Inertial Learning for Improved Dynamic Legged Robot State Estimation

Russell Buchanan, Marco Camurri, and Maurice Fallon

***Abstract*— This paper introduces a learned displacement measurement which improves state estimation algorithms for legged robots in challenging scenarios such as slipping and deformable terrain. By using a deep neural network we show how displacement can be learned from IMU signals alone and then added as an additional measurement to a filtering-based pipeline. This greatly reduces drift for proprioceptive-only state estimation which is critical for legged robots deployed in vision/lidar denied environments such as foggy sewers or dusty mines. In several real experiments in challenging scenarios we show an average reduction of 45% relative pose error and 65% absolute pose error compared to a traditional approach.**

I. Introduction

Recent advances in the locomotion capabilities of legged robots have motivated their use in dull and dirty industrial operations such as routine inspection and monitoring. To perform these tasks autonomously, a robot must have accurate state estimation with limited drift over long periods of time.

This problem has been extensively researched using vision and lidar based-methods [1], [2]. However, the performance of these methods will suffer in vision/lidar denied environments such as mines and sewers due to dust particles and water vapour [3] which can cause a robot's vision and lidar systems to malfunction.

Proprioceptive only methods fuse information from joint kinematics and Inertial Measurement Unit (IMU) data through either filtering [4], [5] or windowed optimization [6]. However, these methods suffer from continuous open-loop drift over time, particularly in the Z position direction. Incorporation of kinematic information into the filter is troublesome because slip events or deformation of the robot's foot or terrain introduce non-Gaussian error. These errors accumulate over time and lead to estimator drift.

Separately, there have been recent advances in pedestrian tracking using inertial data alone. IONet [7] was the first neural network trained to infer the velocity of human motion from the IMU data. More recently inertial-only navigation systems for human motion have been integrated with an Extended Kalman Filter (EKF) for full 6 Degree of Freedom (DoF) state estimation [8]. The key insight of these works was that it is possible to learn a *motion prior* from inertial data which could be robust to changing biases. However, these methods are susceptible to failure if the trained model is applied outside its training domain e.g. if trained using locomotion in an urban setting it would need to be retrained to support tracking of a person travelling in a car or by bike.

The authors are with the Oxford Robotics Institute at the University of Oxford, UK. {russell, mcamurri, mfallon} @robots.ox.ac.uk

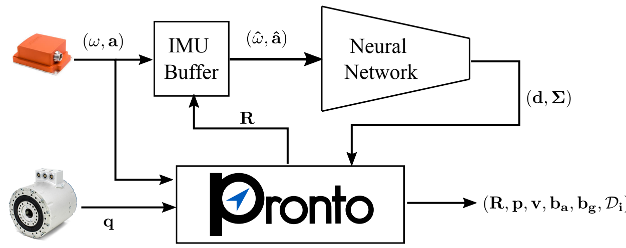

Fig. 1: Block diagram of the full pipeline. IMU and kinematic data are processed by Pronto which provides an estimate of the robot's orientation. This allows a buffer of IMU data to be aligned with gravity and provided as input to the neural network. Displacement measurements and their covariance can then be fused back into Pronto to improve the state estimate.

In this workshop paper we propose a deep neural network capable of learning a motion prior on the locomotion of a legged robot. The network takes as input a buffer of IMU data and outputs a displacement and covariance. Our method is similar to TLIO [8] but we fuse this information with a quadruped state estimator (Pronto [5]) to track the motion of the ANYbotics ANYmal. We show how this general learned displacement measurement can be used in a classical inertial/kinematic filtering-based framework. We present initial results from real experiments with the robot in challenging conditions such as slipping and deformable terrain. We demonstrate a significant reduction in relative pose error, on average 45%, in these situations.

II. Related Works

Here we briefly summarize some relevant works on proprioceptive state estimation for legged robots and inertial-only estimation for pedestrians.

A. Kinematic-Inertial State Estimation in Legged Robots

Lin et al. [9] presented the first state estimator for legged robots which fused information from both kinematics and IMU in an EKF. They worked with a hexapod and assumed three feet were always in contact; adding the inertial data enabled them to execute a more dynamic tripod running gait. Bloesch et al. introduced a new filtering approach called the Two-State Implicit Filter (TSIF) [4]. This method employs residual-based modelling of the available information, eliminating the need for an explicit process model. TSIF is the default state estimator provided with the ANYmal quadruped robot [10] which is used in our experiments.

