Absolute humanoid localization and mapping based on IMU Lie group and fiducial markers

Mederic Fourmy, Thomas Flayols, Florenc Caminade, Dinesh Atchuthan, Nicolas Mansard, Joan Sola

▶ **To cite this version:**

Mederic Fourmy, Thomas Flayols, Florenc Caminade, Dinesh Atchuthan, Nicolas Mansard, et al.. Absolute humanoid localization and mapping based on IMU Lie group and fiducial markers. 2019. hal-02183498v2

HAL Id: hal-02183498

https://hal.science/hal-02183498v2

Preprint submitted on 29 Jul 2019 (v2), last revised 18 Oct 2019 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Absolute humanoid localization and mapping based on IMU Lie group and fiducial markers

Mederic Fourmy[†], Thomas Flayols[†], Florenc Caminade[†], Dinesh Atchuthan[†], Nicolas Mansard[†] and Joan Solà[†*]

***Abstract*— Current locomotion algorithms in structured (indoor) 3D environments require an accurate localization. The several and diverse sensors typically embedded on legged robots (IMU, coders, vision and/or LIDARS) should make it possible if properly fused. Yet this is a difficult task due to the heterogeneity of these sensors and the real-time requirement of the control. While previous works were using staggered approaches (odometry at high frequency, sparsely corrected from vision and LIDAR localization), the recent progress in optimal estimation, in particular in visual-inertial localization, is paving the way to a holistic fusion. This paper is a contribution in this direction. We propose to quantify how a visual-inertial navigation system can accurately localize a humanoid robot in a 3D indoor environment tagged with fiducial markers. We introduce a theoretical contribution strengthening the formulation of Forster's IMU pre-integration, a practical contribution to avoid possible ambiguity raised by pose estimation of fiducial markers, and an experimental contribution on a humanoid dataset with ground truth. Our system is able to localize the robot with less than 2 cm errors once the environment is properly mapped. This would naturally extend to additional measurements corresponding to leg odometry (kinematic factors) thanks to the genericity of the proposed pre-integration algebra.**

I. Introduction

In this work, we are interested in quantifying how accurately a humanoid robot can be localized in a structured 3D environment. The seminal works on localization of legged robots were using leg odometry, quickly followed by contributions fusing the kinematics with inertial measurements [1]. Evidently, odometry measurements can only lead to a drift of the localization. Based on leg odometry, the community has extended the localization performances by improving the behavior of the inertial-kinematic filter [2]–[4], the underlying contact model [5], [6], and by augmenting the odometer with exteroceptive measurements coming from cameras or LIDAR.

The difficulty in fusing inertial, kinematics and exteroceptive measurements stems from the disparity in the properties of each data source. Inertial and kinematic measurements come at high frequency (typically 100 Hz to 1 kHz) and are cheap to process, while images and laser scans are obtained at some few images per second and are expensive to process. On the other hand, inertial measurements are quickly deprecated while images and scans provide absolute information. This implies a rigorous synchronization between the sensors with the risk of decreasing the performances of the inertial estimation when images and laser scans are not carefully merged.

[†] LAAS-CNRS, Université de Toulouse, France
[*] Institut de Robòtica i Informàtica Industrial, Barcelona

These difficulties explain that the first works to merge proprioceptive and exteroceptive sensors for legged localization have been with some staggered approach, first fusing inertial and kinematic measurements at high frequency, and then correcting the localization drift with absolute localization computed from camera and/or LIDAR with low bandwidth and higher delay [7], [8].

Very recently, several concurrent approaches have been proposed to merge all relevant data in a unique estimator. Following the recent results in UAVs localization [9], [10], optimal estimation structured by a factor graph is a very nice framework to formulate the fusion. In [11], a graph-SLAM is proposed to fuse inertial, kinematics and visual data. Inertial measurements are considered using Forster's pre-integration factors [12]. Kinematics data are considered using a 6D factor which is also pre-integrated, but taking into account the hybrid nature of the contact dynamics using an event-based approach. Visual factors are also expressed as 6D constraints obtained by visual odometry. Results are reported on some 3-meters sequences with motion-capture ground truth. In [13], the graph-SLAM also considers inertial measurements through pre-integration, while kinematic measurements are pre-treated by the robot low-level system [2] and integrated directly as 6D factors without further consideration. As this work is applied to a quadruped robot, obtaining this 6D information indeed requires a complex filtering in itself. Finally, the visual information are considered as 2D factors in the image space, obtained from feature (KLT) matching. Impressive experimental results are demonstrated with long outdoor sequences, using a ground-truth obtained from off-line LIDAR reconstruction.

The pros and cons of these two approaches come from the choice of the factors, but the similarities are possibly more important than the differences. Both use a plain Forster pre-integration [12]. Using either visual odometry or feature tracking, both systems cannot natively benefit from the information brought by loop closure, and would fail to exploit known map information. In both cases, the kinematic factor is straightforward to write as a 6D constraint. Finally, both works are able to account for the very different sensor frequencies, while providing a good estimate at the higher frequency if needed.

In this work, we are looking for a solution to localize a humanoid robot indoor, with sufficient accuracy to navigate on some stairs, grasp a handrail or walk on a 30-cm wide beam. As the robot is going to come back again and again in the same environment, we would like to benefit from loop-closure information and localization with respect to

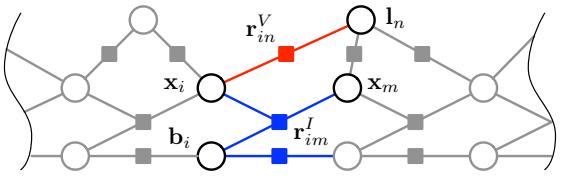

Fig. 1. A typical fraction of the factor graph, involving state blocks corresponding to keyframes $\mathbf{x}_i = (\mathbf{p}_i, \mathbf{v}_i, \mathbf{R}_i)$, biases $\mathbf{b}_i$ and landmark poses $\mathbf{l}_n$. IMU factors (blue) relate consecutive keyframes and the IMU biases. The lower branch controls bias drift along time. Visual factors (red) relate landmarks with poses $(\mathbf{p}_i, \mathbf{R}_i)$.

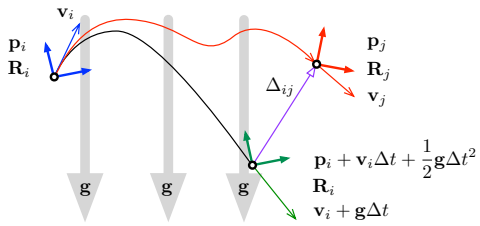

Fig. 2. The free-falling, non-rotating frame $\mathcal{G}_t$ follows a parabolic trajectory governed only by gravity $\mathbf{g}$ and determined by the initial conditions $\mathbf{p}_i$, $\mathbf{v}_i$ and $\mathbf{R}_i$ at time i (blue). The IMU delta $\boldsymbol{\Delta}_{ij}$ between times i and j is defined as the state of the IMU at time j (red) expressed in the free-falling frame $\mathcal{G}_j$ at time j (green).

some known landmarks. While our final goal is to merge in the optimal estimator the measurements coming from all the sensors of the robot, we focus here on contributions validating the use of visual-inertial localization and mapping on a humanoid robot navigating indoor in a 3D environment. For the visual factor, we rely on April tags [14], [15], while proposing a practical contribution to avoid ambiguity issues in the pose estimation of the tags. For the inertial factor, we build upon Forster pre-integration [12] and propose an original and more rigorous theoretical formulation, by exhibiting a Lie topology that is suitable for optimal estimation. This formulation, although leading to very similar results for the inertial factors, would enable an easy generalization to the other high-frequency factors that would typically arise in the humanoid contact (leg odometry based on coders, force sensors, etc). Both inertial and visual factors are processed in a factor graph resulting into a nonlinear maximum-likelihood optimization problem, solved with Ceres [16].

