

Multi-Sensors State Estimation Fusion for Quadruped Robot Locomotion

Chen Yao, Zhenzhong Jia, Shunyao Wang, Yufeng Xu, Xiangru Mou, Peng Kang and Wenfu Xu*

Abstract—Quadruped robots have the ability to walk over obstacles, of particular interest is in their autonomy to move through harsh and unknown conditions that humans cannot enter with certain sensors. However, intrinsic random errors or external impact vibration will inevitably influence these sensors, lowering the accuracy of the state parameter calculation. In this paper, we present an effective 3D state estimation algorithm that incorporates input from several sensors (Inertial measurement unit, joints encoder, camera and LiDAR) to allow our quadruped robot StartDog to balance varying frequency and accuracy for locomotion. The IMU-centric leg odometry that merges inertial and kinematic data considers the foot contact state and lateral slip as a further method to improve the accuracy of the robot's base-link position and velocity. In order to use LiDAR registration to align point clouds, we preprocess the data and obtain initial location information on the map. Further, a modular filtering method is presented, which can combine leg odometry based on the inertial kinematics state estimator of quadruped robots and LiDAR-based localization methods. In the experiments at the indoor building, the motion capture system is used to evaluate the performance of this algorithm. The results show that the quadruped robot system is capable of low error rates and a high frequency of state estimation.

Index Terms—quadruped robot, state estimation, modular filter, multi-sensors fusion

I. INTRODUCTION

QUADRUPED robot is now widely used in exploration, rescue, transportation, etc [10]. Our purpose is to develop an autonomous quadruped robot system with rescue assistance that can successfully cross over obstacles and explore harsh unknown terrain in a catastrophe situation with many equipment sensors. However, intrinsic sensors have the inherent random errors and drift of sensors, different pulse robot's gaits that will lead to shock and vibration, low-frequency sensors since the high frequency is required for a control system, all of these factors makes the accurate and reliable state parameter calculation a key challenge problem. So our focus is on the accurate and reliable 3D state, which can facilitate the control systems in the feedback by regulating the parameters, planning the trajectories, optimizing the controller, and adjusting later slip. To do the work autonomously and efficiently in a certain project, this work must provide correct information from multi-sensors to the 3D estimate algorithm and sense the surroundings to create an environmental map.

*Corresponding author. Chen Yao and Zhenzhong Jia are with the Southern University of Science and Technology (SUSTech), Institute of Robotics, Shenzhen, 518172, GuangDong, China. Shunyao Wang, Yufeng Xu, Xiangru Mou, Peng Kang and Wenfu Xu are with the school of mechanical engineering and automation, Harbin Institute of Technology, Shenzhen, 518172, GuangDong, China. Corresponds to yaoc@mail.sustech.edu.cn, jiaazz@sustech.edu.cn, wfxu@hit.edu.cn.

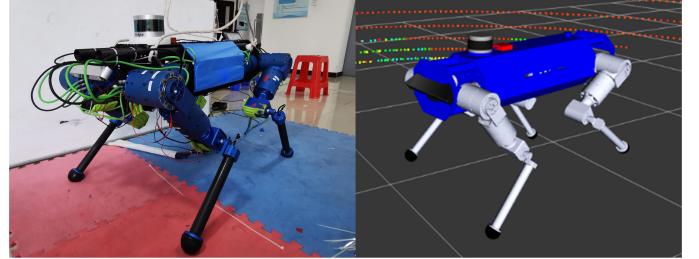


Fig. 1: *Left*: The quadruped robot with 12 degrees of freedom in the test environment and equipped with different sensors. *Right*: StartDog Robot platform shown in RVIZ and can sense the surroundings by LiDAR sensor.

In order to provide a module state estimator for quadruped robot locomotion, we use a variety of sensors, including inertial, kinematic, vision, and LiDAR measurements. To the best of our knowledge, this is the first paper to combine leg odometry and LiDAR localization on a quadruped robot by preprocessing a variety of sensors to enhance performance. The main contributions of this paper are as follows:

- A novel calibration method for camera and robot base-link mainly based on laser tracker is proposed in this paper. Laser tracker can provide high-precision data and is not affected by sight occlusion. Also, the robot does not need to move during the operation.
- An accurate method of detecting foot contact is presented, which focuses mainly on detecting early or late contact. The robot control system will take into account lateral drift or slide to adjust velocity, so our leg odometry can be obtained via reliable contact detection based on EKF.
- A better way of transforming the movement of quadruped robots is achieved with the preprocessing of LiDAR data and initial localization information that matching/aligning point clouds are effective.
- We demonstrate experimentally that the pairing of LiDAR odometry with leg odometry provides a low error rate and high frequency of state estimation in locomotion. Multi-sensors fusion can combine each sensors' advantage to output satisfactory performance.

The structure of this article as follows: Sec. II reveals the most up-to-date state estimate studies. The quadruped robot platform and the sensors' scheme are described in Sec. III. A discussion of the various sensor processing methods and modules is included in Sec. IV. The experimental evaluation is present in Sec. V to demonstrate the performance, and a conclusion is shown in Sec. VI with a plan for the future.

II. RELATED WORK

For the past few years, a quite lot of remarkable work has been shown and these quadruped robots reveal excellent mobility, such as Boston Dynamics BigDog [17] and LS3 [13] robot, MIT cheetah3 [5], ETH ANYmal [1], IIT HYQ [18], and a few more useful applications in complex environments are present. State estimation has been implemented on quadruped robots for a series of significant literature.

Proprioceptive sensors for state estimation mainly focus on the estimator. Notably, ETH [6] proposed a method of base full state estimation based on the fusion of leg encoder and IMU data through an error EKF method[20], joint kinematics was used to correct the state update, and the objectivity of the filter was also analyzed. Cheetah3 [5] focuses mainly on control and planning, but they present a robust estimator through a two-stage sensor fusion that decouples orientation estimation from the estimation of position and velocity. In these methods, state estimation only relies on proprioceptive sensors, and these sensors result in a cumulative increase of errors and drift, as well as producing inaccurate estimation for the locomotion. Typically, using only these proprioceptive sensors for estimation cannot be used when considering the complexities of the task or the environment, so exteroceptive sensors are introduced to mitigate this issue.

Exteroceptive sensors are always utilized in Simultaneous Localization And Mapping (SLAM) for state estimation [7]. Specifically, camera and visual odometry are not employed in our suggested state estimation since they fail under dynamic vibration and variable light circumstances. Instead, LiDAR has become popular and can detect objects accurately over a great distance. The Iterative Closest Point (ICP) method [2] is one of the most popular techniques for point cloud registration, although it requires enough overlap between two point clouds. To get the LiDAR registration, the Normal Distributions Transform (NDT) [3] compares the probability between point clouds and the constructed map. The LiDAR odometry and Mapping (LOAM [24]) approach expose excellent and effective performance and the two-way method can robust obtains the odometry and mapping. Lightweight and Ground-Optimized LiDAR Odometry and Mapping (LeGO-LOAM [19]) developed LOAM with loop closure and a two-stage optimization and receive a more robust output.

