The UFO (Usv For observation) project

Software architecture

Simulation

Data acquisition

Surface drone for observation

Jean-Philippe Babau,
Goulven Guillou, Mickael Kerboeuf,
Pierre-Yves Pillain, Manel Ait-Habouche

Adaptive control of marine applications

- Unpredictable environment
- Strong perturbations
- Safety constraints
- Limited connectivity

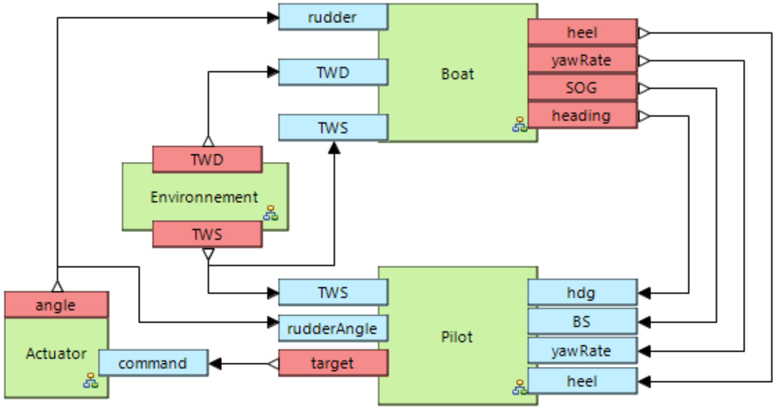
Adaptive software

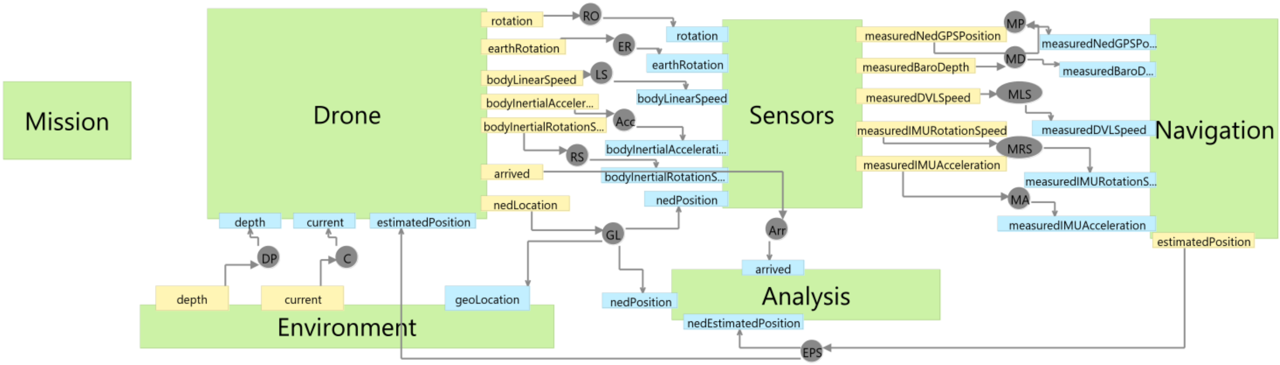
- Embedded software challenges
  - Limited resources (computation, memory, power consumption)
  - Evolution considering heterogeneous platforms, sensors, communication protocols
  - Reuse of software components
  - Configuration depending of the context

Software architecture

- Model-Driven and component-based development
  - Architectural styles (SAIA, IMOCA, AMSA, CARES)
- Code generation