More recently Pronto [5] has been presented as an EKF-based state estimator with the capacity to fuse pose corrections from vision or lidar sensing. This is done by maintaining a history of measurements and applying corrections to the past trajectory asynchronously as exteroceptive measurements arrive. However, like TSIF, when only using proprioceptive data this method suffers from incremental drift in the Z direction over long periods of time.

1) Learned Inertial Navigation: A recent direction of research in inertial navigation systems (INS) is the application of machine learning to infer motion directly from batches of IMU data — typically focused on tracking a mobile phone held by a person. These methods have shown impressive performance when compared to traditional inertial-only pedestrian tracking methods.

Chen et al. introduced IoNet [7], the first data driven INS which estimates odometry using deep recurrent neural networks. They trained a network to relate buffered data from an IMU into 2D displacements and orientation changes. Their method outperformed both naive integration and existing zero velocity update (ZUPT) pedestrian tracking methods. RoNIN [11] is another recent learning based approach which infers 2D velocity and orientation directly from raw IMU data; they claim to outperform IoNet.

TLIO [8] is the first approach to be demonstrated in full 3D and the first to be integrated within a complete filtering framework. An EKF is used to filter the full 6 DoF state and process updates are performed by traditional IMU integration. They trained a network with 40 hours of pedestrian walking data to output displacement and covariance from the buffered IMU data. These measurements are provided to the EKF as relative position measurements and allowed the filter to correct for biases online. The approach showed a significant reduction in odometry drift compared to previous methods.

The application of these methods to robotics has not yet become wide spread, however [12] demonstrated an IMU only state estimation method for vehicles. They exploited the constrained motion model of a car and demonstrated impressive performance on the KITTI dataset.

III. Method

We propose to train a neural network using IMU data while a legged robot walks with a dynamic gait. The network learns a motion prior for the given robot and gait, and outputs displacement and uncertainty on this measurement. This can then be integrated by our filter-based state estimator using Pronto [5] which is designed to fuse displacement measurement. Fig. 1 shows our system architecture.

A. Problem Statement

An IMU provides rotational velocity ω and linear acceleration a in three dimensions. We buffer N samples of this raw data such that the input dimension to our network is $N \times 6$. We propose a neural network which learns a function $\mathcal{F}$ such that

$$(d, \Sigma) = \mathcal{F}(\hat{\omega}, \hat{a}), \tag{1}$$

where $(\hat{\omega}, \hat{a})$ is the buffer after gravity alignment. The variables d and Σ are the resulting displacement measurement and its covariance respectively.

As with TLIO we use a 1D version of the ResNet18 architecture [13]. We also make use of two loss functions. For the first 10 epochs we use Mean Square Error (MSE):

$$\mathcal{L}(d, \hat{d}) = \frac{1}{n} \sum_{i=1}^{n} \left\| d_i - \hat{d}_i \right\|^2, \tag{2}$$

where $\hat{d}$ is output of the network during training and d is the ground truth displacement from Vicon.

Once the network has begun to converge the loss function is changed to Gaussian Maximum Likelihood (GML) which allows for the measurement covariance to be estimated:

$$\mathcal{L}(d, \hat{\Sigma}, \hat{d}) = \frac{1}{n} \sum_{i=1}^{n} -\log\left(\frac{1}{\sqrt{8\pi det(\hat{\Sigma})_i}} e^{\frac{1}{2}\left\| d_i - \hat{d}_i \right\|^2_{\hat{\Sigma}_i}} \right), \tag{3}$$

where $\hat{\Sigma}_i$ is the covariance of the network output. This is computed:

$$\hat{\Sigma}_i = diag(e^{2\hat{u}_{xi}}, e^{2\hat{u}_{yi}}, e^{2\hat{u}_{zi}}), \tag{4}$$

where $\{\hat{u}_{xi}, \hat{u}_{xi}, \hat{u}_{zi}\}$ come from the network uncertainty vector. This process is similar to TLIO and is more fully explained in [8].