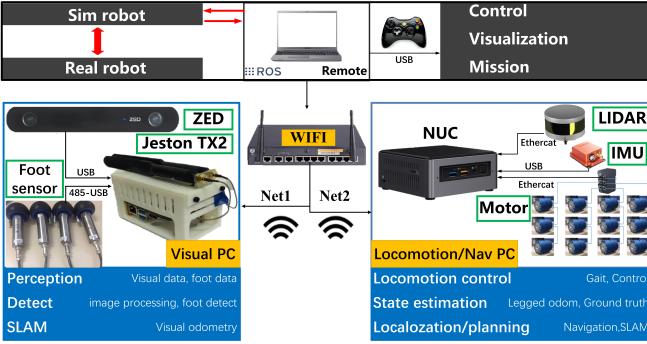


Fig. 2: The hardware and software framework of StartDog. There list all the specifications sensors of the system, also the communication can send and receive through a network.

Fusing with different sensors can reliably improve the performance of state estimation. Chilian *et al.* in [9] mainly uses inertial, leg odometry and visual odometry on a slow crawl six-legged robot. In Fallon's paper[11], inertial, kinematics, LiDAR, and stereo vision are probabilistically fused to produce a single, consistent EKF based estimate named Pronto for the Atlas robot, also a particle filter is used for localization, this estimator can achieve 2cm per 10 steps drift in the quasi-static experiment. Pronto is also implemented for HYQ and WALK-MAN [21], and a combination of the Prone Kinematics-inertial state estimator and LOAM based on the full-size humanoid robot WALK-MAN to perform low drift, high-frequency estimation. Nobili *et al.* in [14],[15] combine inertial, kinematic, stereo vision and LiDAR measurements to produce a modular inertial-driven state estimator which can be directly used to controller. they still compare the performance of MEMS IMU and fiber optic IMU. Based on the recently proposed COCLO (Contact-Centric Leg Odometry) method [23], this new estimator used a Square Root Unscented Kalman Filter to fuse multiple proprioceptive sensors, and this contact-centric approach estimate velocity more accurately. Factor graph optimization [22] method is presented to tightly fuses and smooths inertial, leg odometry, and visual odometry. As well as reduce the dependency on foot contact classifications.

Taking cues from these outstanding works, our approach offers a feasible fusion style of loosely-coupled EKF and provides robust state estimation for our quadruped robots StartDog with multi-sensors.

III. SYSTEM OVERVIEW

In this chapter, we develop a robust multi-sensor fusion system framework to meet our requirements, which include sensor scheme design, notation, calibration, and synchronization.

A. System architecture

The StartDog is a modular electric quadruped robot that supports torque, position, and speed control. Fig 1 shows the view of our StartDog robot. Our platform weighs about 40kg and its overall size is about $1.0m \times 0.27m \times 0.6m$. At the same time, each leg has three degrees of freedom, allowing it to achieve a wide motion space and show high mobility, in particular, the knee joints can turn 360 degrees.

The hardware and software framework of our StartDog are analyzed specifically here and as shown in Fig 2. The framework we have developed is divided into two categories: remote-controlled and robot-controlled to facilitate the high-frequency realization of robotic autonomy. The remote communicates with the robot through WiFi, we can observe the simulation model while using the Xbox to send the order to the real robot. The remote communicates with the robot through WiFi, so we can watch the actual/simulation robot while using the Xbox to send/receive the command to the robot. Through ROS, the onboard computer of the NUC and Nvidia Jetson TX2 communicate in a local network. TX2 acquires data from the zed camera and foot sensors to perceive and identify the environment, etc. On the NUC computer, the data of IMU, LiDAR, and joint encoders are mainly processed and used for locomotion control, state estimation, etc.

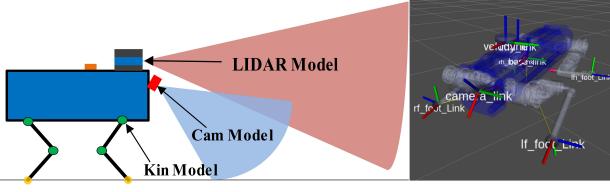


Fig. 3: Sensors configuration for StartDog. *Left:* Sensors installed position and the fields of sensors' view. *Right:* the sensors coordinate system as shown in RVIZ.

B. Sensor Scheme Design

Sensors should be considered that provide accurate information about the quadruped robot, such as proprioceptive (encoders, IMUs, and force sensors), and exteroceptive (camera and LiDAR). Each of the 12 leg joints of the quadruped robot is equipped with an absolute encoder (RLS encoder), which connects through EtherCAT and has a high frequency of 400Hz and a high level of resolution. Based on cost and practicality, our robot adopts the Xsens MTi-G-710 (IMU) to determine acceleration and angular velocity from the base link and is set in the middle place. The air pressure sensor allows detection of foot contact, as well as shock absorption and anti-skidding performance, so these four sensors are directly installed at the end of the leg. Using the VLP-16 LiDAR sensor can provide high precision distance point clouds that are vital for localization. However, it only operates at a frequency of 10Hz, despite the capability to generate 30,000 points per second from the upper deck place's scan. To plan the foothold and the near terrain, the Zed camera is placed on the front and connects with the robot. For improved performance and the fixed integration of the sensors system, the sensor coordinate system configuration is set here and as shown in Fig. 3.

C. Sensor Notations

In this paper, the mathematical notations are used as follow: The matrix will be displayed in bold type, while the coordinate frames will be displayed in lower-case normal italic. For example, ${}^w_b\mathbf{T}$ can represents as a homogeneous transformation of base link frame b corresponding to world frame w . The absolute orientation is represented as a quaternion q_w .

The direction selection of the coordinate system uses the right-hand criterion. We set a coordinate system of the robot base link at the center of mass. Hence, the z-axis is directed upward and perpendicular to the upper plane, also the x-axis is directed forward when relative to the robot framework. The world frame is represented as ${}^w\mathbf{T}$. ${}^L_i\mathbf{D}\mathbf{A}\mathbf{T}$ and ${}^c\mathbf{am}\mathbf{T}$ are the representation of the LiDAR and camera frame .

D. Sensor Calibration

The measurement noise of each sensor is inevitable, and the accumulated error is tough to be completely corrected. In spite of the fact that the transformation between coordinate systems can be easily done with a CAD model and installation design, it is still not possible to obtain a relatively accurate sensor relationship due to factors such as assembly tolerances

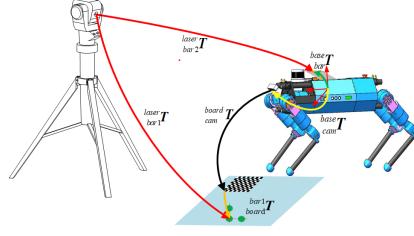


Fig. 4: Calibration method between the camera and robot base-link coordinates.

of each part, properties of materials, motion impact, etc. For the input raw data to be reliable, the appropriate sensors calibration method deserves serious consideration for intrinsic and extrinsic parameters of each type of sensor.

Through a camera, we can obtain precise environmental information and convert it to a global world coordinate system for deep analysis. Thus, the most important thing to know is the coordinate between the camera and robot base link as in [8]. The difference of our calibration method is the use of a *Radian* laser tracker, a camera and a designed calibration board in order to reduce the influence of noise and errors; also, the robot does not need to move and high-accurate coordinate relationships can be achieved as shown in Fig. 4 and Fig. 5.