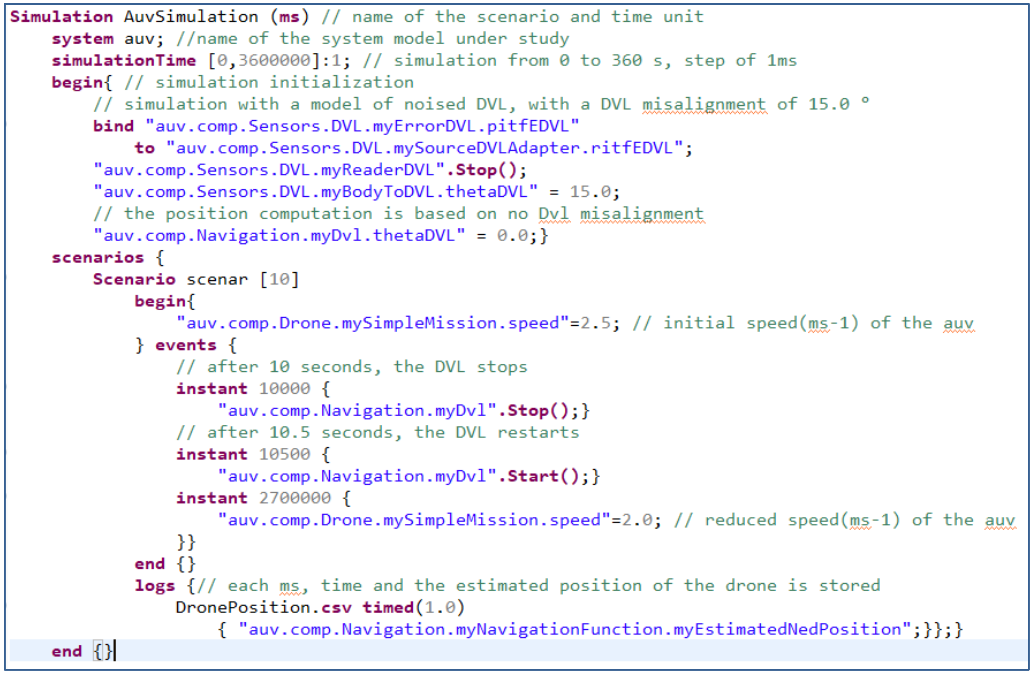
- Embedded system development
- Design of simulator



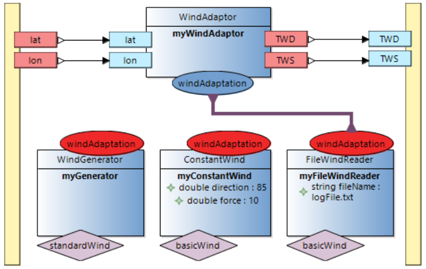


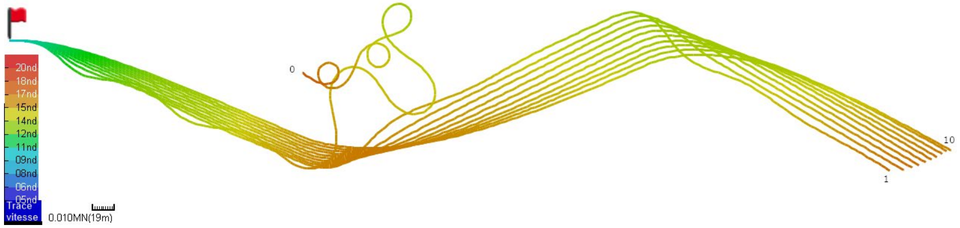
Simulation (AMSA, CARES)

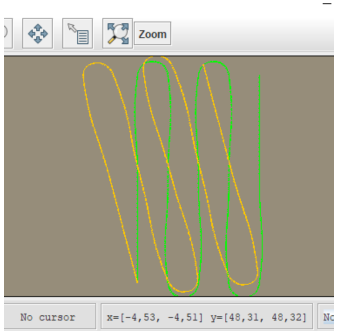
- Configuration
  - Structure
  - Scenarios
- Parameterization
  - Types
  - Parameters
  - Timed events

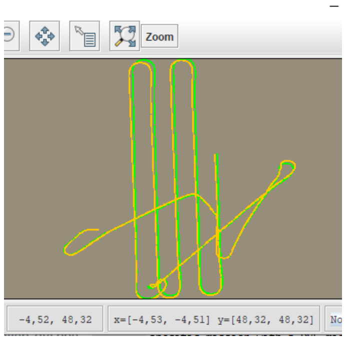


```
Simulation AuvSimulation (ms) // name of the scenario and time unit
    system auv; //name of the system model under study
    simulationTime [0,3600000]:1; // simulation from 0 to 360 s, step of 1ms
    begin{ // simulation initialization
        // simulation with a model of noised DVL, with a DVL misalignment of 15.0 °
        bind "auv.comp.Sensors.DVL.myErrorDVL.pitfEDVL"
            to "auv.comp.Sensors.DVL.mySourceDVLAdapter.ritfEDVL";
        "auv.comp.Sensors.DVL.myReaderDVL".Stop();
        "auv.comp.Sensors.DVL.myBodyToDVL.thetaDVL" = 15.0;
        // the position computation is based on no Dvl misalignment
        "auv.comp.Navigation.myDvl.thetaDVL" = 0.0;}
    scenarios {
        Scenario scenar [10]
            begin{
                "auv.comp.Drone.mySimpleMission.speed"=2.5; // initial speed(ms-1) of the auv
            } events {
                // after 10 seconds, the DVL stops
                instant 10000 {
                    "auv.comp.Navigation.myDvl".Stop();}
                // after 10.5 seconds, the DVL restarts
                instant 10500 {
                    "auv.comp.Navigation.myDvl".Start();}
                instant 2700000 {
                    "auv.comp.Drone.mySimpleMission.speed"=2.0; // reduced speed(ms-1) of the auv
                }}
            end {}
            logs {// each ms, time and the estimated position of the drone is stored
                DronePosition.csv timed(1.0)
                    { "auv.comp.Navigation.myNavigationFunction.myEstimatedNedPosition";}};}
    end {}
```





