

VALINOR: a Lightweight Leg Inertial Odometry for Humanoid Robots

Arnaud Demont*¹, Mehdi Benallegue¹, Thomas Duvinage¹, and Abdelaziz Benallegue²

Abstract: This article presents VALINOR (Velocity-Aided Leg Inertial Nonlinear Odometry and Registration), a method for lightweight yet accurate Leg-Inertial odometry for humanoid robots addressing the challenge of lightweight yet accurate and certifiable state estimation. VALINOR associates Leg odometry with builds upon the Tilt Observer, a computationally efficient complementary filter, which provides accurate estimates of complementary filter that estimates the IMU's tilt and linear velocity with strong mathematical convergence guarantees, associating it with Leg odometry. We introduce an axis-agnostic method for the fusion of the Leg odometry's yaw with the fusion method that coherently combines the Tilt Observer's tilt estimate with the Leg odometry's yaw. We argue that this method is less arbitrary and more mathematically sound than those based on other orientation representations, especially on Euler angles. We present an evaluation of the proposed estimator through validate VALINOR on real-world data on experiments with two humanoid robots. We Results show that, while being 7.5 times faster than the state-of-the-art method used for comparison, VALINOR improves tilt estimation by over 25%, making it a well-suited feedback for balance and walking controllers, achieves estimation performance on par with it. This makes VALINOR a strong candidate for feedback in balance control, as well as for task planning and execution within the robot's local workspace. Beyond this contribution, we hope to highlight the broader need for estimation methods that are not only accurate and efficient, but also grounded in strong mathematical guarantees, towards their safe deployment around humans.

Keywords: Lightweight state estimation, Humanoid robots, Proprioceptive odometry, Tilt estimation, Balancing.

1. INTRODUCTION

Humanoid robot navigation in real-world environments remains a core challenge in robotics. For robots to be deployed at scale in society and industry, whether in healthcare facilities or warehouses, they must operate robustly in dynamic, unstructured, and human-centric environments [1]. This requires strong guarantees on their ability to plan motion, and execute actions in real time. In comparison to other legged robots, the control of humanoids is particularly particularly difficult due to their complex underactuated dynamics controlled through contact with the environment [2]. Their control must therefore be as theoretically grounded as possible, in order to ensure safe, predictable, and certifiable behavior [3].

In the meantime, methods for motion generation continue to gain in complexity and capability. Notably, Model Predictive Control (MPC) [4–6] [4,6], reinforcement learning-based controllers [7, 8], and foundation models [9,10] [10,11] are becoming more prevalent due to their versatility and to their ability to handle long-horizon

planning, complex dynamics, and to their versatility. However, these advances come at the cost of increased computational complexity, running Running them in real-time on resource-limited platforms like embedded systems thus becomes challenging [11–14]. Since the control part is getting heavier, one practical solution is to reduce the computational load of other parts of the pipeline. The state estimation part is one of the cornerstones of the pipeline, since other components depend on its output and often have to wait for it before executing, and thus cannot run in parallel. It is therefore especially critical to try and reduce the computation time of the state estimation part.

State estimation for legged robots has also known notable improvements in the past years, the main trend aiming towards improving its global accuracy by incorporating more and more advanced sensors, especially exteroceptive ones, such as cameras, LiDARs, etc. [15–17] [15,17]. Such methods show remarkable accuracies. However, they also tend to be more and more computationally expensive (e.g. the order of the millisecond for

Arnaud Demont is with the CNRS-AIST JRL (Joint Robotics Laboratory), IRL, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8560 Japan, Université Paris-Saclay, 91190 Gif-sur-Yvette, France, and Laboratoire d'Ingénierie des Systèmes de Versailles, 78140 Vélizy, France (email: arnaud.demont@aist.go.jp). Mehdi Benallegue and Thomas Duvinage are with the CNRS-AIST JRL (Joint Robotics Laboratory), IRL, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8560 Japan (emails: mehdi.benallegue@aist.go.jp, thomas.duvinage@aist.go.jp). Abdelaziz Benallegue is with the CNRS-AIST JRL (Joint Robotics Laboratory), IRL, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8560 Japan, and Laboratoire d'Ingénierie des Systèmes de Versailles, 78140 Vélizy, France (email: abdelaziz.benallegue@uvsq.fr).

* Corresponding author.

[17]).

For this reason, there is still a need for lightweight state estimation methods, which rather than adding new sensors, focus on improving the estimation of key variables at high frequency using the proprioceptive sensors readily available on the robot. The most common sensors for humanoids are joint encoders, inertial measurement units (IMUs) and contact force sensors ~~such as FSR or force/torque sensors~~. IMUs are able to measure angular velocities and linear accelerations, including the gravitational one. It can be used to estimate the vertical direction, which we call *tilt*. ~~This is done~~, often under the assumption that accelerations are negligible compared to the gravitational acceleration (e.g. [18]). Unfortunately, this assumption cannot hold in the case of humanoids, especially with impacts and dynamic motions, which significantly degrades the estimation. IMUs signals can also be integrated to provide estimations of the positions and orientation in which is called inertial odometry, but if done for extended periods of time this would produce fast drifts because of noise and initial estimation errors.

Both issues, high accelerations and odometry drifts can be resolved thanks to a single consideration: legged robots are connected to a stable environment through their contacts. ~~Meaning that we consider contacts to be By modeling the contacts as~~ fixed points in the environment, ~~which provides a partial measurement~~ ~~partial measurements~~ of the robot's kinematics in the world ~~using~~ can be obtained from joint encoders. This consideration ~~only alone~~ makes the tilt fully observable, however dynamic the motion is [19]. The anchor with the environment and the kinematic data can be used also to perform legged odometry (also called "Leg" odometry) using the transformations between successive contacts to rebuild the world-trajectory [20]. This technique can be merged with IMU data and correct for the drift ~~[21–23]~~ [21, 23]. This is called leg-inertial odometry.

Violations of slippage-free contact assumption can degrade the odometry performance ~~[24, 25]~~. Work has thus been done to address the problem of contact slippage, for example by discarding unreliable contacts ~~[24–26]~~ [25, 26], or by attempting to correct the contact reference position [21, 27, 28]. Despite that, the position and yaw orientation of the robot in the world frame remain unobservable [21]. Nevertheless, the methods exploiting these contacts have proven to be more than enough in concrete use-cases when the pose needs to be reliable only locally, like for remotely controlled robots [29] or when task planning is also made in the robot's local environment [30], which means that drifts due to slippage remain limited for these use-cases.

