2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

A Study on Low-Drift State Estimation for Humanoid Locomotion, using LiDAR and Kinematic-Inertial Data Fusion

Vignesh Sushrutha Raghavan[1,2], Dimitrios Kanoulas[1], Chengxu Zhou[1], Darwin G. Caldwell[1], and Nikos G. Tsagarakis[1]

***Abstract*— Several humanoid robots will require to navigate in unsafe and unstructured environments, such as those after a disaster, for human assistance and support. To achieve this, humanoids require to construct in real-time, accurate maps of the environment and localize in it by estimating their base/pelvis state without any drift, using computationally efficient mapping and state estimation algorithms. While a multitude of Simultaneous Localization and Mapping (SLAM) algorithms exist, their localization relies on the existence of repeatable landmarks, which might not always be available in unstructured environments. Several studies also use stop-and-map procedures to map the environment before traversal, but this is not ideal for scenarios where the robot needs to be continuously moving to keep for instance the task completion time short. In this paper, we present a novel combination of the state-of-the-art odometry and mapping based on LiDAR data and state estimation based on the kinematics-inertial data of the humanoid. We present experimental evaluation of the introduced state estimation on the full-size humanoid robot WALK-MAN while performing locomotion tasks. Through this combination, we prove that it is possible to obtain low-error, high frequency estimates of the state of the robot, while moving and mapping the environment on the go.**

I. INTRODUCTION

Humanoid robot navigation in any environment requires the development of efficient sensor data fusion algorithms. This is especially paramount for robots that have the potential to operate in disaster scenarios, where the environment is unstructured and dangerous. The robot needs to be aware both of its own movements and the changes to the environment. For this reason, the robotics community has developed efficient state estimation algorithms [1], [2] to provide humanoids with every possible information about their body state within the surrounding environment. One important state is the pose of the pelvis or the base of the robot during navigation. It is crucial to know the base pose in the world frame, as it helps in building maps of the environment where the robot needs to execute operations.

Several solutions were introduced to solve the robot pose estimation problem, with respect to a fixed world reference frame. The studies in [3], [4], and [5] presented a base pose estimation based on data fusion from Inertial Measurement Units (IMU) and joint kinematics through an External Kalman Filter (EKF). The algorithm in [3] (applied on a quadrupedal robot) used IMU data to predict the robot's state and the data from joint kinematics to correct the state updates. The obtained state included estimates of the velocity, roll, and pitch angles of the base. The algorithm in [4] (applied on a humanoid robot) used separate estimations for the base and the upper body of the robot, with the goal being to reduce the computational complexity of linearizations of the EKF at every time step. While the aforementioned algorithms used the standard non-linear equation of motions, the algorithm in [5] (applied also on a humanoid robot) used the linear inverted pendulum model to estimate the base and Capture Point(CP). These were in turn used for control of joint torques to perform stable walking and preventing falls.

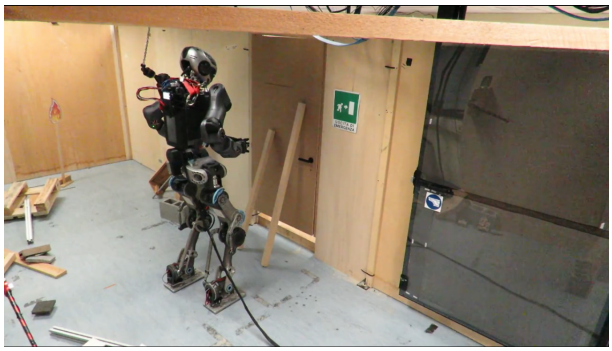

Fig. 1. The WALK-MAN humanoid robot in a disaster environment, created in the lab, before executing a debris removal task after locomotion.

Recently in [6], a sensor data fusion method combining data from IMU, joint kinematics, foot contact sensors and LiDAR named Pronto, was presented. The algorithm is very similar to [3] in terms of calculations for state predictions and corrections. In addition, foot contact force torque sensors were used to determine the foot in contact with the ground. The estimation takes place based on the motion of the pelvis or the base with respect to the stationary foot frame that is in contact with the ground. A very minimal drift of 2cm per 10 steps was achieved with the above mentioned kinematic-inertial state estimator, in quasi-static locomotion with step sizes of 15cm and 36cm.

Drift accumulation is one of the recurring problems of state estimation algorithms based on proprioceptive sensors, such as joint encoders and IMU. While trying to estimate global pose using proprioceptive sensors, the algorithm usually relies constantly on integrating a model using sensor data. The estimation is incrementally accumulative and dependent on previous values. In such estimations, the noisy nature of sensor data causes error in the estimates. This error

[1]Department of Advanced Robotics, Istituto Italiano di Tecnologia (IIT), Via Morego 30, 16163, Genova, Italy {Vignesh.Raghavan, Dimitrios.Kanoulas, Chengxu.Zhou, Darwin.Caldwell, Nikos.Tsagarakis}@iit.it
[2] Department of Information Engineering, University of Pisa

keeps getting accumulated and increases, thereby giving rise to drift. Estimates based solely on proprioception may result into inaccurate environment maps and hence may not be useful for navigation in challenging environments. For this reason, low drift accumulation and very accurate real-time state estimation is required. Typically, drifts are eliminated by using data from exteroceptive sensors, such as LiDAR scanners, monocular/stereo cameras, or other range sensors. LiDARs have become popular due to their ability to give robust, accurate, and repeatable data even in the presence of varying lighting conditions—unlike monocular/stereo cameras or most of the range sensors.

