

ICP Localization and Walking Experiments on a TALOS Humanoid Robot

T. Lasguignes¹, I. Maroger¹, M. Fallon², M. Ramezani², L. Marchionni³, O. Stasse¹, N. Mansard¹

Abstract—This system paper describes the integration and the evaluation of an ICP-based localization system on the TALOS humanoid robot. The new generation of flash LiDAR systems, here an Ouster OS1-64, have made it possible to obtain 3D clouds at 10 Hz. Coupled with an Intel RealSense T265 providing visual-inertial odometry it is possible to localize the robot and use it to generate foot steps in real time to reach specific points. The approach is validated with a Qualisys motion capture system. It is also used to generate real-time walking motion on the TALOS humanoid robot. This paper is an integration paper showing that it is now feasible to accurately guide a commercial research humanoid robot in an environment in real time using a LiDAR system.

Index Terms—LiDAR-based localization, TALOS humanoid robot, walking

I. INTRODUCTION

In order to enable humanoid robots to autonomously perform useful behaviors in a semi structured environment, it is necessary to provide them with a way to autonomously localize themselves in a 3D environment. This is necessary to inspect specific important points in a factory, or to realize some manipulation tasks such as those performed during the DARPA Robotics challenge. The difficulty is to maintain a sufficient precision for the targeted tasks while being in an environment where parts of it are changing due to human activities. A similar problem occurs for autonomous cars navigating in a city. Indeed even if the architectural elements are consistent, the traffic is constantly perturbing the laser measurement. It is therefore important to have a robust, yet efficient, way to localize the robot in a large 3D environment and to update the current status of the map.

In this paper, the problem we are interested in is to assess the localization of the humanoid robot TALOS such that it is able to perform a behavior like manipulating an object or walking over debris. The behaviors can be adaptive through perception, control and planning, they may have various requirements in terms of localization precision.

Finally, the capability to embed the localization software in the robot to make it autonomous is an important aspect to consider.

A. State of the art

The research community working on the Simultaneous Localization And Mapping (SLAM) building problem is very active and consider many instances of the problem.

M. Fallon, L. Marchionni, O. Stasse and N. Mansard acknowledge the H2020 Memmo project, T. Lasguignes acknowledges the ROB4FAM project. I. Maroger acknowledges the ANR Cobot (ANR-18-CE10-0003).

¹ LAAS-CNRS, Université de Toulouse, France

² Oxford Robotics Institute, Univ. of Oxford, UK

³ PAL-Robotics, Barcelona, Spain

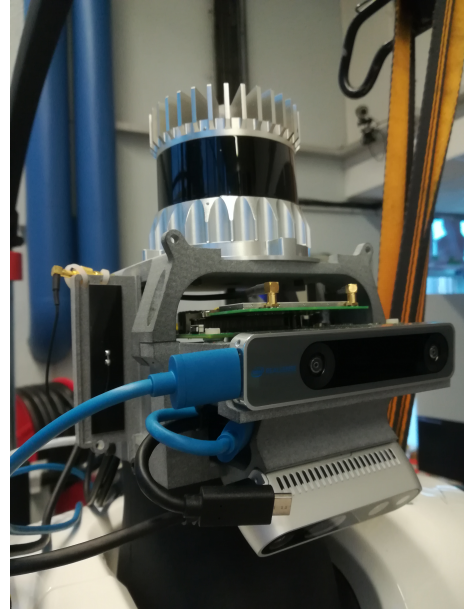


Fig. 1. TALOS equipped with an Ouster OS1-64 LiDAR, an Intel RealSense T265 tracking camera and an Intel RealSense D435i RGB-D camera

Historically, for humanoid robots, it is based on computer vision and is targeting affordable perception systems and therefore uses monocular camera [1]. Since then, the DARPA Robotics Challenge (DRC) has popularized multiple sensors systems including stereo camera and LiDAR [2] such as the MultiSense SLB product from Carnegie Robotics. It was, for instance, recently incorporated in the Walkman humanoid robot from IIT [3].

In [4], a laser-based localization was proposed for a NAO robot in a small multi-level indoor environment, later combined with a monocular camera in [5]. This localization was based on the Monte Carlo Localization technique presented in [6]. Nowadays the general framework of factor graph [7] allows to represent coherently the relationships between the robot state, its control vector and measurements of various sensors. From this representation, it is possible to use a general non-linear solver to solve localization problems, calibration problems, state estimation problems or map building problems. However, sensors having different measurement frequencies and different memory complexities. For instance, in this setting the IMU provides 6 real accelerations and gyrometer measurements at 1kHz, while the LiDAR provides 65536 range measurements at 10Hz. They need to be pre-processed in different ways before being incorporated in such representations.

