

Other RTOS services

Embedded Motion Control 2012

Group 7:

Siddhi Imming

Bart Moris

Roger Pouls

Patrick Vaes



TU / **e**

Technische Universiteit
Eindhoven
University of Technology

Eindhoven, May 29, 2012

Where innovation starts

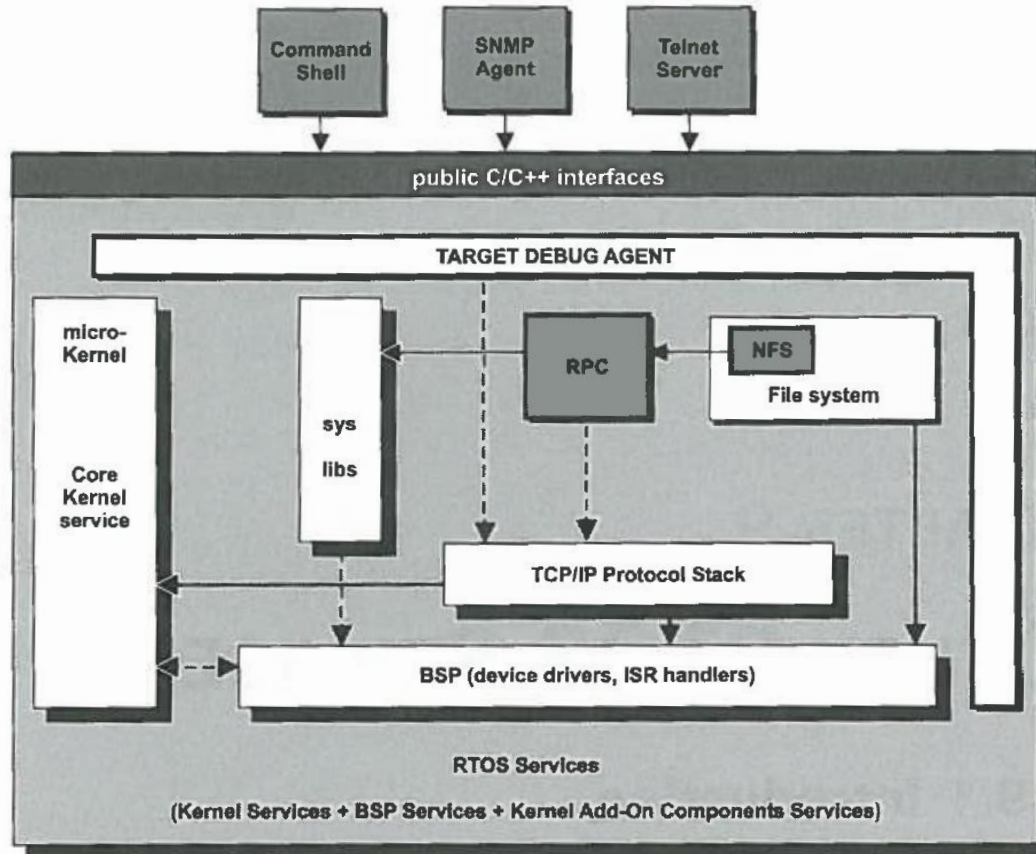
Content

- **Other RTOS services**
- **Connecting two nodes**
- **ROS debugging tools**
- **Gmapping**

Other RTOS services

- **Definition of service in the book is not the same as the definition in ROS**
- **Micro-kernel**
 - **Adjustability**

Other RTOS services



Other ROS services

- **File system component**

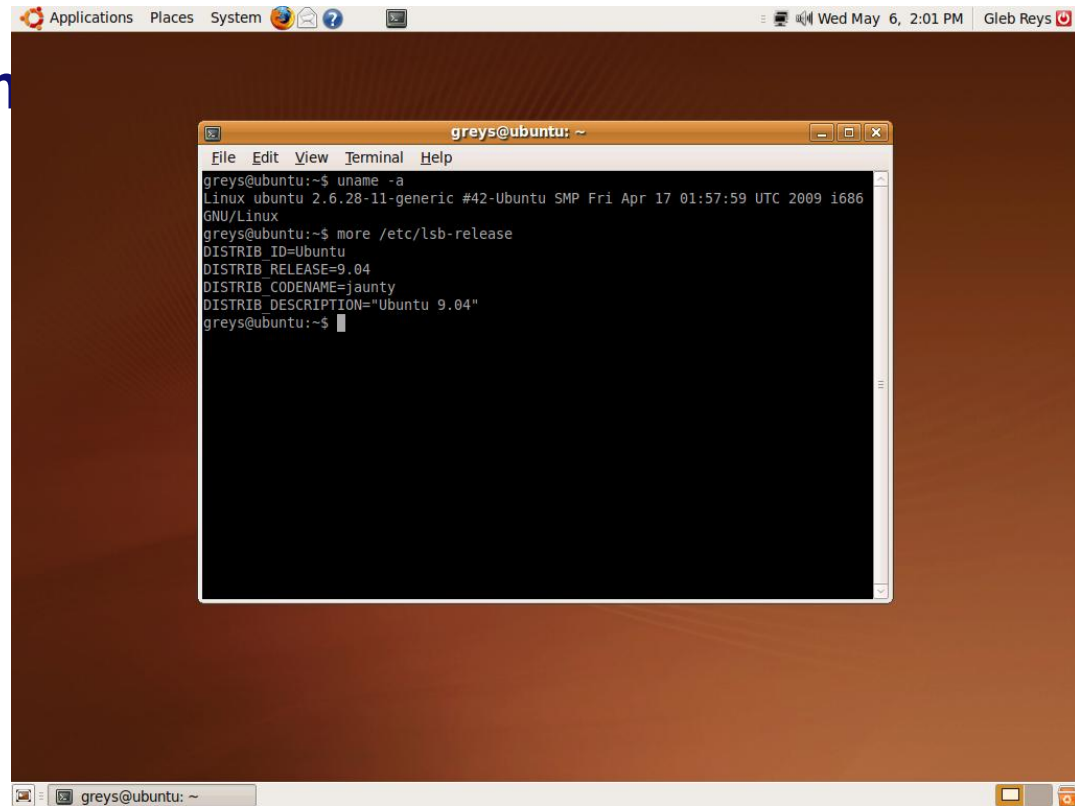
Other ROS services

- File system component



Other ROS services

- File system com
- Command shell



The image shows a screenshot of a Linux desktop environment. The top panel displays the system menu with 'Applications', 'Places', and 'System' options. The system tray on the right shows the date and time as 'Wed May 6, 2:01 PM' and the user 'Gleb Reys'. A terminal window titled 'greys@ubuntu: ~' is open, showing the following commands and output:

```
greys@ubuntu:~$ uname -a
Linux ubuntu 2.6.28-11-generic #42-Ubuntu SMP Fri Apr 17 01:57:59 UTC 2009 i686
GNU/Linux
greys@ubuntu:~$ more /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=9.04
DISTRIB_CODENAME=jaunty
DISTRIB_DESCRIPTION="Ubuntu 9.04"
greys@ubuntu:~$
```

