How ROS works together with the mining industry in i2mine project

Prepared for ROSCON 2014
Sascha Schade
How ROS works together with the mining industry in i2mine project

Outline

- About IMR and RWTH Aachen University
- About i2mine project
- Industrial-grade Sensors
- Mobile Robotic Test Platform
- ROS Integration
- Outlook

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° [280855].
IMR and RWTH Aachen University

- RWTH Aachen University
  - 40,000 students
  - 10,000 staff

- Mining and Metallurgy Machinery
  - Faculty of Georesources and Material Engineering
  - 60 staff
  - Univ.-Prof. Dr.-Ing. Karl Nienhaus, since 1999
The Intelligent Mine of the Future

Introduction

- Changing the Way of Mining
- „Small Invisible Mining“
- Cross Linked Real Time Information Network
- Innovative Mining Methods
- Next Generation Mining Machines
- Autonomous Operation
- Integrated Intelligent Sensor Technologies
- Economical and Safe Exploitation
- Adverse Environmental Impacts
Industrial and academic partners
The Intelligent Mine of the Future

Vision

- Today’s mining operation
- i2mine vision

- Development of methods, technologies and machinery for deep mining
- Autonomous mining operation
- New sensors for highly selective cutting
- Improve health and safety
The Intelligent Mine of the Future

**Future Deep Mining**
*Lean Mining, Minimum Environmental Footprint*

- **Work Packages**
  - **Subtask 4.2.1** Boundary layer and material detection
  - **Subtask 4.2.2** *Collision avoidance* and local underground navigation at the face
  - **Subtask 4.2.3** Machine guidance for precise positioning and cutting trajectory
  - **Subtask 4.2.4** A highly integrated, modular system for process optimization on autonomous mining machinery
# Industrial-grade Sensors

For collision avoidance and navigation

<table>
<thead>
<tr>
<th>3D</th>
<th>IR TOF</th>
<th>UWB</th>
<th>IMU</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="IR TOF" /></td>
<td><img src="image2" alt="UWB" /></td>
<td><img src="image3" alt="IMU" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2D</th>
<th>Laser Scanner</th>
<th>IMR Radar</th>
<th>iDRR (Dual Range Radar)</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image4" alt="Laser Scanner" /></td>
<td><img src="image5" alt="IMR Radar" /></td>
<td><img src="image6" alt="iDRR" /></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1D</th>
<th>iLDR (Linear Dynamic Radar)</th>
<th>Encoder</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image7" alt="iLDR" /></td>
<td><img src="image8" alt="Encoder" /></td>
<td></td>
</tr>
</tbody>
</table>
Concept
Navigation System

Disturbances

Fibre Optic Gyroscope
Accelerometers
IMU
INS

Supporting Sensors
Control input
additional information

Navigation algorithm

Orientation
Position

Surroundings

IMR – Sascha Schade 2014-09-12
Sensor Interfacing

- Industrial-grade Ethernet
- Unified software interface with TCP/IP
- Sensor data accessible with different user tools, in parallel
- No realtime burden for applications
- Data recording with precise time stamp

Technology
- ARM Cortex-M4f (STM32)
- Embedded Linux for Ethernet
Sensor Synchronization

- **Problem**
  - Heterogeneous mix of time-discrete sensors
  - High data rate (400 Hz, 100 Hz)
  - Proprietary interfaces and protocols
  - Arbitrarily placed on vehicle
  - Realtime un-capable user software (Windows, Matlab)

- **Solution**
  - Local clocks
  - Synchronised by Ethernet by micro seconds (PTP)
  - Time stamping of each sensor sample in sensor box
ROS integration

- Without ROS
  - Developing algorithms in Python, Matlab, C++ works well
  - Sensor data available everywhere in research network
    - TCP/IP Server in each sensor box
  - No visualization
  - No data storage, no play back
  - No access to already developed algorithms

- With ROS.org
  - Use rviz
  - ROS bag files
  - Algorithms like navigation stack (SLAM, EKF, ICP from PCL)
ROS integration

- ROS-ify all sensors and actuators
- Dilemma
  - Our sensors include more data than standard ROS messages
    - e.g. temperature, status flags of IMU
    - Measurements campaigns are expensive and rare
    - Do not discard any data as storage and bandwidth come at no cost
    - Wanted to use standard ROS messages for ROS algorithms
- Solution

**Imrmsgs**
- ImrImuLitefLci100c.msg

**Sensor_msgs**
- Imu.msg

**IMR Sensor Box** ➔ **IMR Tcp/IP Stream** ➔ **ROS Client Node** ➔ **ROS Conversion Node** ➔ **ROS EKF SLAM** ➔ **ROS.org**
Mobile Robotic Test Platform
Results
ROS results
ROS Lessons Learned

- Do not change created ROS messages during development
  - This breaks backward compatibility of already recorded data

- Git repository for each node when more than one person is working on the project
  - Use gitslave to provide unified catkin workspace

- Always record visual camera images to interpret data more easily

- Unfortunately not everything always works out of the box
  - Camera
  - Catkin and message generation
  - Need to recompile PCL to support all cameras
Outlook

- Upcoming demonstration phase with industrial partners (one year starting next month)

- Project proposals
  - Nationally funded
  - EU-funded (Horizon 2020)

- ROS will play a key role in these projects
Thank you for your attention!

Contact
Dipl.-Ing. Sascha Schade
IMR RWTH Aachen University
sSchade@imr.rwth-aachen.de
ros@sascha-schade.de