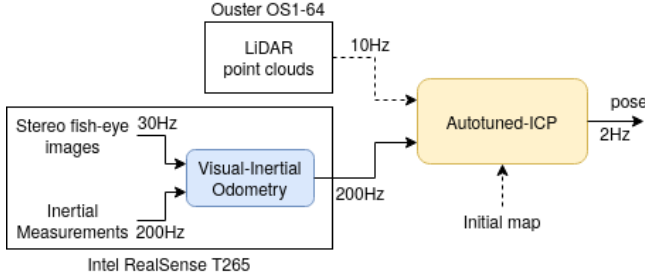


Fig. 2. Bloc diagram of the localization system, dash-arrows shows the flow of point clouds.

For LiDAR-based measurements, Iterative Closest Point (ICP) and 3D points accumulation maps are efficient methods to propose such pre-processing [8]. Such efficient method has been recently refined and deployed in the field of legged robots and more precisely for quadruped robots [9]. Indeed, it is now possible to find reliable industrial robots such as the ANYmal from ANYbotics or Spot from Boston Dynamics quadrupeds robots. This paper focuses on the practical implementation of such subsystem and its implementation for a research humanoid robot platform built by a company, here the TALOS robot from PAL-Robotics. Similar work has been realized using a Dense SLAM system [10], with a HRP-4 humanoid robot and an off-board computer equipped with a GPU. In this work we focus on the localization performance with an board computation.

In this work, we want to localize the TALOS humanoid robot in an indoor industrial-like environment.

B. Problem

For a given map m , the k -th LiDAR measurement $o_k \in \mathbf{R}^{64 \times 1024}$, and Visual Inertial Odometry (VIO) vio_k the goal of the localization system is to find a pose $p_k \in SE(3)$ of the head such that $p_k = ICP(m, o_k, vio_k)$. The overall scheme is presented in Fig. 2.

C. Contributions

The contributions of this paper are the following:

- Introduce a modification of the initial TALOS humanoid robot head design with a flash LiDAR system.
- Successfully implement a LiDAR-based localization combined with a Visual-Inertial Odometry on a TALOS humanoid robot in order to accurately guide it.
- Benchmark the localization system with a Motion Capture system.

II. LOCALIZING IN THE MAP

Fig.2 shows the localization pipeline implemented on the robot that will be presented in the following section.

A. Iterative Closest Point

The ICP algorithm [11] is used to register two point clouds with overlapping parts. It has 4 main steps: pre-filtering,

correspondence estimation, outlier filtering and lastly error-minimization. The issue of this algorithm is the high influence of outliers that can result in convergence to a local minimum.

B. ICP-based localization

We use the localization system from [12] and [9] over a known map.

The Autotuned-ICP (AICP) is an ICP registration method that adjusts the outlier filter of the ICP basis by computing an overlap parameter between the reading cloud and the reference cloud. It allows the registration of consecutive point cloud read to a reference cloud, which in our case will be a pre-built map.

This system uses the ICP implementation proposed by [8] and publicly available under the name *libpointmatcher*¹.

The system needs to be initialized at the start. For this paper, this pose is given by an approximation of the motion capture estimation. Further researches will be conducted on the estimation of this initial pose based on the map and LiDAR data.

C. Visual-Inertial Odometry initialization

Kinematic-Inertial Odometry is effective for high-frequency estimation over a short time interval, whereas Visual-Inertial Odometry, through the use of lower frequencies exteroceptive sensors, remains reliable over a longer period. As both odometries are available on the robot and the experiments does not require a high-frequency state estimation, it was decided to use the Visual-Inertial Odometry as an initialization instead of the Kinematic-Inertial Odometry, used in the initial system, for each iteration.

The Visual-Inertial Odometry is given by an Intel RealSense T265 tracking camera with a frequency of 200Hz. Fig.3 shows the comparison of the VIO to the Motion Capture estimation over 70 seconds, starting from the same initial point.

D. Building the map

In this paper, the indoor environment is known through a given map. In the following sections, different maps are discussed. Some of these maps were tested and compared in the experiment described in Sec.IV.