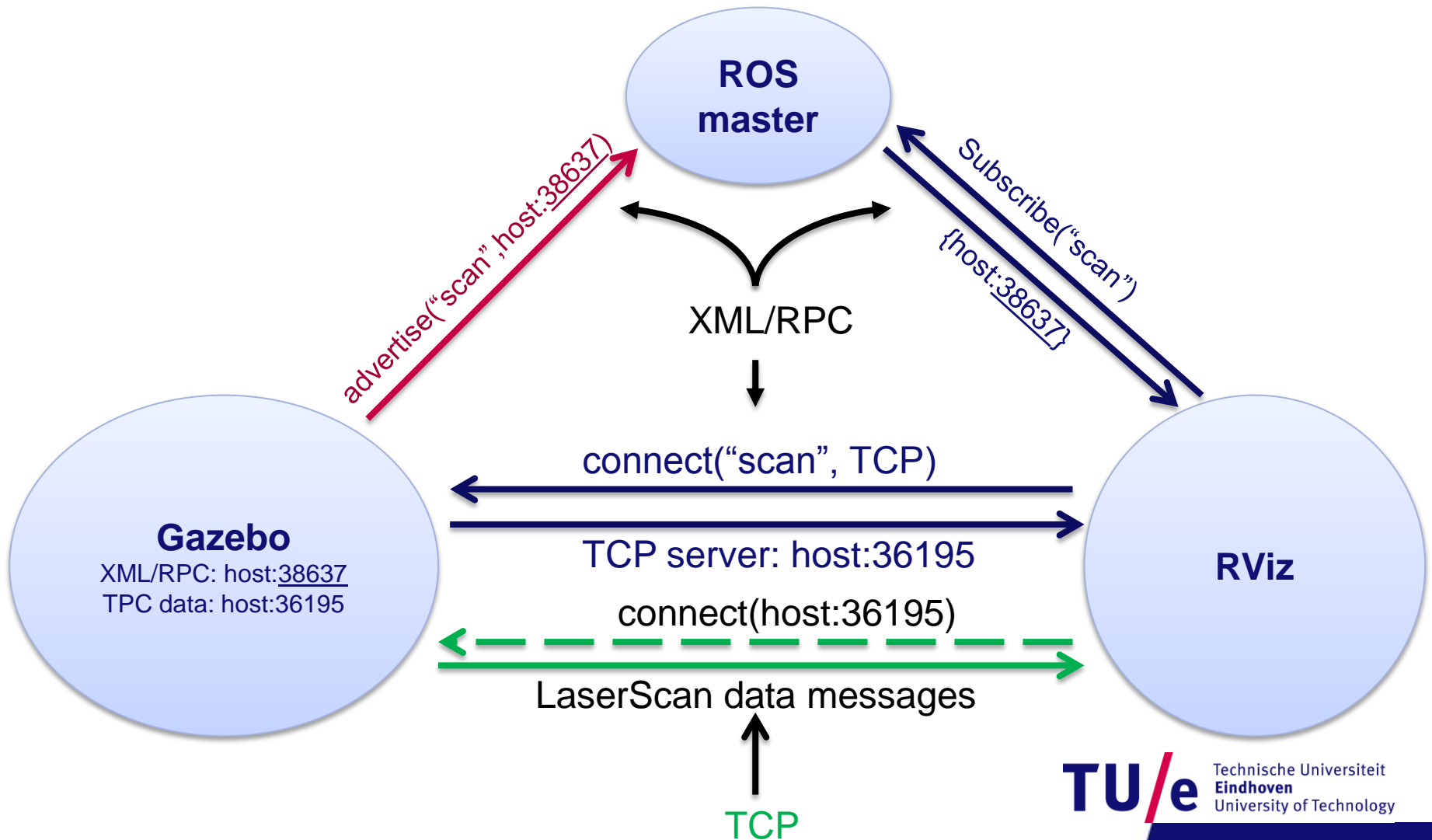
Other ROS services

- **File system component**
- **Command shell**
- **Target debug agent**

Other ROS services

- **File system component**
- **Command shell**
- **Target debug agent**
- **TCP/IP protocol stack**
- **RPC component**

Connecting two nodes



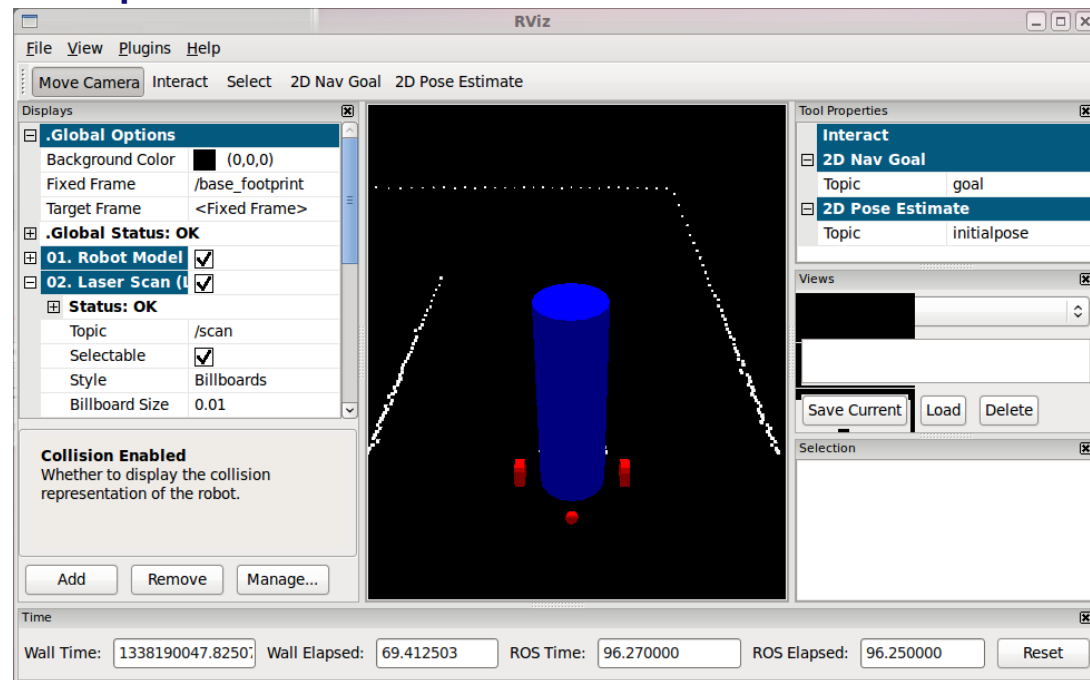
ROS debugging tools

- **Rviz**
- **RXconsole**
- **RXgraph**
- **RXplot**
- **RXbag and Rosbag**
- **Command line tools**

ROS debugging tools

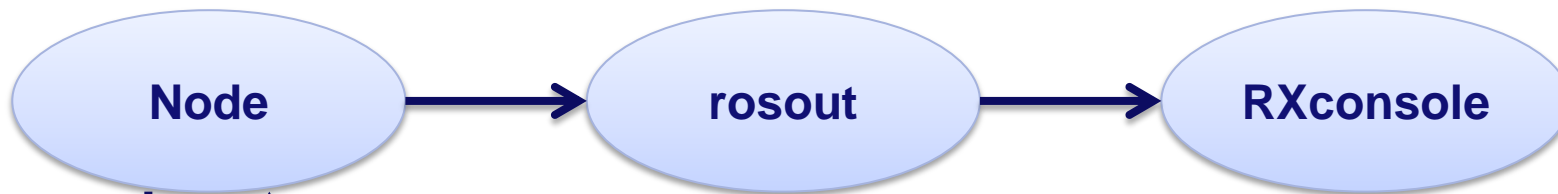
Rviz

- Visualizes messages.
- Can subscribe to multiple topics.
- Can publish messages on topics.



ROS debugging tools

RXconsole



GetLoggerLevel

1. Debug
2. Info
3. Warn
4. Error
5. Fatal

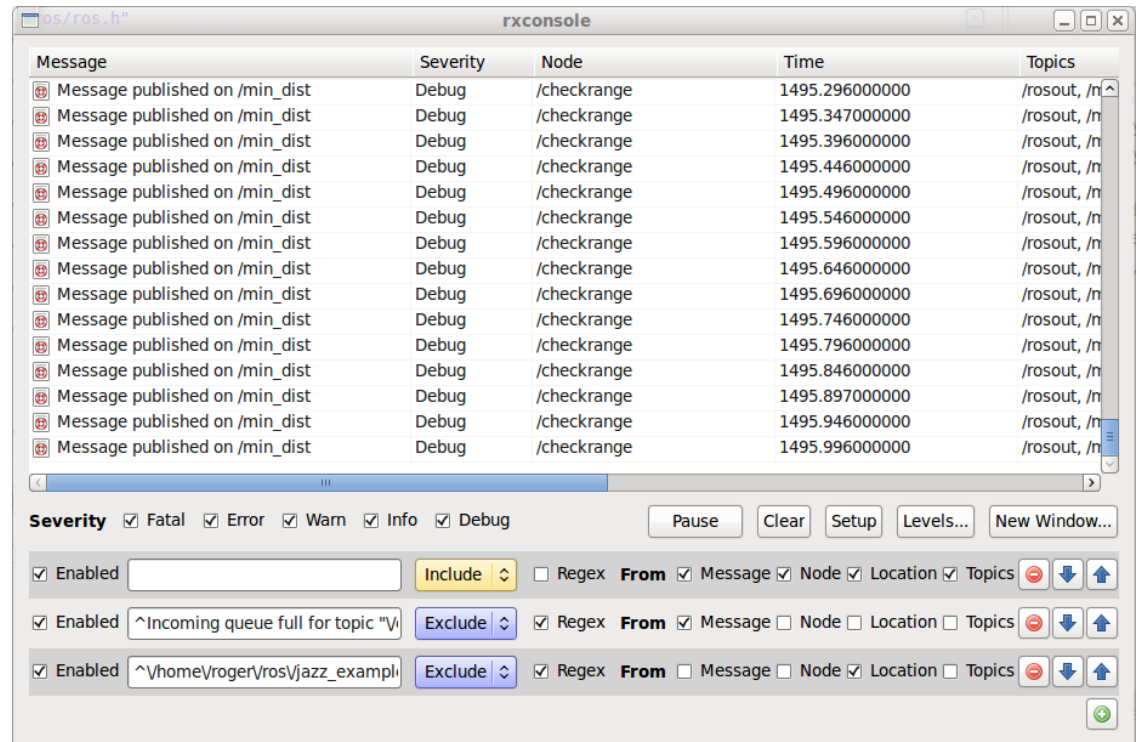
Message:

- Message
- System time
- Message level
- Node
- Topics

ROS Debugging tools

RXconsole

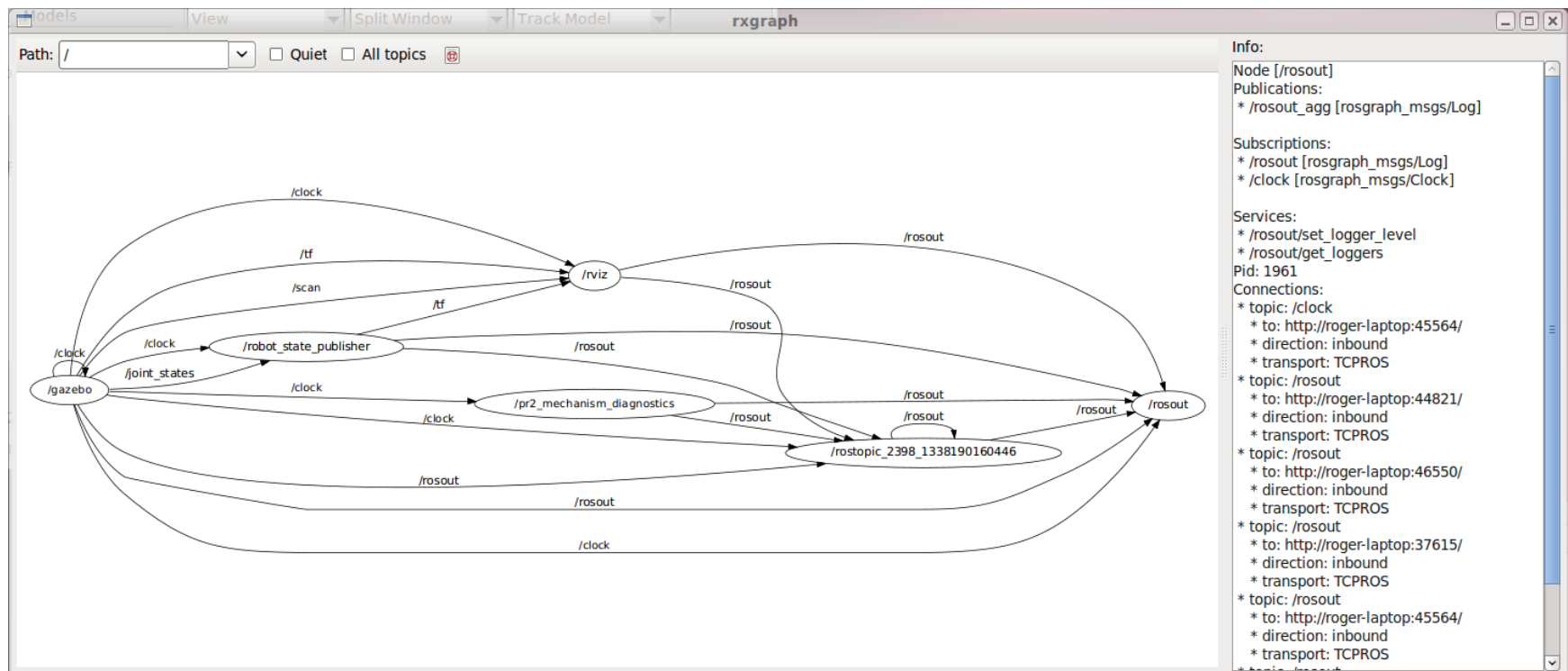
- Display ROS messages
- Set logger level
- Include and exclude messages



ROS debugging tools

RXgraph

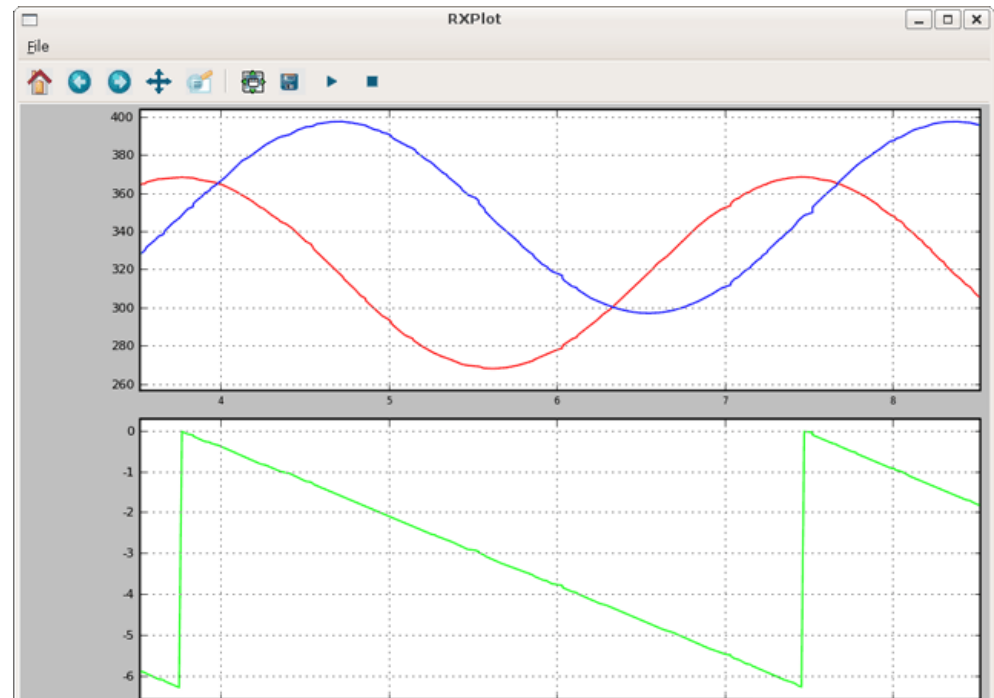
- Visualization of all nodes and connecting topics