B. Training

We recorded 59 logs of between 1 and 5 minutes each for a total 2 hours 24 minutes. Three of the logs (shown in Fig. 2) were selected as test cases and their results are presented in this paper. The remaining 56 were split 75:25 into training and validation datasets. The training set included walking on flat ground, terrain with different thickness of soft padding, different slippery surfaces and elevated terrain. A Vicon motion tracking system was used to generate ground truth 6 DoF pose trajectories which were used as training reference. Total training time was six hours on a laptop with a Quadro P2000 4GB GPU.

IV. Test Experiments

The three test experiments were performed on different terrains as shown in Fig. 2. In each case, the robot was manually controlled to walk in several loops. For each experiment we compare the existing state estimate on the robot TSIF [4] and Pronto [5] — which are both classical kinematics/inertial estimators — to our method which we call from here on Learned Inertial Navigation System (LINS). For all training and experiments we used a buffer size of $N = 400$ (1 s of 400 Hz IMU data). Displacement measurements were estimated at 20 Hz, resulting in subsequent buffers largely overlapping the same data.

A. Soft Terrain

One of the major challenges of state estimation for legged robots is soft or deformable terrain. The specific moment of contact is ill-defined — as the foot needs to compress the terrain before movement stops. The continued downward motion of the foot after the first contact event results in

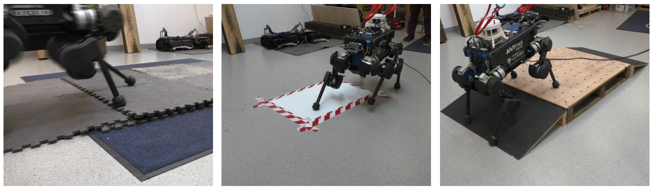

Fig. 2: Test experiments. **Left:** Robot foot compressing into soft terrain. **Middle:** Slip event from walking on slippery terrain. **Right:** Robot climbing over an obstacle.

Mean 5 m Relative Pose Error (RPE) $\mu(\sigma)$[m]			
Data	**TSIF**	**PRONTO**	**LINS**
SLIP	0.10 (0.02)	0.13 (0.02)	**0.05** (0.02)
SOFT-TERRAIN	0.16 (0.06)	0.24 (0.06)	**0.15** (0.07)
TERRAIN	**0.15** (0.04)	0.31 (0.08)	0.22 (0.06)

TABLE I

a hallucinated upward Z drift. Some realistic examples of deformable terrain include sand, mud or tall grass.

To simulate this we placed several layers of foam padding on the floor. This resulted in significantly more vertical drift compared to the normal floor.

B. Slipping

Slipping breaks some kinematic assumptions as the foot is in contact with the terrain but does not have zero relative velocity. This results in error spikes in odometry error when the estimator heavily relies on kinematic sensing. Slipping is also common in field robotics and can be caused by wet or loose terrain.

To create slip events we placed a common office white board on the floor and added dish soap to make a slippery surface. The robot was then commanded to walk over the terrain several times to creating slip events.

C. Obstacle Terrain

Climbing onto terrain is a common challenge for legged robots but does not present significant difficulties for proprioceptive state estimation. Instead we present results from this experiment to demonstrate the network does not simply learn that the robot walks on flat ground and that it can generalize to walking on elevated terrain as well.

We constructed an elevated platform on a pallet. The robot walks blindly as in the other experiments but ramps allow it to climb the obstacle.

V. Results and Discussion

We evaluated performance using two metrics: relative pose error (RPE) and absolute pose error (APE) as defined in [14]. In all experiments adding the learned measurement to Pronto improved the state estimation significantly, even on the obstacle terrain course. On average the 5 m RPE was reduced by 45% as shown in Table I and APE reduced by 65% over 121 m travelled as shown in Table II.

Mean Absolute Pose Error (APE) $\mu(\sigma)$[m]			
Data	**TSIF**	**PRONTO**	**LINS**
SLIP	0.40 (0.24)	0.50 (0.28)	**0.12** (0.06)
SOFT-TERRAIN	0.58 (0.40)	0.78 (0.52)	**0.23** (0.13)
TERRAIN	**0.22** (0.17)	0.40 (0.25)	0.24 (0.12)