1) *Architectural 3D model*: Starting directly from an architectural 3D model can be a source of error. Indeed, as shown on Fig.4, an architectural 3D model can contain occluded walls, e.g. on the left of the figure, or small errors, e.g. the pillars in the middle. When comparing with the points accumulated using the ICP algorithm, mismatches appear with this features and may disrupt the convergence. As the walls in this kind of models are usually parallel in pairs, they create local convergences.

A solution to that problem is to use a prior map that has been extracted from the 3D model taking into account the

¹<https://github.com/ethz-asl/libpointmatcher>

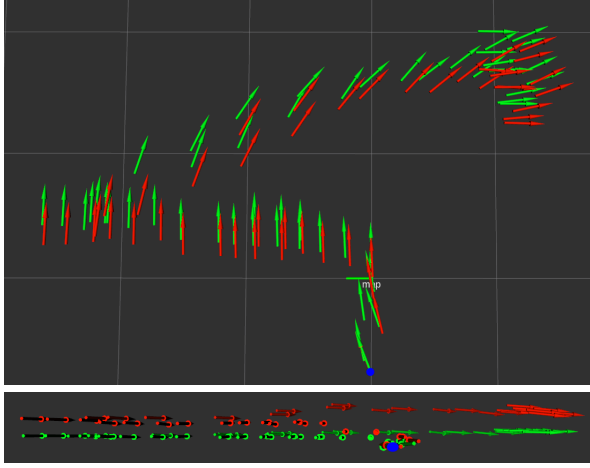


Fig. 3. Downsampling visualization of the Visual-Inertial Odometry (red) and the Motion Capture estimation (green) over the two first targets. Top: view on the $(\vec{x}, \vec{y})$ plane, $\vec{x}$ upward. Bottom: view on the $(\vec{z}, \vec{y})$ plane, $\vec{z}$ upward.

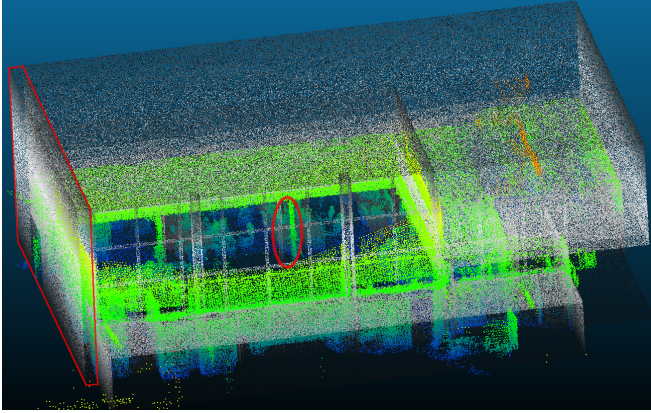


Fig. 4. Comparison between a 3D model (gray scale) and an accumulation of LiDAR scans (color scale). Examples of occluded walls and errors have been highlighted.

occlusions, for example by cleaning manually the model or thanks to a simulation, provided that the sensor model is known. In this work, it was decided to remove the occluded parts manually to compare with the other proposed map introduced in the following section.

2) *Using a scanning device:* The map might be built prior to the robot localization using a scanning device. As no such specific device was available, it was decided to use the data extracted from the LiDAR on the new head of TALOS. Fig.5 displays the map.

This kind of model can present some missing parts, e.g. in this experiment the map misses parts of the floor and the ceiling, but it is easier to produce than an architectural map if an appropriate device is available.

III. WALKING

In order to assess the efficiency of the ICP-based localization system, experiments where the robot has to walk towards multiple given positions were performed. For these experiments, the walking algorithm used to

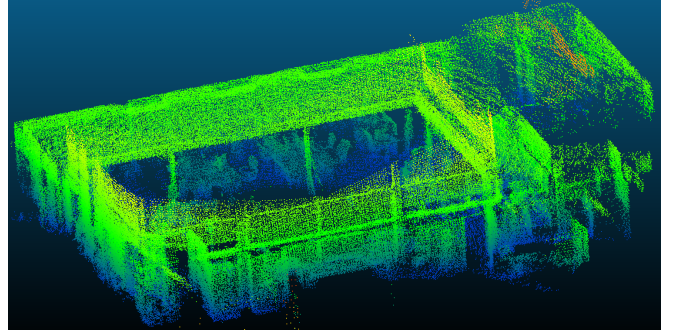


Fig. 5. Map obtained by accumulating multiple LiDAR scans from multiple positions and orientations.