In this work we propose VALINOR (Velocity-Aided Leg Inertial Nonlinear Odometry and Registration), a method for Leg-Inertial odometry for legged robots, which relies on a highly accurate tilt estimate provided by a nonlin-

ear complementary filter [19]. This complementary filter allows for a much faster computation than the conventionally used Kalman Filter (e.g. [27]), and offers mathematical convergence guarantees on the tilt and linear velocity estimate, aligning with our objective of lightweight yet accurate and certifiable state estimation. The main contributions introduced in this paper are as follows:

- The proposed ~~state~~ estimator is 7.5 times faster than the ~~start-of-the-art state-of-the-art~~ method for proprioceptive odometry, while ~~displaying on par odometry accuracy, and even improving significantly tilt estimation achieving odometry accuracy on par with it~~.
- We present the axis-agnostic tilt and yaw fusion, which we show to be more mathematically sound than that based on Euler angle representations.
- We evaluate the proposed method on real experiments with two different humanoid robots.
- ~~The code of the Tilt Observer is open source¹, as well as We provide a ROS wrapper¹ for the open-source Tilt Observer².~~

2. GENERAL NOTATIONS

- The general notation for kinematic variables is ${}^1\bigcirc_2$, expressing the kinematics of ~~the a~~ frame 2 in ~~the a~~ frame 1. To simplify ~~the~~ notation, kinematics in the world frame are written without the \mathcal{W} symbol: ${}^{\mathcal{W}}\bigcirc_2 = \bigcirc_2$.
- ~~We note \hat{x} the estimate of the variable x .~~
- The world frame and the robot's IMU frame are denoted \mathcal{W} and \mathcal{I} , respectively. The frame associated with the i -th contact is denoted \mathcal{C}_i . The anchor point, defined in Section 3, is denoted \mathcal{A} .
- \times is the cross-product operator.
- ~~We define n_c the number of current contacts set with the environment.~~

3. DEFINITION OF THE ANCHOR POINT

We define here the notion of anchor point, since it will be used extensively in the following sections. ~~The position of the anchor point can be merged with the robot's joint encoders and IMU signals to provide an algebraic estimate of the IMU velocity in the world frame. This will be It is notably essential for the design of the Tilt Observer, as explained in Section 4.2 where we show that it is essential for the estimator design.~~

The anchor point, denoted \mathcal{A} , is defined as a point attached to the robot ~~and which that~~ is considered to have zero ~~instantaneous~~ linear velocity in the world frame. This point does not need to be fixed in the

¹

¹https://github.com/ArnaudDmt/state_observation_ros.git

²https://github.com/ArnaudDmt/state_observation

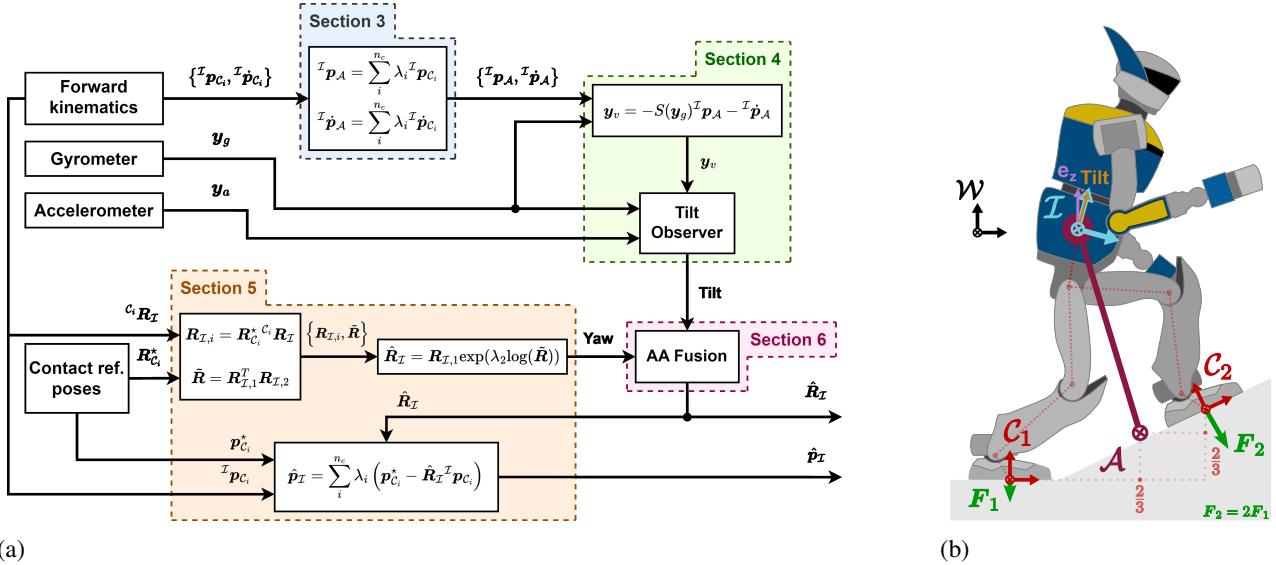


Fig. 1. (a) Summary of VALINOR's pipeline. (b) Illustration of the anchor point position computation and of the reference frames used in VALINOR. \mathcal{W} : world frame; \mathcal{I} : IMU frame; \mathcal{C}_i : frame of the i -th contact. For clarity, only the weighted average of the contact positions is shown. In this example, $\mathbf{F}_2 = 2\mathbf{F}_1$ and so $\lambda_2 = \frac{2}{3}$.

robot frame as well, the only requirement is that its instantaneous position and linear velocity in the frame of the relative to the robot's floating base are known.

Contacts with the environment provide naturally good sets of points known to the robot and having zero world velocity. With the no-slip hypothesis, any point of a contact body would thus be suitable to define an anchor point. However, we need an anchor point that remains always defined and whose position is as continuous as possible, even when contacts are created and broken, as long as there is at least one contact established. Also, the no-slip hypothesis can hold better for some points than others. For instance, when the contact is more likely to slip when contact forces are closer to Coulomb's friction cones, the contact is more likely to slip.

To address these two requirements we determine the kinematics of the anchor point through a weighted average of the kinematics of the n_c current contacts, as shown in Figure 1b. The weighting coefficients are defined such that weaker contacts, which are prone to violate Coulomb's inequality and thus to slip, contribute less to the anchor point's kinematics computation. Given m_{g0} the weight of the robot, we first compute

$$u_i = \frac{\mathbf{F}_{i,z}}{\sqrt{\mathbf{F}_{i,x}^2 + \mathbf{F}_{i,y}^2 + \varepsilon m_{g0}}},$$

the ratio for each contact i the ratio u_i of the normal force to the tangential force, measured by the force / torque sensor at the contact i . The term ε , arbitrarily small, allows to deal with the case where the tangential force is zero.

The contact's The weighting coefficient λ_i is then: of the contact in the average is then computed:

$$u_i = \frac{\mathbf{F}_{i,z}}{\sqrt{\mathbf{F}_{i,x}^2 + \mathbf{F}_{i,y}^2 + \varepsilon m_{g0}}}, \quad \lambda_i = \frac{u_i}{\sum_{j=1}^{n_c} u_i}. \quad (1)$$

Based on this definition, we give the position and linear velocity of the anchor point in the IMU's frame:

$$\begin{aligned} \underline{\tau \mathbf{p}_{\mathcal{A}}} &= \sum_i^{n_c} \lambda_i \underline{\tau \mathbf{p}_{C_i}}, \\ \underline{\tau \dot{\mathbf{p}}_{\mathcal{A}}} &= \sum_i^{n_c} \lambda_i \underline{\tau \dot{\mathbf{p}}_{C_i}}, \end{aligned}$$

$$\underline{\tau \mathbf{p}_{\mathcal{A}}} = \sum_i^{n_c} \lambda_i \underline{\tau \mathbf{p}_{C_i}}, \quad \underline{\tau \dot{\mathbf{p}}_{\mathcal{A}}} = \sum_i^{n_c} \lambda_i \underline{\tau \dot{\mathbf{p}}_{C_i}}, \quad (2)$$

with $\tau \mathbf{p}_{C_i}$ the position and $\tau \dot{\mathbf{p}}_{C_i}$ the linear velocity of the i -th contact in the IMU's frame, which are directly provided by the robot's joint encoders and geometrical model.

$\underline{\tau \mathbf{p}_{\mathcal{A}}}$ and $\underline{\tau \dot{\mathbf{p}}_{\mathcal{A}}}$ are updated at each iteration. As explained in Section 4.2, by combining them with the zero-velocity assumption of the anchor point in the world frame and the gyrometer measurement, we obtain an algebraic estimate of the IMU velocity in the world frame.