The Iterative Closest Point (ICP) [7] method is a commonly used approach to process data from LiDAR sensors to perform localization and motion estimation. Several approaches for robotic applications have been based on ICP, such as [8] and [9]. One major drawback of ICP-based algorithms that are based on range data (e.g., point clouds) is that they need sufficient overlap between the new incoming and the reference data, with which they are compared to. In [10], this issue was solved by tuning the inlier percentage based on the overlap of the range data. This led to more accurate localization and motion estimation, which was demonstrated on the Valkyrie humanoid robot, using the Carnegie Melon Multisense-SL depth/LiDAR sensor. The robot was able to walk to and from a target point repeatedly, without much drift accumulation. Whereas, this was not the case with just the kinematic-inertial state estimator.

Another method to eliminate drift was introduced in [6], where a particle filter-based laser scan localization was used to compare scans to a pre-made LiDAR data-based map. On one side a complete elimination of the drift was achieved, but on the other side, a map needed to be generated prior as input to the state estimation method. This is not ideal for robots that need to move freely in large environments. Similarly, there exist studies like [11] which have used Simultaneous Localization and Mapping (SLAM) on a humanoid robot, which are usually computationally expensive and either they require an initial stop-and-map the environment step, or they need repeatable locations/features.

Other than ICP, several other motion estimation approaches were introduced, which were based on laser range data. Recently, the LiDAR Odometry and Mapping (LOAM) [12] method, achieved impressive estimation results. The LOAM and its vision-based version V-LOAM [13], are currently ranked at the top of the KITII datasets [14]. LOAM extracts repeatable features from consecutive laser scans, such as smooth points belonging to a plane or edge points. It uses such points to run an optimization to minimize the distance between either an edge point and its corresponding edge line, or a plane point and its corresponding plane surface. Through this optimization, it obtains either the motion transform between the laser scans or the odometry and mapping. The method runs in two cores in a quadcore 2.5 GhZ computer with 6 GB RAM and generates accurate LiDAR-based motion estimates. To the best of the authors' knowledge, this method has not been yet applied on humanoid robots.

As the accuracy of LOAM has not been leveraged for humanoid robots, in this paper we investigate the accuracy, robustness, and the computational efficiency of a global state estimator, which combines the LOAM and the Pronto kinematic-inertial state estimator. The computational power consumption analysis is important for having the method working on the on-board computers of a humanoid robot, for long periods of time. The advantage of using the LOAM algorithm to process exteroceptive LiDAR data is that we obtain low-drift estimates that allow the generation of accurate maps without extensive localizations and loop closures. This is useful in challenging environments, where a robot may need to move continuously without stopping to create maps of the environment. We experimentally validate the introduced state estimation algorithm on our full-size humanoid robot WALK-MAN [15], under locomotion tasks. For the exteroceptive sensing, we use the rotating Hokuyo LiDAR sensor, which is part of the CMU Multisense-SL head. We compare the performance of the combined Pronto+LOAM estimator with the Pronto-only and LOAM-only estimations, in an effort to prove that it is possible to perform low drift, high frequency state estimation while simultaneously mapping the environment. We also present an analysis of the computational power consumed to run the combined state estimator.

We acknowledge that two existing, state-of-the art algorithms will be extensively used in this study. Although, through this study we aim to present and experimentally evaluate the accuracy, robustness, and precision of a state estimator that combines Pronto and LOAM for our humanoid robot WALK-MAN, to aid future works in unstructured and dynamic environment locomotion. The main contributions of this paper are as follows;

- We present a subtle 2-way connection to connect LOAM and Pronto estimates, so as to leverage both the drift reducing nature of LOAM and the high frequency estimation of Pronto.
- Through this new combination, we present an algorithm which performs much better that the original individual components.
- We experimentally establish that this combination of Pronto+LOAM provides similar improvements when reduction of drift and pose error are concerned, for different gaits and different duration of experiments.
- We also experimentally establish that it is possible to obtain drift reduction in the pose estimates of pelvis of a humanoid robot, using LiDAR data, with on-the-go map construction.

We first explain the state estimation method that integrates LOAM and Pronto (Sec. II). Following this, we present our experimental evaluation (Sec. III) on the WALK-MAN robot and we finally conclude with some future directions.

II. BASE POSE ESTIMATION FOR HUMANOIDS

As mentioned earlier, in this paper, we attempt to combine the Pronto and LOAM algorithms, on a humanoid robot,

that dynamically walks in the environment. First, we will explain the mathematical notations used in this paper and the robot and environment setup. Then, both state-of-the-art-algorithms will be explained in brief. This will be followed by the details of how we fuse data from the Pronto and LOAM algorithms to create an algorithm which is a new combination of base state estimation and LiDAR mapping.

A. Mathematical Notations

In this paper, the following notations would be used. Vectors will be represented with small-case italics, e.g, v. Matrices will be represented by camel-case bold letters, coordinate time instances with lower-case normal italics and scalars by the normal font. The coordinate frames will be referred as Σ_u, where the letters like u will be used to differentiate the frames. Furthermore, the matrix $\mathbf{H}_Q^P$ represents a homogeneous transformation of frame P with respect to frame Q.

In our robot, the pelvis frame is fixed in the middle of the waist of the robot with the x-axis facing forward and the z-axis upright when the robot stands in its homing position. The pelvis frame will also be referred to as the base of the robot. The LiDAR frame will be represented by Σ_l, while the base frame is represented as Σ_b. The world frame is represented by Σ_w. Quaternions representing rotation from the time-frame $t-1$ to t will be simply represented by $q(t)$. The absolute orientation quaternions and positions of the base w.r.t to the world frame will be simply represented by q_w and x_w respectively.