generate the motion is the one provided by *PAL Robotics*. It was used to send a velocity command to the robot on the topic `/walking_controller/cmd_vel` at a 2 Hz frequency. The velocity command was merely computed as the distances along the $\vec{x}$ and $\vec{y}$ axis in the coordinate frame linked to the robot and angle between the robot and its goal orientation around the $\vec{z}$ axis. Thresholds of 0.1 m/s for the linear velocities and 0.12 rad/s for the angular velocity were applied on the command in order not to reach the robot limits. Moreover, the command were designed so that the robot stops walking when it is at small enough distances and angle from its target, respectively 0.08 m and 0.07 rad. This stopping distance was implemented to prevent the robot to trample on its target instead of stopping on it, this allows swift stops once on spot. Once the robot stops on a target, the robot is programmed to stay still during 8 s before moving on to its next target.

IV. EXPERIMENTS ON THE ROBOT

A. Experimental setup

The experimental room is equipped with a Motion Capture system (MoCap) including 20 infrared Qualisys Miquis M3 cameras sampling at 650 Hz with a 3×10^{-4} m accuracy on the imaged area. This motion capture system was used to record the positions and orientations of the robot head thanks to 5 passive markers placed on its torso and 1 placed on the top of his head. In this study, the MoCap measurements are hypothesized to be the ground truth data. Those measurements were compared to the ones made with the ICP-based localization system in order to assess its accuracy and its performances.

A new head was designed for the TALOS robot and is shown on Fig.1. From above to bellow, it is equipped with an Ouster OS1-64 LiDAR, an Intel RealSense T265 and an Intel RealSense D435i. The latter being unused in the experiment.

The OS1-64 flash LiDAR has a range of 0.8 m to 120 m, an horizontal field-of-view of 360° and a vertical field-of-view of 33.2° with 64 beams. For the experiment, the LiDAR was set to take 1024 horizontal samples per measure and to

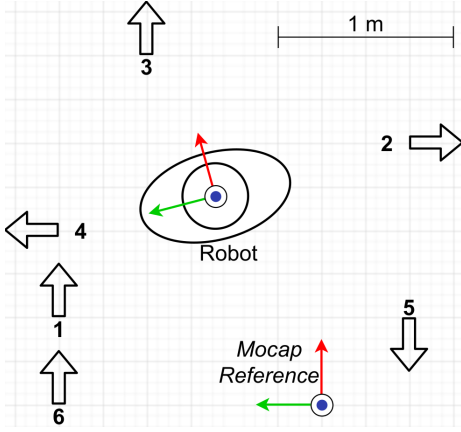


Fig. 6. Goal positions in the experimental room

give point clouds at 10 Hz.

The Intel RealSense T265 is a tracking camera embedding two Fish-Eye camera, an IMU and a VPU. This camera runs a V-SLAM algorithm on the VPU with an output frequency of 200 Hz.

B. Experimental protocol

The performed experiment consists of a series of target the robot has to successively reach. Thus, six goal positions were defined in the experimental room. Those positions are represented on Fig.6. They have been chosen in such a way that the robot ranges the whole MoCap imaged area and faces diverse orientation changes. The position of the robot at the beginning of the experiment is not defined in advance, it could be anywhere in the imaged area. Let us note that, if the robot starts from the origin of the MoCap reference frame, it will have to travel around 10.6 m to perform the whole experiment.

During the experiments, a dataset was recorded on a rosbag including the measurements made by the MoCap, the velocity order sent to the robot, the LiDAR data, the tracking camera images and estimations and the ICP-based localization results.

This experiment was performed 9 times with a TALOS robot. However, due to technical problems and hardware issues on the robot legs and ankles, only 3 experiments have been fully performed and successfully recorded. Among those recorded experiments, the first one was performed with the map built from the architectural 3D model (Sec.II-D.1) whereas the second and third ones were performed using the map built using a scanning device (Sec.II-D.2).

A capture from the video of the first experiment is shown on Fig.7. A video explaining the experimental protocol and showing the second experiment is available at <https://youtu.be/8Sk2eotdgjE>.