It is important to note also that with this definition, although the anchor point is defined constrained to have zero velocity in the world, its instantaneous position may still evolve vary over time.

-0.5pc—Illustration of the anchor point position computation and of the reference frames used in VALINOR. \mathcal{W} : world frame; \mathcal{I} : IMU frame; \mathcal{C}_i : frame of the i -th contact. For clarity, only the weighted average of the contact positions is shown. In this example, $\mathbf{F}_2 = 2\mathbf{F}_1$ and so $\lambda_2 = \frac{2}{3} \cdot 1.5\text{pc}$.

4. TILT OBSERVER WITH PROOF OF CONVERGENCE

The proposed estimator VALINOR relies on a highly accurate estimate of the IMU's tilt provided by a complementary filter, introduced in [19], which we will call *Tilt Observer*.

4.1. Definition of the State Variables

The Tilt Observer is able to provide estimates of the following two variables: provides estimates of:

- $\mathbf{v}_{\mathcal{I},l} \triangleq \mathbf{R}_{\mathcal{I}}^T \mathbf{v}_{\mathcal{I}}$ the linear velocity of the IMU's frame in the world frame, expressed in the frame of the IMU.
- $\mathbf{R}_{\mathcal{I}}^T \mathbf{e}_z$ the tilt of the IMU.

We thus define the state variables:

$$\mathbf{x}_1 \triangleq \mathbf{v}_{\mathcal{I},l}, \quad \mathbf{x}_1 \in \mathbb{R}^3, \quad (3)$$

$$\mathbf{x}_2 \triangleq \mathbf{R}_{\mathcal{I}}^T \mathbf{e}_z, \quad \mathbf{x}_2 \in \mathbb{S}^2. \quad (4)$$

The set $\mathbb{S}^2 \subset \mathbb{R}^3$ is the unit sphere centered at the origin, and defined as

$$\mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x}\| = 1\}$$

$$\mathbb{S}^2 = \{\mathbf{x} \in \mathbb{R}^3 \mid \|\mathbf{x}\| = 1\}.$$

4.2. Definition of the Measurements

The measurements required by the Tilt Observer are:

- \mathbf{y}_g the signal of the IMU's gyrometer.
- \mathbf{y}_a the signal of the IMU's accelerometer.
- \mathbf{y}_v a measurement of $\mathbf{v}_{\mathcal{I},l}$.

Since no sensor provides a direct measurement of $\mathbf{v}_{\mathcal{I},l}$, we obtain \mathbf{y}_v from an algebraic combination of gyrometer signal \mathbf{y}_g and the position and velocity of the anchor point ${}^{\mathcal{I}}\mathbf{p}_{\mathcal{A}}$ from (??) and ${}^{\mathcal{I}}\dot{\mathbf{p}}_{\mathcal{A}}$ from (??) respectively, giving in the IMU's frame given by Eq. (2):

$$\mathbf{y}_v = -\mathbf{y}_g \times {}^{\mathcal{I}}\mathbf{p}_{\mathcal{A}} - {}^{\mathcal{I}}\dot{\mathbf{p}}_{\mathcal{A}} \quad (5)$$

4.3. Definition of the Filter

The state dynamics of our the system can be written as:

$$\dot{\mathbf{x}}_1 = -\mathbf{y}_g \times \mathbf{x}_1 - g_0 \mathbf{x}_2 + \mathbf{y}_a, \quad (6)$$

$$\dot{\mathbf{x}}_2 = -\mathbf{y}_g \times \mathbf{x}_2, \quad (7)$$

with g_0 the gravitational acceleration constant. We write a complementary filter which uses A complementary filter can then be defined, using the system's dynamics as a feed-forward, and corrects correcting the estimated state using the velocity measurement \mathbf{y}_v :

$$\begin{cases} \dot{\hat{\mathbf{x}}}_1 = -\mathbf{y}_g \times \hat{\mathbf{x}}_1 - g_0 \hat{\mathbf{x}}_2' + \mathbf{y}_a + \alpha_1 (\mathbf{y}_v - \hat{\mathbf{x}}_1), \\ \dot{\hat{\mathbf{x}}}_2' = -\mathbf{y}_g \times \hat{\mathbf{x}}_2 - \frac{\alpha_2}{g_0} (\mathbf{y}_v - \hat{\mathbf{x}}_1), \\ \dot{\hat{\mathbf{x}}}_2 = -(\mathbf{y}_g - \gamma \hat{\mathbf{x}}_2 \times \hat{\mathbf{x}}_2') \times \hat{\mathbf{x}}_2. \end{cases}$$

$$\begin{cases} \dot{\hat{\mathbf{x}}}_1 = -\mathbf{y}_g \times \hat{\mathbf{x}}_1 - g_0 \hat{\mathbf{x}}_2' + \mathbf{y}_a + \alpha_1 (\mathbf{y}_v - \hat{\mathbf{x}}_1), \\ \dot{\hat{\mathbf{x}}}_2' = -\mathbf{y}_g \times \hat{\mathbf{x}}_2 - \frac{\alpha_2}{g_0} (\mathbf{y}_v - \hat{\mathbf{x}}_1), \\ \dot{\hat{\mathbf{x}}}_2 = -(\mathbf{y}_g - \gamma \hat{\mathbf{x}}_2 \times \hat{\mathbf{x}}_2') \times \hat{\mathbf{x}}_2. \end{cases} \quad (8)$$

$$\begin{cases} \dot{\hat{\mathbf{x}}}_1 = -\mathbf{y}_g \times \hat{\mathbf{x}}_1 - g_0 \hat{\mathbf{x}}_2' + \mathbf{y}_a + \alpha_1 (\mathbf{y}_v - \hat{\mathbf{x}}_1), \\ \dot{\hat{\mathbf{x}}}_2' = -\mathbf{y}_g \times \hat{\mathbf{x}}_2 - \frac{\alpha_2}{g_0} (\mathbf{y}_v - \hat{\mathbf{x}}_1), \\ \dot{\hat{\mathbf{x}}}_2 = -(\mathbf{y}_g - \gamma \hat{\mathbf{x}}_2 \times \hat{\mathbf{x}}_2') \times \hat{\mathbf{x}}_2. \end{cases} \quad (9)$$

$$\begin{cases} \dot{\hat{\mathbf{x}}}_1 = -\mathbf{y}_g \times \hat{\mathbf{x}}_1 - g_0 \hat{\mathbf{x}}_2' + \mathbf{y}_a + \alpha_1 (\mathbf{y}_v - \hat{\mathbf{x}}_1), \\ \dot{\hat{\mathbf{x}}}_2' = -\mathbf{y}_g \times \hat{\mathbf{x}}_2 - \frac{\alpha_2}{g_0} (\mathbf{y}_v - \hat{\mathbf{x}}_1), \\ \dot{\hat{\mathbf{x}}}_2 = -(\mathbf{y}_g - \gamma \hat{\mathbf{x}}_2 \times \hat{\mathbf{x}}_2') \times \hat{\mathbf{x}}_2. \end{cases} \quad (10)$$

$\hat{\mathbf{x}}_1$ is the estimate of \mathbf{x}_1 and $\hat{\mathbf{x}}_2'$ is an intermediate estimate of \mathbf{x}_2 , which converges exponentially, but is not restricted to lie on the sphere \mathbb{S}^2 . This estimate drives the convergence of the tilt's final estimate $\hat{\mathbf{x}}_2$, which is explicitly projected onto \mathbb{S}^2 so as to respect its Lie group structure. The positive scalar gains α_1 , α_2 and γ are positive scalar gains. $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$ are estimates of \mathbf{x}_1 and \mathbf{x}_2 , respectively define the convergence rates of the three state estimates. Insights on their tuning are provided in [31]. Note that the filter's convergence guarantees hold regardless of the gains' tuning.