B. Robot and Environment Setup

The WALK-MAN humanoid robot [15] will be used for the proof of concept and the experimental evaluation in the paper. WALK-MAN is a 102 kg robot and is 195 cm tall. It has an IMU attached at its waist, and encoders for every joint. It is also equipped with a CMU Multisense-SL sensor, which consists of a stereo camera and a rotating Hokuyo LiDAR scanner. The primary aim of WALK-MAN is to function in disaster scenarios. An environment similar to the one shown in Fig. 1 will be used for our experiments. A motion capture OptiTrack camera system will be used to track markers attached rigidly to the waist of the robot and thereby provide ground-truth for the base state estimation.

C. Pronto-EKF

Pronto is an EKF-based algorithm with two primary steps, namely state vector prediction and correction update. The state vector consists of the following elements:

- 3D positions of the base frame Σ_b in the world frame: x_w.
- Global orientation quaternion: q_w.
- Global Linear velocity of the body frame Σ_b expressed in body coordinates: v_b.
- Angular velocity in the body frame: ω_b.
- Accelerometer and gyro biases: b_a, b_g.

The complete state vector s is $[x_w, q_w, v_b, \omega_b, b_a, b_g]$. The accelerations a_b and angular velocity w_b, obtained from the IMU, are used to predict the state vector $\hat{s}$, using standard non-linear kinematic models, similar to those presented in [16]. The corrections to the state vector $\hat{s}$ are provided by the joint kinematics. The calculations for determining these corrections are made based on the foot in contact with the ground. The process is described below.

The initial position of the base in the fixed world frame Σ_w, is set to [0,0,L], where L is the height of the base from the ground at the beginning of the experiment. The joint-kinematics module has low level filters of its own and provides transforms of each joint frame with respect to the base frame Σ_b. From the initial position and the joint kinematics module output, we know the initial pose of the feet with respect to the world frame. Using these initial poses and joint kinematics, we ascertain the pose of the feet in contact with the ground, with respect to the world frame Σ_w.

As mentioned earlier, the calculations for the correction update is made with respect to the foot in contact with the ground, the global pose of which is known. To determine which foot is in contact with the ground, foot contact force/torque sensors are used. A Schmitt trigger [17] with a threshold for the vertical force value F was used. When the vertical or Z component of the foot sensor data sensed a force greater than F, it was ascertained that the foot corresponding to that particular force sensor was in contact with the ground. For our robot, F was equal to 300 N. A simple state machine based on the force values obtained from the foot contact sensors of the two feet was used to ascertain which foot was in contact with the ground. Let the transformation matrix of the foot in contact with the ground w.r.t to the world frame be $\mathbf{H}_w^f$, where f represents the foot frame. From the joint kinematics, we acquire $\mathbf{H}_f^b$ which is the transform of the base frame with respect to the stationary foot. The base position and orientation in the world frame can be calculated as follows:

$$\begin{aligned} &\mathbf{H}_w^b = \mathbf{H}_w^f \times \mathbf{H}_f^b \\ \text{with} \quad &\mathbf{H}_w^b = [\mathbf{R}(q_w)|x_w] \end{aligned} \tag{1}$$

where $\mathbf{R}$ is the rotation matrix corresponding to the quaternion q_w. Two base positions at times t and $t-1$ are obtained and the linear velocity expressed in world coordinates, calculated as follows, is used for correcting the predicted state vector $\hat{s}$:

$$v_w = \frac{x_w(t) - x_w(t-1)}{\delta t} \tag{2}$$

where δt is the difference in time of the position estimates obtained from the joint kinematics. v_w is used as a measurement to update the estimated state $\hat{s}$.

With the above described kinematic-inertial fusion, the study in [6] achieved estimates which drifted at the low rate of 2cm every 10 steps. To eliminate this drift, a particle filter localization was used with a pre-created LiDAR scans based map, to provide corrections directly to the predicted estimate of x_w.

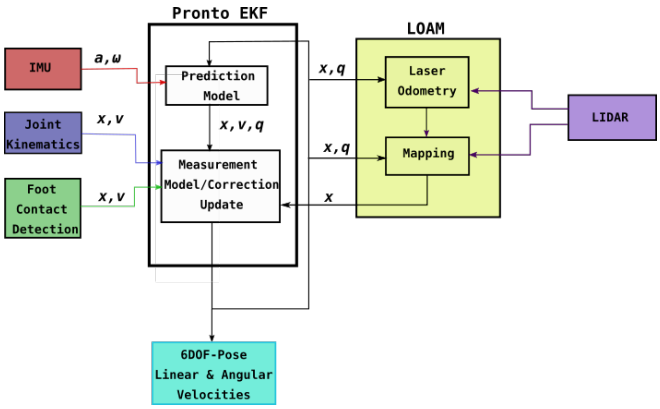

Fig. 2. The block diagram detailing an overview combination structure of the Pronto and LOAM algorithms. The estimation of the particular base state contributed to by the sensor or processing block is represented by the italic letters. The position is represented by x, the linear velocity by v, the acceleration by a, the angular velocity by ω and the orientation quaternion by q.