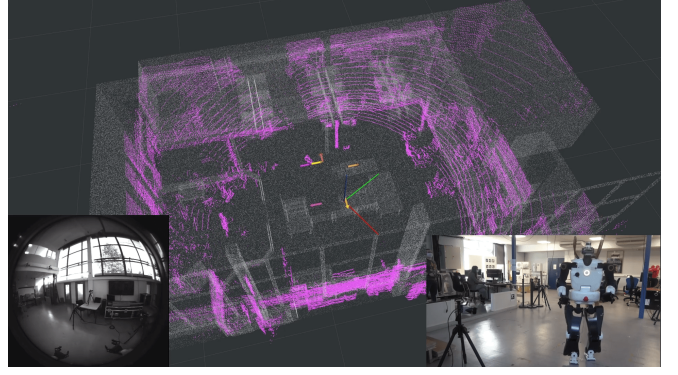


Fig. 7. Frame extracted from the video of the first experiment. The video shows the ICP data (gray: prior map, pink: read point cloud aligned, axis: estimated pose) and the targeted poses (arrows) in the middle, the image from a sensor of the Intel RealSense T265 on the left and the video of the experimental room and the robot on the right.



Fig. 8. Visualization of the recorded data on RViz (in red the MoCap set and in green the ICP set)

V. DISCUSSION

A. Comparison between the motion capture system and the ICP-based localization system

During the experiments on the robot, two data sets were collected simultaneously : one with the positions and orientations of the robot head recorded by the MoCap and another recorded by the ICP localization system. As both localization systems do not have the same sampling frequency, the raw data sets cannot be compared directly. Indeed, the MoCap set includes around 40 times the amount of data of the ICP one as Fig.8 shows. This is why both data sets were interpolated in order to have the same length.

First of all, for the three data sets, a delay of the ICP system over the MoCap can be measured shifting the two data sets to minimize the difference between the (x, y, z) positions and the θ orientation measured by the two localization systems. This delay scores between 0.51 s and 0.62 s depending on the experiment and is mainly due to the computation time of the ICP. In what follows, this

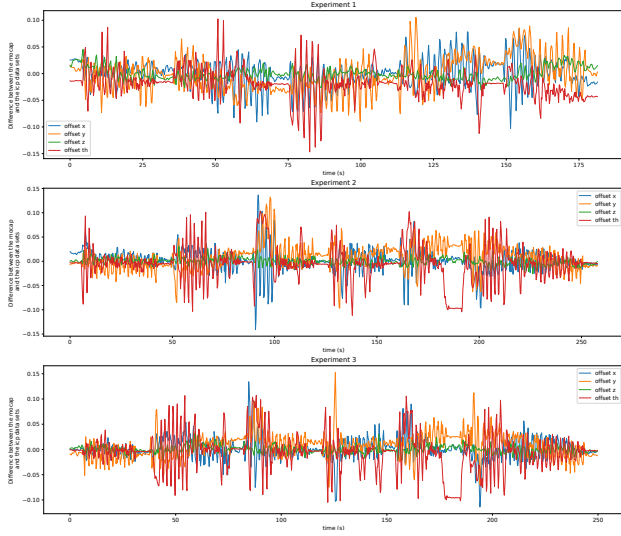


Fig. 9. Evolution of the errors between the data sets during the 3 experiments (x in blue, y in orange, z in green and θ in red)

	On $\vec{x}$ (mm)	On $\vec{y}$ (mm)	On $\vec{z}$ (mm)	Around $\vec{z}$ (rad)
Exp. 1	18.7 ± 17.2	24.9 ± 19.3	9.2 ± 7.2	0.025 ± 0.019
Exp. 2	14.8 ± 17.3	19.9 ± 17.8	5.4 ± 4.4	0.023 ± 0.027
Exp. 3	13.8 ± 15.9	18.9 ± 15.7	5.6 ± 4.8	0.024 ± 0.027

TABLE I

COMPUTED ERRORS BETWEEN THE MoCAP AND THE ICP DATA SETS
FOR ALL THE EXPERIMENTS

delay has been removed from the ICP set.