II. State Estimation for the Humanoid

In graph-based optimization, the problem is well represented as a bipartite graph, where one type of node refers to the variables, and the other type called *factors* represent the geometrical constraints between variables, produced by the measurements. The state $\mathbf{x}$ is modeled as a multi-variate Gaussian distribution. In the case of landmark-based visual-inertial SLAM (see Fig. 1), $\mathbf{x}$ includes robot poses and velocities $\mathbf{x}_i = (\mathbf{p}_i, \mathbf{v}_i, \mathbf{R}_i)$ and sensor biases $\mathbf{b}_i$, both at selected keyframes i along the trajectory, and landmark poses $\mathbf{l}_n \in SE(3)$. Bias are considered constant between keyframes and are taken at the i-th keyframe. In line with the recent works on the subject, we write the MAP optimization as the least-squares minimization (Fig. 1),

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_i \left\|\mathbf{r}^I_i(\mathbf{x})\right\|^2_{\boldsymbol{\Sigma}^I_i} + \sum_j \left\|\mathbf{r}^V_j(\mathbf{x})\right\|^2_{\boldsymbol{\Sigma}^V_j}, \quad (1)$$

with $\{\mathbf{r}^I, \boldsymbol{\Sigma}^I\}$ and $\{\mathbf{r}^V, \boldsymbol{\Sigma}^V\}$ indicating the residuals and covariances of respectively the inertial (IMU) and visual factors. These residuals are computed differently depending on the nature of the measurements and the state blocks they relate to. They are described in the following two chapters.

III. Pre-integrated IMU factors on dedicated Lie group

In key-frame based optimization for SLAM, IMU pre-integration was first proposed by Lupton in [17] as a means to avoid repeatedly integrating all the IMU data at each iteration of the optimizer. Lupton's seminal work used the Euler angles for orientation, and was improved 10 years later by Forster [12], who proposed a formulation in the more proper $SO(3)$ rotation Lie group. Forster's method is considered the standard to this date, and it is the one used in all major recent works in the subject, see *e.g.* [11], [13].

We however consider that there is room for improvement in the following aspects. First, neither Lupton nor Forster provide an interpretation of the IMU delta measurements, and define them as a mere algebraic construction. Second, the formulation in [12] is complicated, involving a number of large sums and products along the IMU data sequence. Third, the way Jacobians are obtained is somewhat cumbersome, leaving the reader with insufficient intuition on what is going on behind the proposed formulae. As a whole, it does not appear easy to generalize such methods to other motion pre-integration cases.

In order to give a response to these topics, our approach to IMU pre-integration differs from [12] in the following aspects. First, we provide a clear physical interpretation to the IMU deltas. Second, we present a recursive formulation, meaning that we provide equations to be applied every time an IMU sample is acquired. Moreover, one integration step is broken down in different distinguishable stages, which contributes to clarity. Third, we obtain slightly more accurate integration that stems from the exponential map of the new proposed Lie group comprising the full IMU delta (*i.e.*, not only rotation). Fourth, thanks to the abstraction provided by the Lie theory layer, our approach to Jacobians and uncertainty propagation is more compact and intuitive. And fifth and importantly, our Lie formulation easily generalizes to the pre-integration on other kinds of manifold. First steps in exploiting this generalization are explored in [18].

A. The IMU deltas matrix Lie group $\mathcal{D}$

We introduce a new matrix Lie group representation of the IMU deltas. The complete IMU pre-integration theory, including the computation of the residual, is based on this new Lie structure. The theoretical material for the Lie development in this section can be found in our report [19]. For some developments and formulae related to the particular IMU case, please refer to the appendix.

1) Definition and interpretation of the IMU deltas: The IMU deltas, as introduced in [12], [17] can be interpreted [20] as the motion increments, in terms of position, velocity and orientation, between the current IMU frame and another frame, that started at the IMU state at time i, $\mathbf{x}_i = (\mathbf{p}_i, \mathbf{v}_i, \mathbf{R}_i)$, and falls freely and without rotating at the acceleration of gravity (Fig. 2),

$$\begin{aligned} \Delta\mathbf{p}_{ij} &= \mathbf{R}_i^\top(\mathbf{p}_j - \mathbf{v}_i\Delta t_{ij} - \tfrac{1}{2}\mathbf{g}\Delta t_{ij}^2) \\ \Delta\mathbf{v}_{ij} &= \mathbf{R}_i^\top(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t_{ij}) \\ \Delta\mathbf{R}_{ij} &= \mathbf{R}_i^\top\mathbf{R}_j \\ \Delta t_{ij} &= t_j - t_i \ . \end{aligned} \tag{2}$$

2) The IMU deltas matrix Lie group $\mathcal{D}$: We propose a matrix form of the Lie group of IMU deltas as,

$$\boldsymbol{\Delta} = \begin{bmatrix} \Delta\mathbf{R} & \Delta\mathbf{v} & \Delta\mathbf{p} \\ \mathbf{0} & 1 & \Delta t \\ \mathbf{0} & 0 & 1 \end{bmatrix} \in \mathcal{D} \subset \mathbb{R}^{5\times 5} \ . \tag{3}$$

Group identity, inverse and composition stem from regular matrix identity, inverse (with $\Delta\mathbf{R}^{-1} = \Delta\mathbf{R}^\top$) and product,

$$\boldsymbol{\Delta}_{\mathcal{E}} = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & 0 \\ \mathbf{0} & 0 & 1 \end{bmatrix} = \mathbf{I}_{5\times 5} \tag{4}$$

$$\boldsymbol{\Delta}^{-1} = \begin{bmatrix} \Delta\mathbf{R}^\top & -\Delta\mathbf{R}^\top\Delta\mathbf{v} & -\Delta\mathbf{R}^\top(\Delta\mathbf{p} - \Delta\mathbf{v}\Delta t) \\ \mathbf{0} & 1 & -\Delta t \\ \mathbf{0} & 0 & 1 \end{bmatrix} \tag{5}$$

$$\boldsymbol{\Delta}\cdot\boldsymbol{\delta} = \begin{bmatrix} \Delta\mathbf{R}\delta\mathbf{R} & \Delta\mathbf{v} + \Delta\mathbf{R}\delta\mathbf{v} & \Delta\mathbf{p} + \Delta\mathbf{v}\delta t + \Delta\mathbf{R}\delta\mathbf{p} \\ \mathbf{0} & 1 & \Delta t + \delta t \\ \mathbf{0} & 0 & 1 \end{bmatrix} . \tag{6}$$

3) Lie algebra $\mathfrak{d}$ and exponential map: The Lie algebra elements $\boldsymbol{\tau}^\wedge$ and their isomorphic Cartesian $\boldsymbol{\tau}$ have the forms

$$\boldsymbol{\tau}^\wedge = \begin{bmatrix} [\boldsymbol{\theta}]_\times & \boldsymbol{\rho} & \boldsymbol{\upsilon} \\ \mathbf{0} & 0 & \Delta t \\ \mathbf{0} & 0 & 0 \end{bmatrix} \in \mathfrak{d}, \quad \boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\rho} \\ \boldsymbol{\upsilon} \\ \boldsymbol{\theta} \\ \Delta t \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{v}\Delta t \\ \mathbf{a}\Delta t \\ \boldsymbol{\omega}\Delta t \\ \Delta t \end{bmatrix} \in \mathbb{R}^{10}, \tag{7}$$

with $\mathbf{v} \triangleq \dot{\Delta\mathbf{p}}$, $\mathbf{a} \triangleq \dot{\Delta\mathbf{v}}$ and $[\boldsymbol{\omega}]_\times \triangleq \Delta\mathbf{R}^{-1}\dot{\Delta\mathbf{R}}$. Operators $\wedge$ and $\vee$ are defined so that $\boldsymbol{\tau}^\wedge = (\boldsymbol{\tau})^\wedge$ and $\boldsymbol{\tau} = (\boldsymbol{\tau}^\wedge)^\vee$.