D. LOAM

The LOAM algorithm uses data from a rotating 2D laser scanner and has two primary steps. The first step is the accumulative motion estimation or odometry. Motion is estimated and accumulated in a set of three scans of one single sweep. The algorithm extracts plane points and edge points in each scan using a simple smoothness coefficient. Based on this, it minimizes the distance between an edge point and the corresponding line or a plane point and the corresponding plane. In this way the motion performed during the three scans is obtained. This motion is accumulated with the previous motions to provide odometry. The initial transform of the points to the laser frame Σ_l is done using a rough orientation and position accumulation based on the accelerations and angular velocities obtained from an IMU attached to the rotating laser scanner. This rough pose is also used as initial pose for the minimization optimization to estimate the motion. This estimated odometry is published at 10Hz.

The second step is the mapping. The mapping considers scans from one complete sweep of the rotating laser scanner and hence runs at a lower frequency of approximately 1Hz. The mapping module uses the pose of the LiDAR accumulated by the odometry module, and matches features, namely the edges and plane points, obtained using the smoothness coefficient from one sweep with features of the next sweep and further refines the LiDAR pose using this matching. The matched points are now transformed and associated to an additive map expressed in a fixed world frame. The refined pose is published as an odometry after mapping.

E. Pronto+LOAM

As mentioned earlier, in this paper we present a novel combination of the kinematic-inertial Pronto with LOAM. We present the results based on the experiments on the WALK-MAN robot. The robot has all the necessary configurations needed to use the Pronto algorithm. The open source version of the Pronto algorithm presented in [6], was adapted to a ROS based implementation. The LOAM algorithm was already implemented in ROS based packages. Hence, very little modification was necessary.

The IMU on the WALK-MAN robot provided data at 700Hz, while joint kinematics was obtained in the form of joint frame transforms at about 1000Hz. The overall estimate publishing rate of the Pronto module alone on the WALK-MAN robot data was 700Hz. In the original Pronto algorithm, LiDAR based localization on a pre-created map was used to provide corrections for the position state x_w.

Instead of using a pre-created map and a particle filter-based localization, we used the odometry-after-mapping estimates obtained from the LOAM module, to provide corrections to the x_w predicted state. We made a subtle two-way connection between the Pronto and the LOAM modules, which is the main contribution of the presented work. LOAM by itself was considered a superior algorithm owing to its high accuracy in benchmark tests. To the best of the authors' knowledge combining LOAM with an additional filter is new. The shortcomings of the individual algorithms for base pose estimation for a humanoid robot provided ample motivation to connect the LOAM estimation to the EKF. In this way, the drift is reduced, while the very high frequency estimates from kinematic-inertial estimation are not lost. This combination of two established state-of-the-art methods to formulate Pronto+LOAM is novel for a humanoid robot as it does not need stop-and-go-mapping. It creates maps on-the-go and the accumulated errors and drift are very low. The modifications to create the aforementioned two-way connection will be explained briefly.

On the LOAM side, instead of using the LiDAR attached IMU to transform the laser scan points to the laser frame, the final corrected estimates of position and orientation from the Pronto module were used to transform points to the world frame Σ_w. The estimates from Pronto also act as initial points for the minimization optimization for the odometry and mapping modules.

The LOAM module finally publishes the pose of the LiDAR with respect to the world frame Σ_w. Let the transformation matrix representing this pose be $\mathbf{H}_w^{l\prime}$. The pose/transform of the base frame Σ_b with respect to the laser frame Σ_l, is obtained from joint kinematics as $\mathbf{H}_l^b$. We then obtain the pose of the base as estimated by the LOAM module as $\mathbf{H}_w^{b\prime} = \mathbf{H}_w^{l\prime} \times \mathbf{H}_l^b$. We use the positions from $\mathbf{H}_w^{b\prime}$ directly, to provide corrections to position state x_w in the EKF of of the Pronto module.

A block diagram of the flow of information between the modules can be seen in Fig. 2. A summary of the roles of the modules and sensor data in the EKF can be found in the Table I

III. Experimental Evaluation

In this section we will detail the experimental evaluation of the Pronto+LOAM state estimator. The feet of the WALK-MAN robot may slip on contact with the ground due to the low grip and foot impact during walking at 1 step/s.

TABLE I

ROLES OF THE SENSOR DATA FUSION COMPONENTS

Sensor/ Module	Position	Linear Velocity	Angular Velocity	Orientation
IMU	Prediction	Prediction	Direct Measure-ment	Correction Update
Joint Kinemat-ics+Foot Contact	–	Correction Update	–	–
LiDAR/ LOAM	Correction Update	–	–	–

The algorithms presented in [18], [19] were used to perform the walking experiments on the WALK-MAN robot. The XBotCore platform on the robot [20] was the main communication interface used to obtain all joint states and sensor data synchronized and in ROS compatible formats. We used two experiments to demonstrate that the state estimator performs accurately for different gaits at higher walking speeds than the robots presented in the literature like the Valkyrie in [10] and the Atlas robot in [6].

A. *Experimental Setup*

Throughout the experiments, we set the robot step time to 1 s meaning that every step of walking takes 1 s to execute. In the first experiment, we let the robot walk with a step size of 0.05 m in the following sequence: walk forward, slightly turn, and walk backwards. This sequence is repeated multiple times and its duration is 699 s. In the second experiment, we let the robot walk with a step size of 0.10 m. The robot starts by turning on the spot multiple times and the experiment ends with the robot walking backwards. This experiment's duration is 180 s. The Fig. 3 shows the robot during the experiment after some motion, the body and the world frames and the corresponding RVIZ visualization during the experiment. In the rest of the paper, we notate the first experiment as *Exp-5* and the second one as *Exp-10*, based on the step size (in cm) of the walking motion, for the purposes of brevity. We compare three estimations: 1) the kinematic-inertial Pronto-only, 2) LOAM-only, and finally 3) the combined estimator of Pronto+LOAM.