4.4. Advantages of the Tilt Observer

The use of Using a complementary filter for the Tilt Observer presents notable strengths in comparison to other methods like the commonly used offers key advantages over common approaches such as the Kalman Filter. First, it allows us to easily work naturally operates in the frequency domain when determining the gains. This is particularly, which is particularly suitable for our model since the assumption of fixed contacts used to obtain the velocity measurement \mathbf{y}_v is more valid in low frequency than in high frequency. In (6), we thus use the IMU measurements for at low frequencies. In Eq. (8), IMU measurements thus provide the high frequency variation feed-forward of $\hat{\mathbf{x}}_1$, and while \mathbf{y}_v for ensures its low frequency variation correction. Second, one iteration of the filter is computationally extremely cheap since it only consists in computing three equations, it also doesn't involve any which involve no matrix multiplication or inversion and is thus computationally extremely cheap, as will be shown in Section 7.2.1. Finally, the formulation as a complementary filter allows to conduct a convergence analysis of the estimation error, providing strong mathematical guarantees on the estimator's performances. Especially here, it has been shown in [19] that:

- The dynamics of the estimation error is autonomous, and thus does not depend on the motion of the robot.
- The intermediate estimator $\{\hat{x}_1, \hat{x}'_2\}$ is *globally exponentially stable*, with respect to the origin $(0, 0)$.
- The full estimator is *almost globally asymptotically stable*, and locally *exponentially stable*. It provides a good filtering of noise and guarantees to respect the normality constraint.

5. LEG ODOMETRY FOR POSITION AND YAW ESTIMATION

While the tilt of the IMU's frame in the world frame is estimated by the Tilt Observer, its position and yaw are obtained using Leg odometry. Once a contact i is created, its *initial pose* $\{p_{C_i}^*, R_{C_i}^*\}$ in the world is obtained by *via* forward kinematics from the *current IMU estimated IMU frame pose* $\{\hat{p}_{\mathcal{I}}, \hat{R}_{\mathcal{I}}\}$:

$$R_{C_i}^* = \hat{R}_{\mathcal{I}}^T R_{C_i}, \quad (11)$$

$$p_{C_i}^* = \hat{p}_{\mathcal{I}} + \hat{R}_{\mathcal{I}}^T p_{C_i}, \quad (12)$$

with $\{^T p_{C_i}, ^T R_{C_i}\}$ the pose of the i -th contact in the IMU's frame *pose and provided by* the robot's *geometrical model and joint encoders*. We call it the *The contact's reference pose* $\{p_{C_i}^*, R_{C_i}^*\}$, which we consider *reference pose is considered* constant to enforce *the contact's anchoring role*. This pose is then its anchoring role, and is used to recover the pose of the IMU's frame in the world frame. With the proposed pipeline, we thus leverage both the accuracy and mathematical guarantees provided by the Tilt Observer, and the robustness to drift provided by the Leg odometry. Similarly to the computation of the anchor point in Section 3, the contribution of the contacts to the Leg-leg odometry is weighted to *trust more contacts which are the least favor those less* prone to slippage, *in order to mitigate* its effect.

To obtain the yaw. First, we estimate the IMU's orientation from contact information. For the two most reliable contacts³, we compute the weighted average between the reference orientation of the two most reliable contacts⁴ IMU frame's orientation from the contact's reference orientation. For each contact i , we write:

$$R_{\mathcal{I},i} = R_{C_i}^T R_{\mathcal{I}}, \quad (13)$$

with $R_{\mathcal{I},i}$ the orientation of the IMU's frame in the contact frame, provided by the robot's geometrical model and joint encoders. We then compute the average $R_{\mathcal{I},\text{avg}}$ between both contacts' error between the obtained orientations, which is used to compute their weighted

³The two contacts with the highest ratio u defined in Eq. (1).

⁴The two contacts with the highest ratio u_i defined in (1).

average, using the formalism defined by SO(3) the Lie group of rotation matrices:

$$\tilde{R} = R_{\mathcal{I},1}^T R_{\mathcal{I},2} \quad (14)$$

$$\underline{\underline{R}}_{\mathcal{I},\text{avg}} = R_{\mathcal{I},1} \exp(\lambda_2 \log(\tilde{R})). \quad (15)$$

exp and log are the *exponential and logarithm maps of SO(3)*. We note *exponential and logarithm maps*. Note that for small angles, we can use the approximation:

$$\log(\tilde{R}) \simeq \frac{1}{2} (\tilde{R} - \tilde{R}^T),$$

Once the average orientation $\log(\tilde{R}) \simeq \frac{1}{2} (\tilde{R} - \tilde{R}^T)$.

Once the IMU's orientation $\hat{R}_{\mathcal{I}}$ has been computed, we merge the corresponding yaw with the tilt estimated by the Tilt Observer using a new axis-agnostic fusion of tilt and yaw, that we describe in Section 6, which preserves the estimated tilt and its mathematical convergence guarantees with little computations.

The position of the IMU's frame in the world frame is then obtained from the The obtained orientation is then used to compute the IMU frame's position, together with the n_c contact reference poses:

$$\hat{p}_{\mathcal{I}} = \sum_i^{n_c} \lambda_i (p_{C_i}^* - \hat{R}_{\mathcal{I}}^T p_{C_i}). \quad (16)$$

We note that this position is obtained using the IMU's orientation estimate that merges the estimated tilt with the yaw coming from the odometry. It takes then Doing so, our position estimate thus takes full profit from the accurate tilt provided by the Tilt Observer.

6. TILT-YAW FUSION

The Tilt Observer of Section 4 provides \hat{x}_2 , an accurate estimate of the IMU tilt noted $\ell = R_1^T e_z$, while the contact-based Leg odometry of Section 5 supplies an orientation $R_1, R_2 = \hat{R}_{\mathcal{I}}$, whose yaw reflects the robot's heading. We now merge these two pieces into a single rotation R , which would preserve our tilt estimate, especially its mathematical convergence guarantees.

6.1. Why not Euler angles?

Classically in these situations, we use the Euler angles representation of the rotations, $R_1 = R_z(\psi_1)R_y(\theta_1)R_x(\phi_1)$ and $R_2 = R_z(\psi_2)R_y(\theta_2)R_x(\phi_2)$, the textbook fusion is

$$R_{\text{cl}} = R_z(\psi_2)R_y(\theta_1)R_x(\phi_1). \quad (17)$$

This solution has two properties

- tilt is conserved, i.e. $R_{\text{cl}}^T e_z = \ell$.
- among all rotations \hat{R} such that $\hat{R}^T e_z = \ell$, R_{cl} minimises the horizontal error $\|\hat{R}^T e_x - R_2^T e_x\|^2$.

Despite its simplicity, Eq. (17) rests on two tacit assumptions that are seldom met in the leg–inertial setting considered here:

- The optimality of (17) is defined with respect to the IMU axis e_x only. This makes sense when e_x is the heading carrier, for example with a GPS-based heading for wheeled vehicles. In practice the IMU can be installed with an arbitrary yaw offset, and our yaw information actually comes from leg odometry (average contact frames, as in VALINOR where yaw information comes from Leg odometry) (Section 5), not from that axis. Treating e_x as “special” therefore introduces a systematic bias.
- Because (17) uses a ZYX decomposition of \mathbf{R}_1 , it inherits the Euler-angle singularity at $\theta_1 = \pm 90^\circ$. Exactly there the pitch axis aligns with e_z , roll and yaw collapse, and the horizontal projection of e_x disappears, leaving the yaw numerically indeterminate. Such attitudes are frequent in humanoids that perform multicontact motions.

