Bremen MSS Datasets - Technical Description

v1.0, December 2023

ABSTRACT

This white-paper provides a technical description of the Bremen Mechanical Scanning Sonar (MSS) Datasets. It complements the related academic paper "Scanning Sonar Data from an Underwater Robot with Ground Truth Localization" (*currently under review*) by providing additional information on the technical details and the usage of the datasets.

1 License Information

The data will be released under a *Creative Commons Attribution-NonCommercial (CC BY-NC)* license[1] as soon as a related academic paper is published. The data is hence currently not yet released for free use. In case you found the URL/data by coincidence and you are interested in using it, please contact Andreas Birk (abirk@constructor.university). You can also simply regularly re-visit the web-page, the data will be as mentioned released once a related academic paper is published and this section will be accordingly updated.

2 Overview

The *Bremen MSS Datasets* provide Mechanical Scanning Sonar (MSS) data recorded from an Unmanned Underwater Vehicle (UUV) and related data from navigation sensors. The UUV is localized with ground truth using an optical tracking system. The collection consists of 14 datasets of which 13 are related to different robot motions. There is 1 additional calibration dataset where the UUV is standing still. The calibration is provided to be able to tune 3rd party localization methods on a robot location that is fixed for a while, respectively to analyze the ground truth beyond the analysis provided by the Qualisys system itself. In all datasets, ground-truth localization is provided.

A list of the datasets with a description of the sensor data therein and the environment-conditions in which it was recorded can be found in the related academic paper. The data can be accessed via:

- https://robotics.constructor.university/bremen-mss-datasets/

The data recording is based on the widely used Robot Operating System (ROS) running on the UUV. Every dataset is hence provided as a ROS2 bag. In addition, the data is provided as YAML-files to allow an easy usage with a wide range of software for data science and for mathematical processing independent of ROS. The YAML-files come in two versions, namely once with all sensor data included in one single file per dataset and once as a collection of files per dataset where the data of each sensor is stored in a separate file.

In order to use the ROS2 bags, we provide a ROS2 package including the message definitions. Furthermore, the markers-messages of the Qualisys motion capture system used for the underwater ground truth tracking need to be installed. The required ROS2 packages can be downloaded from:

- https://github.com/constructor-robotics/datasetMessageDefinitions.git
- https://github.com/MOCAP4ROS2-Project/mocap.git
- https://github.com/MOCAP4ROS2-Project/mocap_msgs.git

3 Coordinate Systems and Sensor Locations

In Fig. 1 the BlueROV UUV used in the experiments is shown as an CAD model, which is used in the following to illustrate the coordinate frames and their locations relative to the vehicle frame for which ground truth is provided. The x-, y-, respectively z-axis is colored in red, green, respectively blue in each frame for the ease of reference.

[1] https://creativecommons.org/licenses/by-nc/4.0/

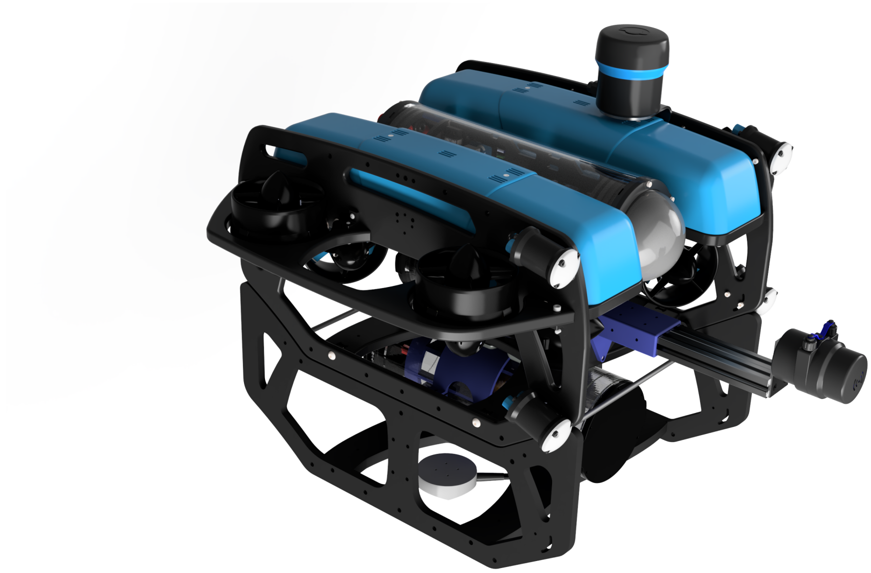


Figure 1. BlueROV2 CAD model, which is used to illustrate the coordinate frames including their locations.