The exponential map transfers tangent elements to the group; the logarithmic map is its inverse,

$$\boldsymbol{\Delta} = \mathrm{Exp}(\boldsymbol{\tau}) \triangleq \exp(\boldsymbol{\tau}^\wedge) = \begin{bmatrix} \mathrm{Exp}(\boldsymbol{\theta}) & \mathbf{Q}\boldsymbol{\upsilon} & \mathbf{Q}\boldsymbol{\rho} + \mathbf{P}\boldsymbol{\upsilon}\Delta t \\ \mathbf{0} & 1 & \Delta t \\ \mathbf{0} & 0 & 1 \end{bmatrix} \tag{8}$$

$$\boldsymbol{\tau} = \mathrm{Log}(\boldsymbol{\Delta}) \triangleq \log(\boldsymbol{\Delta})^\vee = \begin{bmatrix} \mathbf{Q}^{-1}(\Delta\mathbf{p} - \mathbf{P}\mathbf{Q}^{-1}\Delta\mathbf{v}\Delta t) \\ \mathbf{Q}^{-1}\Delta\mathbf{v} \\ \mathrm{Log}(\Delta\mathbf{R}) \\ \Delta t \end{bmatrix} \tag{9}$$

where $\mathrm{Log}()$ is obtained by identifying terms in (3) and (8). Matrices $\mathbf{P}$ and $\mathbf{Q}$ are provided in the appendix.

4) Jacobians, uncertainty: For general functions $f : \mathcal{M} \to \mathcal{N}; y = f(x)$, we propagate uncertainty normally via the Jacobians $\mathbf{J}_x^y \triangleq \frac{\partial y}{\partial x}$, *i.e.*, $\boldsymbol{\Sigma}_y = \mathbf{J}_x^y\,\boldsymbol{\Sigma}_x\,\mathbf{J}_x^{y\top}$. These Jacobians map the tangent spaces of the mannifolds $\mathcal{M}, \mathcal{N}$ at x and y, and in case of vector spaces they resort to the classical Jacobian. They also satisfy the chain rule, which we use extensively in our developments. We provide ample reference and justification of this approach in the technical report [19].

A comment is however necessary for the present IMU case. It relates to the uncertainty of the last component of the tangent space (7), which is the time Δt. This component has no uncertainty by definition. Having it in the covariances would imply singularity and result in the risk of a number of well-known numerical issues. We therefore systematically marginalize this time component out of the covariances, simply by removing the last row and column.

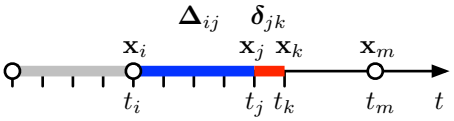

Fig. 3. The pre-integrated delta $\boldsymbol{\Delta}_{ij} \in \mathcal{D}$ contains all motion increments from time i up to time j. The current delta $\boldsymbol{\delta}_{jk} \in \mathcal{D}$ contains the motion from time j to k, computed from the last IMU measurement at time k, so that $\boldsymbol{\Delta}_{ik} = \boldsymbol{\Delta}_{ij} \cdot \boldsymbol{\delta}_{jk}$. Pre-integration is complete when $k = m$.

B. Pre-integrated IMU factors

An IMU factor is created between consecutive keyframes $\mathbf{x}_i$ and $\mathbf{x}_m$ by integrating all IMU data from t_i to t_m (Fig. 3). The factor is pre-integrated during the data acquisition phase, and then used in a second phase to compute the motion residuals at each iteration of the optimization solver. Both phases are described hereafter.

1) IMU pre-integration: The following pipeline of operations is performed to recursively pre-integrate IMU data into a unique measurement.

At the reception of each IMU measurement $\mathbf{y}_k = (\mathbf{a}, \boldsymbol{\omega})_k$, start by correcting it with available bias estimates $\overline{\mathbf{b}}_i = (\mathbf{a}_b, \boldsymbol{\omega}_b)_i$, to produce the tangent vector $\boldsymbol{\tau} = \boldsymbol{\nu}\delta t$. For this, set the velocity part of $\boldsymbol{\nu}$ to zero as the IMU is by definition at zero speed with respect to the moving frame (see comment B.2 in the appendix). Obtain at the same time the respective Jacobians,

$$\boldsymbol{\tau}_k = \begin{bmatrix} \mathbf{0} \\ \mathbf{a} - \mathbf{a}_b \\ \boldsymbol{\omega} - \boldsymbol{\omega}_b \\ 1 \end{bmatrix} \delta t, \quad \mathbf{J}_{\mathbf{y}}^{\boldsymbol{\tau}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ 0 & 0 \end{bmatrix} \delta t, \quad \mathbf{J}_{\mathbf{b}}^{\boldsymbol{\tau}} = -\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \\ 0 & 0 \end{bmatrix} \delta t \tag{10}$$

Second, use the exponential map to obtain the current delta step $\boldsymbol{\delta}_{jk}$ in the group manifold, and obtain Jacobian

$$\boldsymbol{\delta}_{jk} = \mathrm{Exp}(\boldsymbol{\tau}_k) \ , \qquad \mathbf{J}_{\boldsymbol{\tau}}^{\boldsymbol{\delta}} = \mathbf{J}_r(\boldsymbol{\tau}_k) \ . \tag{11}$$

Third, use group composition (6) to update the pre-integrated delta; obtain Jacobians

$$\overline{\boldsymbol{\Delta}}_{ik} = \overline{\boldsymbol{\Delta}}_{ij} \cdot \boldsymbol{\delta}_{jk} \ , \quad \mathbf{J}_{\boldsymbol{\Delta}_{ij}}^{\boldsymbol{\Delta}_{ik}} = \mathbf{Ad}_{\boldsymbol{\delta}_{jk}}^{-1} \ , \quad \mathbf{J}_{\boldsymbol{\delta}_{jk}}^{\boldsymbol{\Delta}_{ik}} = \mathbf{I} \ , \tag{12}$$

where $\mathbf{Ad}_{\boldsymbol{\delta}}$ is the adjoint and $\mathbf{J}_r$ is the right Jacobian —see the appendix and [19] for reference. Fourth, propagate the delta covariance

$$\boldsymbol{\Sigma}_{ik}^{\Delta} = \mathbf{J}_{\boldsymbol{\Delta}_{ij}}^{\boldsymbol{\Delta}_{ik}} \boldsymbol{\Sigma}_{ij} \mathbf{J}_{\boldsymbol{\Delta}_{ij}}^{\boldsymbol{\Delta}_{ik}\top} + \mathbf{J}_{\mathbf{y}}^{\boldsymbol{\Delta}_{ik}} \boldsymbol{\Sigma}_{\mathbf{y}} \mathbf{J}_{\mathbf{y}}^{\boldsymbol{\Delta}_{ik}\top} \ , \tag{13}$$

with $\boldsymbol{\Sigma}_{\mathbf{y}}$ the covariance of the IMU measurements $\mathbf{y}$, and $\mathbf{J}_{\mathbf{y}}^{\boldsymbol{\Delta}_{ik}} = \mathbf{J}_{\boldsymbol{\delta}}^{\boldsymbol{\Delta}_{ik}} \mathbf{J}_{\boldsymbol{\tau}}^{\boldsymbol{\delta}} \mathbf{J}_{\mathbf{y}}^{\boldsymbol{\tau}}$ computed using the chain rule. Finally, integrate the Jacobian of the delta with respect to the biases

$$\mathbf{J}_{\mathbf{b}}^{\boldsymbol{\Delta}_{ik}} = \mathbf{J}_{\boldsymbol{\Delta}_{ij}}^{\boldsymbol{\Delta}_{ik}} \mathbf{J}_{\mathbf{b}}^{\boldsymbol{\Delta}_{ij}} + \mathbf{J}_{\boldsymbol{\delta}_{jk}}^{\boldsymbol{\Delta}_{ik}} \mathbf{J}_{\boldsymbol{\tau}}^{\boldsymbol{\delta}} \mathbf{J}_{\mathbf{b}}^{\boldsymbol{\tau}} \ . \tag{14}$$