B. *Analysis of Results*

The results of the robot base position (X,Y,Z) tracking for the two experiments can be seen in Fig. 4 and Fig. 6. As expected in both experiments, the estimation of the kinematic-inertial Pronto-only algorithm, drifted from the ground truth to a great extent. Fig. 5 and Fig. 7 show the translation error magnitude comparison of the three estimators. As can be seen from the illustrated magnitudes, the error of the Pronto+LOAM estimation initially rises up, but is eventually brought down by the corrections from the LOAM algorithm, thereby keeping the error magnitude low and bounded, unlike the kinematic-inertial Pronto-only estimation.

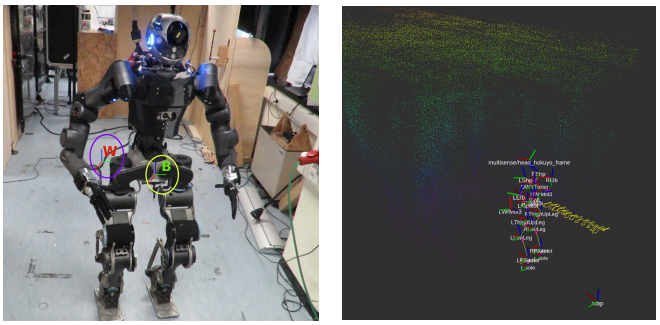

Fig. 3. Left: the WALK-MAN robot with the two 3D frames, labelled as W (world) and B (body). Right: the visualization of robot frames (right) inside the generated point cloud map, during the experiment. The estimated (Pronto+LOAM) robot pose and the joint frames are visualized, with the yellow curvy line to visualize the estimated robot base frame path.

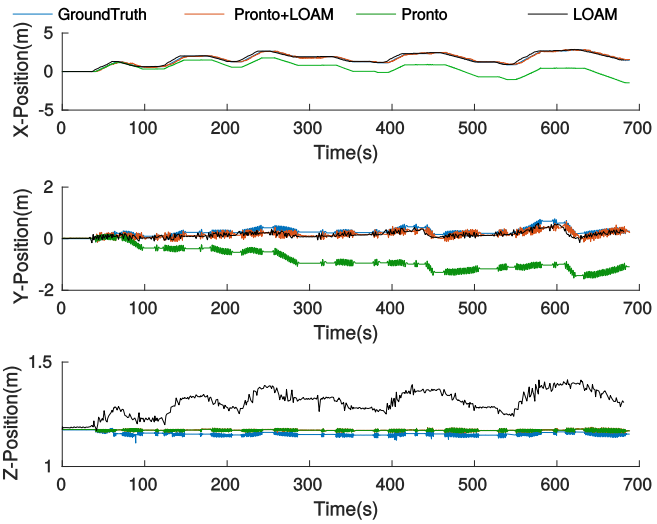

Fig. 4. 3D translation tracking comparison of Pronto-only, LOAM-only, and Pronto+LOAM methods, with respect to the ground truth for *Exp-5*. From the first two subplots the drift in the Pronto-Only estimation can be clearly observed. Whereas, the LOAM and Pronto+LOAM have low drifts. From the second subplot, it can be clearly seen that the black line fails to follow all the motion curves in the Y-tracking. This is due to the low-frequency estimation of the LOAM algorithm

The subtle difference in the two experiments can be illustrated by the fact that the drift in position of the Pronto estimation of the WALK-MAN base pose for *Exp-5* is 0.28cm/s of walking motion, whereas for *Exp-10*, it is around 0.7cm/s. In the original Pronto kinematic-inertial only algorithm, low drifts of 2 cm per ten steps were reported for the Atlas robot. In our case, a similar calculation leads to 2.8cm/10 steps for *Exp-5* and and 7 cm/10 steps for *Exp-10*. The step sizes used in study [6] were longer (15 cm and 36 cm steps), but smaller drift rate was observed. This difference could be attributed to the three key factors namely, difference in the systems of the robots used for the experiments, slight differences in adapting the Pronto kinematic-inertial only algorithm for our WALK-MAN robot and finally the walking controller and gait used for locomotion experiments. The difference in the robotic systems, the controllers and gait generators, can lead to differences in the walking style and trajectory of impact

TABLE II

ROOT MEAN SQUARED VALUES FOR X,Y,Z POSITION ERRORS(M) AND YAW(DEGREES), AND FINAL DRIFT(M)

Quantity	*Exp-10*			*Exp-5*		
	Pronto+LOAM	Pronto	LOAM	Pronto+LOAM	Pronto	LOAM
X-Position	0.0855	0.3642	0.0807	0.0634	1.4304	0.2092
Y-Position	0.1215	0.4126	0.1526	0.1282	1.130	0.1423
Z-Position	0.0084	0.0080	0.0578	0.0191	0.0191	0.1597
Yaw	6.4723	11.3931	17.0841	-	-	-
Final Drift	0.1573	0.7916	0.2441	0.0956	3.2388	0.3914