On the sophisticated designed calibration board, target balls can be placed randomly so that a relative pose relationship can be observed between the corner of the calibration box on the bottom left and these target balls as shown in Fig. 5. In detail, using the laser tracker, the coordinate data of three targets at different postures can be accurately measured, then the three-point criterion is used to identify the direction. By matching the images input to the left camera, the PNP determines the camera and calibration board conversion coordinates. The final transformation of the pose relationship is obtained by iteratively calculating the images at different postures:

$${}^c\mathbf{am}\mathbf{T} = {}^b\mathbf{bar2}\mathbf{T} \cdot {}^laser\mathbf{bar2}\mathbf{T} \cdot {}^laser\mathbf{bar1}\mathbf{T} \cdot {}^b\mathbf{bar1}\mathbf{T} \cdot {}^b\mathbf{board}\mathbf{T} \quad (1)$$

This method only requires moving the calibration board and recording data from the laser tracker and camera, which can reduce the error caused by robot motion, and does not require complex data processing. In addition, we think that the laser tracker in combination with the designed calibration board is likely to deliver superior accuracy and robustness over the motion capture system, so we can be confident that reliability and robustness will be ensured.

E. Sensor Synchronization

Due to the large difference in frequency, bandwidth, and interface signals, data processing mainly involves sensors synchronization. The focus is on synchronizing foot sensors, IMU, and encoders at the same time, which can provide effective state estimation data. Software synchronization is done using the time synchronization that comes to form ROS (Robot Operating System). By using this method, we are able to achieve synchronous message output from multiple sources at the same time, also it can be more flexible and inexpensive than hardware synchronization using FPGA chips.

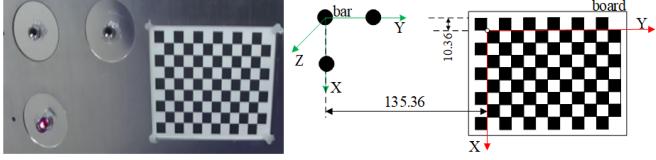


Fig. 5: *Left*: The target balls and the calibration board, we record and replace balls' postures randomly; *Right*: The corresponding specifications of the coordinate between the target balls and the corner of the calibration box on the bottom left.

IV. APPROACH

The quadruped robot's autonomous system requires low-latency and accurate estimation of state elements, especially the accuracy of velocity estimation in a controller; therefore, care should be taken to minimize drift error and high frequency. Our approach can realize a state estimator only based on the measurement of proprioceptive sensors of IMU and kinematic with a higher frequency. Additional external sensors, such as LiDAR and cameras, can also enhance state measurement accuracy and ensure redundancy, although their frequency can be low. Therefore, it is necessary to verify the trade-off between frequency and accuracy and design a complementary frame that combines these sensors.

A. Leg Odometry

This section presents our inertial-kinematic estimator that is based on [6]. State estimation elements are defined as follows for the robot's base link:

$$\mathcal{X} = [{}^w P, {}^w V, {}^w \theta, b_a, b_\omega]$$

The 15 state elements (3D position ${}^w P$, velocity ${}^w V$ and orientation ${}^w \theta$) are all expressed in the world coordinate. The velocity can also be expressed in the base link coordinate (when robot not move and the foot velocity is zero, velocity ${}^w V$ equivalent to ${}^b V$). The orientation refers to Euler angles (quaternion can be transformed through exponential coordinate). At the same time, IMU can deal with online calibration of acceleration and angular velocity biases b_a and b_ω by EKF, so the acceleration and angular velocity can be corrected in each update loop.

1) IMU estimation: Using two IMU frames angular and acceleration velocity, we estimate the prediction state of the current pose by using the mean value theorem of integrals[16]:

$$\begin{aligned} {}_b C_I(t + \Delta t) &= {}_b C_I(t) \text{Exp}(w'_t(t) \Delta t) \\ {}_b V_I(t + \Delta t) &= {}_b V_I(t) + a'_t \Delta t \\ {}_b P_I(t + \Delta t) &= {}_b P_I(t) + {}_b V_I(t) \Delta t + 0.5 a'_t(t) \Delta t^2 \end{aligned} \quad (2)$$

where ${}^w V_I$ and ${}^w P_I$ are the IMU velocity and position, respectively, ${}_b C_I$ represents the homogeneous transformation matrix of attitude which can be gained by quaternion exponential operation. Δt means the timestamp from the IMU frequency. w'_t and a'_t will be taken as the average angular and linear velocity at time t while can get from the mean value at the time t and $t + \Delta t$.

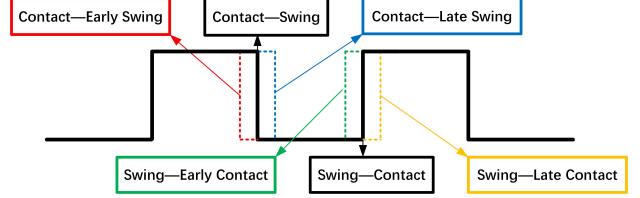


Fig. 6: contact state modes of the finite state machine (FSM).

2) Foot Estimation: By using AD sampling, the foot sensors determine the corresponding air pressure. As soon as the pressure value reaches the threshold value of the vertical component, the corresponding sensor state is identified. Considering these bad actual surface conditions, the movement always needs reliable contact with the ground. The force sensor-based method has too much noise, and the accuracy of the threshold cannot be assured. In addition, the gait phase is based on the ideal foot state. As early or late contact can also cause instability in walking, it is essential to detect the contact state more precisely by adding the contact event trigger.

Combining contact detection of the foot sensors and gait phase, a finite state machine (FSM) for each leg movement is defined in Fig. 6. We divide the contact state into 6 modes: *contact and early swing*, *contact and swing*, *contact and late swing*, *swing and early contact*, *swing, and contact*, *swing and late contact*. The black line indicates the planned gait phase and other models can be derived from the comparison of foot sensors and gait phase. When these data are combined, we can obtain accurate estimates of which foot is on the ground.

By selecting the trot gait, the robot can walk forward under the flat test ground. The swing and support times are both set to 0.3s in the actual experiment. Fig. 7 shows the relevant performance parameters from the left front leg (force, z-axis position, true contact state, etc.).

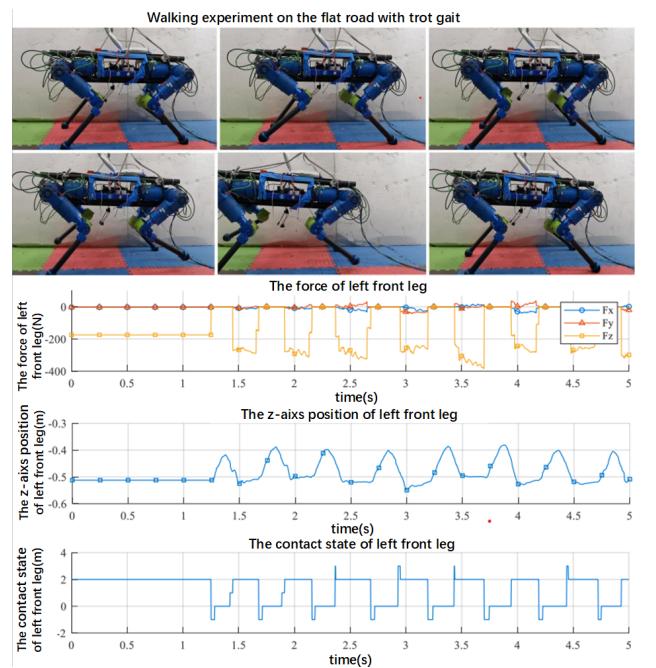


Fig. 7: Correlation state of left foreleg in trot gait.