Pre-integration starts after each keyframe with $\overline{\boldsymbol{\Delta}}_{ii} = \mathbf{I}$, $\boldsymbol{\Sigma}_{ii}^{\Delta} = \mathbf{0}$ and $\mathbf{J}_{\mathbf{b}}^{\boldsymbol{\Delta}_{ii}} = \mathbf{0}$, using $\overline{\mathbf{b}}_i = \mathbf{b}_i$ the current best estimate of the bias at time i. Pre-integration is complete when $k = m$, which yields $\overline{\boldsymbol{\Delta}}_{im}$, $\boldsymbol{\Sigma}_{im}^{\Delta}$ and $\mathbf{J}_{\mathbf{b}}^{\boldsymbol{\Delta}_{im}}$.

2) IMU factor residual: Computation of the residual is done through the following steps. Use the pre-integrated Jacobian $\mathbf{J}_{\mathbf{b}}^{\boldsymbol{\Delta}_{im}}$ to correct the pre-integrated delta $\overline{\boldsymbol{\Delta}}_{im}$ to account for the new bias estimate $\mathbf{b}_i \neq \overline{\mathbf{b}}_i$,

$$\boldsymbol{\Delta}_{im}(\mathbf{b}_i) = \overline{\boldsymbol{\Delta}}_{im} \cdot \mathrm{Exp}(\mathbf{J}_{\mathbf{b}}^{\boldsymbol{\Delta}_{im}}(\mathbf{b}_i - \overline{\mathbf{b}}_i)) \ . \tag{15}$$

Use (2) as $\boxminus$ to compute the expected delta from $\mathbf{x}_i$ to $\mathbf{x}_m$,

$$\widehat{\boldsymbol{\Delta}}_{im}(\mathbf{x}_i, \mathbf{x}_m) = \mathbf{x}_m \boxminus \mathbf{x}_i \ . \tag{16}$$

Compute the residual in the tangent of $\mathcal{D}$ at $\boldsymbol{\Delta}_{im}$,

$$\mathbf{r}^{\Delta}_{im}(\mathbf{x}_i, \mathbf{x}_m, \mathbf{b}_i) = \mathrm{Log}(\boldsymbol{\Delta}_{im}(\mathbf{b}_i)^{-1} \cdot \widehat{\boldsymbol{\Delta}}_{im}(\mathbf{x}_i, \mathbf{x}_m)) \in \mathbb{R}^9 \ , \tag{17}$$

and drop the Δt part from the residual after the $\mathrm{Log}()$ —see comment in Section III-A.4.

3) Bias drift: A second part of the IMU residual concerns bias drift (see Fig. 1). This is straightforward,

$$\mathbf{r}^{B}_{im} = \mathbf{b}_m - \mathbf{b}_i \in \mathbb{R}^6 \ , \qquad \boldsymbol{\Sigma}^{B}_{im} \in \mathbb{R}^{6\times 6} \ . \tag{18}$$

The complete 15-DoF IMU residual can be put simply as

$$\mathbf{r}^{I}_{im} = \begin{bmatrix} \mathbf{r}^{\Delta}_{im} \\ \mathbf{r}^{B}_{im} \end{bmatrix} \in \mathbb{R}^{15}, \quad \boldsymbol{\Sigma}^{I}_{im} = diag(\boldsymbol{\Sigma}^{\Delta}_{im}, \boldsymbol{\Sigma}^{B}_{im}) \ , \tag{19}$$

where it might be worth noticing, for computational aspects in solving (1), that $\left\|\mathbf{r}^{I}\right\|^2_{\boldsymbol{\Sigma}^{I}} = \left\|\mathbf{r}^{\Delta}\right\|^2_{\boldsymbol{\Sigma}^{\Delta}} + \left\|\mathbf{r}^{B}\right\|^2_{\boldsymbol{\Sigma}^{B}}$.

C. IMU Lie group versus Forster's method

Mathematically, and disregarding methodology, the main difference between our method and Forster's [12] is to be found in the exponential map. To see it, let us consider small rotation increments $\boldsymbol{\theta} = \boldsymbol{\omega}\delta t$ captured at each single IMU sample. In such cases, the matrices $\mathbf{P}, \mathbf{Q}$ appearing in the exponential map (8) and detailed in (31) can be approximated by $\mathbf{P} \approx \frac{1}{2}\mathbf{I}$ and $\mathbf{Q} \approx \mathbf{I}$. The exponential becomes,

$$\mathrm{Exp}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{a} \\ \boldsymbol{\omega} \\ 1 \end{bmatrix}\delta t\right) \approx \begin{bmatrix} \mathrm{Exp}(\boldsymbol{\omega}\delta t) & \mathbf{a}\delta t & \frac{1}{2}\mathbf{a}\delta t^2 \\ \mathbf{0} & 1 & \delta t \\ \mathbf{0} & 0 & 1 \end{bmatrix} \ , \tag{20}$$

where we find the terms $\mathbf{a}\delta t$ and $\frac{1}{2}\mathbf{a}\delta t^2$, which should sound familiar from Forster's method. In effect, with this approximation, if we now compact all the steps (10–12) of our integration into a cumulative expression,

$$\boldsymbol{\Delta}_{ik} = \prod_{j=i+1}^{k} \mathrm{Exp}\left(\begin{bmatrix} \mathbf{0} \\ (\mathbf{a}_j - \mathbf{a}_{bi}) \\ (\boldsymbol{\omega}_j - \boldsymbol{\omega}_{bi}) \\ 1 \end{bmatrix}\delta t\right) \ , \tag{21}$$

it is possible (although tedious) to show that both Forster's and our method are exactly equivalent when $\boldsymbol{\omega}\delta t \to 0$.

IV. Visual Tag Exteroceptive Factors

A. Factor residual

Knowing the intrinsic matrix K of the camera, assuming that the image distortions are corrected and that we know the size of the fiducial markers, the apriltag library [14] provides us with the relative transformation between the camera at time i and the tag n, ${}^{ci}\mathbf{T}_n \in SE(3)$. This measurement can be used in the context of graph-SLAM to define a 6-DoF factor between the key frame IMU pose at time i, ${}^{w}\mathbf{T}_i$ extracted from $\mathbf{x}_i$, and the tag's pose ${}^{w}\mathbf{T}_n$, extracted from $\mathbf{l}_n$, which is used as a landmark and estimated concurrently with the trajectory (see Figs. 1 and 4). A key aspect of the library is that it also provides a unique id for each tag, which solves the otherwise hard problems of feature association and loop closure.

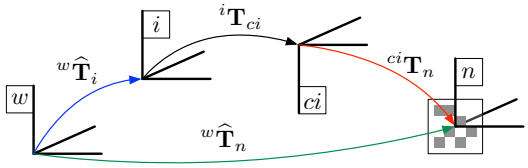

Fig. 4. Kinematic chain of the reference frames involved in one tag observation and their transforms $\mathbf{T} \in SE(3)$. w: world frame; i: IMU frame at time i; ci: camera frame at time i; n: tag n frame. The $SE(3)$ tag measurement is highlighted in red.