To avoid these two issues we adopt an *axis-agnostic* fusion that uses solely the measured tilt ℓ , allows any horizontal vector to encode yaw, and stays remains regular for all configurations except the physically undefined case $\ell = -e_z$.

The construction is detailed next.

6.2. Yaw without Euler angles

The yaw contained in \mathbf{R}_2 can be extracted without decomposing the matrix into Euler angles. Let \mathbf{r}_2 denote the third column of \mathbf{R}_2 as $\mathbf{r}_2 = \mathbf{R}_2 e_z$. This column describes the world coordinates of the body z -axis, i.e. $\mathbf{r}_2 = \mathbf{R}_2 e_z$, which is not the tilt of \mathbf{R}_2 . Using that row we It is used to build the horizontal vector \mathbf{m}_2 :

$$\underline{\mathbf{r}_2} = \mathbf{m}_2 = \begin{cases} \mathbf{e}_x, & \text{if } r_{2,x}^2 + r_{2,y}^2 \simeq 0, \\ \frac{1}{\sqrt{r_{2,x}^2 + r_{2,y}^2}} [r_{2,y} - r_{2,x} 0]^\top, & \text{otherwise.} \end{cases} \quad (18)$$

where $r_{2,x}$ and $r_{2,y}$ are the first and second components of \mathbf{r}_2 . The condition $r_{2,x}^2 + r_{2,y}^2 \simeq 0$ occurs when the tilt $\mathbf{R}_2^T e_z$ is vertical (upward or downward), so we take $\mathbf{m} = \mathbf{e}_x$ to avoid the singularity. Otherwise, the vector \mathbf{m}_2 always exists and is unique.

Rotating \mathbf{m}_2 by \mathbf{R}_2^T results in another vector $\tilde{\mathbf{m}}_2 = \mathbf{R}_2^T \mathbf{m}_2$, which is horizontal as well. This means that \mathbf{m}_2 is an invariant horizontal vector.

We define the yaw of \mathbf{R}_2 , as the angle between these two vectors \mathbf{m}_2 and $\tilde{\mathbf{m}}_2$, which is computed as

$$\theta = \text{atan2}\left(m_{2,x}\tilde{m}_{2,y} - m_{2,y}\tilde{m}_{2,x}, m_{2,x}\tilde{m}_{2,x} + m_{2,y}\tilde{m}_{2,y}\right), \quad (19)$$

where the x and y subscripts denote the first and second components of the corresponding vector. The use of horizontal invariant vectors to define the yaw angle guarantees

that we only consider rotations around the vertical axis, giving the most sound definition of yaw.

When the tilt $\mathbf{R}_2^T e_z$ is vertical, any horizontal vector is invariant. In such a case, taking $\mathbf{m} = \mathbf{e}_x$ amounts at mimicking Euler yaw angle. When this tilt is downward, meaning \mathbf{R}_2 is a rotation of π around a horizontal axis, the concept of yaw itself is ill-defined. When the tilt is upward, any horizontal vector gives the same yaw, so using Euler’s yaw angle by taking $\mathbf{m}_2 = \mathbf{e}_x$ is a valid choice.

Fortunately we don’t have to compute this angle explicitly to perform the merge. It can be done directly using the tilt ℓ and the second matrix \mathbf{R}_2 .

6.3. Axis-agnostic triad fusion

6.3.1 Select a proper reference vector

Similarly to the yaw computation above, we start with the selection of the reference vector. However, the fusion makes this step slightly more subtle than just identifying the yaw of a single matrix. This is because the two rotations could have a very different tilt. This could lead the tilt vector ℓ to be nearly colinear with the rotated horizontal invariant vector $\mathbf{R}_2^T \mathbf{m}_2$. This would make the fusion singular.

The solution is to find a reference vector that is both horizontal and orthogonal to the real up direction $\mathbf{r} = \mathbf{R}_2 \ell = [v_x \ v_y \ v_z]^\top$. Consequently, a proper reference vector can be defined as

$$\mathbf{m} = \begin{cases} \mathbf{m}_2, & \text{if } v_x^2 + v_y^2 \simeq 0, \\ \frac{1}{\sqrt{v_x^2 + v_y^2}} [v_y - v_x 0]^\top, & \text{otherwise.} \end{cases}$$

$$\mathbf{m} = \begin{cases} \mathbf{m}_2, & \text{if } v_x^2 + v_y^2 \simeq 0, \\ \frac{1}{\sqrt{v_x^2 + v_y^2}} [v_y - v_x 0]^\top, & \text{otherwise.} \end{cases} \quad (20)$$

The degenerate case occurs only when \mathbf{r} is vertical, corresponding to $\mathbf{R}_2^T e_z = \ell$, which means the rotations nearly have the same tilt. In this situation we know that ℓ and $\mathbf{R}_2^T \mathbf{m}_2$ are nearly orthogonal. We thus safely use the yaw angle definition of the previous section.

We define the vector \mathbf{m} expressed in the frame of \mathbf{R}_2 as $\mathbf{m}_l = \mathbf{R}_2^\top \mathbf{m}$.

6.3.2 Build two right-handed bases.

Using cross products, form

$$\mathbf{B}_1 = [\mathbf{m} \times \mathbf{e}_z \quad \mathbf{e}_z \times \mathbf{m} \times \mathbf{e}_z \quad \mathbf{e}_z],$$

$$\mathbf{B}_1 = [\mathbf{m} \times \mathbf{e}_z \quad \mathbf{e}_z \times \mathbf{m} \times \mathbf{e}_z \quad \mathbf{e}_z], \quad (21)$$

$$\tilde{\mathbf{B}}_2 = \begin{bmatrix} \frac{\mathbf{m}_l \times \ell}{\|\mathbf{m}_l \times \ell\|} & \frac{\ell \times \mathbf{m}_l \times \ell}{\|\mathbf{m}_l \times \ell\|} & \ell \end{bmatrix}. \quad (22)$$

$$\mathbf{B}_2 = \begin{bmatrix} \frac{\mathbf{m}_l \times \ell}{\|\mathbf{m}_l \times \ell\|} & \frac{\ell \times \mathbf{m}_l \times \ell}{\|\mathbf{m}_l \times \ell\|} & \ell \end{bmatrix}.$$

Each matrix is orthonormal: the first column is the chosen horizontal vector, the third column is the vertical direction, and the second column completes a right-handed frame.

6.3.3 Fuse tilt and yaw.

The desired rotation is obtained by rotating from the basis $\{\mathbf{m}_2, \cdot, \ell\}$ to $\{\mathbf{m}, \cdot, \mathbf{e}_z\}$:

$$\mathbf{R} = \mathbf{B}_1 \mathbf{B}_2^\top. \quad (23)$$

6.4. Properties

The rotation obtained with (23) shows Eq. (23) has the following properties:

- *Tilt preservation*: $\mathbf{R}^\top \mathbf{e}_z = \ell$.
- *Optimal horizontal alignment*: for any rotation $\hat{\mathbf{R}}$ satisfying $\hat{\mathbf{R}}^\top \mathbf{e}_z = \ell$, the choice (23) minimises

$$\|\hat{\mathbf{R}}^\top \mathbf{m} - \mathbf{m}_l\|^2.$$

$\|\hat{\mathbf{R}}^\top \mathbf{m} - \mathbf{m}_l\|^2$. The proof follows exactly the one-parameter maximisation used for the Euler fusion, but with the task-specific direction \mathbf{m} in place of the body axis \mathbf{e}_x .