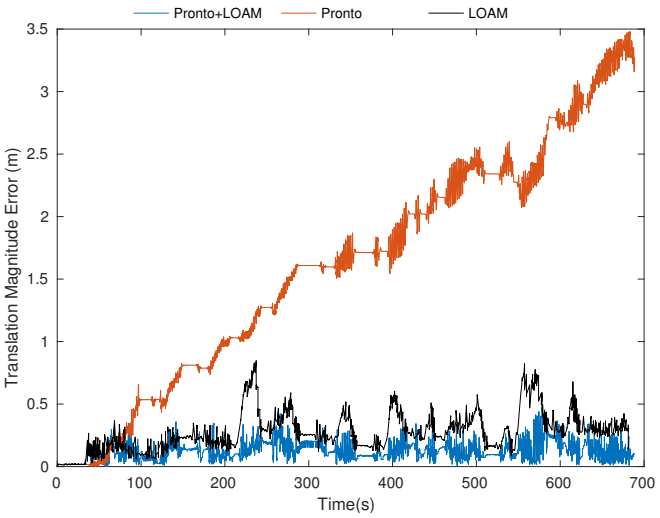

Fig. 5. 3D translation magnitude error comparison of Pronto-only, LOAM-only, and Pronto+LOAM methods for *Exp-5*. It can be observed that the for the majority of the time, the error in estimation by Pronto+LOAM(blue line) is lower than that of LOAM only(black line) estimation.

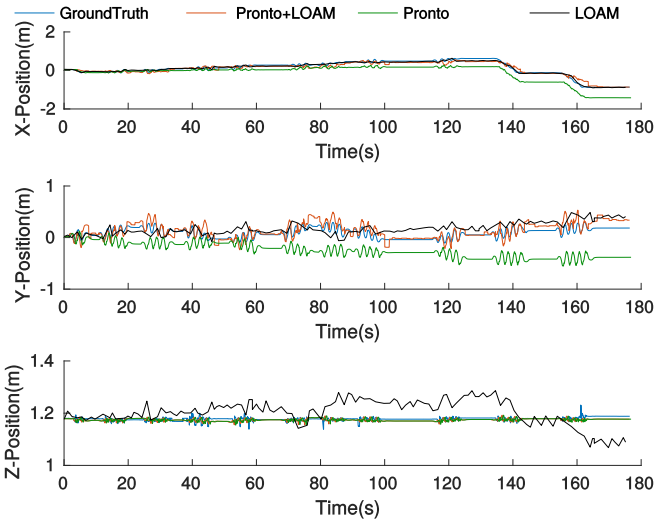

Fig. 6. 3D translation tracking comparison of Pronto-only, LOAM-only, and Pronto+LOAM methods with respect to the ground truth for *Exp-10*.

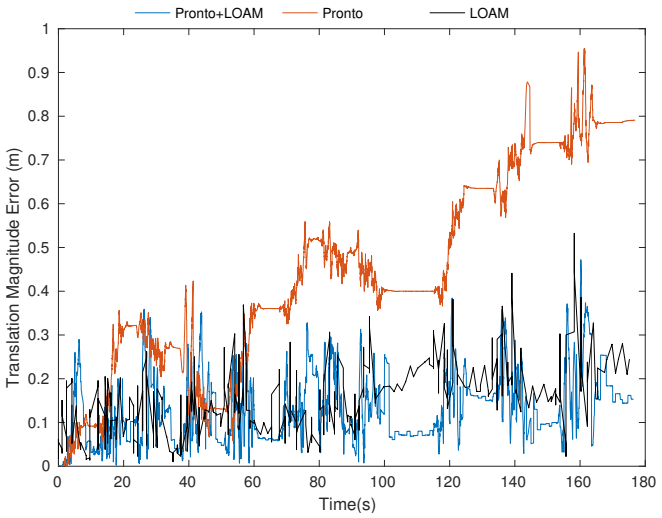

Fig. 7. 3D translation magnitude error comparison of Pronto-only, LOAM-only, and Pronto+LOAM methods for *Exp-10*.

with the ground while locomoting. Along with difference in walking speeds, the aforementioned factors may have caused more slippage during the experiments with the WALK-MAN robot. As mentioned earlier, the robot foot slips on contact with the ground. The slips were visually more observable in *Exp-10* case. Even though in *Exp-5* more steps are taken to cover the same distance, the rate of drift observed in the Pronto estimation is lower. We attribute this to the fact that higher speed of walking causes more slippage, which cannot be determined by the kinematic-inertial estimator. Hence, higher drift rate is observed in *Exp-10*.

Different step sizes were used in the two experiments, and the duration of the *Exp-5* was longer than that of *Exp-10* by ~ 8 minutes. In spite of these difference in speeds, from the translation error magnitude illustrated in Figs. 5 and 7, we can observe that, for our experiments, the Pronto+LOAM estimator reduces the rate of drift accumulation to a great extent. Furthermore, we can also observe that, for both the presented experiments, the Pronto+LOAM estimator always brings back the instantaneous translation error magnitude to a value well below 0.2 m. The similar improvement in performance and reduction in drift by the Pronto+LOAM estimation when compared with the original algorithms, for the two different experiments, proves that the combined estimator is suitable for varying gaits and speeds of locomotion.