The factor's residual is defined in $\mathfrak{se}(3)$ as the discrepancy between the expected relative pose ${}^{ci}\widehat{\mathbf{T}}_n = {}^{i}\mathbf{T}_{ci}^{-1}\,{}^{w}\widehat{\mathbf{T}}_i^{-1}\,{}^{w}\widehat{\mathbf{T}}_n$ and the measurement ${}^{ci}\mathbf{T}_n$ (Fig. 4):

$$\mathbf{r}^{V}_{in}(\mathbf{x}_i, \mathbf{l}_n) = \mathrm{Log}({}^{ci}\mathbf{T}_n^{-1}\,{}^{i}\mathbf{T}_{ci}^{-1}\,{}^{w}\widehat{\mathbf{T}}_i^{-1}\,{}^{w}\widehat{\mathbf{T}}_n) \in \mathbb{R}^6, \tag{22}$$

where ${}^{i}\mathbf{T}_{ci}$ is the IMU-to-camera transform.

B. Factor covariance

We associate a covariance matrix $\boldsymbol{\Sigma}^{V}_{in} \in \mathbb{R}^{6\times 6}$ to this residual. Each camera-tag pose is retrieved by a PnP algorithm [21] on its 4 corners. Thus a natural way to proceed is to consider the effects of pixel noise on the recovered transformation. This we do as follows. Each of the four tag corners ${}^{n}\mathbf{p}_j,\ j \in \{1,2,3,4\}$ in tag frame is projected to the image according to

$$\mathbf{u}_j = ph({}^{ci}\mathbf{T}_n \cdot {}^{n}\mathbf{p}_j), \qquad \mathbf{J}_j = \mathbf{J}^{\mathbf{u}_j}_{{}^{ci}\mathbf{T}_n} \ , \tag{23}$$

where $ph : \mathbb{R}^3 \to \mathbb{R}^2$ is the pinhole projection function and $\mathbf{J}_j \in \mathbb{R}^{2\times 6}$ is the Jacobian with respect to the measured transform, computed according to the Lie theory [19]. We stack the four pixels $\mathbf{u}_j$ into $\mathbf{u} \in \mathbb{R}^8$, and the four Jacobians $\mathbf{J}_j$ of into $\mathbf{J} \in \mathbb{R}^{8\times 6}$. Through covariance propagation, we get the relation between $\boldsymbol{\Sigma}^{V}_{in}$ and $\boldsymbol{\Sigma}_{\mathbf{u}}$

$$\boldsymbol{\Sigma}_{\mathbf{u}} = \mathbf{J}\boldsymbol{\Sigma}^{V}_{in}\mathbf{J}^{\top} \ , \tag{24}$$

which is the inverse of what we want. Since $\mathbf{J}$ is full-column rank, we can compute its pseudo-inverse $\mathbf{J}^{+} = (\mathbf{J}^{\top}\mathbf{J})^{-1}\mathbf{J}^{\top}$ to invert (24) to find $\boldsymbol{\Sigma}^{V}_{in} = \mathbf{J}^{+}\boldsymbol{\Sigma}_{\mathbf{u}}\mathbf{J}^{+\top}$. Then, assuming uncorrelated pixel noises, $\boldsymbol{\Sigma}_{\mathbf{u}} \in \mathbb{R}^{8\times 8}$ is a diagonal matrix with terms equal to $\sigma^2 = n^2$, n being a number of pixels accounting for the pixelization noise and motion blur, as in [15]. We obtain finally,

$$\boldsymbol{\Sigma}^{V}_{in} = n^2(\mathbf{J}^{\top}\mathbf{J})^{-1} \ . \tag{25}$$

We note that, contrary to [15], we do not need to rely on a heuristic to initialize the landmark covariance since the present formulation defines directly a covariance on $SE(3)$.

C. Ambiguity in the pose estimation

During our preliminary tests, we encountered the problem of an ambiguous measured pose: in the presence of pixel noise, the pose estimation of planar tags returned by a PnP algorithm can jump between two close solutions. This problem was addressed in [22] which provides an implementation that retrieves both ambiguous poses, each with its own reprojection error. When expressed in camera frame, both solutions share the tag position and differ only in its orientation. We typically want to select the solution with smaller error. However, if the reprojection errors e_1 and e_2 are too close (we test for $\frac{e_2}{e_1} < h$ with $e_1 \leq e_2$ and h an empirical threshold), there is the risk of selecting the wrong tag orientation, something that would greatly hamper the optimizer. In this case, we increase the rotational part of the covariance matrix by a great factor so that it does not influence the estimation.

D. Related works

Two Apriltag based visual-inertial SLAM systems have been implemented in the previous years. In [23], the authors rely on a EKF in which state propagation in naturally handled by the IMU and each marker detection is used in an update step where the reprojection error of its 4 corners provides a 8D innovation vector. A closer solution to ours was very recently proposed in [15] and is also based on graph SLAM optimization benefiting from Forster's IMU pre-integration from GTSAM. As explained previously, the Apriltag factor formulation is different from ours and the algorithm is tested on large datasets consisting only of smooth motions.

V. Results

A. Experimental setup

We have gathered several datasets in the experimental arena of the humanoid robots at LAAS-CNRS, a 3D environment about $10m \times 5m$ made of flat floor, stairs of various slopes and a 30cm wide beam. The robot environment was augmented with about 20 fiducial "April-tag" markers (about 20 cm width). The tags have been randomly dispatched in the environment. They are fixed during a run, but may vary significantly between two sets of data, and their locations are not calibrated —that is, we do not have ground-truth localization of the tags.

Each dataset is composed of 3 sequences:

- a sequence of RGB images captured at 33 Hz with a synchronized camera
- a sequence of IMU measures captured at 200 Hz
- a sequence of motion-capture (MoCap) measurements used as ground truth.

The visual-inertial sensor (VIS) is comprised of a Memsic IMU running at 200 Hz and an Imagine Source camera. IMU and camera are hardware synchronized: the image acquisition is triggered by a micro-controller (STM32) synchronized with the IMU. We have validated that there is less than 2 ms synchronization error by the hardware (shutter time), that this delay is stable, and it is compensated in the dataset. The camera and the IMU are collocated, with less than 10 cm of distance between IMU and camera focal. Although our implementation of the least-squares estimator is able to calibrate the sensors, we have not tried to calibrate the camera-to-IMU extrinsic parameters. In each sequence, we have taken care that the camera is navigating in a comfortably-dense field of tags, even if it may not have always a tag in its field of view.

The motion-capture data have been obtained from a calibrated 3D marker attached to the camera. MoCap and visual-inertial data are not synchronized at capture time and have been aligned in post-process by maximizing the velocity norm cross-correlation between MoCap and estimated state sequences.

The datasets are available at https://gepgitlab.laas.fr/loco-3d/wolf-data/.

TABLE I
Datasets description and results

Description	Duration	Length	MTE[1]	STE[2]
Handheld loop	62 s	20.7 m	27	10
HRP2 turning then walking	72 s	15.4 m	30	16
HRP2 climbing stairs	53 s	6.5 m	12	6
HRP2 descending stairs	20 s	2.6 m	30	12
HRP2 walking along two loops	226 s	58.14 m	-	-

[1] Mean translation error [mm]
[2] Std. dev. of translation error [mm]

B. Localization precision

We consider five datasets which are summarized in table I. They cover different tasks on which a consistent estimation of the robot movement is necessary. The first one is a relatively long sequence consisting of two loops with the VIS handheld. This is used to test the long term localization of the robot, which is interesting for navigation. Secondly, we made the LAAS Gepetto team HRP2 walk and turn around on a short distance to evaluate the resilience of the filter to the vibrations of the robot. Finally, two more challenging datasets are recorded while the robot is climbing and descending stairs. Especially on the latter, the locomotion causes impacts that on one hand bring the IMU close to its dynamic range saturation, and on the other hand provokes images with motion blur. Finally, we walked our robot around two loops to present a larger dataset during which acquisition MoCap failed unfortunately (Fig. 8). Note that during these experiments, the estimator was not used for feedback control. In order to compare our results with the ground truth, we need to align the trajectories. State estimation using a visual inertial setup has 4 unobservable DoFs since it measures pitch and roll through gravity and therefore only needs to be aligned to the mocap data via a position and yaw transformation. This alignment is done using the library [24], by which some of the presented graphs are produced. Figures 5, 6, 7 and 9 present a comparison of estimation and mocap data position trajectories both with a qualitative visualization and a metric error function of the time. The red and black dots correspond respectively to the beginning and end of the trajectories. For each case, key frames are created at a frequency of 6.6 Hz (every 5