3) Leg Kinematic: When the average of the combined legs which are in reliable contact with the fixed ground is taken, the robot base position can be computed using the angle of the joints. Based on the characteristics of quadruped robot locomotion, we propose the following leg kinematic methods that select the change amount between joint kinematics in a given frame to calculate the base position (see Alg. 1, Alg. 2).

With algorithm 1, the last frame is seen as the given frame in each timestamp, so we can get the change amount and transform the data from the base link coordinate to the world coordinate. By adding it to the initial position, the final position of the base link will be determined.

Algorithm 1 Kinematic estimation for each timestamp data

Input : joints data α , foot sensor data s_{fj} , plan gait p_{fj} , the orientation of IMU data Iw , timestamp i , leg j .

- 1: Set the initial startup position as ${}^w P_K$.
- 2: **if** $ContactDetect(s_{fj}, p_{fj}) = \text{true}$ **then**
- 3: **return** Leg contact $FC_j = 1$.
- 4: **end if**
- 5: Get the leg position change amount at each timestamp for $\Delta_{fi}^b P = Fkin(\alpha_{i+1}) - Fkin(\alpha_i)$
- 6: Combined the contact state and each leg change
- 7: Obtain the position change of the robot base link in each iteration $\Delta_{fi}^b P = - \sum_{j=1}^4 FC_j \Delta_{fj}^b P$.
- 8: Transform the position in world frame at each timestamp $\Delta P_{K-i} = {}^w C \Delta_{fi}^b P = Exp(w\Delta t) \Delta_{fi}^b P$.
- 9: Compute the position of base link ${}^w P_K + = \Delta P_{K-i}$.
- 10: **return** ${}^w P_K$.

For algorithm 2, the initial frame of each leg connected to the ground can also be seen as the given frame in each timestamp. By setting each leg's switch as a conversion volume, the base link position can be updated.

Algorithm 2 Kinematic estimation for each leg switch data

Input : joints data α , foot sensor data s_{fj} , plan gait p_{fj} , the orientation of IMU data Iw , timestamp i , leg j .

- 1: Set the initial startup position as ${}^w P_K$.
- 2: **if** $ContactDetect(s_{fj}, p_{fj}) = \text{true}$ **then**
- 3: **return** Leg contact $FC_j = 1$.
- 4: **return** switch joint angle $\alpha_{f0} = \alpha_i$
- 5: **end if**
- 6: Get the leg position change amount at each timestamp for $\Delta_{fi}^b P = Fkin(\alpha_{i+1}) - Fkin(\alpha_{f0})$
- 7: Obtain the position change of the robot base link in each iteration: $\Delta_{fi}^b P = - \sum_{j=1}^4 FC_j \Delta_{fj}^b P$.
- 8: Transform the position in world frame at each timestamp $\Delta P_{K-i} = {}^w C \Delta_{fi}^b P$.
- 9: **if** foot state of last frame is switch **then**
- 10: **return** conversion volume $f_K + = \Delta P_{K-i}$
- 11: **end if**
- 12: Get the position of base link ${}^w P_K = f_K + \Delta P_{K-i}$.
- 13: **return** ${}^w P_K$.

Alg. 1 mainly considers the change amounts between 2 nearest frames, so the error only exists within these 2 frames. During a gait cycle, both the new frame and the initial frame associated with the switched foot are used in Alg. 2, and the error will be the accumulation of both frames. In order to estimate the robot's base-link position, we used Alg. 1 since it provides low error and reliable accuracy.

By using the Jacobian method, the differential derivative calculation will increase the base link velocity error for movement impact, as well as the friction resistance, the installation clearance of the joints, and the sensor reading error. This paper proposes a technique to modify the base-link velocity: The linear velocity of robot base-link takes into account not only the position at various timestamps, but also the lateral slide in dynamic locomotion, so the accurate and smooth estimate forum as follows:

$${}^w V_K = ({}^w P_K(t) - {}^w P_K(t - nT)) / nT + \bar{w}_I R_l \quad (3)$$

Where R_l is the distance between the base link and the tip of the foot by the kinematics, T represents the interval of timestamp, we choose $n=15$ as the experiment parameter.

4) Leg Odometry: The leg odometry is always affected by the proprioceptive sensors since there is a calculation error as well as kinematic drift and noise in the IMU readings or integration. In order to improve the performance, we implemented the EKF framework, which takes advantage of the reliable sensors readings in order to estimate the robot's state, which was inspired by [20],[21]. The magnitude of these parameters often differs greatly and can cause divergence or failure, so the error value is used as a state variable to make sure the input state is relatively small, The error parameters are set as $[\delta P, \delta V, \delta \theta, \delta b_a, \delta b_\omega]$.

Using prediction and update model equations, a data fusion framework is constructed to produce a more accurate estimation. In this study, the global foot contact position is not reliable obtained for the moment controlled is considered instead of position controlled, and we cannot get this data from motion capture, therefore, the error of joint kinematics and IMU is used to update the state vector instead of the foot placement[6].

As shown in Fig. 8, the feedback correction method is selected to obtain the leg odometry. To compensate for the state, the feedback value calculated by the EKF will be directly fed back into the IMU prediction model. By using this method, the calculation result can be more accurate and the drift can be reduced further.

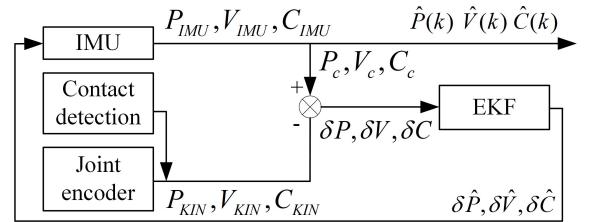


Fig. 8: Flow chart of the fusion method for Leg odometry, which combines the IMU, contact detection, and joint encoder data into the error EKF frame.

B. LiDAR-based Estimation

The leg odometry can be obtained just below its initial motion coordinate (the Odom coordinate system), but if there are no exteroceptive sensors to improve the performance, the result will be a severe drift over time with relatively large errors [6]. We do not use cameras and visual odometry to estimate state because they are ineffective in dynamic locomotion and low light environments. Here, the visual module only provides the initial localization and detection. LiDAR must be the focus of more efforts, and in the following, we define the LiDAR-based localization approach for the robot base-link state estimation according to a map frame.

1) Initial localization: Upon starting the robot in the real experiment, initial localization can determine the initial position of the robot's base link relative to the global map, and different ways can be used to obtain this transformation matrix. The global map is built through the mapping method with the pre-processing point clouds [19]. Additionally, during the initialization process of the simulation environment, the initial posture can be obtained directly by subscribing to the *model-state* topic. *Apriltag* is an effective method for ensuring the initial pose can be obtained from a camera reliably and stably both in simulation and indoor environments.