- *Singularity*: the construction is undefined only when $\ell = -\mathbf{e}_z$; at that upside-down pose, yaw itself has no meaning.
- *Computation*: implementation requires only vector cross products and normalisations, with no Euler-angle extraction, keeping the cost compatible with real-time embedded execution.

The axis-agnostic fusion is therefore adopted as the default in **VALINOR**; it avoids the arbitrary preference for the body axis \mathbf{e}_x , takes into account the difference of tilt and avoids the gimbal-lock issues of Eq. (17).

7. EXPERIMENTAL EVALUATION

We identified the following criteria that a state estimator should satisfy to serve in our intended use case, namely, providing feedback for remote control [29] or task planning and execution in the robot's local workspace [30]:

Criterion 1 High computation speed, to leave more processing time for the rest of the pipeline.

Criterion 2 Accurate tilt and local lateral (horizontal) velocity estimation, which are crucial for stabilization.

Criterion 3 Low local drift (over small displacements) in translation and yaw. This would correlate with lower absolute drift, but the latter cannot be used as a metric because of its lower determinism by proprioceptive odometry.

The experiments presented in this Section assess **VALINOR** based on these criteria. We compare it with the Right-Invariant EKF (RI-EKF) from [27], a state-of-the-art method for legged robot proprioceptive odometry. This estimator has demonstrated its effectiveness in practical applications, notably as real-time feedback for remote control [29]. It is therefore particularly relevant to compare ourselves to this method.

7.1. Description of the experiments

The proposed estimator has been evaluated across **VALINOR** and the RI-EKF have been evaluated across two experimental scenarios on two different humanoid robots⁴:

- Experiment 1: A walk on a flat ground over about 18 meters with the robot RHP Friends [32], as shown in Figure ??2a. This experiment was repeated 5 times, for a total distance of about 90 meters.
- Experiment 2: A multi-contact motion over about 2 meters with the robot HRP-5P [33]. This motion involved an additional contact at the robot's left hand, and non-coplanar contacts on tilted obstacles, as shown in Figure ??Figures 2b and 2c. This experiment was repeated 4 times for a total distance of about 8 meters.

-0.5pe

-0.5pe Picture of HRP-5P during the multi-contact motion. The imprints of the successive contacts are represented by colored trapezoids: Red: Right foot; Green: Left foot; Blue: Left hand. Figure reused from [28]. -1.5pe

7.2. Experimental Results

We compare

7.2.1 Computational Performance Evaluation

To evaluate the computation speed performance of **VALINOR**, our proposed estimator, with the Right-Invariant EKF (RI-EKF) by [27], on both their computation and their estimation performance. For a fair comparison, both estimators received the same contact state information, from a Schmitt Trigger on the Ground Reaction Force (with thresholds set to 10% and 15% of the robot's weight). Ground truth pose we compared its average computation time per iteration over the four multi-contact experiments (involving three contacts), to

⁴We used the This dataset built to evaluate the Kinetics Observer was also used in [28]

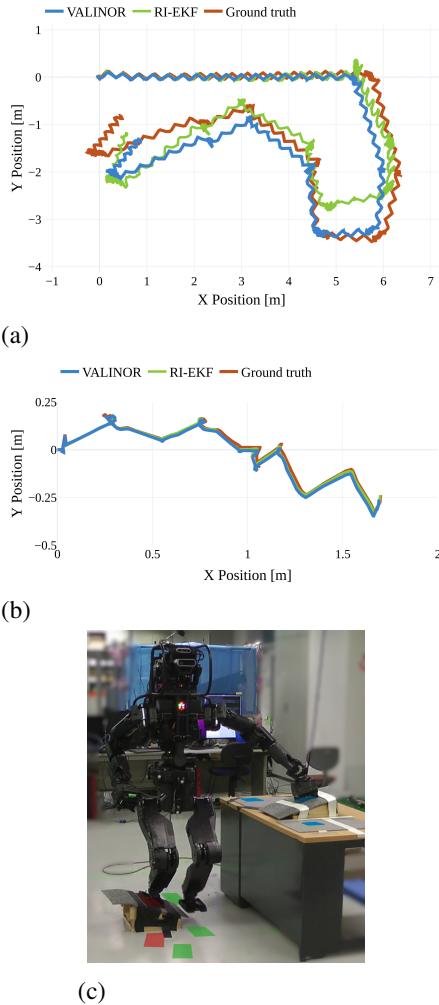


Fig. 2. ~~Ground truth and estimated RHP-Friend robot's pelvis trajectories during: (a) the first experiment of walk on flat ground (b) the multi-contact motion (c) HRP-5P performing the multi-contact motion. Contact imprints are represented by colored trapezoids. In red: Right foot. In green: Left foot. In blue: Left hand.~~

-1.5pe

that of the RI-EKF. Both estimators were run on the same laptop, equipped with an Intel Xeon E-2176M CPU (2.70GHz x 12). In average, an iteration of VALINOR was computed in 2.547 μ s, against 19.315 μ s for the RI-EKF. The proposed method is thus more than 7.5 times faster than the state-of-the-art estimator. Across all sequences, the average computation times had a standard deviation of 0.043 μ s, indicating consistency from one sequence to another. We also note that an additional contact in VALINOR implies only minor additional computations in Eqs. (1), (2), and (16), making our method highly scalable with respect to the number of contacts.

7.2.2 Estimation Accuracy Evaluation

In both experiments, the odometry performance of VALINOR and of the RI-EKF are evaluated using the ground truth trajectory of the robot's pelvis (to which the IMU is fixed on both robots) is provided by a motion capture system (OptiTrack, 16 Prime^X with 13 cameras), and its ground truth velocity is obtained by finite differences from the ground truth position then filtered with a zero-phase low-pass filter. The pose of the IMU's frame, estimated by both Valinor and the RI-EKF estimators, is transformed into that of the pelvis using simple rigid body transformation. For fairness in the comparison, both estimators receive the same contact state information, from a Schmitt Trigger on the Ground Reaction Force (with thresholds set to 10% and 15% of the robot's weight).

The odometry performance of Valinor and the RI-EKF is evaluated against the ground truth trajectory. Since the focus is on accuracy in local translations and yaw (Criterion ③), performance is assessed using the Relative Error (RE), as defined in [34]. RE The latter evaluates the estimation drift over segments of fixed distances. It is particularly relevant for proprioceptive odometry here, as it is not affected by global drift, which is unavoidable without exteroceptive sensors. RE therefore provides valuable and easily interpretable insight into the estimator's expected local accuracy of the estimator over a given distance.

We compute the Relative Error for lateral (x, y) and vertical (z) translations separately, since these components are generally influenced by different factors. Accuracy in lateral translation would mainly rely on that of local displacements and of yaw estimation, whereas accuracy in vertical translations would be affected by the accuracy of the estimation of the vertical local displacement, of the tilt, and the reliability of the contact height initialization. Similarly, angular errors in tilt and yaw are also evaluated independently, without being affected by global drift. The error e_ℓ on the tilt estimate is

Table 1. Mean and standard deviation (in parentheses) of the error on the estimations by VALINOR and the RI-EKF, during the two experiments on real robots. The 1 m and the 0.3 m Relative Errors are presented for the walk on flat ground and the multi-contact motion, respectively. The best result for each metric is highlighted in bold.