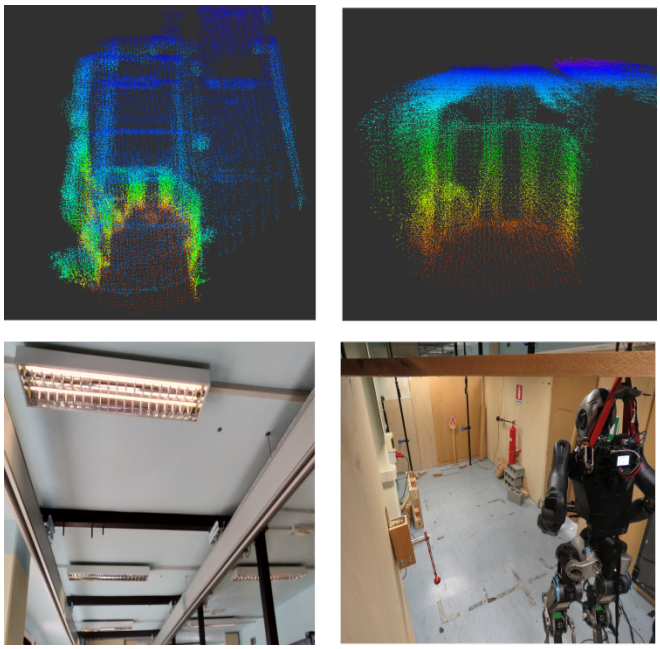

Fig. 8. Two views of the map created by the Pronto+LOAM estimation during *Exp-5*. The top two images are the point could maps of the locations in the lab represented in the corresponding images below them.

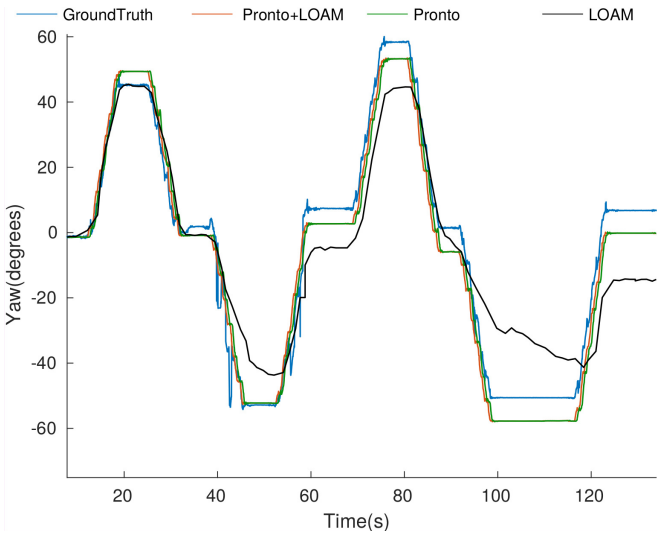

Fig. 9. Yaw tracking of the Pronto+LOAM, Pronto-only, and LOAM-only estimations, compared with the ground truth. The Pronto+LOAM and Pronto-only yaw estimations (yellow and red lines) are almost identical.

Furthermore, we obtained comparisons with ground truth for the base-yaw with respect to the world frame for the *Exp-10* experiment, in which multiple turning motions were executed by the robot. The results can be seen in Fig. 9. It can be observed that the LOAM-only algorithm performs worse than the Pronto-only and Pronto+LOAM algorithm when it comes to yaw tracking. This observation provided sufficient motivation to not use yaw estimates from the LOAM algorithm for the EKF corrections. As both Pronto-only and Pronto-LOAM rely only on IMU data for yaw estimation, they have identical results. The low accuracy tracking of yaw by LOAM module for the particular case of our humanoid robot will be investigated and improved upon in future works.

We present the Root Mean Squared (RMS) errors comparison for the positions and yaw in Table II. We also present the final positional drift at the end of the experiment in Table II. The light blue coloured cells indicate best performances in *Exp-10* while the pink coloured cells indicate the best performances in *Exp-5*. From the various plots and the Table II it can clearly be seen that, in terms of errors, the combination of Pronto+LOAM performs much better in almost all cases when compared with the original state-of-the art algorithms. As LOAM publishes estimates at lower frequencies ($\leq$ 1Hz), its estimates are not useful for dynamic control and walking, which is one the major areas of future applications of this work. The low frequency estimates lead to higher errors for the LOAM-only estimation and also causes the algorithm to miss many motions performed between two consecutive estimates. Whereas, the Pronto-only algorithm faces the problem of accumulating the drift. Although when we combine both the Pronto and LOAM, we are able to take advantage of the high frequency estimates of Pronto and correct for the drift using LOAM. The benefits of the Pronto-LOAM can be easily seen not only in the RMS values but also in the final position drifts of the estimations. Furthermore, it is to be noted that the Pronto+LOAM estimation had a very low drift of only 9.56 cm for a $\sim$ 11 minute long experiment with an approximate walking path length of $\sim$ 10.7m. This experiment did not have just monotonous one direction straight walk but back and forth walking with small turns. This provides sufficient proof of the robustness and precision of the Pronto+LOAM algorithm for long experiments.

The experimental data was collected in the form of "rosbag" from the robot and they were run on a laptop with a 2.50 GHz 4 core i7-6500U CPU and 16 GB RAM. The average memory used when the Pronto only estimation was being executed was 54 MiB. The combined estimation of Pronto+LOAM used up to 96 MiB of memory, while the LOAM-only estimation used only 48 MiB in average. It is to be noted that, the complete Pronto+LOAM estimation used 3 CPUs, and the average usage percentages of each of the CPUs were approximately 93%, 74% and 34%, respectively. The LOAM-only estimation published at an average rate of 0.690 Hz. The maximum time without a new published estimate was 5.06s. This was slightly improved in the Pronto+LOAM estimator, where the LOAM module published estimates at 0.705 Hz, with the maximum time period without a published estimate being 3.57 s. The low memory usage of the Pronto+LOAM estimation makes it suitable for on-board devices on the robot, leaving open the problem of achieving high publishing frequency of the estimates on an on-board computer with lower number of processors.

