Bringing Intervals into ROS, the Robot Operating System

Vincent Drevelle, Téva Demangeot

SWIM 2018, 25-27 July 2018, Rostock (Germany)

Univ Rennes, Inria, CNRS, IRISA
Rennes, France
vincent.drevelle@irisa.fr
The *interval* ROS package is still a work in progress...

This talk is about motivation and current development.
Outline

Intervals and Robotics

ROS, the Robot Operating System

The *interval* ROS package

> Messages
> Visualization
> Bridge with Ibex

Live demo
Intervals and robotics

Representation of uncertainty
- Epistemic uncertainty
- Stochastic uncertainty: noise

Representation of domains / sets
- Maps
- Bounded error estimation

Applications
- Control under uncertainty
- Bounded-error state estimation
- Fault detection and isolation
- Mapping, SLAM (data association)
- Design (under constraints and uncertainty)
- ...

Inria
inventors for the digital world
Intervals in field robotics: Underwater mapping

Île des Morts (near Brest, France, 2014)

> V. Drevelle, with L. Jaulin and S. Rohou
Intervals in field robotics: Drone localization

Camera-based localization
IRISA
(Rennes, France, 2017)
I.F. Kenmogne
Intervals in field robotics: Vehicle positioning

GPS with FDI (Paris, France, 2010)
V. Drevelle & Ph. Bonnifait

GPS with FDI and onboard sensors fusion (Versailles, France, 2010)
V. Drevelle & Ph. Bonnifait
Intervals in field robotics

Use intervals in real robots / systems

- Sensors
- Estimation algorithms
- Control algorithms
- Actuators

Interval data

- Data logging
- Monitoring and visualization
- Data exchange

Interval data
ROS, the Robot Operating System

History

- Started in 2007 in Stanford, development continued by Willow Garage since 2008
- 2010: ROS 1.0
- 2013: Open Source Robotics Foundation (OSRF), now Open Robotics
- 2018: Melodic Morenia version, see ros.org
What is ROS?

ROS stands for “the robot operating system”...
...but ROS is not an operating system!

**ROS is robotics middleware**

A collection of software frameworks for robot software development.

**Main features**

A ROS application is a network of **nodes** (sensor/actuator interfaces, processing, display) interconnected with a message passing infrastructure

- Nodes are distributed, and can be written in different programming languages
- Packages and dependencies management
- Collection of software components, targeted to robotics applications needs
ROS communication main features

**Master node: rosmaster**
Coordination node, name server

**Topics**
Publish / subscribe *message* passing

**Services**
Remote procedure call

**Parameter server**
Share and update parameters for all nodes connected to the ROS network

**Supported languages for nodes**
- Main client libraries: C++, Python, Lisp
- JSON-RPC interface with ROSBridge
- Contributed client libraries for Java, Matlab, Javascript, etc

Introspection (messages, topics, graph)
ROS tools and libraries

**Command line tools**
Examine the graph, data flow, display live data, node and message type information, deployment...

**Visualization tools**
- **Rviz:** 3D
- **rqt:** 2D plot, ROS network graph, image viewer, ...

**Data recording and playback**
“.bag” dataset file format. rosbag (console) and rqt_bag (graphical) utilities.

**Transform library (tf2)**
Transform coordinates between coordinates frames (e.g. world, base, head, arm, gripper).
Keeps track of a transform database over time.

**Drivers**
For sensors (cameras, lasers...) and actuators (servomotors, ...)

**Robotics algorithms**
Localization, EKF, SLAM, planning...
Messages

Standard and custom datatypes defined in a .msg text file
Compiled to C++, Python and Lisp classes

A message is a data structure made of:
  > Basic types (int32, float64, ...)
  > Other messages types
  > Arrays of basic types/messages

Standard sensor and geometry messages already available (Point, Vector3, Pose, Path, Image, LaserScan...)

```bash
rosmsg show geometry_msgs/Pose

geometry_msgs/Point position
  float64 x
  float64 y
  float64 z

geometry_msgs/Quaternion orientation
  float64 x
  float64 y
  float64 z
  float64 w
```
Topics

Data channel on which nodes exchange data related to a particular information (e.g. the speed reading of the front-left wheel of the robot, or, the map of the free space, etc...)

