# IMU

ROBOTICS

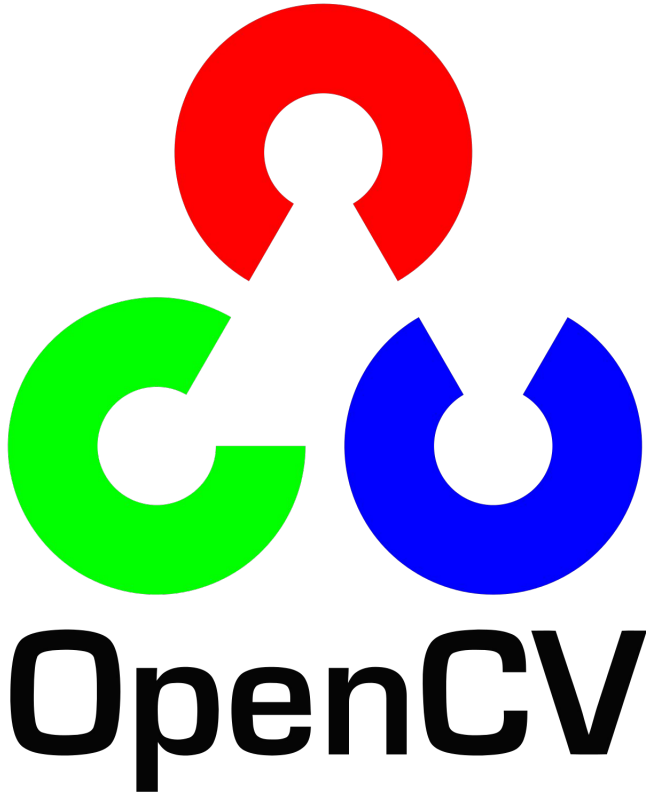
POLITECNICO
MILANO 1863

# IMU

-Accelerometer

-Gyroscope

-Magnetometer



# IMU

-Retrieve orientation from IMU data:

-Fuse different sources:

  -Acc+gyro: 3D orientation

  -Acc+gyro+mang: Absolute orientation

ROS package imu_tools implements a set of filters to fuse IMU data and retrieve robot orientation

http://wiki.ros.org/imu_tools

# IMU

```
sudo apt-get install ros-kinetic-imu-tools
```

Than we run the complementary filter node using a launch file and setting the proper parameters

Default configuration subscribe to `imu/data_raw` and publish on `imu/data`

# IMU

```
<!-- ComplementaryFilter configuration -->
<launch>
<node pkg="imu_complementary_filter" type="complementary_filter_node"
      name="complementary_filter_gain_node" output="screen">
    <param name="do_bias_estimation" value="true"/>
    <param name="do_adaptive_gain" value="true"/>
    <param name="use_mag" value="false"/>
    <param name="gain_acc" value="0.01"/>
    <param name="gain_mag" value="0.01"/>
   </node>
</launch>
```

# IMU

The imu tools package also offers a rviz plugin for imu visualization:

open rviz, then go to add->by display type->rviz imu plugin->imu

Now on the left window, under imu-> Topic select the imu topic

In the rviz visualization window you can see the imu axis

# Robot Localization

ROBOTICS

POLITECNICO
MILANO 1863

# Robot Localization

Collection of state estimation nodes:

-ekf_localization_node and ukf_localization node for imu and odometry sensors fusion

-navsat_transform_node for gps data integration

Unlimited number of sensors

15 dimension state estimation

# Robot Localization

Collection of state estimation nodes:

-ekf_localization_node and ukf_localization node for imu and odometry sensors fusion

-navsat_transform_node for gps data integration

Unlimited number of sensors

15 dimension state estimation

# Robot Localization

Requires sensors to be properly mounted on the vehicle:

REP 103 as a reference

Data using ENU

acceleration:

* Measure 9.81 meters per second squared for the Z axis.

* If the sensor is rolled 90 degrees (left side up), the acceleration should be 9.81 meters per second squared for the Y  axis.

* If the sensor is pitched 90 degrees (front side down), it should read 9.81 meters per second squared for the X axis.

Requires a properly formed TF tree

# State estimation nodes

To add a sensor:

<param name="sensor" value="/topic"/>

The configure it:

```
<rosparam param="sensor_config">[false, false, false,
                                 false, false, false,
                                 true,  false, false,
                                 false, false, false,
                                  false, false, false]</rosparam>
```

Set to true the values you want to use from your sensor

# State estimation nodes

frequency: frequency at which the filter will produce data

sensor_timeout: time in s after we consider a sensor to time out

two_d_mode: if we work in 2D, it automatically set to false all third dimension components of sensors initialization

# State estimation nodes

[sensor]_differential: if you only have pose sensors and you want to retrieve velocity

[sensor]_relative: compute pose from initial point (if you want everythin to start at (0,0,0))

imuN_remove_gravitational_acceleration: if you imu does not automatically substract g force