TABLE II

A. Soft Terrain

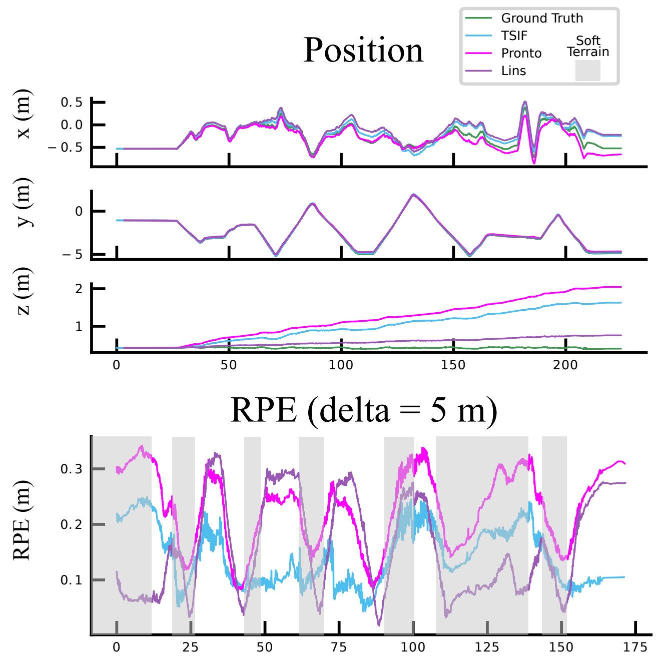

Fig. 3: Results from soft terrain experiment. The upper plot shows position over time and the bottom plot shows RPE over time. The RPE plot begins at the time when the robot first travels 5 m.

The results for the soft terrain are shown in Fig. 3. The periods where the robot walked on the soft terrain are highlighted in grey. It is clear in these areas the learned measurement has the greatest impact in reducing the RPE. Additionally the drift in Z positions which can be seen in both TSIF and Pronto in the upper graft, is drastically reduced by the addition of the learned factor.

B. Slipping

Results from the slipping experiment demonstrate the greatest reduction in drift by adding the learned measurement. This is shown in Fig.4.

C. Obstacle Terrain

The obstacle terrain experiment is primarily to demonstrate the generalization of our method to non planar terrain. The upper plot of Fig. 5 shows an improvement in Z drift while still showing increases and decreases in elevation while the robot climbs over the terrain. In initial experiments

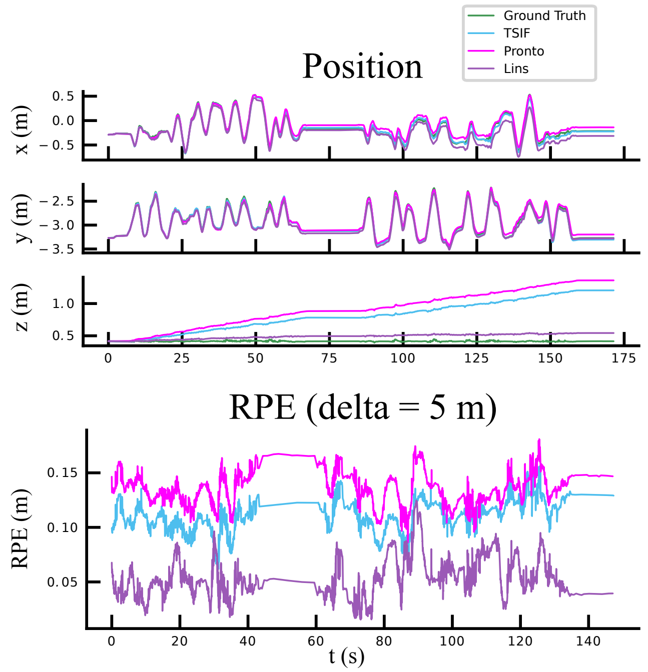

Fig. 4: Results from slipping terrain experiment. The upper plot shows position over time and the bottom plot shows RPE over time. The RPE plot begins at the time when the robot first travels 5 m.

we found if the training dataset did not include enough examples of the robot walking over elevated terrain, the network assumed constant planar elevation. The resulting state estimate would then ignore the elevation gain and loss from climbing on terrain. These results show the network has generalized beyond flat ground.

VI. Conclusions

In this paper we introduced a learned displacement measurement which, when used in a filtering framework, greatly reduces drift in proprioceptive state estimation. In challenging situations such as slipping and deformable terrain we see the greatest benefit and in this paper we show in real experiments an improvement of 45% relative pose error and 65% absolute pose error over 121 m travelled. This work shows great promise in improving proprioceptive state estimation which is vital for legged robots in vision denied environments. In future work we intend to show more general usability of this measurement in optimization frameworks and combine with exteroceptive sensors for highly robust state estimation.