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel combination of two state-of-the-art algorithms to perform state estimation and mapping for the humanoid robot WALK-MAN. The Pronto and LOAM algorithms were combined with a subtle two-way connection and the robustness and accuracy of the combined global state estimation algorithm was tested and compared with the original algorithms on real experimental data obtained from WALK-MAN. We were able to prove that it is possible to consistently achieve low drifts, low error and high frequency base pose estimation. This is achieved while doing on-the-go state estimation and mapping for the humanoid robot WALK-MAN, by combining the aforementioned algorithms. Similar performance improvement from the original algorithms was seen in two separate experiments, where the step size differed (5cm and 10cm), proving its robustness to varying speeds. We also presented the computational power consumption analysis for the combined estimator. The results of this analysis are encouraging and in the future we look to achieving similar high frequency and high accuracy estimation, when the algorithm is ported to the on-board computer of the robot. We intend to use this framework in future work, where the robot will be capable of traversing unknown terrain without the need for repeatable features, or stop-and-go-mapping algorithms. Future work will also involve testing the combined Pronto+LOAM estimator with the robot walking on rough and non-flat surfaces. Furthermore, the Pronto+LOAM estimates will be used for dynamic navigation planning on non-flat and irregular surfaces as well as for dynamic control, while walking on irregular surfaces [21], using also our newly developed real-time dense surface mapping and tracking system [22].

ACKNOWLEDGMENT

This work is supported by the CogIMon (grant agreements no 644727) EU projects.

REFERENCES

[1] S. Piperakis, M. Koskinopoulou, and P. Trahanias, "Non-Linear State Estimation for Humanoid Robot Walking," in *IEEE Robotics and Automation Letters (RA-L)*, June 2018.

[2] T. Flayols, A. D. Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse, "Experimental Evaluation of Simple Estimators for Humanoid Robots," in *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 889–895.

[3] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State Estimation for Legged Robots-Consistent Fusion of Leg Kinematics and IMU," *Robotics*, vol. 17, pp. 17–24, 2013.

[4] X. Xinjilefu, S. Feng, W. Huang, and C. G. Atkeson, "Decoupled State Estimation for Humanoids using Full-Body Dynamics," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 195–201.

[5] X. Xinjilefu, S. Feng, and C. G. Atkeson, "Center of Mass Estimator for Humanoids and its Application in Modelling Error Compensation, Fall Detection and Prevention," in *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 67–73.

[6] M. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-Free Humanoid State Estimation Fusing Kinematic, Inertial and Lidar Sensing," in *14th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2014, pp. 112–119.

[7] P. J. Besl and N. D. McKay, "Method for Registration of 3-D Shapes," in *Sensor Fusion IV: Control Paradigms and Data Structures*, vol. 1611. International Society for Optics and Photonics, 1992, pp. 586–607.

[8] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-World Data Sets," *Autonomous Robots*, vol. 34, no. 3, pp. 133–148, 2013.

[9] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse Iterative Closest Point," in *Computer graphics forum*, vol. 32, no. 5. Wiley Online Library, 2013, pp. 113–123.

[10] S. Nobili, R. Scona, M. Caravagna, and M. Fallon, "Overlap-based ICP Tuning for Robust Localization of a Humanoid Robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4721–4728.

[11] G. Oriolo, A. Paolillo, L. Rosa, and M. Vendittelli, "Vision-based Odometric Localization for Humanoids using a Kinematic EKF," in *12th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2012, pp. 153–158.

[12] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, vol. 2, 2014.

[13] ——, "Visual-lidar odometry and mapping: Low-drift, robust, and fast," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2174–2181.

[14] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] N. G. Tsagarakis, D. G. Caldwell, F. Negrello, W. Choi, L. Baccelliere, V. Loc, J. Noorden, L. Muratore, A. Margan, A. Cardellino, *et al.*, "WALK-MAN: A High-Performance Humanoid Platform for Realistic Environments," *Journal of Field Robotics*, vol. 34, no. 7, pp. 1225–1259, 2017.

[16] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, "State Estimation for a Humanoid Robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, 2014, pp. 952–958.

[17] O. H. Schmitt, "A Thermionic Trigger," *Journal of Scientific Instruments*, vol. 15, no. 1, p. 24, 1938.

[18] C. Zhou, X. Wang, Z. Li, and N. Tsagarakis, "Overview of Gait Synthesis for the Humanoid COMAN," *Journal of Bionic Engineering*, vol. 14, no. 1, pp. 15–25, 2017.

[19] C. Zhou, Z. Li, X. Wang, N. Tsagarakis, and D. Caldwell, "Stabilization of Bipedal Walking Based on Compliance Control," *Autonomous Robots*, vol. 40, pp. 1041–1057, 2016.

[20] L. Muratore, A. Laurenzi, E. M. Hoffman, A. Rocchi, D. G. Caldwell, and N. G. Tsagarakis, "Xbotcore: A real-time cross-robot software platform," in *IEEE International Conference on Robotic Computing (IRC)*. IEEE, 2017, pp. 77–80.

[21] D. Kanoulas, C. Zhou, A. Nguyen, G. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Vision-based Foothold Contact Reasoning using Curved Surface Patches," in *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017, pp. 121–128.

[22] D. Kanoulas, N. G. Tsagarakis, and M. Vona, "rxKinFu: Moving Volume KinectFusion for 3D Perception and Robotics," in *IEEE-RAS 18th International Conference on Humanoid Robotics (Humanoids)*, 2018.