ROS debugging tools

RXplot

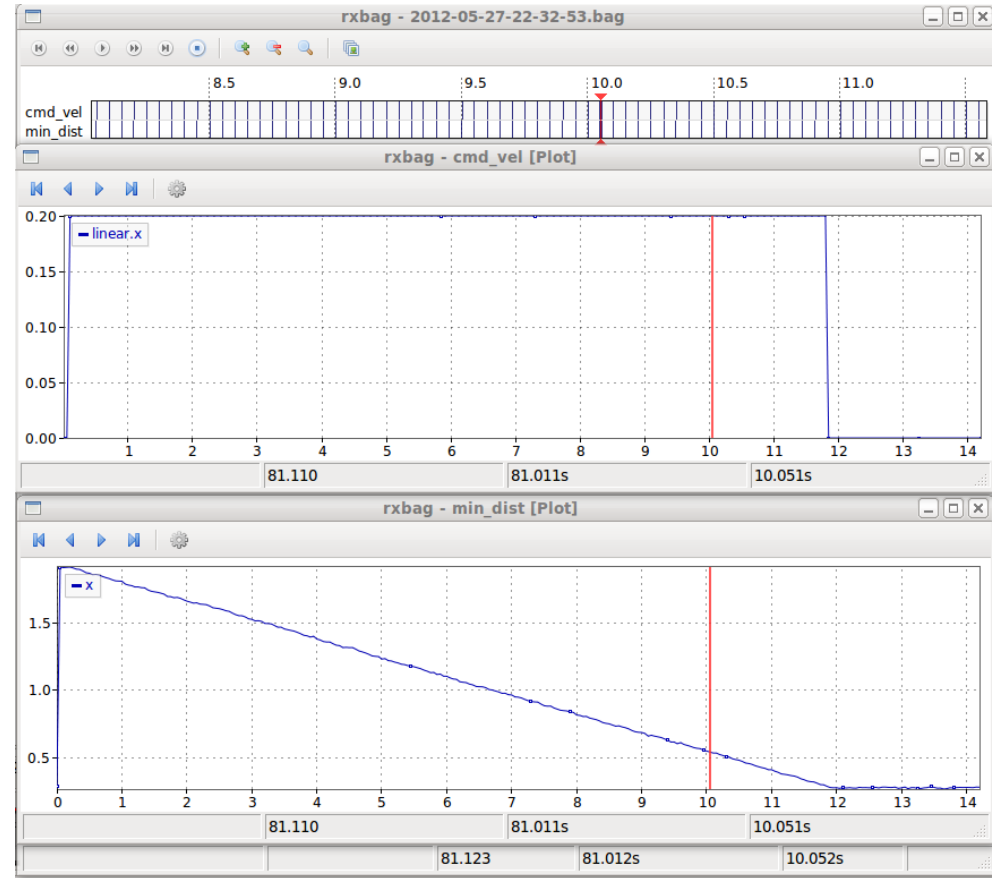
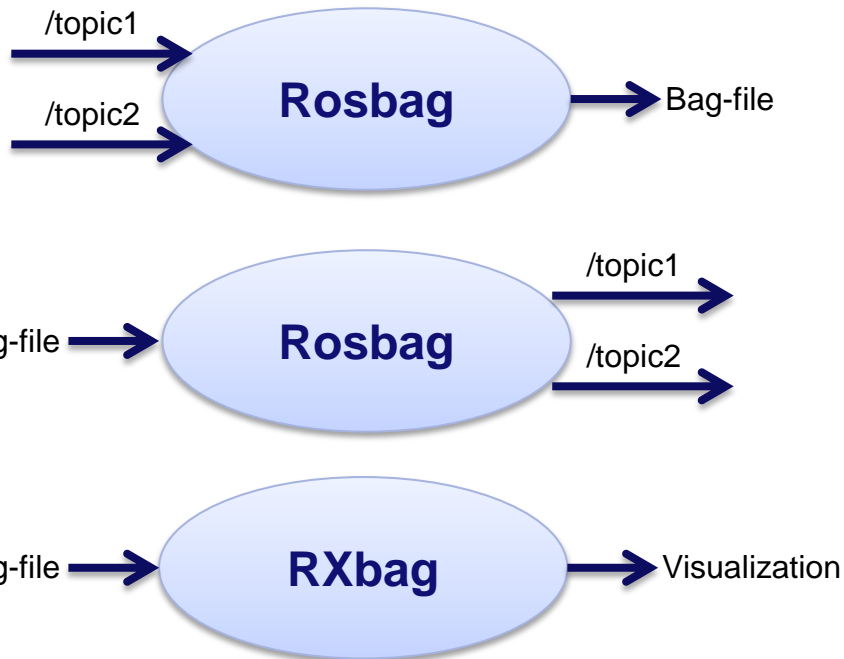
- Subscribes to topic and plots data from topic field



ROS debugging tools

RXbag and Rosbag

- Record and playback topic messages



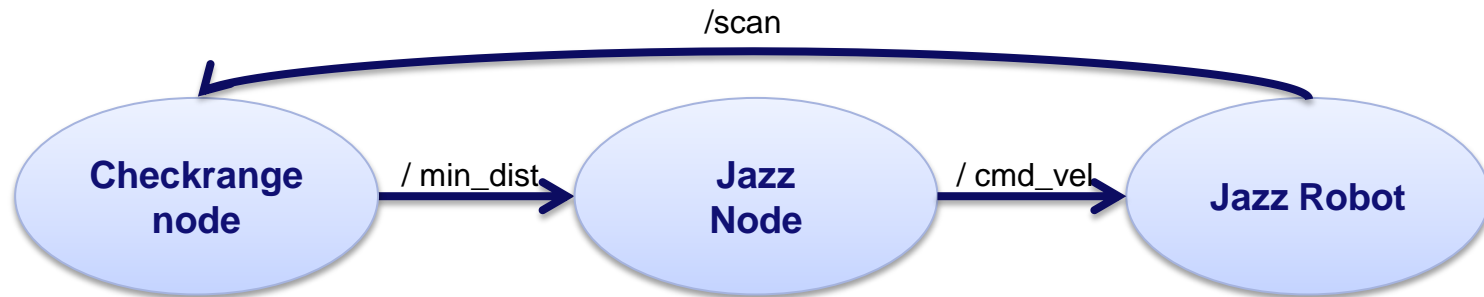
ROS debugging tools

Command line tools

- **rostopic** **Topics**
- **rosservice** **Services**
- **roscpp** **Nodes**
- **rosmmsg** **Messages**
- **rossrv** **Service files**
- **roswtf** **Report with warnings and errors**

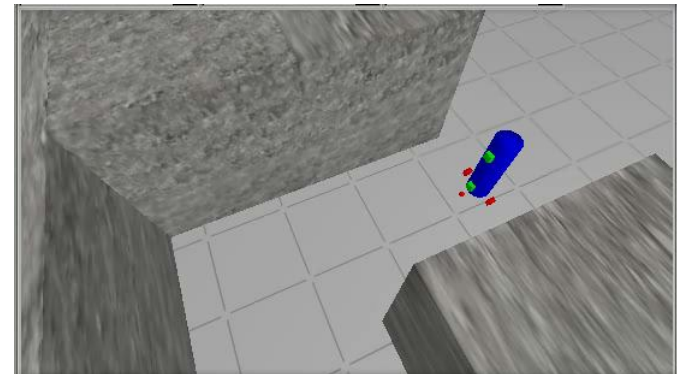
ROS Debugging example

- **Example: simple collision avoidance**



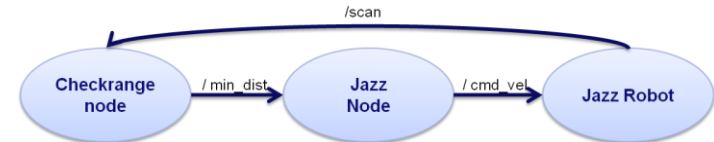
- Receive laserdata
- Calculate distance
- Send distance

- Receive distance
- Compare with minimal treshold (0.3m)
- Send velocity message



ROS Debugging example

Using RXconsole



Checkrange node:

```
scan_pub.publish(min_dist);
ROS_DEBUG("Message published on /min_dist");
```

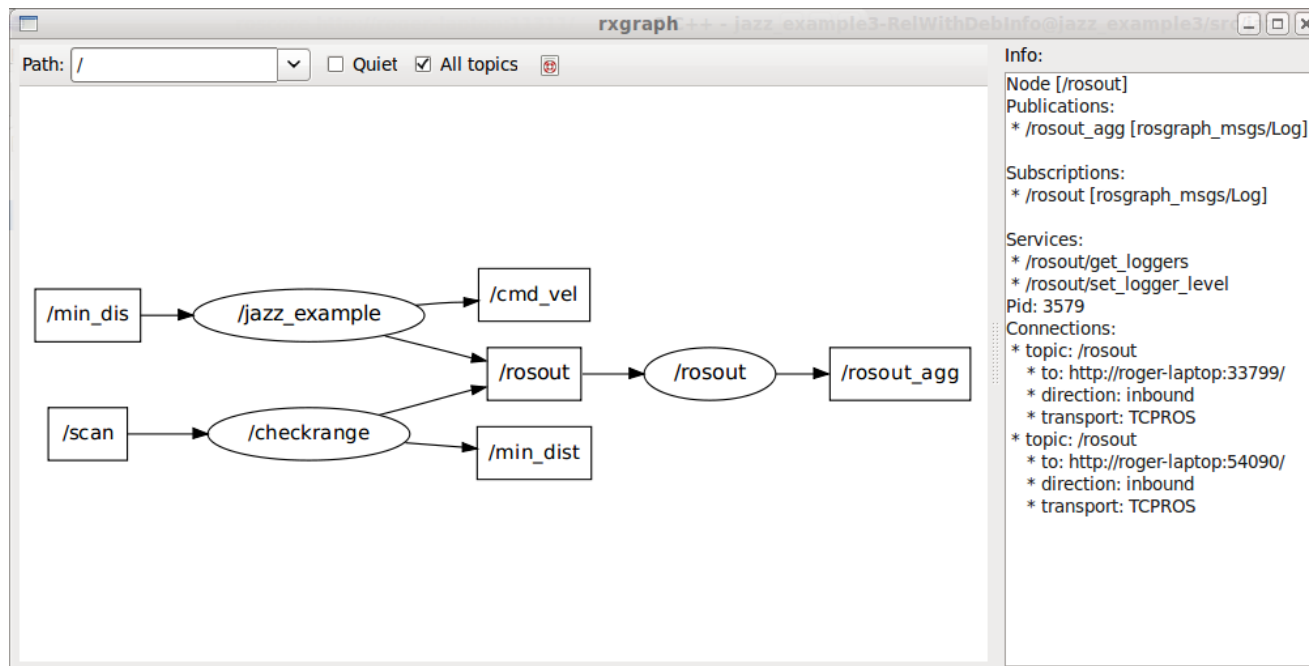
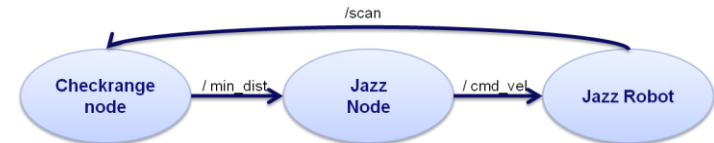
Jazz node:

```
void min_distcallback(...) {
    ROS_DEBUG("Message received on /min_dist");
}
```