Then, the differences between the two data sets on the position and on the orientation can be computed. These differences will now be referred as offsets. The evolution of these offsets over the whole experiments is shown on Fig.9. These offsets are of two kinds :

- Structural offsets due to a difference between the coordinate frames of both localization systems. We make the assumption that these offsets are constant as it seems to be the case on Fig.9 and can easily be computed as the mean of the differences between the data sets over the experiment duration. Thus, these offsets scores 0.19 m, 0.058 m, 0.097 m respectively along the $\vec{x}$, $\vec{y}$ and $\vec{z}$ axis and 1.12×10^{-4} rad around the $\vec{z}$ axis for the first experiment. These offsets are lower for the second and third experiment (respectively 0.045 m, 0.021 m, 0.058 m and 0.048 rad) because the coordinate systems were reset after the first experiment. In what follows, these structural offsets are considered equal to zero as they have been removed to the ICP data set as the MoCap is taken as the ground truth. The results after removing these offsets are represented on Fig.10.
- Errors due to real differences of results between the two localization systems. They can be computed as the mean of the difference between the two data sets after removing the structural offset. These errors are presented in Tab.I.

During the 3 experiments, the robot started from 3 different positions with diverse orientations. As the results shown on Tab.I are similar whatever the experiment is, we can conclude that the ICP-based localization system does not depend on the starting pose. Moreover, the robot travelled different distances according to the experiments, respectively 8.93 m, 11.05 m and 10.69 m. As Fig.10 shows, the error does not seem to increase with the travelled distance, at least for distances around 10 m .

B. Accuracy on the goal positions

The goal positions can easily be identified on the data sets as they match with the points with zero velocity. To assess the accuracy of the ICP-based localization system on achieving the goal positions, the difference between the measured position and the desired goal position can be computed. The absolute average error on the goal positions ($\pm$ standard deviation) is 0.025 ± 0.016 m on the $\vec{x}$ axis, 0.039 ± 0.042 m on the $\vec{y}$ axis and 0.028 ± 0.012 rad around the $\vec{z}$ axis. All these errors are lower than the tolerance admitted on stops stated in Sec.III which means that the accuracy of the ICP localization system is satisfactory for the targeted application. Moreover, with such accuracy, more complex localization tasks could be considered. For example, in future works, the ICP-based localization system may be used to place the robot in front of an object to perform a task requiring accuracy such as drilling or crossing debris.

C. Effect of the map on the ICP-based localization system

As Tab.I shows, there is no significant difference (less than the standard deviations) in the errors between the first experiment and the second and third experiments whereas the ICP-based localization system was performed with different maps. This means that, even if the map built from the architectural 3D model is a priori less accurate than the map built using a scanning device, the ICP localization system has an accurate behavior in both cases.

VI. CONCLUSION

We presented a pipeline to localize the TALOS robot in an indoor environment based on LiDAR and Visual-Inertial odometry. The VIO is used to initialize the ICP steps registering read point clouds to a map point cloud. The system was successfully deployed on the robot and tested in a walking situation. The results were compared to a MoCap system considered here as the ground truth.

The ICP-based localization system shows an average error of 0.016 m, 0.021 m and 0.0067 m along the $\vec{x}$, $\vec{y}$ and $\vec{z}$ axis and of 0.024 rad around the $\vec{z}$ axis with respect to the motion capture system taken as the ground truth. Those results show that the ICP is accurate enough to target, in future works, more challenging localization tasks such as precisely placing the robot in front of an object to interact with it.