		RE Translation [m]		RE Orientation [°]		Linear velocity [m.s ⁻¹]	
		Lateral Vertical { x, y }	Vertical z	Tilt	Yaw	Lateral { x, y }	Vertical z
<u>Walk on flat ground</u>	VALINOR	0.032 (Proposed) (0.021)	0.015 (0.012)	0.49 (0.29)	1.14 (1.17)	0.019 (0.021)	0.007 (0.008)
	RI-EKF [27]	0.047 (0.057)	0.011 (0.026)	0.68 (0.49)	0.96 (0.90)	0.015 (0.017)	0.008 (0.010)
<u>Multi-contact motion</u>	VALINOR	0.007 (Proposed) (0.005)	0.002 (0.002)	0.23 (0.17)	0.40 (0.32)	0.010 (0.011)	0.004 (0.006)
	RI-EKF [27]	0.012 (0.016)	0.004 (0.006)	0.57 (1.02)	0.47 (0.65)	0.009 (0.009)	0.005 (0.007)

computed with:

$$e_{\ell} = \arccos(\ell_{gt}^T \hat{\ell}),$$

with using $e_{\ell} = \arccos(\ell_{gt}^T \hat{\ell})$, with ℓ_{gt} the ground truth tilt, and $\hat{\ell}$ the estimated one. Finally, we also assess the estimation of the IMU's linear velocity in the world. This velocity is expressed in the frame of the IMU such that it is not affected by errors in the orientation estimate. The velocity error is also separated into lateral and vertical components. Finally, local linear velocity estimation is assessed using the Mean Absolute Error between the estimated velocity and the ground truth, the latter obtained by finite differences from the ground-truth position, filtered with a zero-phase low-pass filter.

7.3. Experimental Results

7.2.1 Computational Performance Evaluation

To evaluate the computation speed performance of Valinor, we compared its average computation time per iteration over the four multi-contact experiments, to that of the RI-EKF. Both estimators were run on the same laptop, on an Intel Xeon E-2176M CPU (2.70GHz x 12). In average, an iteration of Valinor was computed in 2.547 s, against 19.315 s for the RI-EKF. The proposed method is thus more than 7.5 times faster than the state-of-the-art estimator, while proposing equivalent estimation accuracy, as shown below.

7.2.1 Estimation Accuracy Evaluation

Table 1 regroups summarizes the evaluation results of Valinor's estimation performance compared to those of the RI-EKF during the two experiments on real robots. The presented errors are computed over all the sequences of sequences in each scenario: 5 for the walk on flat ground, and 4 for the multi-contact motion.

We first note Based on the obtained results, we observe that the relative strengths of both estimators were consistent across both experiments, and that overall, they both displayed satisfactory estimation performance. Indeed, without the use of any exteroceptive sensors, their errors in translation estimates remain of the order of the centimeter per meter walked, those on orientation estimates remain around 1° per meter walked, and finally their error in velocity estimates remain below 2 cm. s⁻¹. Now we give more insights into the obtained results: across the two experiments. In the following, we detail these strengths, indicating the number of the criterion to which each is favorable in a circle.

- VALINOR estimated more accurately lateral translations local lateral translations ③ more accurately than the RI-EKF in both experiments, improving the estimation by over 30% and 40% compared to the RI-EKF respectively. Additionally, as visible in Figure ??-3, it improved the estimation of the tilt ② by about 28% in Experiment 1 and by 60% in Experiment 2, reaching an average error of only 0.23°. For both translations and tilt estimates, our method displayed a significantly lower standard deviation of the error, proving an improved estimation consistency.
- The RI-EKF estimated lateral velocities local lateral velocities ② and yaw ③ more accurately than VALINOR, with a during Experiment 1, with 20% smaller the error. But we believe this difference has lower significance due to the large value of the standard deviation, and 16% smaller errors, but both estimators displayed comparable performance in both estimates during Experiment 2.
- Both estimators showed similar performance for the estimation of the yaw and of vertical translation and velocity in vertical translation ③ and velocity estimation.

As shown To elaborate on the vertical translation results, one can notice in Figure ??, we noticed 3 that both estimators drifted upwards during the walk on flat ground, which involved longer walks and many more steps than the multi-contact motion. One of the reasons, a large number of steps. One contributing factor is the quality of the pitch estimation, an error in pitch estimation causing drift during forward walking. Another factor, inherent to Leg odometry, is its reliance on the quality of the contact detection. For example, detecting contacts consistently too early (when they are not perfectly still yet), would cause the estimate to systematically detecting contacts too early or too late can cause drift on each step. Conversely, drift can also occur when detecting contacts consistently too late, notably due to the robot's structural deformation under the contact force, etc. While investigating, we could indeed reduce the vertical drift contact forces. In general use, this vertical drift could be reduced by tuning the thresholds of the Schmitt Trigger used for contact detection contact detection threshold, which is arbitrarily set. However, we decided to leave them changed to to ensure a fair comparison. The other main factor explaining the vertical drift is the quality of the pitch estimate, which significantly affects vertical translation estimates during long forward walks, we kept the threshold unchanged and left this point for discussion. Moreover, absolute drift is not a reliable metric for proprioceptive odometry (see Criterion ③).

To conclude, both VALINOR and the RI-EKF displayed relative strengths with respect to each other, with regard to the criteria we defined. Overall, they provided highly satisfactory estimation performance which makes them well suited for local task planning and execution. Without the use of any exteroceptive sensors, their errors in translation estimates remain of the order of the centimeter per meter walked, those in orientation estimates remain around 1° per meter walked, and finally those in velocity estimates remain below 2 cm.s⁻¹.

It is noteworthy that, while reducing the computation time by more than a factor 7.5, VALINOR could consistently improve tilt and lateral translation estimation significantly.

0.5pe Estimated and ground truth height of the robot's pelvis during the first trial of the walk on flat floor. 1.5pe

8. CONCLUSION

In this paper, we proposed VALINOR, a lightweight estimator for humanoid robots. By combining Leg odometry with the Tilt Observer, we could perform accurate proprioceptive odometry whose accuracy rivals with the state-of-the-art method. We notably improved significantly the estimation of the lateral translations and of the tilt, while being over 7.5 times faster. Through this contribution, we aim to highlight the importance and the

effectiveness of lightweight state estimation methods, especially in the current state of control for legged robots where computational efficiency is becoming a predominant challenge. We also seek to drive interest towards methods with mathematical guarantees, which we believe are essential for the safe industrial and societal deployment of legged robots.

We note that by relying on Leg odometry and thus on the assumption of fixed contacts in the world, position and yaw estimation would drift from the actual ones in case of contact slippage [21], as visible in Figure 2a. Our method is particularly well suited to our intended use case of high-speed feedback for control in a local task space, but for tasks requiring minimal drift in absolute pose, methods using exteroceptive measurements [15, 17], although slower, would of course be more appropriate.

Further work would aim at reducing the sensitivity to slippage and other failure by improving the coupling between the Inertial and the Leg odometry, notably such that the position and yaw estimates leverage better IMU measurements while preserving high computational efficiency. Other extensions could investigate consider the incorporation of other sensors into the framework. Finally, we could investigate the handling of biases in the measurements, which is known to be a considerable challenge in proprioceptive odometry, either in theoretical terms or practical performances.

DECLARATIONS

Conflict of Interest

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

Authors' Contributions

Arnaud Demont and Mehdi Benallegue developed the proposed estimator's theoretical foundations. Abdelaziz Benallegue supervised the study and validated the theoretical framework. Arnaud Demont and Thomas Duvinage conducted the experiments. Arnaud Demont implemented the estimator and processed the experimental results. Arnaud Demont and Mehdi Benallegue wrote the manuscript in consultation with Thomas Duvinage and Abdelaziz Benallegue.