# navsat_transform_node

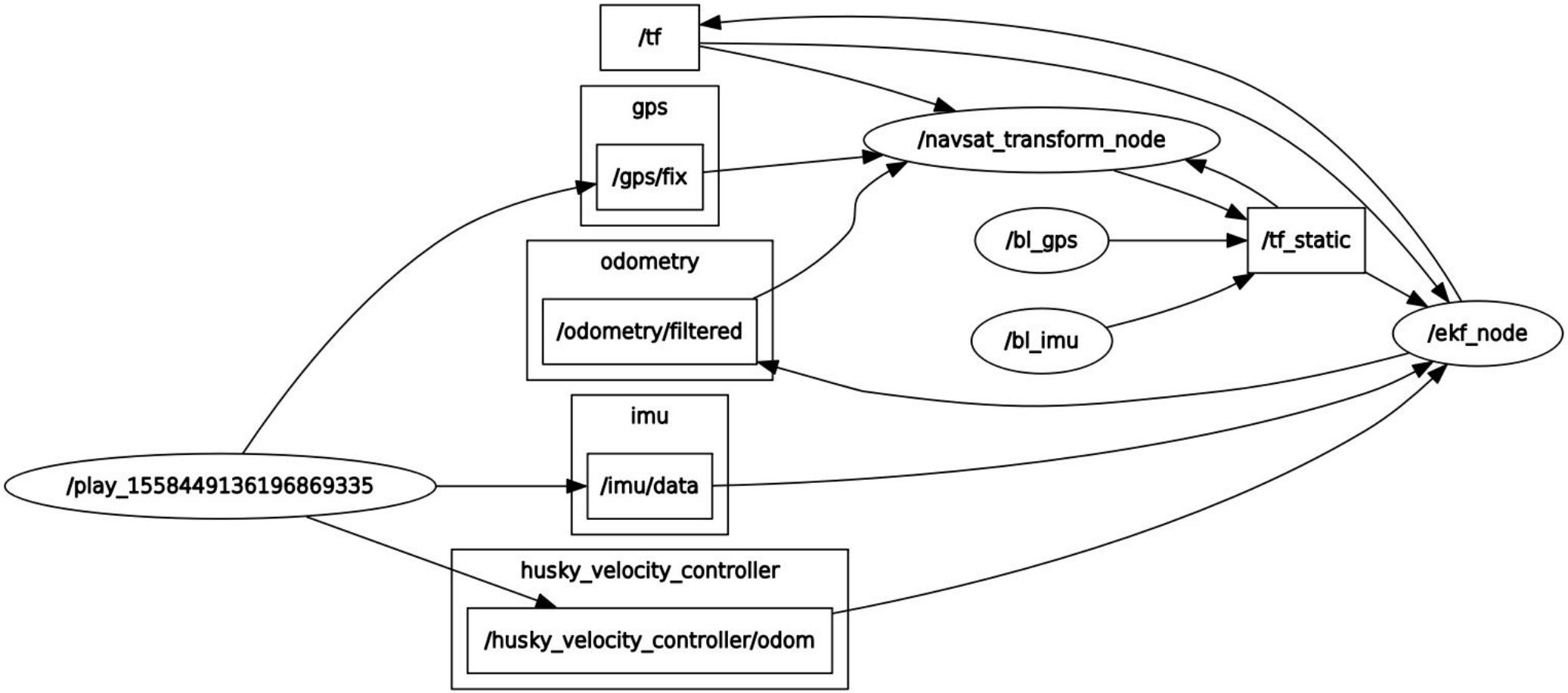
magnetic_declination_radians: correct the magnetometer based on position

yaw_offset: yaw should be 0 when facing est, otherwise you have to correct it

use_odometry_yaw: if you don’t want to use IMU for initial heading

datum: can specify the origin

# Basic configuration



# Data visualization (mapviz)

ROBOTICS

POLITECNICO
MILANO 1863

# mapviz

Rviz works fine if you want to visualize robot data, but it does not support gps data visualization

Mapviz is a ros package built especially for gps visualization with great integration with maps

```
sudo apt-get install ros-kinetic-mapviz*
```

# mapviz

To start mapviz use its launch file, do not simply start using rosrun, instead use:

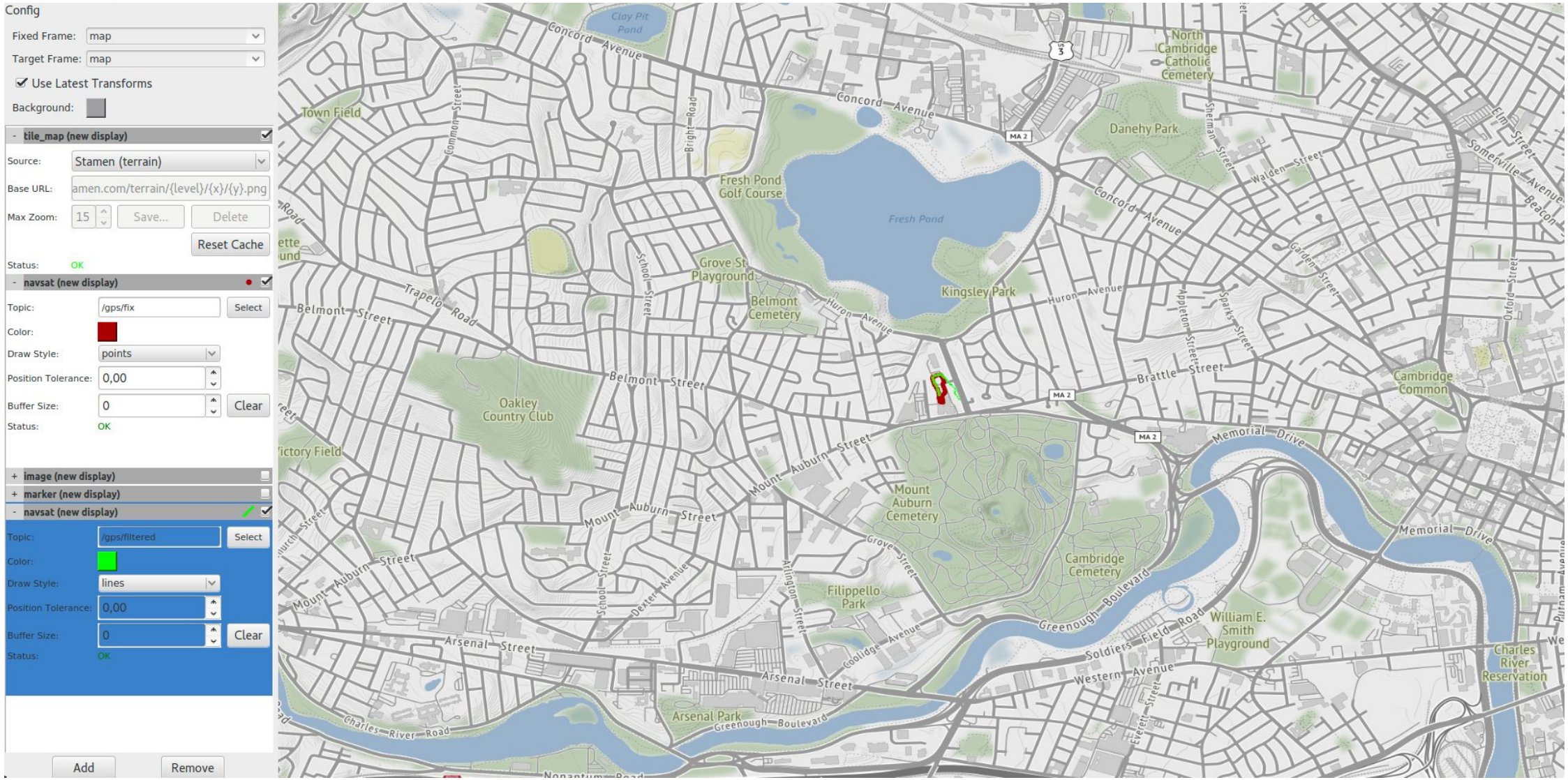
```
roslaunch mapviz mapviz.launch
```

Similarly to rviz you can select the tipe of sources, for GPS data select navsat, than select the topic you want to subscribe

And also superimpose a map, selecting the source on tile_map menu

In the robot localization project folder is also saved a file with the configuration for this project, so you can directly laod it

# mapviz


Config
Fixed Frame: map
Target Frame: map
Use Latest Transforms
Background:
tile_map (new display)
Source: Stamen (terrain)
Base URL: amen.com/terrain/{level}/{x}/{y}.png
Max Zoom: 15
Save...
Delete
Reset Cache
Status: OK
navsat (new display)
Topic: /gps/fix
Select
Color:
Draw Style: points
Position Tolerance: 0,00
Buffer Size: 0
Clear
Status: OK
image (new display)
marker (new display)
navsat (new display)
Topic: /gps/filtered
Select
Color:
Draw Style: lines
Position Tolerance: 0,00
Buffer Size: 0
Clear
Status: OK
Add
Remove
Concord Avenue
Clay Pit Pond
Town Field
Common Street
Bright Road
Fresh Pond Golf Course
Fresh Pond
Grove St Playground
Belmont Cemetery
Kingsley Park
Danehy Park
North Cambridge Catholic Cemetery
Sherman Street
Walden Street
Somerville Avenue
Beacon
Elm Street
Garden Street
Oxford Street
Huron Avenue
Trapelo Road
Belmont Street
School Street
Oakley Country Club
Victory Field
Brattle Street
Cambridge Common
Appleton Street
Sparks Street
Memorial Drive
Mount Auburn Street
Mount Auburn Cemetery
Cambridge Cemetery
Grove Street
Filippello Park
Arlington Street
Dexter Avenue
Greenough Boulevard
Soldiers Field Road
Western Avenue
William E. Smith Playground
Charles River Reservation
Arsenal Street
Arsenal Park
Charles River Road
Everett Street
Coolidge Avenue
MA 2
US 3