- Unidirectional communication
- Strongly typed by message type
- Accessed by its name (e.g. /car/wheel_speed_fl)

- Nodes that generate data **publish** messages to the relevant topic(s)
- Nodes that consume data **subscribe** to the relevant topic(s)

There can be multiple publishers and subscribers to a topic
Bringing intervals into ROS

ROS
De-facto « 1st choice » middleware for roboticians
Lots of debugging tools (visualization, graph introspection, etc)
Large user community, packages browser, documentation and tutorials available
Used in robotics classes

Intervals
Available methods for state estimation, mapping, SLAM, FDI
Specific visualization needs (boxes, pavings...)
Alternative to the classical probabilistic methods (most of the time Gaussian)

Intervals in ROS
● Ease the use of interval methods with real robots / data
● Increase the visibility of interval methods
The *interval* ROS Package

**Messages**

- Intervals ✓
- Pavings ✓

**Visualisation**

- 3D with Rviz ✓
- 2D with RQT
- Vibes connector

**Computation**

- Link with Ibex ✓
- Tutorials (SIVIA, ...) ✓
- Basic interval estimator
- Basic paver
Interval Messages

**Standardize data exchange between interval computation nodes**

- Enable log and replay of interval results
- Basic interval types, as a foundation for application specific interval data
- Conversion to/from interval libraries types

```
interval_msgs/Interval
  > float64 lb
  > float64 ub

interval_msgs/IntervalVector
  > interval_msgs/Interval[] box

interval_msgs/SubPaving
  > interval_msgs/IntervalVector domain
  > interval_msgs/IntervalVector[] subpaving
```
Interval Messages

Geometrical and time messages

PointInterval (for position), PoseInterval, Vector3Interval (e.g. for speed)
TimeInterval, DurationInterval

Stamped versions of messages

Messages with an additional standard header

- Sequence number
- Timestamp
- Coordinate frame name
ROS intervals: Visualization

Display interval quantities (boxes) and subpavings published as ROS messages

2D display (Work in progress)

- 2D view in rqt
  - Qt-based graphical user interface development framework in ROS

- Bridge with Vibes
  - Easy to use interval display system
  - enstabretagnerobotics.github.io/VIBES

3D display

- 3D view in RViz
RViz interval display plugins

**RViz**

ROS 3-D visualization tool
Off-the-shelf display of standard robotics quantities: points, poses, maps, paths...

Uses **stamped** message datatypes
  > All geometry is transformed to a common frame before display
  > Allows mixed display of global frame objects (e.g, map) with robot/sensor body frame data (e.g, measurements)

**Interval RViz Plugins**

*interval_rviz_plugin* package
Rviz plugins to display Interval messages in 3D
  > interval Points (position box), Poses, Vector3 (speed box)
  > SubPavings and Pavings (position)
ROS intervals: link with Ibex lib

Ibex library

*ibex-lib.org (Chabert et al.)*
C++ library for constraint processing over real numbers
  - Interval arithmetics
  - Contractors / Separators
  - Solvers

ROS Interval Ibex

Conversion between Ibex types and ROS interval messages
Display of Set solutions

Configurable set-inversion solver (work in progress)
  - Plug and play interval solver within a ROS graph
  - Dynamic observation model setting (no compilation needed)
Demo

Robot positioning from ranging beacons, by using SIVIA

Robot simulation \rightarrow \textit{true\_position} \rightarrow RViz

\textit{range\_measurements} \rightarrow \text{SIVIA} \rightarrow \text{point estimate} \rightarrow \text{solution paving}
The *interval* ROS Package

Demo

RViz

rosmaster and nodes

Graph of nodes and topics
Outlook

Hosted at INRIA GitLab

> Unstable version, under development
> Open source (BSD license)
> [https://gitlab.inria.fr/rainbow-intervals/ros_interval](https://gitlab.inria.fr/rainbow-intervals/ros_interval)

Short term goal

> Finalize rqt 2D view, and simple Ibex paver
> First release by the end of September 2018

What’s next

Get indexed in the ROS package directory, for further visibility

Integration with other robotics related interval libraries (e.g Dynibex)