Message	Severity	Node	Time	Topics
Message published on /min_dist	Debug	/checkrange	1495.296000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.347000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.396000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.446000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.496000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.546000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.596000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.646000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.696000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.746000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.796000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.846000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.897000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.946000000	/rosout, /m
Message published on /min_dist	Debug	/checkrange	1495.996000000	/rosout, /m

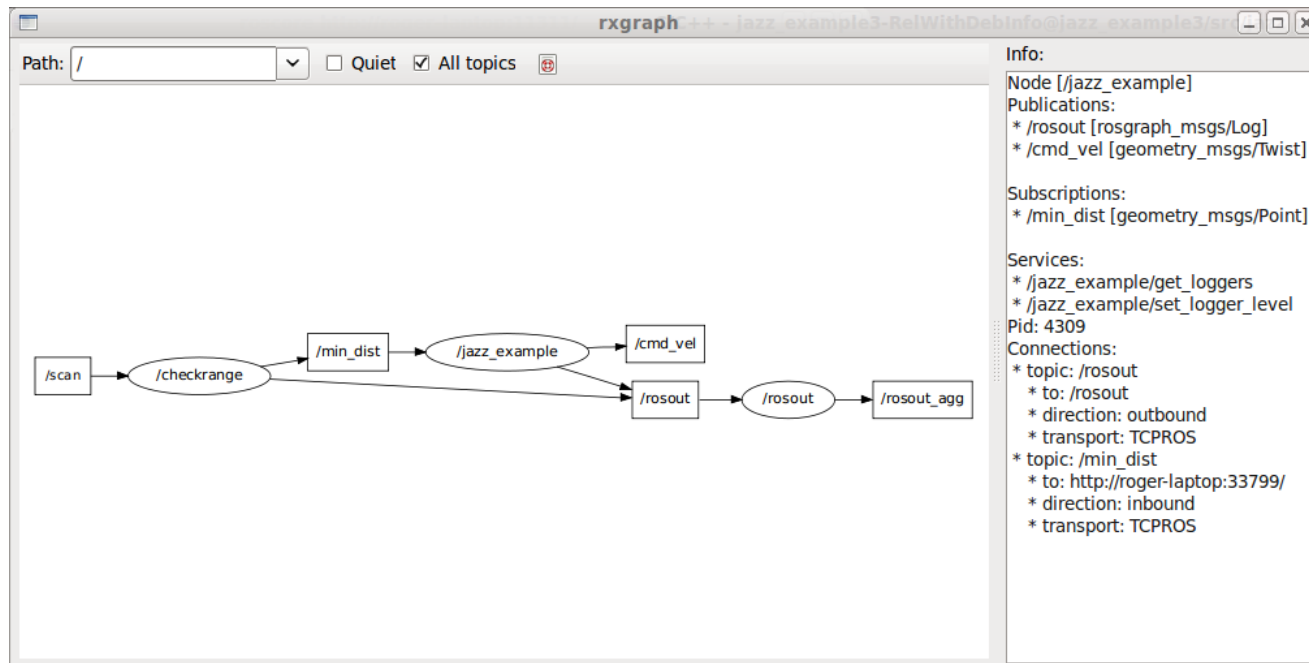
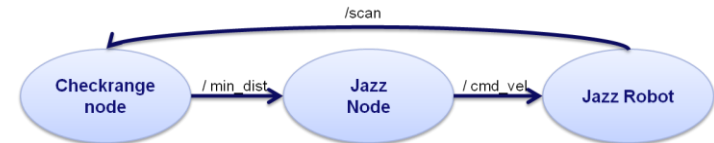
ROS Debugging example

Using RXgraph



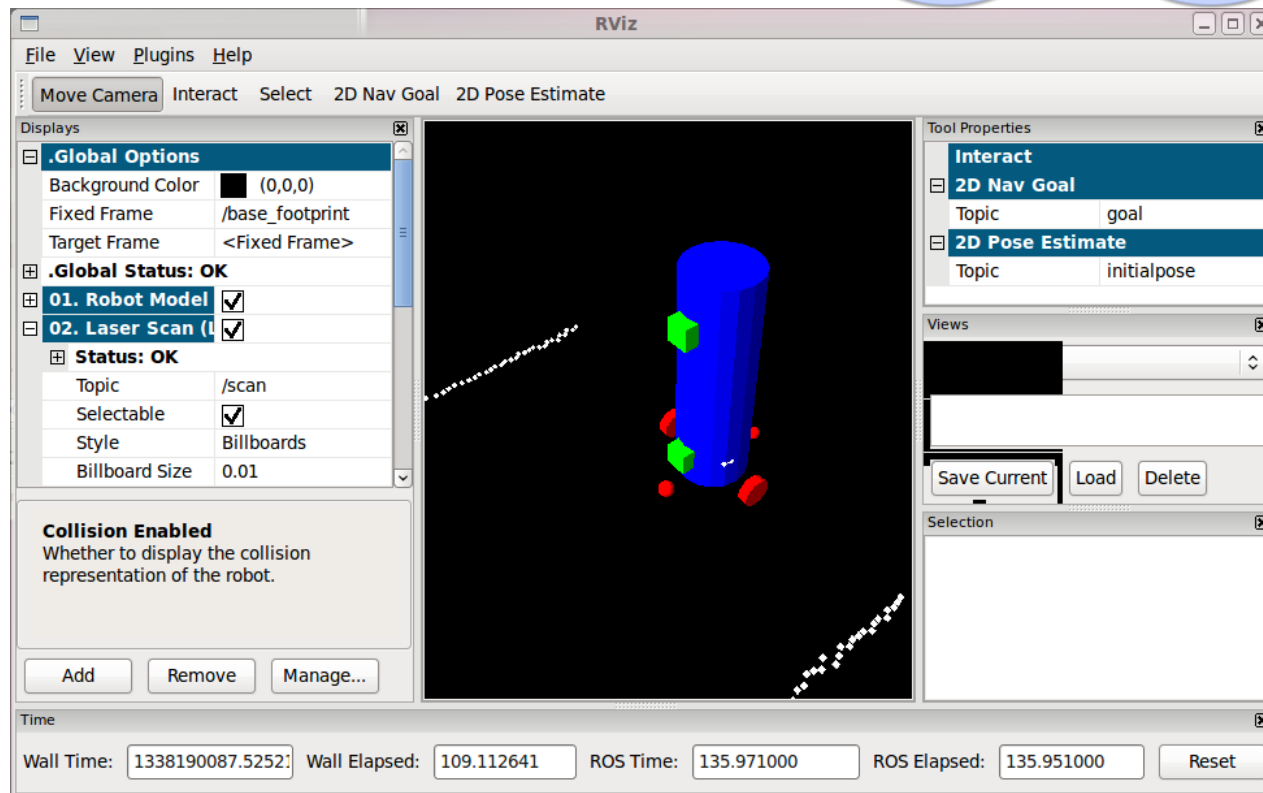
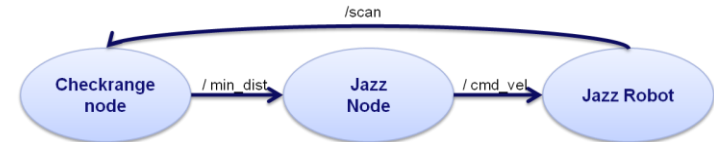
ROS Debugging example