During the mapping initialization process, the image's position in the map coordinate system ${}_{cam}^{map}\mathbf{T}$ can be determined. Similarly, we can also know the posture of the image in the motion initialization process ${}_{odom}^{map}\mathbf{T}$. With different coordinate transforms, we can measure the relative initial posture information with Odom to map coordinate ${}_{odom}^{map}\mathbf{T}$, the corresponding coordinate are shown in Fig 9.

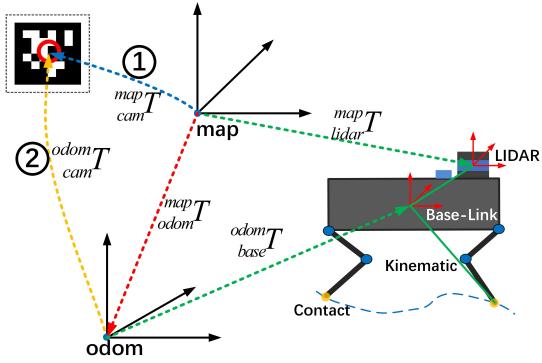


Fig. 9: The relationships of coordinate system.

From initial localization, we can also obtain the transformation ${}_{odom}^{map}\mathbf{T}$, and the transformation ${}_{base}^{odom}\mathbf{T}$ can directly get from leg odometry in the Odom coordinate system. Therefore, the information of leg odometry ${}_{base}^{map}\mathbf{T}$ is shown in the map coordinate system for later deal and comparison.

2) Point cloud preprocess: Laser point clouds always have distortions and cluttered points during the scanning process. Interferences in these systems will cause a too long matching time, weak matching, and need to be optimized through pre-filtering, which leverages the number of false alignments and ensures a balance between match efficiency and accuracy. The laser point cloud can be preprocessed by using the following three steps:

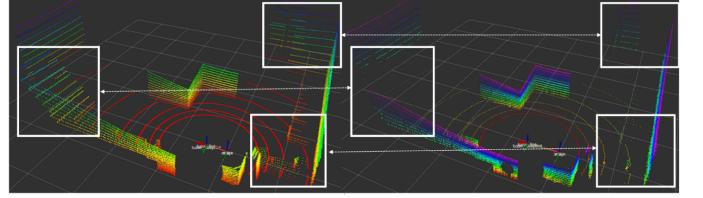


Fig. 10: Left: the raw point cloud in a square environment. right: the point cloud after outlier removed.

1. Remove outlier point cloud: The raw point clouds can be divided into the ground, cluster, and outlier according to environmental characteristics [19].

To compute the angle change among the extracted point clouds, the point cloud with a small angle will be set as the ground point cloud [12]. Using a vertical distance between the two point clouds, we cluster all the point clouds once the ground points have been removed. Further, ignore clusters with a small number of point clouds, which are defined as outliers (here we set 30 points as the threshold). The comparison of the different processing results is shown in Fig 10.

2. Downsample point cloud: A large number of laser point clouds will also lead to matching delay and error, so it is necessary to carry out reasonable downsampling to reduce the number of the processed point cloud. After a series of tests and comparisons, we decide to save nearly half of the processed point clouds, while retaining their shape and details.

3. limited point cloud in an interesting range: The point cloud which is too far away will lead to large distortion and false recognition, so the laser point cloud within the range of $0.5 - 10m$ is retained here.

3) Point cloud registration: After selecting the pre-processing point cloud and obtaining the initial position in the map, we need to predict motion between two frame laser point clouds. Our method of calculating pose from the LiDAR sensor is based on the Normal Distributions Transform (NDT) registration method [4] after analyzing few methods of 3D laser point cloud matching. The NDT algorithm of scanning alignment produces the global state by evaluating input parameters such as the preprocessing laser point cloud, predicted motion estimate (the measurements of leg odometry ${}_{base}^{map}\mathbf{T}_{lo}$ are used to determine the increment and continuously accumulate the previous position), and initial localization data.

C. Sensor Fusion

In order to balance accuracy and real-time for state estimation, it is necessary to consider and integrate the approach of proprioceptive and exteroceptive sensors, and ensure relatively high-frequency and high-precision output through the multi-sensor fusion module. This fusion framework also needs to further optimize the trajectory and provide the smoothness to prevent the failure of different data.

Gathered sensors data from the measurement systems, we correct the state estimate of the kinematic-inertial odometry by the LiDAR data association in a loosely coupled style from robot locomotion. Table I summarizes the sensors data and required state components of modules in the EKF fusion, this will allow us to track information between the modules.

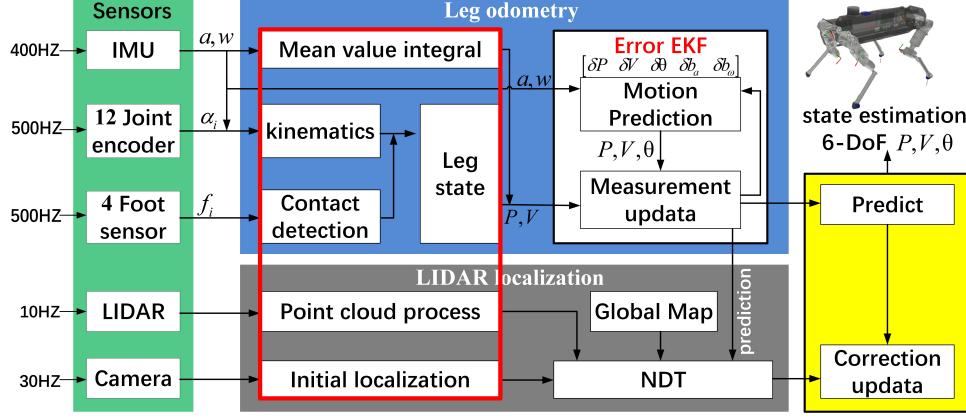


Fig. 11: the block diagram for state estimation structure, the input data are all come from sensors and the position, velocity and orientation of robot base-link in map coordinate will be obtained.

TABLE I: SENSOR DATA AND STATE COMPONENTS.

Sensor	Frequency	position prediction	velocity prediction	orientation prediction
IMU	400			
Leg encoder	100	—	update	—
Foot sensor	100	—	update	—
LiDAR	10	update	—	update
Camera	30	update	—	—

The multi-sensor fusion framework is represented in Fig. 11, where the red rectangle represents the preprocessing of the sensor data, and the flow of information among different modules signifies the transmission of the 6-DoF state estimation. The IMU prediction will be sent to the control system at each iteration of leg odometry to minimize latency, as well as provide relatively stable measurements. LiDAR registration and initial visual localization are run at much lower frequencies, so they are used to correct global localization. Finally, we obtain an accurate and stable state estimation by using the error EKF filtering method.

Using different sensors and considering both frequency and accuracy, we build a quadruped state estimation architecture. The total output trajectory of sensor data can be illustrated in Fig 12. When there are higher-precision sensor data, the data in the filter will be updated, and the subsequent measurement trajectory will have to be recalculated, otherwise, the relatively high-accuracy sensor data will always be used. As we combine and arrange data, we obtain the highest accuracy with the slowest update frequency in the loosely coupled EKF method.