For the sake of simplicity, the coordinate system that provides the reference frame for the UUV is in the geometric center of the BlueROV. This frame is shown in Fig. 2. The standard convention for underwater vehicles is used for the BlueROV2, i.e., a North-East-Down (NED) coordinate system, respectively a right-handed frame with the x-axis pointing forward and the z-axis pointing down. The same convention is used for the world- or main-frame in which the vehicle-frame is localized.

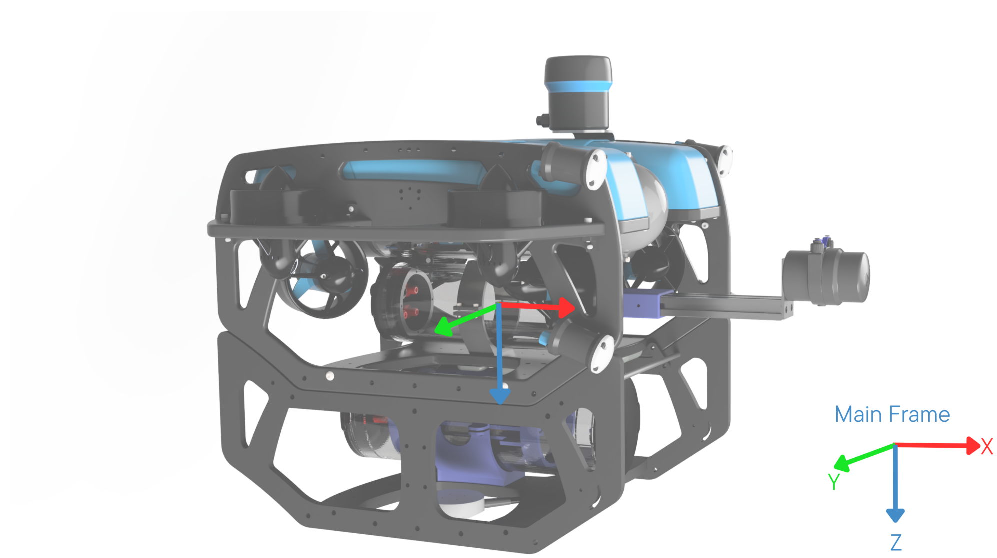
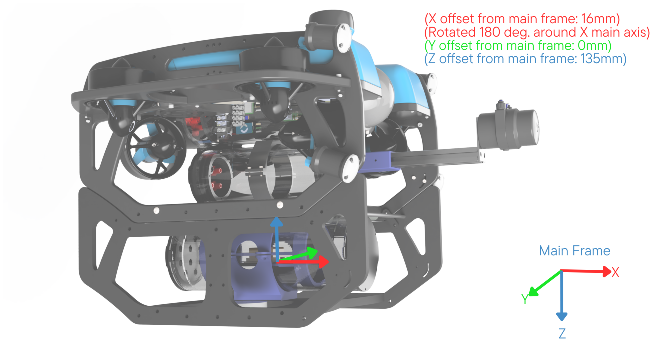
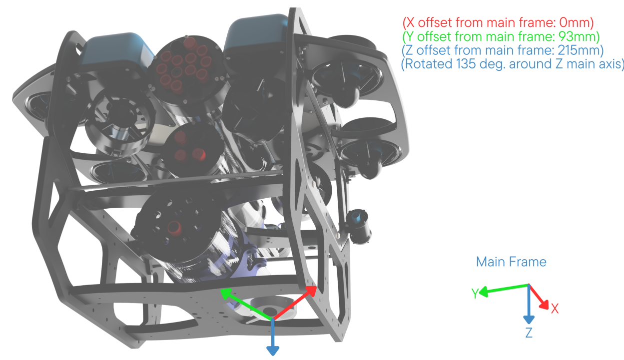


Figure 2. Reference Coordinate System of the BlueROV2. It is placed at the geometrical center of the robot.