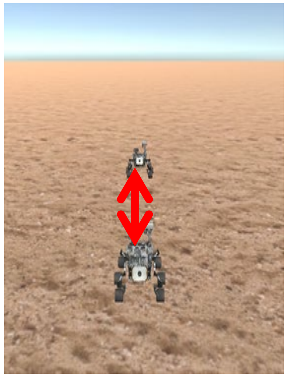


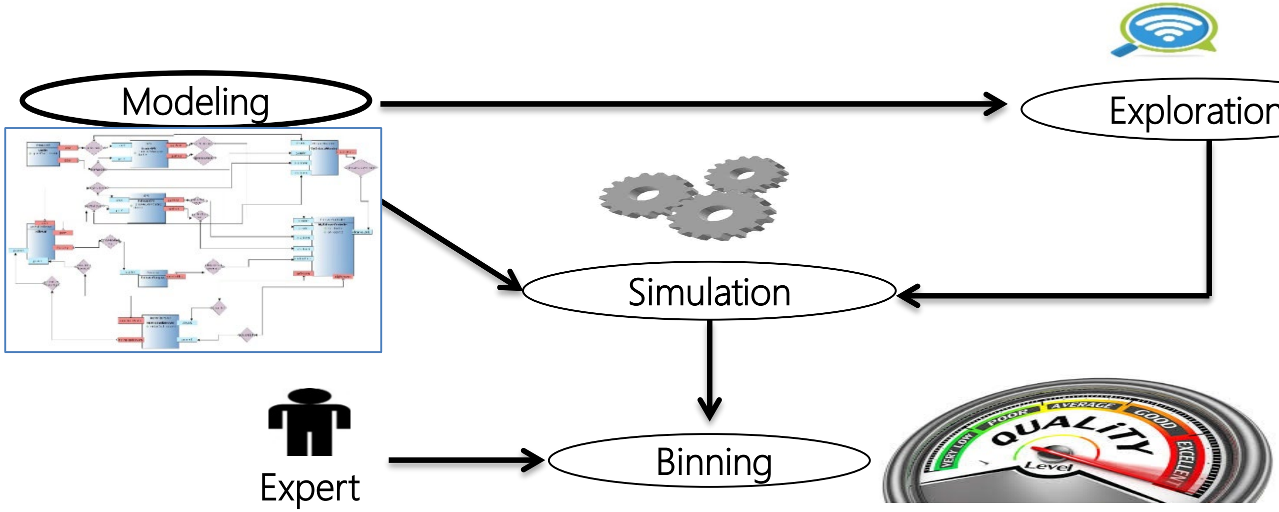


Control parameter tuning

- Objective: automate the tuning of control parameters
- The proposed approach is composed of 8 steps

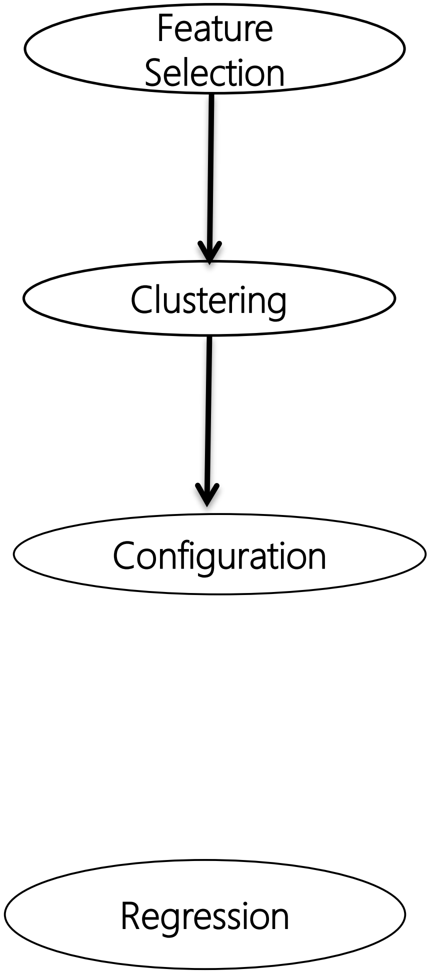
- Steps related to model simulation
  - Explore the controller behavior by simulating different configurations
  - Evaluate the simulation results regarding a control objective