Using RXgraph



ROS debugging example

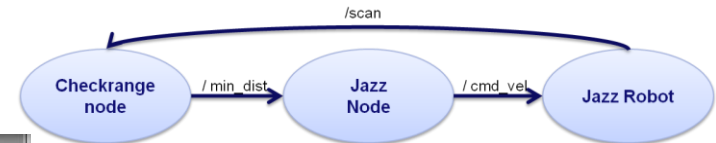
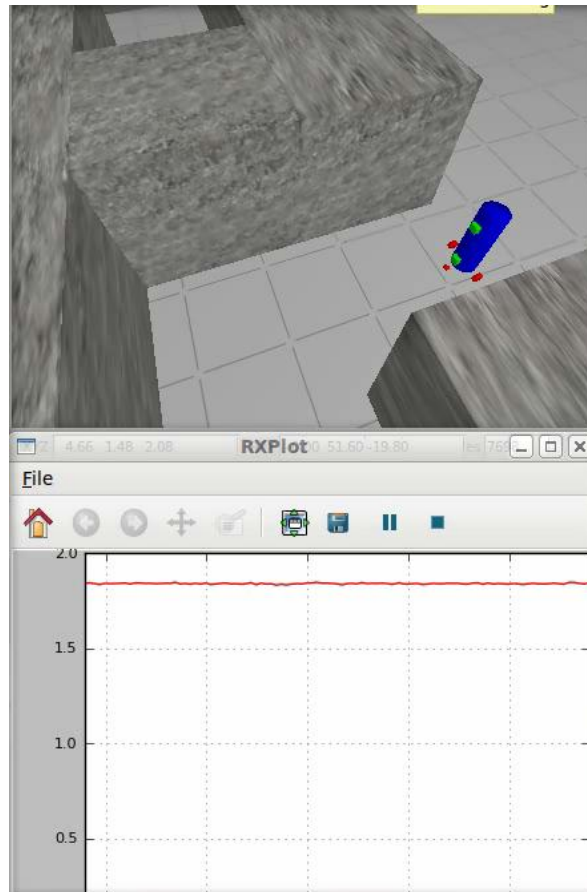
- Rviz



ROS Debugging example

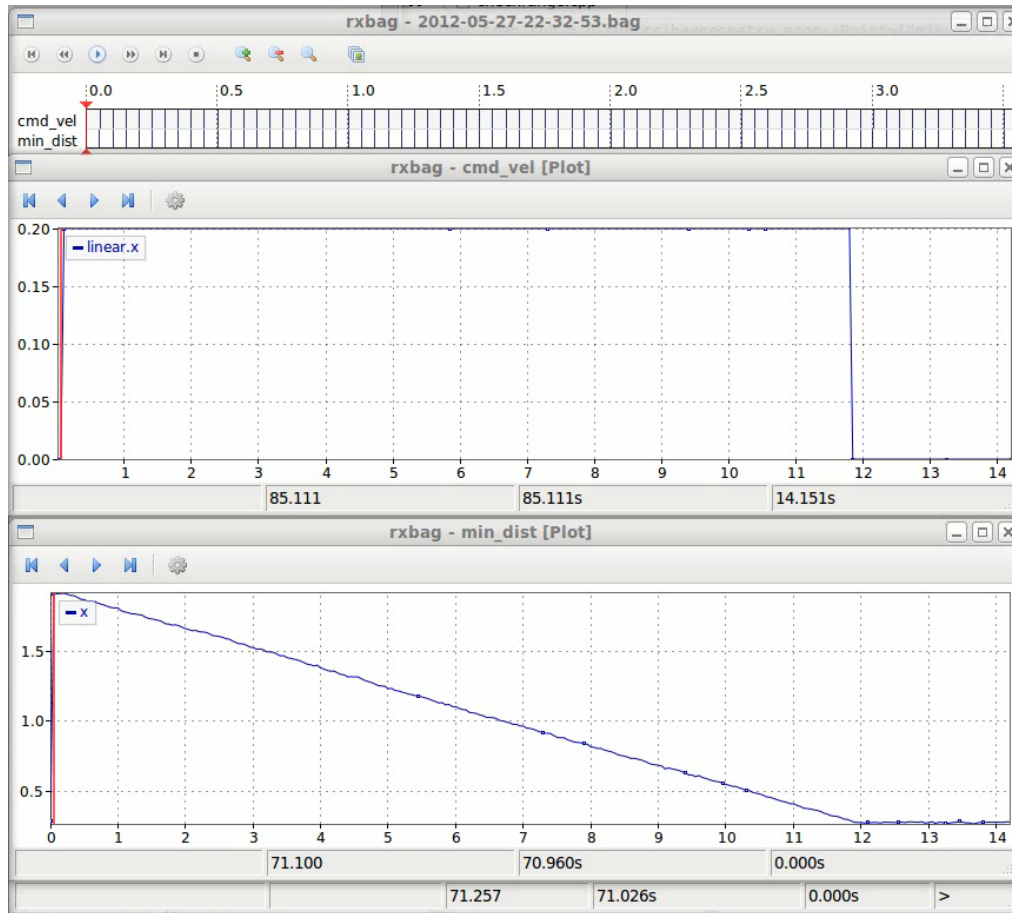
Using RXplot

```
rxplot /min_dist/x
```



ROS Debugging example

Using RXbag and Rosbag

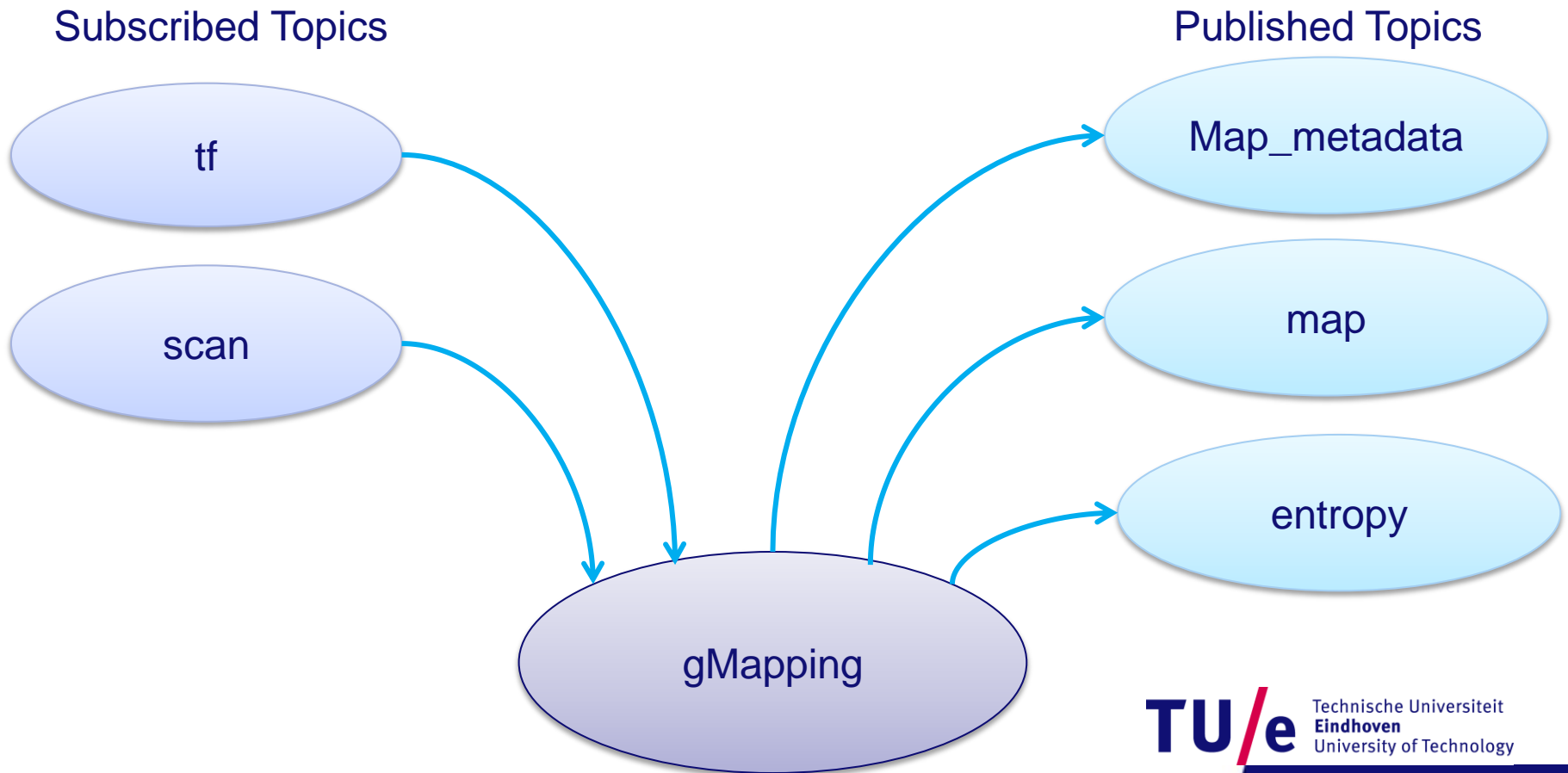


Introduction gMapping



- **Purpose**
 - Making a map and localize the robot on the map.
- **Hardware Requirements**
 - Robot that provides odometry.
 - A horizontally-mounted, fixed laser range-finder.