REFERENCES

- [1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

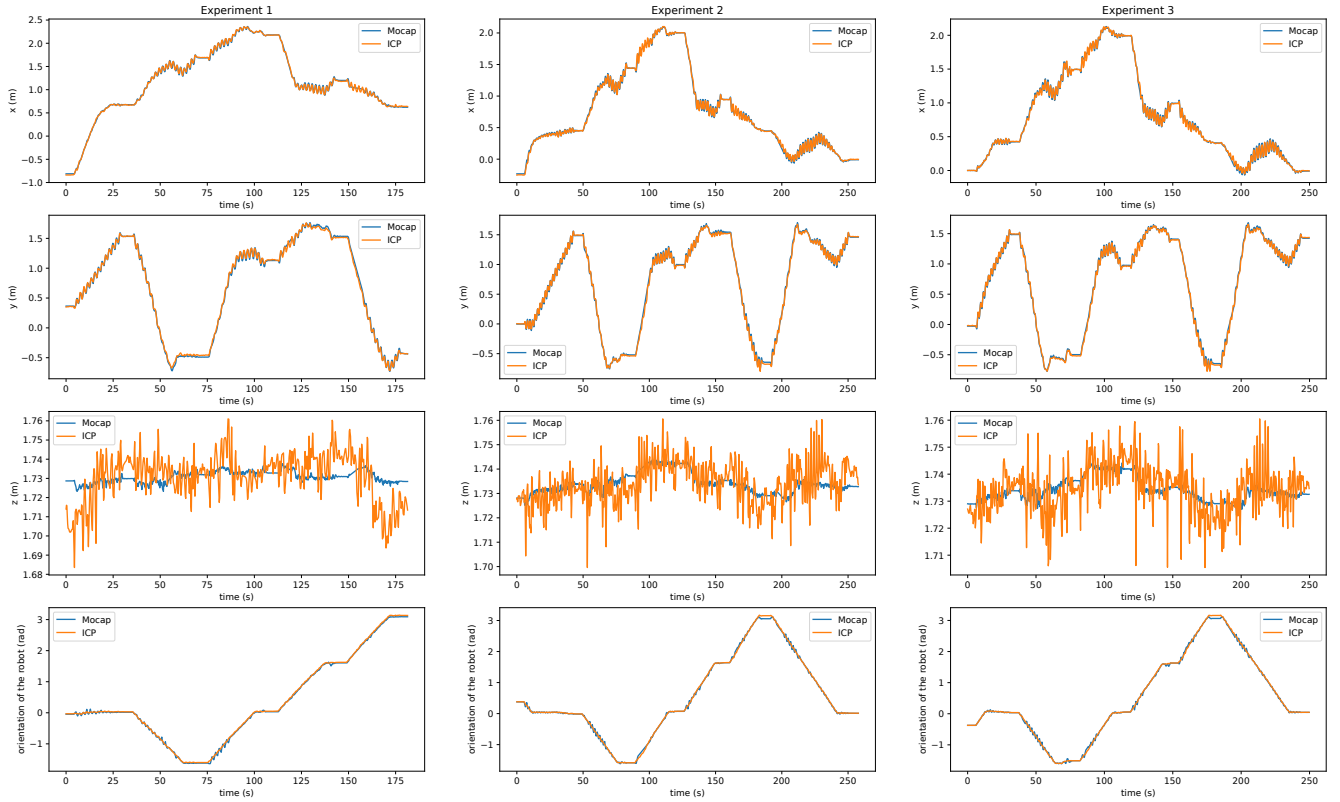


Fig. 10. Position (x on the first row, y on the second row, z on the third row) and orientation (θ on the last row) of the robot measured by the MoCap (in blue) and by the ICP localization system with the removed delays and structural offsets (in orange) during the 3 experiments

- [2] M. Fallon and al., “An architecture for online affordance-based perception and whole-body planning,” Tech. Rep., 2015.
- [3] N. G. Tsagarakis and al., “Walk-man: A high-performance humanoid platform for realistic environments: A high-performance humanoid platform for realistic environments,” *J. Field Robotics*, no. 7, p. 1225–1259, 2017.
- [4] A. Hornung, K. M. Wurm, and M. Bennewitz, “Humanoid robot localization in complex indoor environments,” in *IROS*, 2010, pp. 1690–1695.
- [5] S. Oßwald, A. Hornung, and M. Bennewitz, “Improved proposals for highly accurate localization using range and vision data,” in *IROS*, 2012, pp. 229–254.
- [6] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte carlo localization for mobile robots,” in *ICRA*, vol. 2, 1999, pp. 1322–1328.
- [7] F. Dellaert and M. Kaess, “Factor graphs for robot perception,” *Foundations and Trends in Robotics*, vol. 6, no. 1-2, 2017.
- [8] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, “Comparing icp variants on real-world data sets,” *Autonomous Robots*, no. 3, p. 133–148, Feb. 2013.
- [9] S. Nobili, R. Scona, M. Caravagna, and M. Fallon, “Overlap-based icp tuning for robust localization of a humanoid robot,” in *ICRA*, 2017, pp. 4721–4728.
- [10] A. Tanguy, D. D. Simone, A. I. Comport, G. Oriolo, and A. Kheddar, “Closed-loop MPC with dense visual SLAM - stability through reactive stepping,” in *ICRA*, 2019, pp. 1397–1403.
- [11] P. J. Besl and N. D. McKay, “A method for registration of 3-d shapes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992.
- [12] M. Ramezani, G. Tinchev, E. Iuganov, and M. Fallon, “Online lidar-slam for legged robots with robust registration and deep-learned loop closure,” in *ICRA*, 2020.