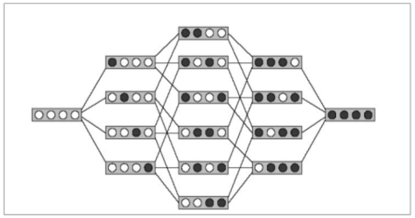


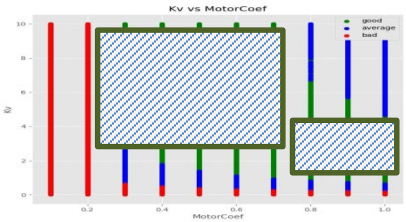


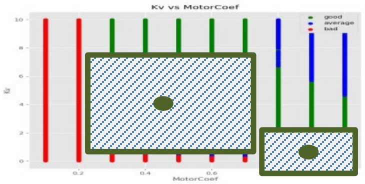
Control parameter tuning

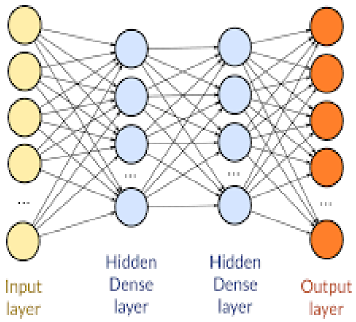
- Mode definition steps
  - Identify contextual parameters that have a significant impact on performances of the controller
- Regrouping contexts and parameterization
- One representative valuation of control parameters for each operational mode
- Online adaptation
  - Assign adequate control parameters for unknown contexts