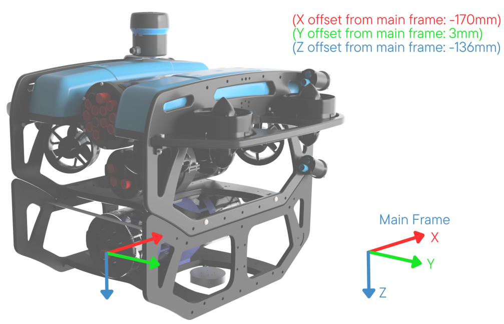
For each sensor, the according sensor frame is shown in Fig. 3. On the top right of each figure, the placement of the frame relative to the UUV-frame is provided, i.e., information on the translation and rotation components.



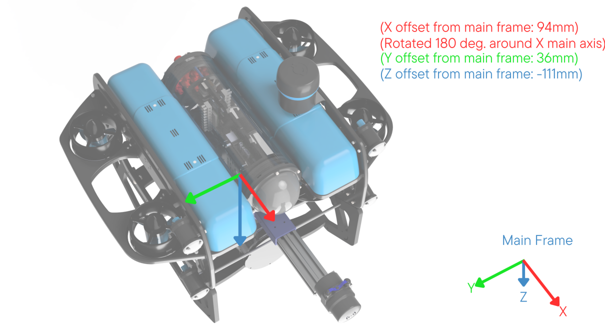
(a) MTi-300 IMU



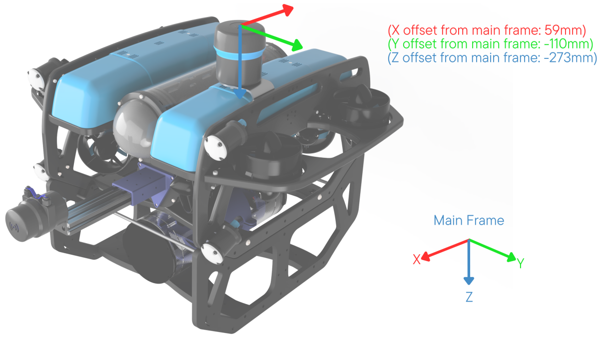
(b) A50 DVL



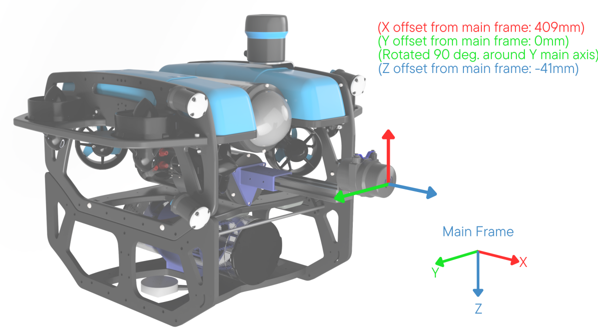
(c) Bar30 pressure



(d) PX4 IMU



(e) Ping360 MSS



(f) Micron DST MSS

Figure 3. The frames of the different sensors that are recorded in the datasets.

3.1 Inertial Measuring Unit (IMU)

The main IMU is an Xsens MTi-300, which is a high quality MEMS-based sensor. It is mounted inside the watertight bottom tube of the BlueROV2. The according coordinate system is shown in Fig. 3a.

3.2 Doppler Velocity Log (DVL)

The DVL used in the experiments is a A50 from Waterlinked. It is one of the lowest cost DVLs on the market. It is placed at the bottom right of the vehicle, pointing downward to measure the motion over ground. It is rotated in yaw due to mounting and cabling reasons. Its coordinate system is shown in Fig. 3b.

3.3 Pressure Sensor

The pressure sensor is a Bar30 High-Resolution sensor from BlueRobotics. It provides the absolute pressure, i.e., an indication of the water depth at which the vehicles is located. The sensor is mounted in the back of the UUV. Its coordinate system is shown in Fig. 3c.