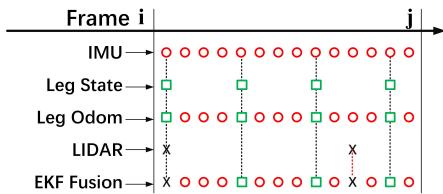


Fig. 12: Measurements from input sensors in the same time frame, and the different sensor data can be incorporated into the fusion frame as shown.

V. EXPERIMENTAL EVALUATION

By applying the above-mentioned specific method to the quadruped robot, we can obtain adequate experimental evaluation and reasonably reliable verification, by using the built experimental system providing appropriate reference truth data, we can demonstrate that the algorithm meets our needs.

A. Experimental System

Robot simulation platform and complex actual quadruped robot StartDog are built mostly with ROS. The gazebo parameters of *robotstate* are taken as the ground truth data in simulation. We employed the motion capture system, which is mounted on the robot's upper deck to record the 'actual' pose information, to obtain real-time ground truth data, as well as to further evaluate the parameters of our algorithms. The simulation and actual test experiment system are detailed shown in Fig. 13.

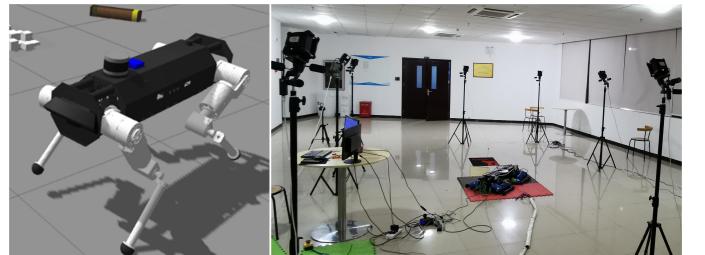


Fig. 13: The simulation and experiment environment.

B. Experiment Error Evaluation

In order to assess the accuracy of the state estimation, it is necessary to calculate the dispersion degree of the relative error between the different reference values and the calculated value. For experimental evaluation, the root mean square error and unit distance drift can be used. The root mean square error (RMSE) is widely used to evaluate velocity and attitude, and this is defined as the root mean square of the absolute error between velocity estimates and real values:

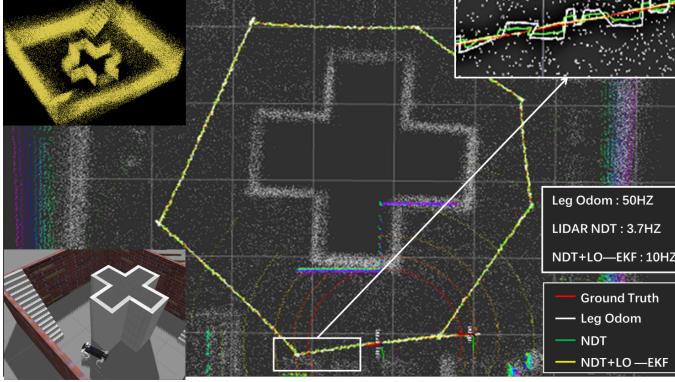


Fig. 14: Different evaluation for state estimation are shown to compare the accuracy and frequency in simulation.

$$RMSE = \sqrt{\frac{\sum_{k=1}^N |x_k - x_k'|^2}{N}} \quad (4)$$

Drift per Distance Traveled (DDT) represents the evaluation index of the change for displacement parameters. Basically, it is defined as the difference between the different values at different sampling times. The DDT error in one direction is defined as: For displacement parameters, Drift per Distance Traveled (DDT) represents the evaluation index.

$$DDT_x = 100 \frac{\frac{1}{N} \sum_{k=1}^N |x_k - x_k'|}{\sum_{k=1}^{N-1} |x_{k+1} - x_k|} = 100 \frac{\bar{e}_x}{\sum_{k=1}^{N-1} \Delta x} \quad (5)$$

C. Experiment and analysis

1) Simulation Experiment for Fusion: By using a keyboard, we can control the quadruped robot to walk in the simulation and record the gazebo measurements and sensor data. We can then analyze the fusion data to assess the performance of the robot. The sensor fusion algorithm is tested and Fig.14 shows the details during walking, which includes the global map, simulation square environment, corresponding trajectories of different localization methods, as well as the comparisons of output frequency and accuracy.

Analyzing the tracking accuracy of different localization methods, we can see that the moving trails are identical, but the data noise of leg odometry is relatively high, whereas sensor fusion gets a satisfactory accuracy. In addition, this fusion method improves and maintains the output frequency. As a result, accuracy, frequency, and sensor fusion are all effectively balanced by the sensor fusion approach.

2) Actual Experiment for Fusion: The real experiment of the multi-sensor fusion approach is carried out by using the trot gait for our quadruped robot startDog in a flat environment. The remote controls the robot moves in a step of 0.2m per cycle of the leg order of 1-4-2-3.

To transform the different localization data to the map coordinate system and output the different trajectory comparisons during robot motion experiment, including the estimation of IMU, leg kinematics and leg odometry, laser-based localization, the motion capture system and sensor fusion, as shown in

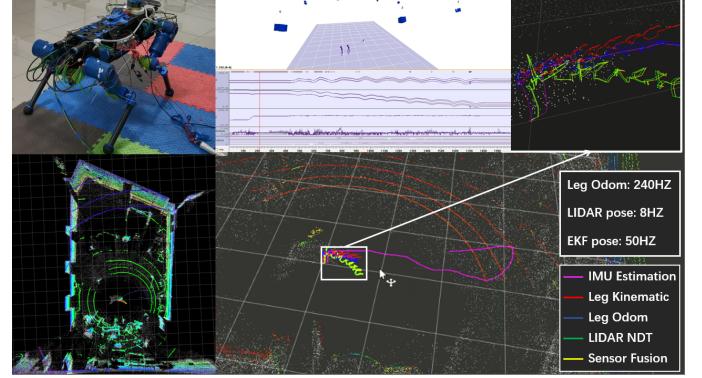


Fig. 15: Different localization trajectories compared in RVIZ and the detail show the various of experiment visualization.

Fig. 15. From different sensor data and coordinate systems, it is possible to obtain the following transformation relationship:

$$\overset{map}{base}\mathbf{T} = \overset{map}{odom}\mathbf{T} \cdot \overset{odom}{base}\mathbf{T} \quad (6)$$

$$\overset{map}{imu}\mathbf{T} = \overset{map}{mbase}\mathbf{T} \cdot \overset{base}{imu}\mathbf{T} \quad (7)$$

where $\overset{map}{odom}\mathbf{T}$ can directly obtain from the initial localization through camera as mentioned before, $\overset{base}{imu}\mathbf{T}$ present the relationship between IMU and robot base link. Similarly, combined with the IMU and leg kinematics in the map coordinate system, we can know the leg odometry in the map coordinate system.