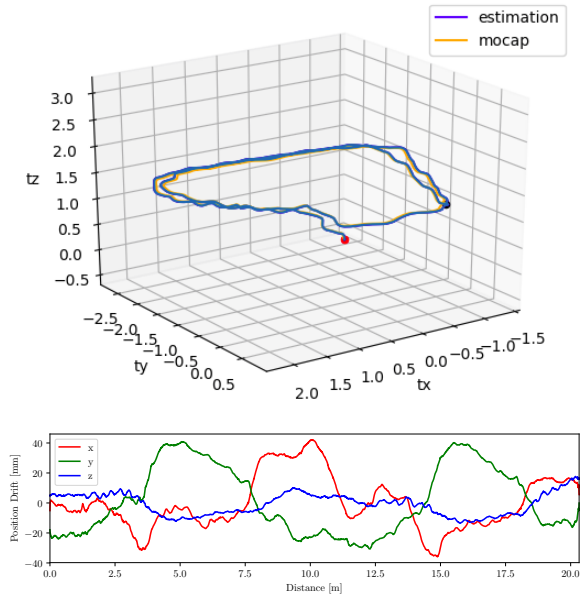

Fig. 5. Handheld loop. Above: estimated trajectory vs mocap. Below: translation error (mm) as a function of time

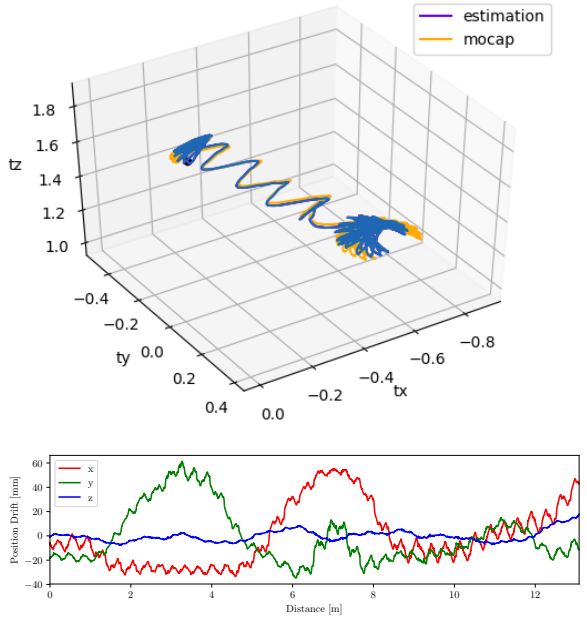

Fig. 6. HRP2 walking on flat ground. Above: estimated trajectory vs mocap. Below: translation error (mm) as a function of time

images) if tags are detected in the corresponding image. In all cases, our estimator achieves errors consistently below a few centimeters. The biggest errors are obtained for the walking datasets in figure 6 where the two humps correspond to phases where the robot is turning on itself which results in a fast rotation of the landmarks with respect to the robot.

C. Velocity estimation

A high rate estimation of a humanoid robot velocity and in particular of its center of mass is critical for balance

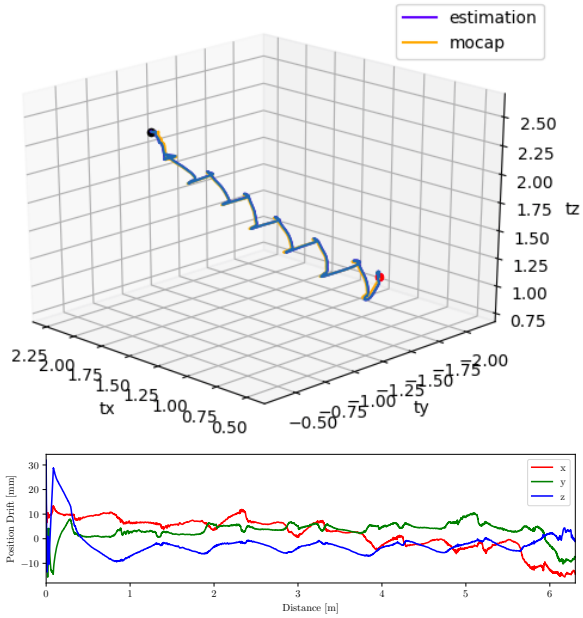

Fig. 7. HRP2 climbing stairs. Above: estimated trajectory vs mocap. Below: translation error (mm) as a function of time

controllers. It can be recovered from motion capture through numerical differentiation of the positions, but this results in a quite noisy time series as it can be observed at the beginning of Fig. 10 (above): the robot is not moving though velocities as high as 50 cm/s can be measured. It is especially visible when hard impacts make the robot shake at approximately 10 Hz. The estimated velocity follows a familiar oscillatory damped system behaviour while the mocap estimation is more erratic. In this sense, our estimator seems to be more fit for feedback control than the use of the MoCap.

VI. Conclusion

In this paper, we have proposed a visual-inertial localization system for a humanoid robot navigating in a structured indoor environment. We have proposed an original theoretical contribution by reformulating the Forster pre-integration using a dedicated Lie group. While the formulation only marginally improves the performance of the state-of-the-art for handling inertial measurements, we believe it brings several improvements, in particular the possibility to easily extend the pre-integration principle to other high-frequency sensors typically available on legged robots. We proposed to integrate the camera using artificial landmarks. Compared to other visual-inertial localization systems also using fiducial markers, we have proposed a practical contribution to handle the ambiguity in the pose estimation of the landmarks. Using artificial landmarks is an interesting solution for humanoid robots navigating indoor in an professional environments, in particular a lab, that can be easily augmented with the markers. Finally, we have proposed an experimental validation, based on 4 new datasets, that are released with the paper. We have demonstrated unprecedented accuracy during a 3D locomotion, with about 1cm error in average for climbing 5 steps of a stair.

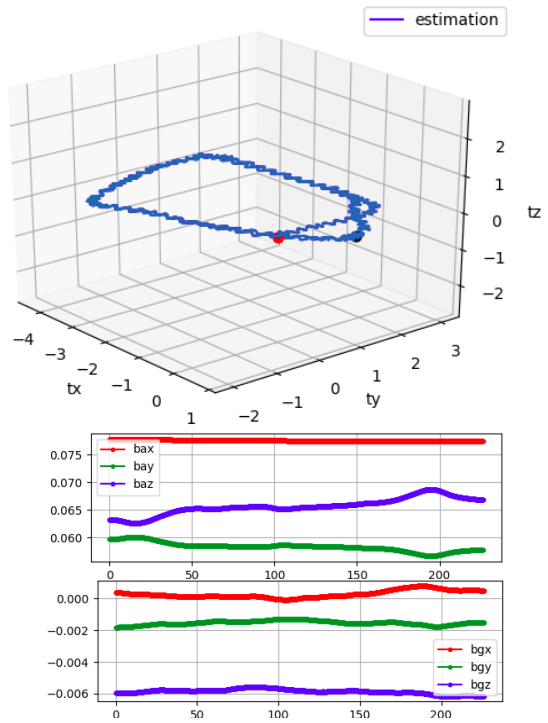

Fig. 8. HRP2 walking, 2 loops. Top: Estimated trajectory, groundtruth missing to MoCap failure. Middle, Bottom: accelerometer and gyroscope biases estimation (respectively in $m.s^{-2}$ and $rad.s^{-1}$). The estimates does not change much along the trajectory which is representative of the relatively low bias drift of the MEMSIC IMU.