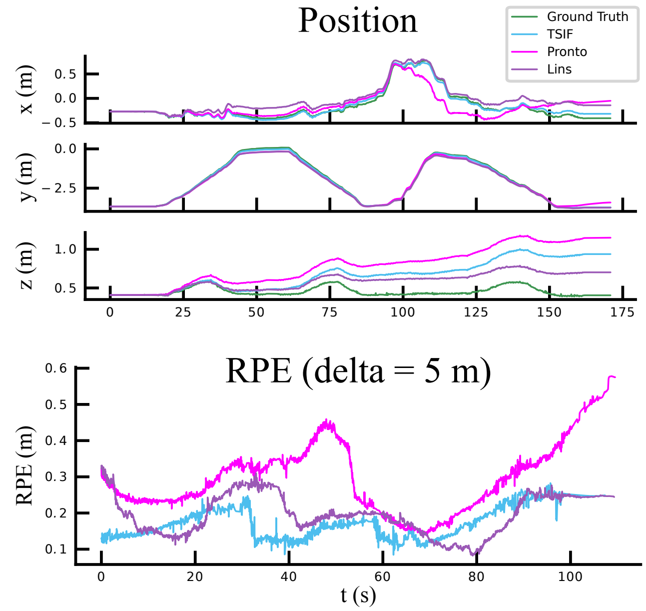

Fig. 5: Results from obstacle terrain experiment. The upper plot shows position over time and the bottom plot shows RPE over time. The RPE plot begins at the time when the robot first travels 5 m.

References

[1] D. Wisth, M. Camurri, and M. Fallon, “Preintegrated velocity bias estimation to overcome contact nonlinearities in legged robot odometry,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 392–398.

[2] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carlone, C. Guaragnella, and A.-a. Aghamohammadi, “Locus: A multi-sensor lidar-centric solution for high-precision odometry and 3d mapping in real-time,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 421–428, 2021.

[3] H. Kolvenbach, D. Wisth, R. Buchanan, G. Valsecchi, R. Grandia, M. Fallon, and M. Hutter, “Towards autonomous inspection of concrete deterioration in sewers with legged robots,” *Journal of Field Robotics*, 2020.

[4] M. Bloesch, M. Burri, H. Sommer, R. Siegwart, and M. Hutter, “The two-state implicit filter recursive estimation for mobile robots,” *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 573–580, 2018.

[5] M. Camurri, M. Ramezani, S. Nobili, and M. Fallon, “Pronto: A Multi-Sensor State Estimator for Legged Robots in Real-World Scenarios,” *Frontiers in Robotics and AI*, vol. 7, 2020.

[6] R. Hartley, M. G. Jadidi, L. Gan, J. K. Huang, J. W. Grizzle, and R. M. Eustice, “Hybrid Contact Preintegration for Visual-Inertial-Contact State Estimation Using Factor Graphs,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018, pp. 3783–3790.

[7] C. Chen, X. Lu, A. Markham, and N. Trigoni, “IoNet: Learning to cure the curse of drift in inertial odometry,” *AAAI Conference on Artificial Intelligence*, pp. 6468–6476, 2018.

[8] W. Liu, D. Caruso, E. Ilg, J. Dong, A. I. Mourikis, K. Daniilidis, V. Kumar, and J. Engel, “Tlio: Tight learned inertial odometry,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5653–5660, 2020.

[9] P.-C. Lin, H. Komsuo, and D. E. Koditschek, “Sensor Data Fusion for Body State Estimation in a Hexapod Robot With Dynamical Gaits,” *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 932–943, 2006.

[10] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, “ANYmal - a highly mobile and dynamic quadrupedal robot,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[11] S. Herath, H. Yan, and Y. Furukawa, “Ronin: Robust neural inertial navigation in the wild: Benchmark, evaluations, new methods,” in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020, pp. 3146–3152.

[12] M. Brossard, A. Barrau, and S. Bonnabel, “AI-IMU Dead-Reckoning,” *IEEE Trans. on Intelligent Vehicles*, vol. 5, no. 4, pp. 585–595, 2020.

[13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[14] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, “A benchmark for the evaluation of rgb-d slam systems,” in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580.