3.4 Flight Computer with Inertial Measuring Unit (IMU)

For control and low-level state estimation, a Pixhawk 4 flight computer from HolyBro with PX4 software is used on the UUV. it is located in the upper tube of the BlueROV2. The Pixhawk 4 has an own IMU, which is less performing than the Xsens MTi-300. The data can nonetheless be of interest for comparison, respectively the analysis of methods operating on very low-cost IMU data. Also, the data from both IMUs can be fused. The related coordinate system is shown in Fig. 3d.

3.5 Horizontal Mechanical Scanning Sonar (MSS)

A Ping360 MSS from BlueRobotics is mounted in a horizontal scanning orientation. It is positioned on top of the BlueROV2, such that a 360^o scan can be performed without any obstructions from the robot. The related coordinate system can be seen in Fig. 3e.

3.6 Vertical Mechanical Scanning Sonar (MSS)

A Micron DST MSS from Tritech is mounted in a horizontal scanning orientation on the UUV. It is positioned vertically in front of the BlueROV to avoid any interferences with the vehicle structure. Its coordinate system is shown in Fig. 3f.

4 Message Definitions

As mentioned before, the message definitions are provided as a ROS2-package at

- https://github.com/constructor-robotics/datasetMessageDefinitions.git

For reference, the message definitions are also provided here as listings for each sensor.

4.1 Inertial Measuring Unit (IMU)

For the Xsens MTi-300 IMU, the message definition is:

```
Header header

geometry_msgs/Quaternion orientation
float64[9] orientation_covariance # Row major about x, y, z axes

geometry_msgs/Vector3 angular_velocity
float64[9] angular_velocity_covariance # Row major about x, y, z axes

geometry_msgs/Vector3 linear_acceleration
float64[9] linear_acceleration_covariance # Row major x, y z
```

Listing 1. IMU

4.2 Doppler Velocity Log (DVL)

For the Waterlinked A50, the message definitions are:

```
uint64 timestamp #in ns
waterlinked_a50/TransducerReport report
```

Listing 2. DVL

```
float64 vx
float64 vy
float64 vz
float64 fom
float64 altitude
waterlinked_a50/Transducer[] transducers
bool velocity_valid
int32 status
string format
string type
float64 time
float64[9] covariance
uint64 time_of_validity
uint64 time_of_transmission
```

Listing 3. TransducerReport

```
int32 id
float64 velocity
float64 distance
float64 rssi
float64 nsd
bool beam_valid
```

Listing 4. Transducer

4.3 Pressure Sensor

For the BlueRobotics Bar30 High-Resolution pressure sensor, the message definition is:

```
Header header           # timestamp of the measurement

float64 fluid_pressure  # Absolute pressure reading in Pascals.

float64 variance        # 0 is interpreted as variance unknown
```

Listing 5. Water Pressure

4.4 Flight Computer with IMU

For HolyBro Pixhawk 4 IMU, the message definition is:

```
uint64 timestamp      # time since system start (microseconds)

int32 RELATIVE_TIMESTAMP_INVALID = 2147483647   # (0x7fffffff) If one of the relative timestamps is set
    to this value, it means the associated sensor values are invalid

# gyro timstamp is equal to the timestamp of the message
float32[3] gyro_rad                    # average angular rate measured in the FRD body frame XYZ-axis
    in rad/s over the last gyro sampling period
uint32 gyro_integral_dt                # gyro measurement sampling period in microseconds

int32 accelerometer_timestamp_relative  # timestamp + accelerometer_timestamp_relative = Accelerometer
    timestamp
float32[3] accelerometer_m_s2          # average value acceleration measured in the FRD body frame XYZ-
    axis in m/s^2 over the last accelerometer sampling period
uint32 accelerometer_integral_dt       # accelerometer measurement sampling period in microseconds

uint8 CLIPPING_X = 1
uint8 CLIPPING_Y = 2
uint8 CLIPPING_Z = 4

uint8 accelerometer_clipping    # bitfield indicating if there was any accelerometer clipping (per axis)
     during the integration time frame
uint8 gyro_clipping             # bitfield indicating if there was any gyro clipping (per axis) during
    the integration time frame

uint8 accel_calibration_count   # Calibration changed counter. Monotonically increases whenever
    accelerometer calibration changes.
uint8 gyro_calibration_count    # Calibration changed counter. Monotonically increases whenever rate
    gyro calibration changes.
```