While the localization is already satisfactory with respect to the needs of the control, the proposed method is only one step toward the final localization that we aim at. As recently proposed by other teams, we finally want to also fuse the information of contact in the estimator. This should be done following the pre-integration approach to account for the high-frequency of contact information. Thanks to the proposed Lie approach, we should be able to finely handle the available contact information, without the help of a staggered kinematic estimator but by directly merging the raw sensor values. We also hope that the released benchmark will be exploited by the community to challenge our estimator.

APPENDIX
ELEMENTS OF THE IMU DELTA MATRIX LIE GROUP

A. Tangent space and Lie algebra $\mathfrak{d}$

Following [19], the tangent space of $\mathcal{D}$ at the point $\boldsymbol{\Delta}$ is found by taking the time derivative of the group constraint, $\boldsymbol{\Delta}^{-1}\boldsymbol{\Delta} = \mathbf{I}$. Noting $\dot{\bullet} \triangleq \frac{\partial \bullet}{\partial t}$, this yields after a few manipulations

$$\boldsymbol{\Delta}^{-1}\dot{\boldsymbol{\Delta}} = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \Delta\mathbf{R}^\top\mathbf{a} & \Delta\mathbf{R}^\top(\mathbf{v}-\Delta\mathbf{v}) \\ \mathbf{0} & 0 & 1 \\ \mathbf{0} & 0 & 0 \end{bmatrix}, \tag{26}$$

with $\mathbf{v} \triangleq \dot{\Delta\mathbf{p}}$, $\mathbf{a} \triangleq \dot{\Delta\mathbf{v}}$ and $[\boldsymbol{\omega}]_\times \triangleq \Delta\mathbf{R}^\top\dot{\Delta\mathbf{R}}$. The Lie algebra $\mathfrak{d}$ is the tangent space at the identity $\boldsymbol{\Delta} = \mathbf{I}$. Its elements $\boldsymbol{\nu}^\wedge \triangleq \dot{\boldsymbol{\Delta}}|_{\boldsymbol{\Delta}=\mathbf{I}}$ and their isomorphics $\boldsymbol{\nu}$ in Cartesian space are given by,

$$\boldsymbol{\nu}^\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \mathbf{a} & \mathbf{v} \\ \mathbf{0} & 0 & 1 \\ \mathbf{0} & 0 & 0 \end{bmatrix} \in \mathfrak{d} \quad \underset{\wedge}{\overset{\vee}{\rightleftharpoons}} \quad \boldsymbol{\nu} = \begin{bmatrix} \mathbf{v} \\ \mathbf{a} \\ \boldsymbol{\omega} \\ 1 \end{bmatrix} \in \mathbb{R}^{10}. \tag{27}$$

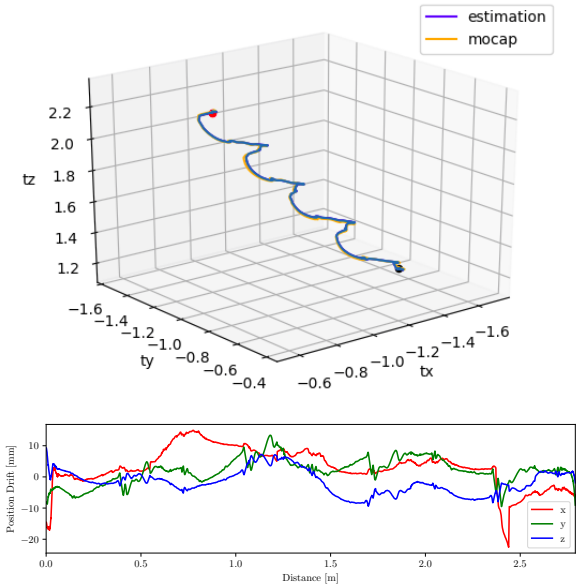

Fig. 9. HRP2 descending the stairs. Above: estimated trajectory vs mocap. Below: translation error (mm) as a function of time

This tangent $\boldsymbol{\nu}^\wedge$ corresponds to the 'velocity' of the group element. Any point in the Lie algebra can be obtained after moving at constant velocity during a period Δt, that is, $\boldsymbol{\tau}^\wedge = \boldsymbol{\nu}^\wedge\Delta t \in \mathfrak{d}$ —see (7).

B. The exponential map

1) The general case: Eq. (26) can be written as $\dot{\boldsymbol{\Delta}} = \boldsymbol{\Delta}\cdot\boldsymbol{\nu}^\wedge$. This is an ordinary differential equation whose integral for constant $\boldsymbol{\nu}$ yields the exponential map [19],

$$\boldsymbol{\Delta}(t) = \exp\left(\boldsymbol{\nu}^\wedge t\right). \tag{28}$$

This gives a direct expression of the integral of information of the type $(\mathbf{v}, \mathbf{a}, \boldsymbol{\omega})$ onto the deltas manifold. Se below for the $(\mathbf{a}, \boldsymbol{\omega})$ case.

The closed form of the exponential map is obtained through Taylor expansion (see *e.g.* [19] for examples). At $t = \Delta t$ we have,

$$\boldsymbol{\Delta}(\Delta t) = \exp(\boldsymbol{\nu}^\wedge\Delta t) \triangleq \sum_n \frac{1}{n!}(\boldsymbol{\nu}^\wedge\Delta t)^n. \tag{29}$$

Exploiting the cyclic pattern of the powers of $[\boldsymbol{\omega}]_\times$, this results in

$$\exp\left(\begin{bmatrix} [\boldsymbol{\omega}]_\times & \mathbf{a} & \mathbf{v} \\ \mathbf{0} & 0 & 1 \\ \mathbf{0} & 0 & 0 \end{bmatrix}\Delta t\right) = \begin{bmatrix} \exp([\boldsymbol{\omega}]_\times\Delta t) & \mathbf{Q}\mathbf{a}\Delta t & \mathbf{Q}\mathbf{v}\Delta t + \mathbf{P}\mathbf{a}\Delta t^2 \\ \mathbf{0} & 1 & \Delta t \\ \mathbf{0} & 0 & 1 \end{bmatrix} \tag{30}$$

with (we skip proofs for space reasons)

$$\mathbf{Q}(\boldsymbol{\theta}) = \mathbf{I} + \frac{1-\cos\theta}{\theta}[\mathbf{u}]_\times + \frac{\theta-\sin\theta}{\theta^2}[\mathbf{u}]_\times^2 \tag{31}$$

$$\mathbf{P}(\boldsymbol{\theta}) = \frac{1}{2}\mathbf{I} + \frac{\theta-\sin\theta}{\theta^2}[\mathbf{u}]_\times + \frac{\cos\theta+\frac{1}{2}\theta-1}{\theta^2}[\mathbf{u}]_\times^2, \tag{32}$$

where $\boldsymbol{\theta} = \boldsymbol{\omega}\Delta t$, $\theta = \|\boldsymbol{\theta}\|$ and $\mathbf{u} = \boldsymbol{\theta}/\theta$ form the angle-axis representation of the rotation step $\boldsymbol{\omega}\Delta t$.