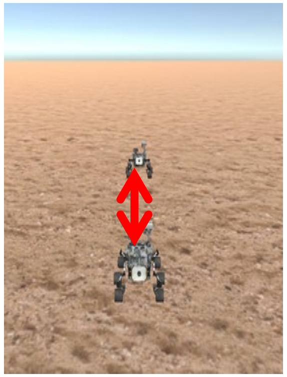


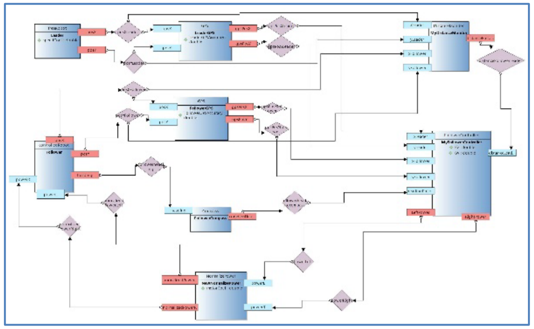




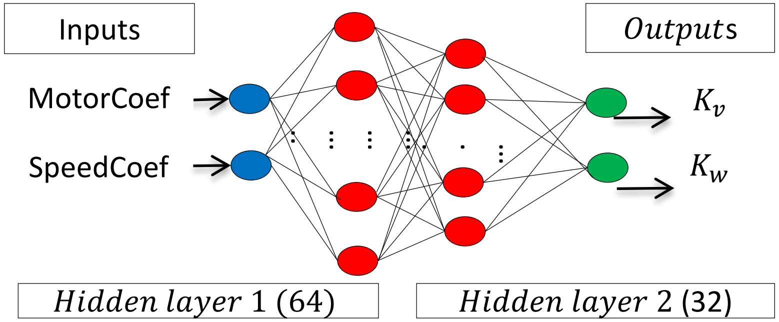
Control parameter tuning

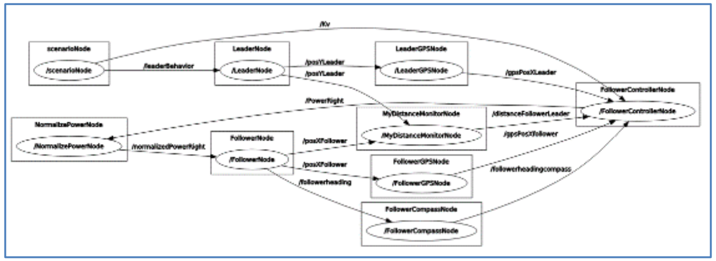
- Leader / Follower application

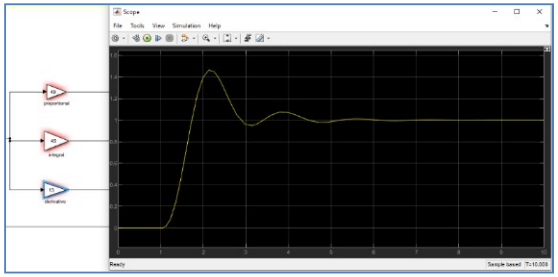


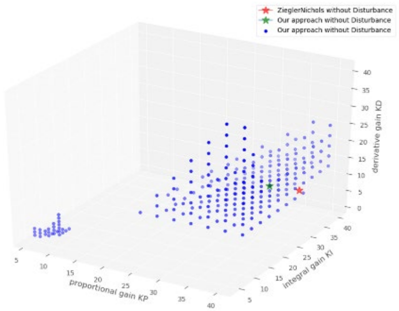


ROS code generation







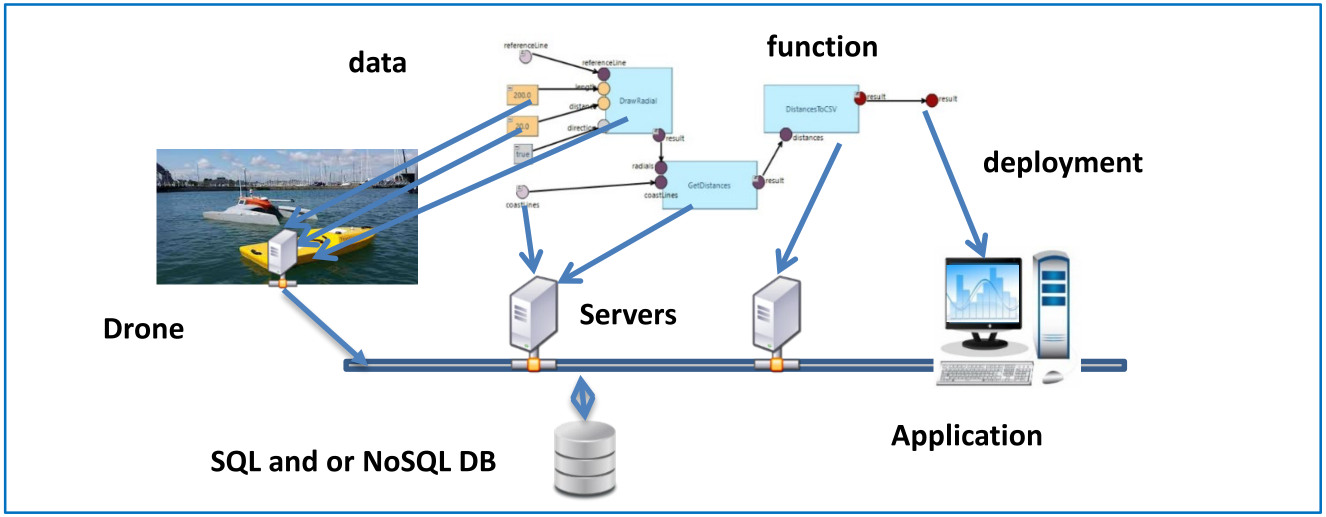


Data acquisition (in progress)

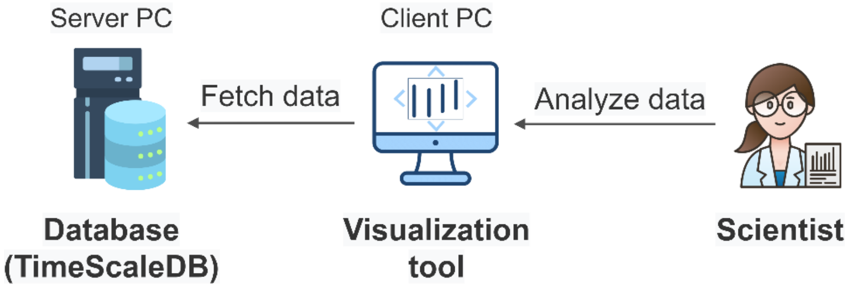
- A *framework* for data acquisition provided by marine drones

- Efficient storage and data navigation
  - BigData DB for time series
  - Optimized framework
  - Metadata