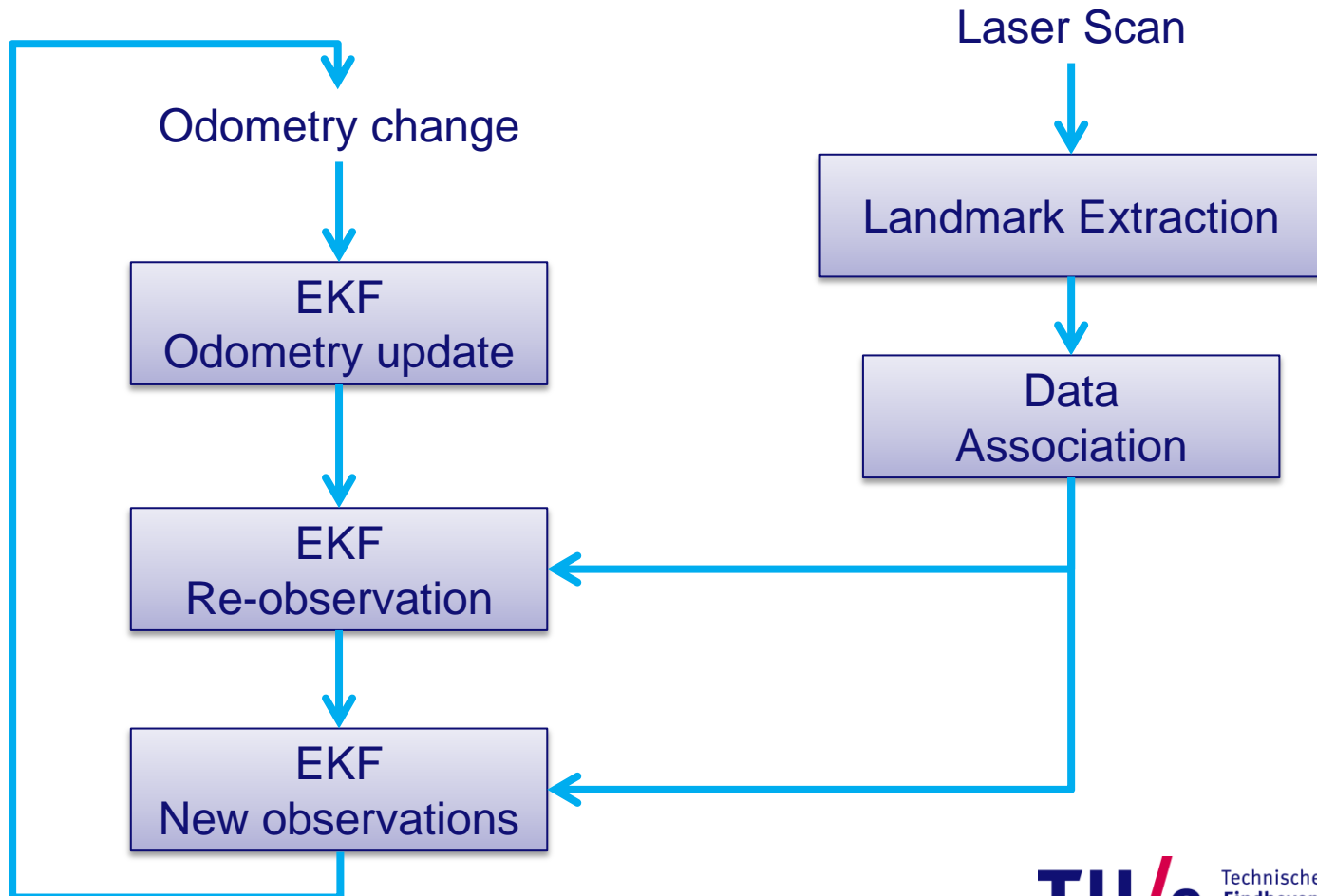
Introduction gMapping



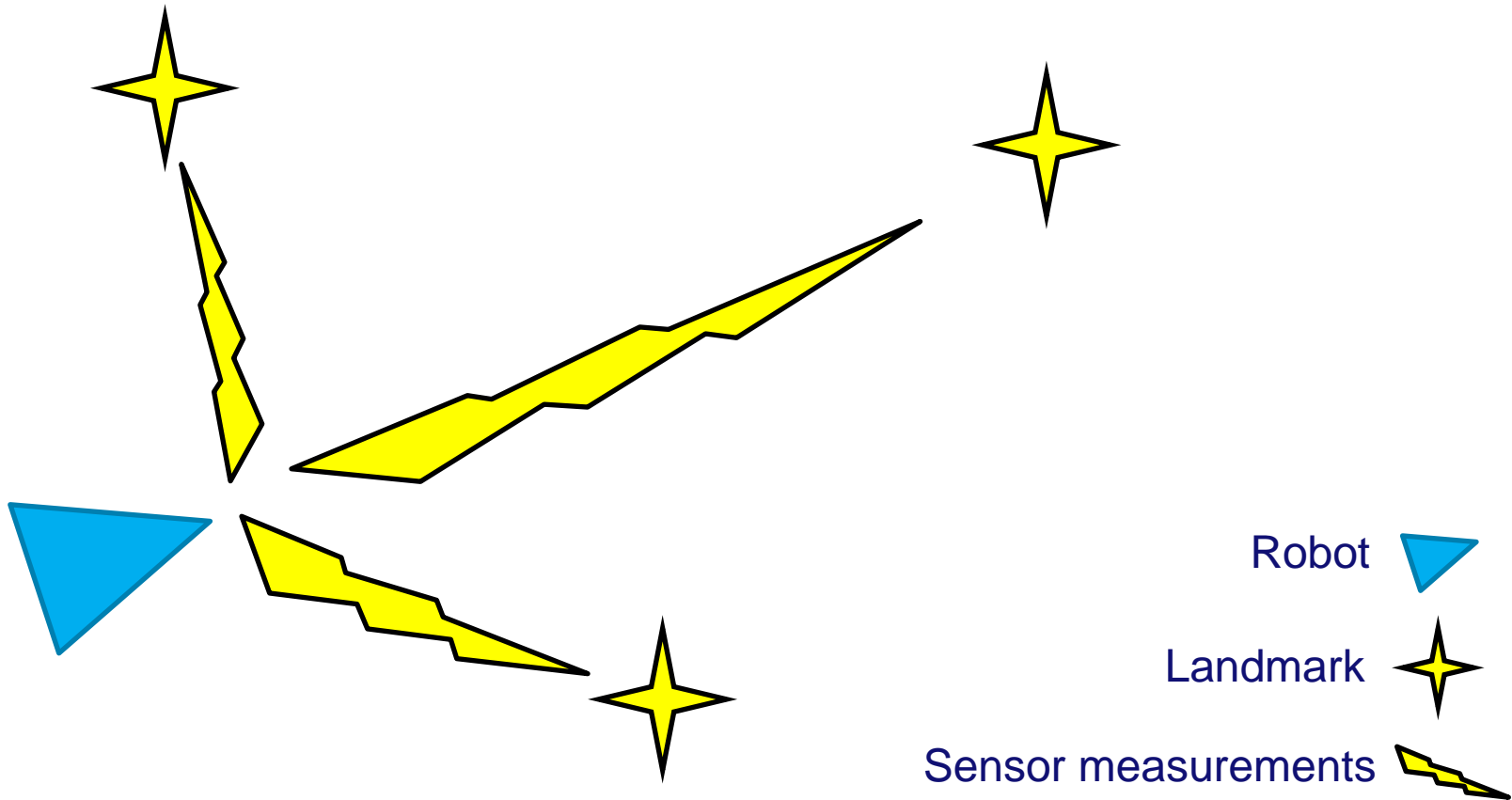
Open SLAM

- **gMapping is based on open SLAM**
- **SLAM Simultaneous Localization and Mapping**
- **SLAM is applicable for both 2D and 3D motion.**
 - **For his lecture we only consider 2D because of our maze problem.**
- **SLAM consists of multiple parts:**
 - **Landmark extraction**
 - **Data association**
 - **State estimation**
 - **State update**
 - **Landmark update**