2) The IMU case of $\mathbf{v} = 0$: We defined the IMU deltas as the motion relative to the free-falling frame, which has initial velocity $\mathbf{v}_i$. Thus the tangent velocity $\mathbf{v} = \dot{\Delta\mathbf{p}}$ is zero at the start of the integration step. Since the exponential $\mathrm{Exp}(\boldsymbol{\nu}\Delta t)$ assumes a constant tangent vector $\boldsymbol{\nu} = (\mathbf{v}, \mathbf{a}, \boldsymbol{\omega}, 1)$ during the interval Δt, we have that $\mathbf{v} = 0$ during the full step. This gives immediately

$$\exp\left(\begin{bmatrix} [\boldsymbol{\omega}]_\times & \mathbf{a} & \mathbf{0} \\ \mathbf{0} & 0 & 1 \\ \mathbf{0} & 0 & 0 \end{bmatrix}\Delta t\right) = \begin{bmatrix} \exp([\boldsymbol{\omega}]_\times\Delta t) & \mathbf{Q}\mathbf{a}\Delta t & \mathbf{P}\mathbf{a}\Delta t^2 \\ \mathbf{0} & 1 & \Delta t \\ \mathbf{0} & 0 & 1 \end{bmatrix}. \tag{33}$$

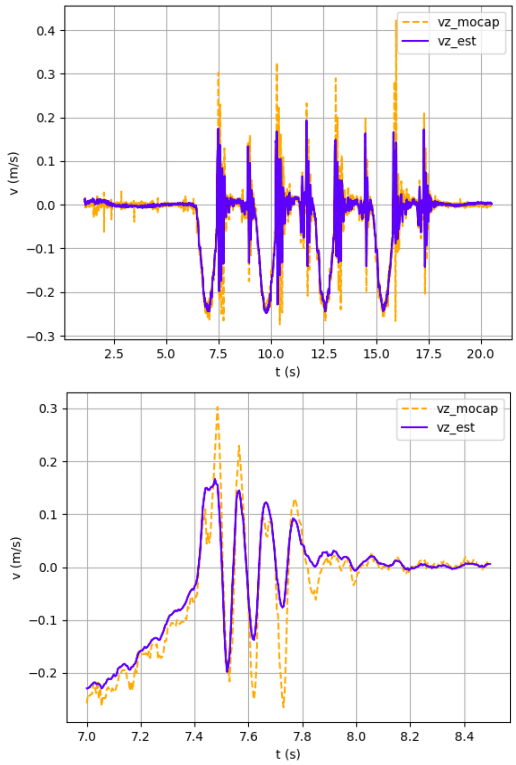

Fig. 10. HRP2 descending stairs. Above: velocity along z axis. Below: close up on one footstep landing showing damped vibration at around 10 Hz.

C. The adjoint and small adjoint matrices

Following the general methodology explained in [19], the adjoint matrix is obtained by identifying the linear terms in $\mathbf{Ad}_{\boldsymbol{\Delta}}\boldsymbol{\tau} = (\boldsymbol{\Delta}\boldsymbol{\tau}^{\wedge}\boldsymbol{\Delta}^{-1})^{\vee}$. We get after long but relatively easy calculations,

$$\mathbf{Ad}_{\boldsymbol{\Delta}} = \begin{bmatrix} \Delta\mathbf{R} & -\Delta\mathbf{R}\Delta t & [\Delta\mathbf{p}-\Delta\mathbf{v}\Delta t]_{\times}\Delta\mathbf{R} & \Delta\mathbf{v} \\ \mathbf{0} & \Delta\mathbf{R} & [\Delta\mathbf{v}]_{\times}\Delta\mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Delta\mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 1 \end{bmatrix} \in \mathbb{R}^{10\times 10} . \quad (34)$$

Similarly, from [25] the small adjoint matrix can be computed by identifying the linear terms in $\mathbf{ad}_{\boldsymbol{\tau}}\boldsymbol{\sigma} = (\boldsymbol{\tau}^{\wedge}\boldsymbol{\sigma}^{\wedge} - \boldsymbol{\sigma}^{\wedge}\boldsymbol{\tau}^{\wedge})^{\vee}$ which for $\boldsymbol{\tau} = (\boldsymbol{\rho}, \boldsymbol{\upsilon}, \boldsymbol{\theta}, \Delta t) \in \mathfrak{d}$ yields,

$$\mathbf{ad}_{\boldsymbol{\tau}} = \begin{bmatrix} [\boldsymbol{\theta}]_{\times} & -\mathbf{I}\Delta t & [\boldsymbol{\rho}]_{\times} & \boldsymbol{\upsilon} \\ \mathbf{0} & [\boldsymbol{\theta}]_{\times} & [\boldsymbol{\upsilon}]_{\times} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & [\boldsymbol{\theta}]_{\times} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 \end{bmatrix} \in \mathbb{R}^{10\times 10} . \quad (35)$$

D. The right Jacobian

The right Jacobian $\mathbf{J}_r$ is the Jacobian of $\mathrm{Exp}()$ as described in [19]. Lacking at the moment a closed form for it, we take the general methodology for the left Jacobian described in [25], and transform it to the right using $\mathbf{J}_r(\boldsymbol{\tau}) = \mathbf{J}_l(-\boldsymbol{\tau})$ [19],

$$\mathbf{J}_r(\boldsymbol{\tau}) = \mathbf{J}_l(-\boldsymbol{\tau}) = \sum_i \frac{\mathbf{ad}_{-\boldsymbol{\tau}}{}^i}{(i+1)!} = \sum_i \frac{(-\mathbf{ad}_{\boldsymbol{\tau}})^i}{(i+1)!} . \quad (36)$$

This sum can be truncated at the desired degree of accuracy.

References

[1] P.-C. Lin, H. Komsuoglu, and D. 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.

[2] 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," 2013.

[3] 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*, 2014.

[4] T. Flayols, A. Del Prete, P. Wensing, A. Mifsud, M. Benallegue, and O. Stasse, "Experimental evaluation of simple estimators for humanoid robots," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, 2017.

[5] G. Bledt, P. M. Wensing, S. Ingersoll, and S. Kim, "Contact model fusion for event-based locomotion in unstructured terrains," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[6] N. Rotella, S. Schaal, and L. Righetti, "Unsupervised contact learning for humanoid estimation and control," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018.

[7] S. Nobili, M. Camurri, V. Barasuol, M. Focchi, D. Caldwell, C. Semini, and M. Fallon, "Heterogeneous sensor fusion for accurate state estimation of dynamic legged robots," in *Robotics: Science and Systems Foundation*, 2017.

[8] M. Fallon, "Accurate and robust localization for walking robots fusing kinematics, inertial, vision and LIDAR," *Interface Focus*, Jun. 2018.

[9] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 3281–3288.

[10] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[11] 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 International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[12] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2017.

[13] D. Wisth, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *arXiv preprint arXiv:1904.03048*, 2019.

[14] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016.

[15] G. He, S. Zhong, and J. Guo, "A lightweight and scalable visual-inertial motion capture system using fiducial markers," *Autonomous Robots*, pp. 1–21, 2019.

[16] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org.

[17] T. Lupton and S. Sukkarieh, "Efficient integration of inertial observations into visual SLAM without initialization," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2009.

[18] J. Deray, J. Andrade-Cetto, and J. Solà, "Joint on-manifold self-calibration of odometry model and sensor extrinsics using pre-integration," in *European Conference on Mobile Robots*, 2019.

[19] J. Solà, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," Institut de Robòtica i Informàtica Industrial, Barcelona, Tech. Rep. IRI-TR-18-01, 2018.

[20] D. Atchuthan, A. Santamaria-Navarro, N. Mansard, O. Stasse, and J. Solà, "Odometry based on auto-calibrating inertial measurement unit attached to the feet," in *2018 European Control Conference (ECC)*, June 2018, pp. 3031–3037.

[21] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o(n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, p. 155, Jul 2008. [Online]. Available: https://doi.org/10.1007/s11263-008-0152-6

[22] T. Collins and A. Bartoli, "Infinitesimal plane-based pose estimation," *International Journal of Computer Vision*, vol. 109, no. 3, pp. 252–286, 2014.

[23] M. Neunert, M. Bloesch, and J. Buchli, "An open source, fiducial based, visual-inertial motion capture system," in *2016 19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 1523–1530.

[24] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry."

[25] E. Eade, "Derivative of the exponential map," Tech. Rep., 2018.