Listing 6. PX4 IMU

4.5 Horizontal Mechanical Scanning Sonar (MSS)

For the horizontally scanning BlueRobotics Ping360 MSS, the message definition is:

```
std_msgs/Header header            # header info
float32 angle            # the measurement angle [rad]
uint8 gain  # Sonar Gain
uint16 number_of_samples
uint16 transmit_frequency # [kHz]
uint16 speed_of_sound # [m/s]
uint8 range      #  range value [m]
uint8 step_size  #  steps for each measurement(from 1 to 10)
uint8[] intensities    # intensity data [0-255].  This is the actual data received from the sonar
```

Listing 7. Ping360 MSS

4.6 Vertical Mechanical Scanning Sonar (MSS)

For the vertically scanning Tritech Micron DST MSS, the message definitions are:

```
std_msgs/Header header
float32 angle
float32 bin_distance_step
micron_driver_ros/IntensityBin[] bins
```

Listing 8. Micron Scan Lines

```
float32 distance
uint8 intensity
```

Listing 9. Intensity Bins

4.7 Ground Truth Measurements

The ground truth measurement definitions are based on the mocap message definitions. See Sec. 2 for the ROS2 package. The message definitions are:

```
std_msgs/Header header
uint32 frame_number
mocap_msgs/RigidBody[] rigidbodies
```

Listing 10. Rigid Bodies

```
string rigid_body_name
Marker[] markers
geometry_msgs/Pose pose
```

Listing 11. Rigid Body

```
int8 USE_NAME=0
int8 USE_INDEX=1
int8 USE_BOTH=2

int8 id_type
int32 marker_index
string marker_name
geometry_msgs/Point translation
```

Listing 12. Marker

5 Directory Structure

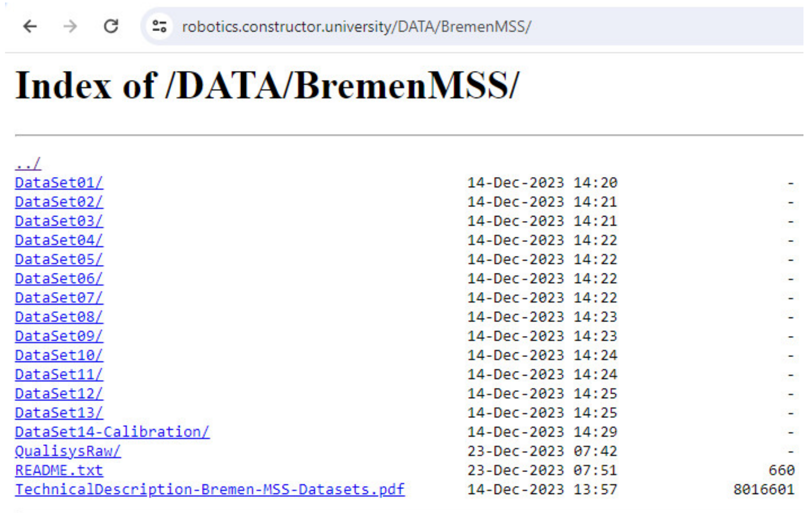
The datasets can be accessed via the following landing-page:

- https://robotics.constructor.university/bremen-mss-datasets/

This web-page provides some basic information and it points to the following browsable web-directory where the actual data is located:

- https://robotics.constructor.university/DATA/BremenMSS/

It is of course also possible to immediately start at the browsable web-directory. Fig. 4 shows the file-structure of the root of this web-directory. The folders "DataSet*N*" contain the files of the 13 datasets. The "DataSet14-Calibration" is the special case where the robot stands still, which can be used for calibration purposes if needed. Raw data from the Qualisys optical motion tracking system is found in the folder "QualisysRaw". It is only provided for the sake of completeness. Processed ground truth localization data is in addition provided for each of the datasets in the respective folders. Furthermore, a "README.txt" provides license information and the file "TechnicalDescription-Bremen-MSS-Datasets.pdf" is a copy of this white-paper.