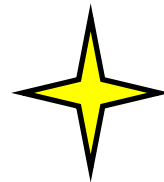
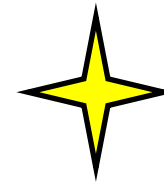
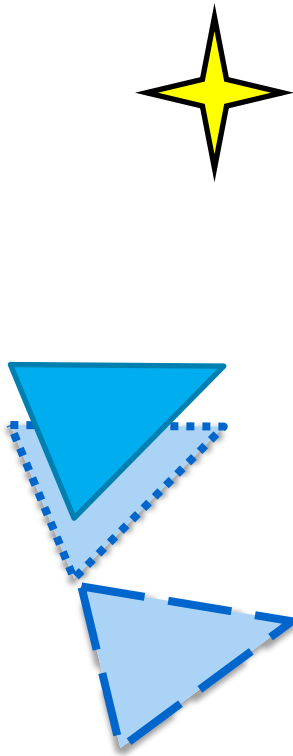
SLAM process



SLAM process



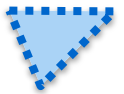
SLAM process



Odometry position



Estimate position



Actual position



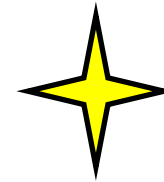
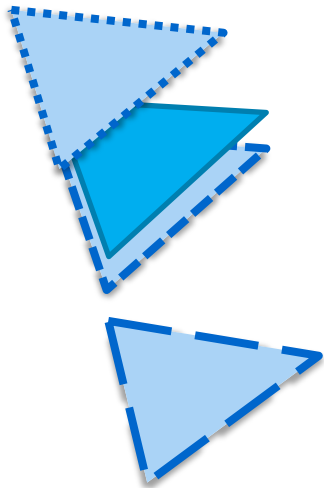
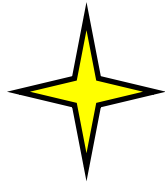
Landmark



Extended Kalman Filter (EKF)

- Recursive data processing algorithm.
- Extended Kalman filter is used for non-linear systems.
- Extended Kalman Filter filters the data based on:
 - The process noise.
 - A known error of the odometry (used in prediction step).
 - The measurement noise.
 - A known error of the range and bearing (used in measurement step).
 - Kalman gain relies more on odometry or scan data.

Kalman Gain



Estimated position with Kalman gain



Odometry position



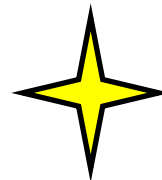
Measured position



Actual position



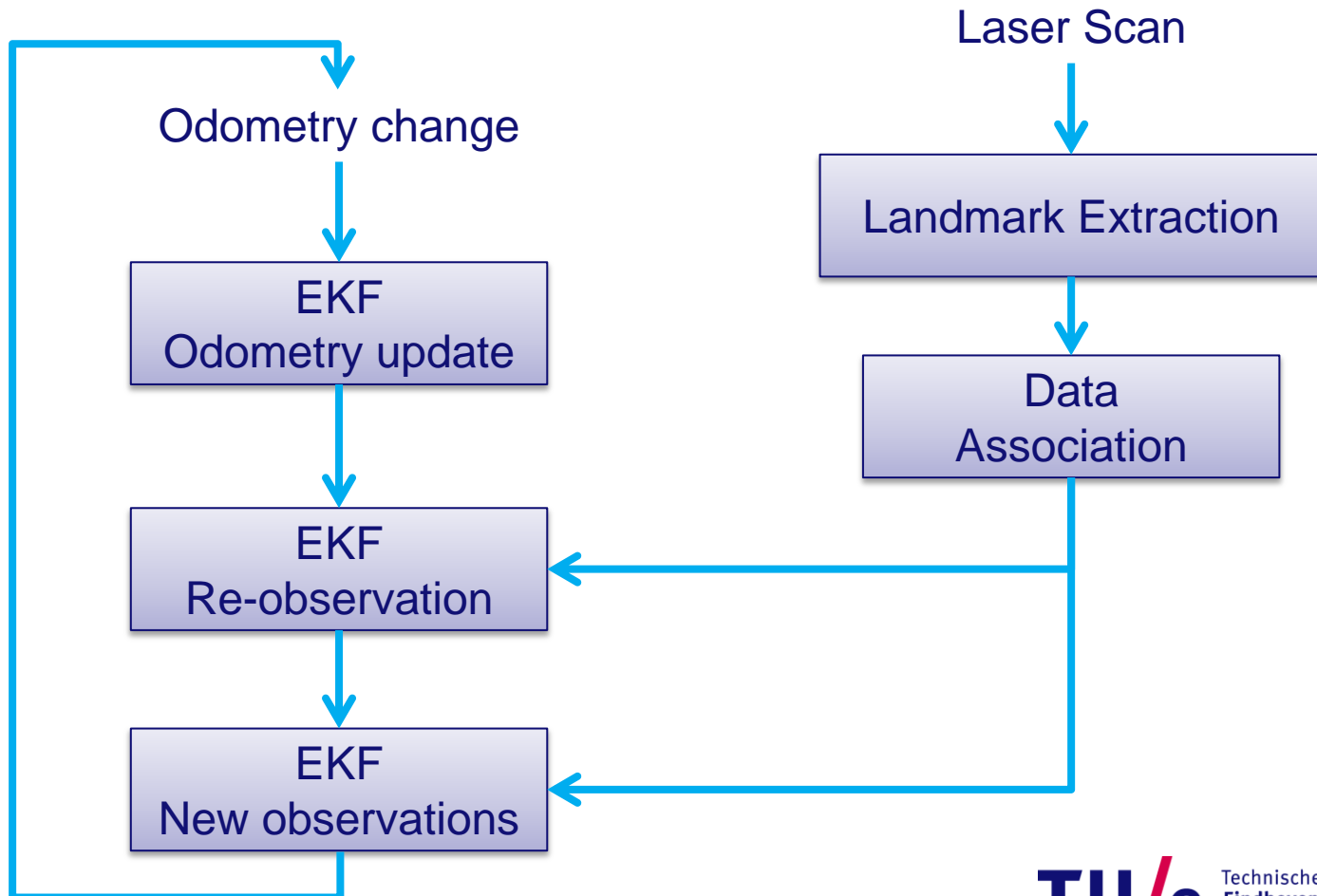
Landmark



Extended Kalman Filter

- **Overview of the process**
 1. **Update the current state estimate using the odometry data.**
 2. **Update the estimated state from re-observing landmarks.**
 3. **Add new landmarks to the current state.**

SLAM process



Using gMapping

Add this to the launch file.

```
<node pkg="robot_pose_ekf" type="robot_pose_ekf" name="robot_pose_ekf">  
  <param name="freq" value="10.0"/>  
  <param name="sensor_timeout" value="1.0"/>  
  <param name="publish_tf" value="true"/>  
  <param name="odom_used" value="true"/>  
  <param name="imu_used" value="false"/>  
  <param name="vo_used" value="false"/>  
  <param name="output_frame" value="odom"/>  
</node>
```

```
<node pkg="gmapping" type="slam_gmapping" name="slam_gmapping" output="screen">  
  <roscparam>  
    # Time between updates in the map. We keep it small for better  
    # user feedback during mapping. Increasing this will decrease CPU  
    # usage. However, map updates are single-CPU bound.  
    map_update_interval: 0.05  
  </roscparam>  
</node>
```

References

- **Welch, G and Bishop, G. 2001. “An introduction to the Kalman Filter”, [Link](#)**
- **S. Riisgaard and M. R. Blas, “SLAM for Dummies (A Tutorial Approach to Simultaneous Localization and Mapping),“ [Link](#)**

Questions