This visualization in Fig. 15 shows the different frequency optimization results as well as the trajectory visualization. While the leg odometry shows high frequency (Note: the frequency of leg odometry is different between the simulation and the actual environment in order to test and control the quadruped robot more easily), there is a clear drift in true trajectories when compare with the ground truth data, while the LiDAR-based localization shows a reliable track with discontinuous lines. The fusion of multiple sensor data will further improve accuracy, as well as ensure smooth trajectory and promote the frequency of sensors fusion.

3) Quantitative analysis for Fusion: After fusion has output a reliable frequency, more attention needs to be paid to the accuracy for state estimation. Since the initial position data of the target balls are only recorded in the motion capture coordinate system and cannot communicate with ROS, to further quantitative compare the accuracy of the robot walk trajectory and for error analysis, it is possible to project the marker and fusion data into a fixed coordinate system. In order to ensure the unity of the data process and start from zero in the trajectory visualization, the coordinates of the marker0 are chosen when the robot in the initial state, and then these state quantities are converted to the initial marker0 coordinate system. The corresponding different coordinate systems under the actual experiment are shown in Fig. 16. For subsequent treatment, project the measurement information into the target marker coordinate system:

$$\overset{bar0}{bar-i}\mathbf{T} = \overset{mc}{bar0}\mathbf{T}^{-1} \cdot \overset{mc}{bar-i}\mathbf{T} \quad (8)$$

$$\overset{mmc}{odom}\mathbf{T} = \overset{mc}{bar0}\mathbf{T} \cdot \overset{bar0}{base}\mathbf{T} \cdot \overset{base}{odom}\mathbf{T} \quad (9)$$

$$\bar{base}^0 \mathbf{T} = \bar{mc}_{bar0}^0 \mathbf{T}^{-1} \cdot \bar{mc}_{odom}^0 \mathbf{T} \cdot \bar{odom}_{base}^0 \mathbf{T} \quad (10)$$

$$\bar{imu}^0 \mathbf{T} = \bar{base}^0 \mathbf{T} \cdot \bar{imu}^0 \mathbf{T}, \quad (11)$$

$$\bar{laser}^0 \mathbf{T} = \bar{base}^0 \mathbf{T}^{-1} \cdot \bar{mc}_{odom}^0 \mathbf{T} \cdot \bar{map}_{laser}^0 \mathbf{T} \cdot \bar{map}_{laser}^0 \mathbf{T} \quad (12)$$

Where $\bar{mc}_{bar-i}^0 \mathbf{T}$ means the transformation of marker0 relative to the motion capture system in different timestamp. $\bar{base}^0 \mathbf{T}$ represent the relationship between marker0 in designed board and robot base link at the beginning. The fixed coordinate transformation relationship of robot Odom and coordinate the motion capture system is represented as $\bar{odom}^0 \mathbf{T}$. Other different transformations can be found in the above part.

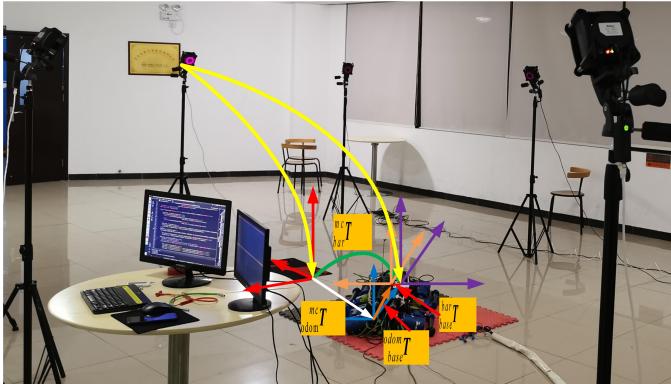


Fig. 16: The relationship between different transformation coordinate systems during actual experiment.

Fig. 18 shows the measurement results of the robot's position and orientation relative to the marker0 coordinate system. When comparing with our sensor fusion approach, both the position and the orientation can be closely matched to the ground truth data from the motion capture system. Because of the kinematics cumulative error as well as the unknown characteristic on yaw direction, the rotation around the z-axis and the position in the z-axis always have a large noise and a non-negligible drift. Although the position accuracy of the x-axis and y-axis has greatly improved, it has been affected by the computation error from leg kinematics as well as inner kinematic errors from body locomotion that can be addressed by the essential movement calibration. Another improvement is in pitch and roll, which are mainly derived from the pre-process IMU. Therefore, our fusion algorithm matches the ground truth in 6 directions with low errors.

We also compute the DDT for the position error and the RMSE for the orientation error during the quadruped robot trotting a few times, comparing leg kinematics(Kin), leg odometry(Kin+EKF), LiDAR-based localization(LiDAR), and sensor fusion(LiDAR+EKF). The error accuracy of the above different methods (DDT and RMSE) related to motion capture measurements is shown in Fig. 17. According to the results, the maximum position error is less than 5cm, and the maximum angle error is less than 1 degree. These errors are less than the performance index of the requirement. As a result, sensor fusion provides state estimation information effectively from these quantitative analyses.

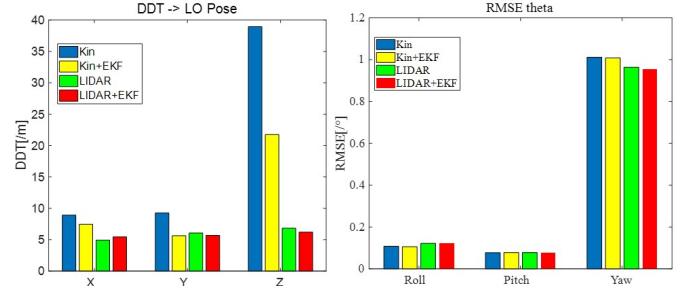


Fig. 17: left: Different positions are estimated for trot logs using DDT.. Right: Estimation of the robot's different velocity for trot logs by RMSE.

CONCLUSION

In this paper, we introduced a novel and modular state estimation algorithm to balance frequency and accuracy for quadruped robot, which fuses inertial, encoder, camera and LiDAR data by Extended Kalman Filter, also a series of robust and effective preprocessing approaches for different sensors measurements is proposed. A motion capture system was used to evaluate the performance of the fusion algorithm on our experimental StartDog robot. Using the proposed algorithm, a consistently low error rate and high estimation frequency can be achieved. For future work, we will transplant all the systems to the onboard computer, additionally, we aim to perform factor graph optimization on state estimation, as well as test our algorithm in the outside environment with dynamic shock.

ACKNOWLEDGMENT

This work is supported by the Legged Robotics Group (LRG) of Harbin Institute of Technology, Shenzhen(HITSZ) and RObot Mobility and mAnipulation (ROMA) lab at Southern University of Science and Technology(SUSTech).

It's worth noting that most of the experimental work in this essay was performed while pursuing a master's degree at Harbin Institute of Technology in Shenzhen with a group of interesting guys. I regret that a few mistakes exist in this paper and I cannot reboot our quadruped robot to do more experiments to fix these errors at now. So this paper will still be open and hope you will know these disadvantages to do more excellent work for the quadruped robot.

REFERENCES

- [1] C. Dario Bellicoso, Marko Bjelonic, Lorenz Wellhausen, Kai Holtmann, Fabian Gunther, Marco Tranzatto, Peter Fankhauser, and Marco Hutter. Advances in real-world applications for legged robots. *Journal of Field Robotics*, 35(8):1311–1326, 2018.
- [2] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [3] P. Biber and W. Strasser. The normal distributions transform: a new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*, volume 3, pages 2743–2748 vol.3, 2003.
- [4] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003.