- Acquisition
  - Reuse of generic algorithms
  - Platform independent
  - Embedded (or not) deployment



Data storage and visualization



Server PC

Client PC

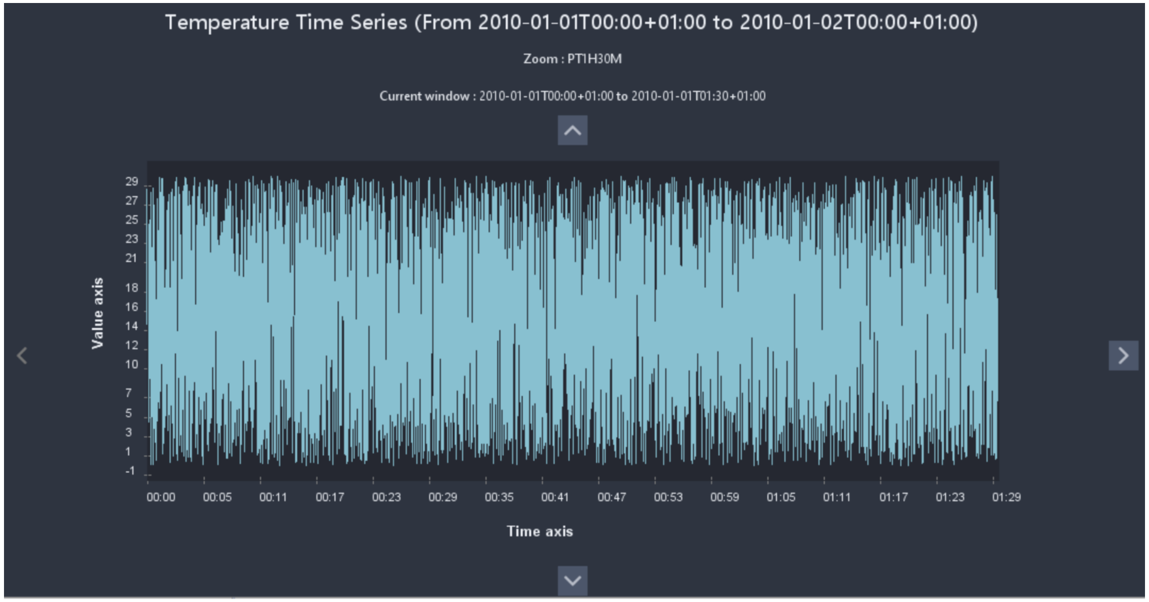
Fetch data

Analyze data

Database (TimeScaleDB)

Visualization tool

Scientist

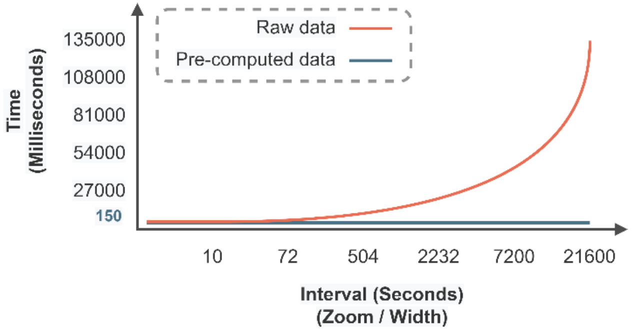


Temperature Time Series (From 2010-01-01T00:00+01:00 to 2010-01-02T00:00+01:00)

Zoom : PT1H30M

Current window : 2010-01-01T00:00+01:00 to 2010-01-01T01:30+01:00

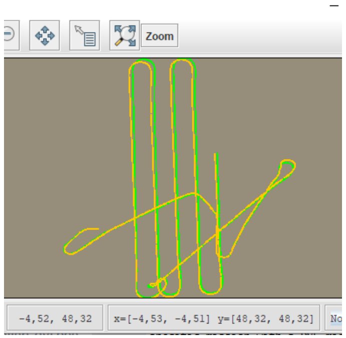
- Min-Max visualization
  - *Mean*
  - *Boxplot*
- Navigation
  - Zoom in / out
  - Previous / next
- Pre-computed tables
  - Depending on zoom scale and number of pixels
- Prefetch
  - Zoom in/out, previous, next



Surface drone application

- Transect and swath
  - Current
  - Waves
  - Wind

- Trajectory control tuning
- Environment estimation
- Safe control



```
-4,52, 48,32    x=[-4,53, -4,51] y=[48,32, 48,32]    No
```