame
- Scale: force magnitude
Visualisation concept - State

Change of robot colour according to the acknowledge string

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

⇒ Intuitive and direct feedback on current state and success of transition request
Visualisation concept - State

Visualisation of active Virtual Fixtures

- Geometries: plane, cone or path
- Change of colour due to active current zone
  - Free
  - Aware
  - Locked
Tutorial

LightWeightRobotIGT – Getting Started

Sebastian Tauscher
Institute of Mechatronic Systems
Leibniz Universität Hannover
Sebastian.Tauscher@imes.uni-hannover.de