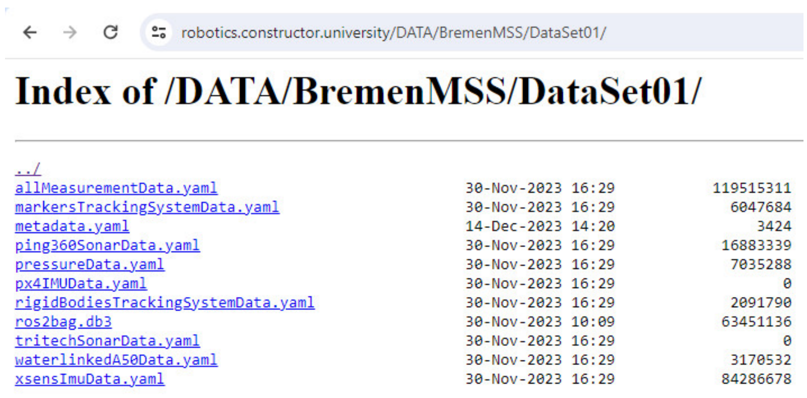


robotics.constructor.university/DATA/BremenMSS/

Index of /DATA/BremenMSS/

../		
DataSet01/	14-Dec-2023 14:20	-
DataSet02/	14-Dec-2023 14:21	-
DataSet03/	14-Dec-2023 14:21	-
DataSet04/	14-Dec-2023 14:22	-
DataSet05/	14-Dec-2023 14:22	-
DataSet06/	14-Dec-2023 14:22	-
DataSet07/	14-Dec-2023 14:22	-
DataSet08/	14-Dec-2023 14:23	-
DataSet09/	14-Dec-2023 14:23	-
DataSet10/	14-Dec-2023 14:24	-
DataSet11/	14-Dec-2023 14:24	-
DataSet12/	14-Dec-2023 14:25	-
DataSet13/	14-Dec-2023 14:25	-
DataSet14-Calibration/	14-Dec-2023 14:29	-
QualisysRaw/	23-Dec-2023 07:42	-
README.txt	23-Dec-2023 07:51	660
TechnicalDescription-Bremen-MSS-Datasets.pdf	14-Dec-2023 13:57	8016601

Figure 4. The files in the root directory at
https://robotics.constructor.university/DATA/BremenMSS/.



robotics.constructor.university/DATA/BremenMSS/DataSet01/

Index of /DATA/BremenMSS/DataSet01/

../		
allMeasurementData.yaml	30-Nov-2023 16:29	119515311
markersTrackingSystemData.yaml	30-Nov-2023 16:29	6047684
metadata.yaml	14-Dec-2023 14:20	3424
ping360SonarData.yaml	30-Nov-2023 16:29	16883339
pressureData.yaml	30-Nov-2023 16:29	7035288
px4IMUData.yaml	30-Nov-2023 16:29	0
rigidBodiesTrackingSystemData.yaml	30-Nov-2023 16:29	2091790
ros2bag.db3	30-Nov-2023 10:09	63451136
tritechSonarData.yaml	30-Nov-2023 16:29	0
waterlinkedA50Data.yaml	30-Nov-2023 16:29	3170532
xsensImuData.yaml	30-Nov-2023 16:29	84286678

Figure 5. The files in one of the dataset folders.

Fig. 5 shows the files that can be found in each of the "DataSet*N*" folders including the 14^{th} dataset where the robot stands still. As mentioned, there is a ROS2 bag "ros2bag.db3" with all the sensor recordings for each dataset. Furthermore, there are YAML-files, once as a single file "allMeasurementData.yaml" with all data and as separate files for the different sensors. As described in the related academic paper, not all datasets include all sensors. Nonetheless, there are according YAML-files in all folders for the sake of simplicity and consistency of the software provided on GitHub. The according separate file for that particular sensor is hence empty if a sensor is not included.

The data is also provided as a single Zip-file, which can be downloaded from the following locations:

- https://robotics.constructor.university/DATA/bremen-mss-datasets.zip
- *A Zenodo URL with DOI will be provided for the final version*

The latter provides persistent access to the data with a Digital Object Identifier (DOI). Please note that this zip-file is a large file with about 5.5 GB.