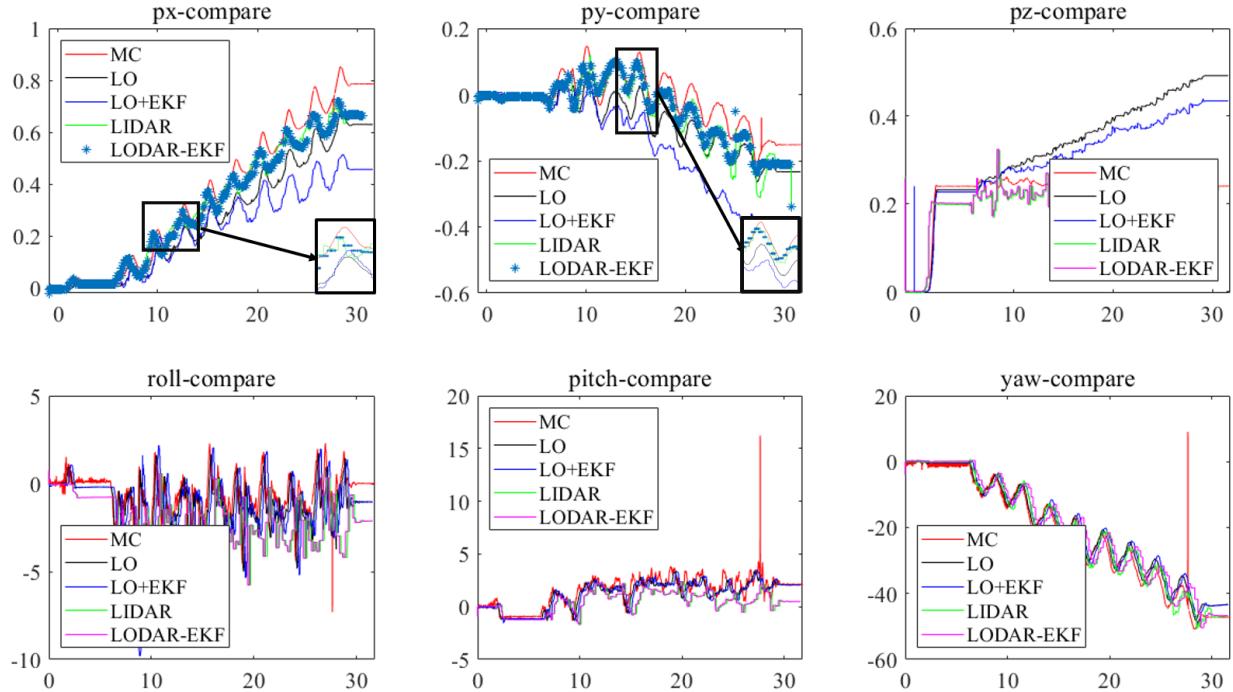


Fig. 18: Different track comparison results for robot position and velocity in position-x, position-y, position-z, roll, pitch and yaw direction.

- [5] Gerardo Bledt, Matthew J. Powell, Benjamin Katz, Jared Di Carlo, Patrick M. Wensing, and Sangbae Kim. MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2245–2252, 2018.
- [6] Michael Bloesch, Marco Hutter, Mark A Hoepflinger, Stefan Leutenegger, Christian Gehring, C David Remy, and Roland Siegwart. State estimation for legged robots-consistent fusion of leg kinematics and imu. *Robotics*, 17:17–24, 2013.
- [7] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.
- [8] Marco Camurri, Stephane Bazeille, Darwin G. Caldwell, and Claudio Semini. Real-time depth and inertial fusion for local SLAM on dynamic legged robots. In *2015 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 259–264, San Diego, CA, USA, 2015. IEEE.
- [9] Annett Chilian, Heiko Hirschmüller, and Martin Görner. Multisensor data fusion for robust pose estimation of a six-legged walking robot. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2497–2504, 2011.
- [10] Jeffrey Delmerico, Stefano Mintchev, Alessandro Giusti, Boris Gromov, Kamil Melo, Tomislav Horvat, Cesar Cadena, Marco Hutter, Auke Ijspeert, Dario Floreano, et al. The current state and future outlook of rescue robotics. *Journal of Field Robotics*, 36(7):1171–1191, 2019.
- [11] Maurice F. Fallón, Matthew Antone, Nicholas Roy, and Seth Teller. Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 112–119, 2014.
- [12] Shinpei Kato, Eiji Takeuchi, Yoshio Ishiguro, Yoshiki Ninomiya, Kazuya Takeda, and Tsuyoshi Hamada. An Open Approach to Autonomous Vehicles. *IEEE Micro*, 35(6):60–68, 2015.
- [13] Jeremy Ma, Sara Susca, Max Bajracharya, Larry Matthies, Matt Malchano, and Dave Wooden. Robust multi-sensor, day/night 6-DOF pose estimation for a dynamic legged vehicle in GPS-denied environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 619–626, St Paul, MN, USA, 2012. IEEE.
- [14] Simona Nobile, Marco Camurri, Victor Barasuol, Michele Focchi, and Maurice Fallon. Heterogeneous Sensor Fusion for Accurate State Estimation of Dynamic Legged Robots. In *Robotics Science and Systems*, 2017.
- [15] Simona Nobile, Raluca Scona, Marco Caravagna, and Maurice Fallon. Overlap-based ICP tuning for robust localization of a humanoid robot. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4721–4728, Singapore, Singapore, 2017. IEEE.
- [16] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *Robotics, IEEE Transactions on*, 34(4):1004–1020, 2018.
- [17] Marc Raibert, Kevin Blankespoor, Gabriel Nelson, and Rob Playter. BigDog, the Rough-Terrain Quadruped Robot. *IFAC Proceedings Volumes*, 41(2):10822–10825, 2008.
- [18] C. Semini, J. Buchli, M. Frigerio, T. Boaventura, and D. G. Caldwell. Hyq - a dynamic locomotion research platform. In *Int'l Workshop on Bio-Inspired Robots*, 2011.
- [19] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.
- [20] Joan Sola. Quaternion kinematics for the error-state kalman filter. *arXiv preprint arXiv:1711.02508*, 2017.
- [21] Paweł Wawrzyniński, Jakub Możaryn, and Jan Klimaszewski. Robust estimation of walking robots velocity and tilt using proprioceptive sensors data fusion. *Robotics and Autonomous Systems*, 66:44–54, 2015.
- [22] David Wisth, Marco Camurri, and Maurice Fallon. Robust Legged Robot State Estimation Using Factor Graph Optimization. *IEEE Robotics and Automation Letters*, 4(4):4507–4514, 2019.
- [23] Shuo Yang, Hans Kumar, Zhaoyuan Gu, Xiangyuan Zhang, Matthew Travers, and Howie Choset. State Estimation for Legged Robots Using Contact-Centric Leg Odometry. *arXiv:1911.05176 [cs, eess]*, 2019.
- [24] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2):401–416, 2017.