Funding

This paper is based on results obtained from a project of Programs for Bridging the gap between R&D and the IDEal society (society 5.0) and Generating Economic and social value (BRIDGE)/Practical Global Research in the AI x Robotics Services, implemented by the Cabinet Office, Government of Japan, and partially funded by the Japan Science and Technology Agency (JST) with the

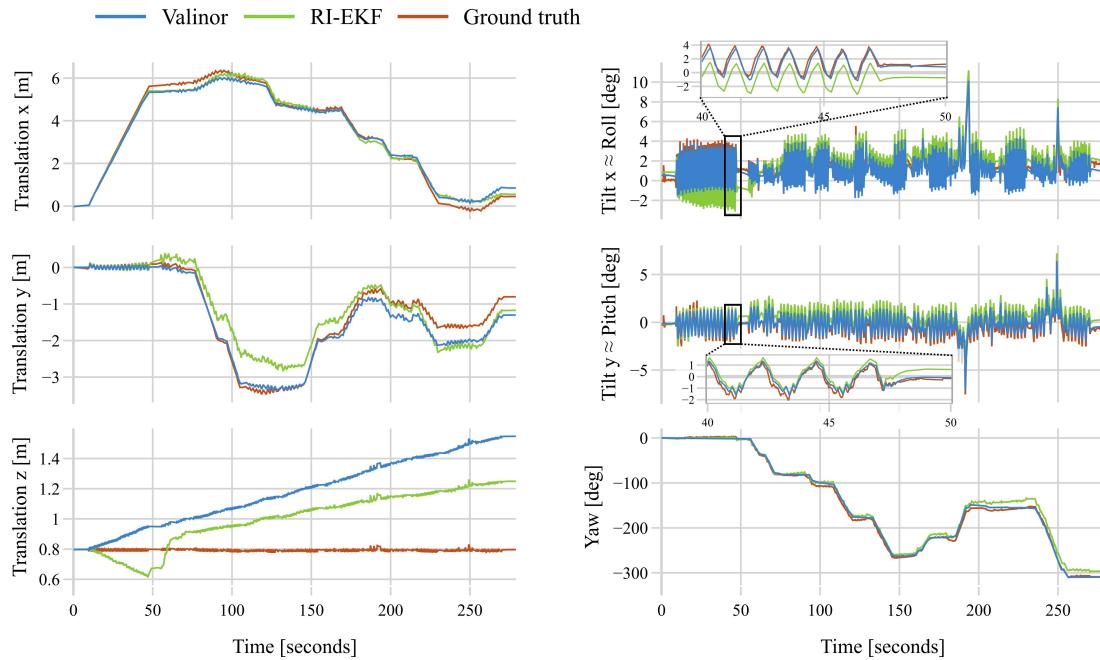


Fig. 3. Estimated and ground truth roll and pitch angles pose of the robot's pelvis RHP-Friends during the first trial of the walk on flat floor. Zoomed-in views are provided for the dense tilt plots, with a black rectangle indicating the corresponding area in the original plot.

JST-Mirai Program, grant number JPMJMI21H4.

REFERENCES

- [1] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, “Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot,” *Autonomous Robots*, vol. 40, 07 2015.
- [2] T. Sugihara and M. Morisawa, “A survey: dynamics of humanoid robots,” *Advanced Robotics*, vol. 34, no. 21-22, pp. 1338–1352, 2020.
- [3] A. Chlipala, *Certified programming with dependent types: a pragmatic introduction to the Coq proof assistant*. MIT Press, 2013.
- [4] S. Katayama, M. Murooka, and Y. T. and, “Model predictive control of legged and humanoid robots: models and algorithms,” *Advanced Robotics*, vol. 37, no. 5, pp. 298–315, 2023.
- [5] E. Dantec, M. Naveau, P. Fernbach, N. Villa, G. Saurel, O. Stasse, M. Taix, and N. Mansard, “Whole-body model predictive control for biped locomotion on a torque-controlled humanoid robot,” in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, Nov 2022, pp. 638–644.
- [6] A. Dallard, M. Benallegue, N. Scianca, F. Kanehiro, and A. Kheddar, “Robust Bipedal Walking with Closed-Loop MPC: Adios Stabilizers,” Feb. 2024, media material :<https://youtu.be/hfg7dTpuqu4>.
- [7] J. Peters, S. Vijayakumar, and S. Schaal, “Reinforcement learning for humanoid robotics,” *Proceedings of the third IEEE-RAS international conference on humanoid robots*, pp. 1–20, 01 2003.
- [8] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, “Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control,” *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 840–888, 2025.
- [9] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, Linxi, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. LLontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu, “Gr0ot n1: An open foundation model for generalist humanoid robots,” *CoRR*, vol. abs/2503.14734, March 2025.
- [10] K. Kawaharazuka, T. Matsushima, A. Gambardella, J. Guo, C. Paxton, and A. Zeng, “Real-world robot applications of foundation models: A review,” *Advanced Robotics*, vol. 38, no. 18, pp. 1232–1254, 2024.
- [11] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman, B. Ichter, D. Driess, J. Wu, C. Lu, and M. Schwager, “Foundation models in robotics: Applications, challenges, and the future,” *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 701–739, 2025.
- [12] M. N. Zeilinger, D. M. Raimondo, A. Domahidi, M. Morari, and C. N. Jones, “On real-time robust model predictive control,” *Automatica*, vol. 50, no. 3, pp. 683–694, 2014.

- [13] R. Findeisen and F. Allgöwer, “Computational delay in nonlinear model predictive control,” *IFAC Proceedings Volumes*, vol. 37, no. 1, pp. 427–432, 2004.
- [14] P. Thodoroff, W. Li, and N. D. Lawrence, “Benchmarking real-time reinforcement learning,” in *NeurIPS 2021 Workshop on Pre-registration in Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 181. PMLR, 13 Dec 2022, pp. 26–41.
- [15] D. Wisth, M. Camurri, and M. Fallon, “Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots,” *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 309–326, 2022.
- [16] M. Fallon, “Accurate and robust localization for walking robots fusing kinematics, inertial, vision and lidar,” *Interface focus*, vol. 8, no. 4, p. 20180015, 2018.
- [17] Y. Kuang, T. Hu, M. Ouyang, Y. Yang, and X. Zhang, “Tightly coupled lidar/imu/uwb fusion via resilient factor graph for quadruped robot positioning,” *Remote Sensing*, vol. 16, no. 22, 2024.
- [18] R. Mahony, T. Hamel, and J.-M. Pflimlin, “Nonlinear complementary filters on the special orthogonal group,” *IEEE Transactions on automatic control*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [19] M. Benallegue, R. Cisneros, A. Benallegue, Y. Chitour, M. Morisawa, and F. Kanehiro, “Lyapunov-Stable Orientation Estimator for Humanoid Robots,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6371–6378, Oct. 2020.
- [20] P.-C. Lin, H. Komsuoglu, and D. Koditschek, “A leg configuration measurement system for full-body pose estimates in a hexapod robot,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 411–422, June 2005.
- [21] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, “State estimation for legged robots-consistent fusion of leg kinematics and IMU,” *Robotics*, vol. 17, pp. 17–24, 2013.
- [22] P. Wawryński, J. Možaryn, and J. Klimaszewski, “Robust estimation of walking robots velocity and tilt using proprioceptive sensors data fusion,” *Robotics and Autonomous Systems*, vol. 66, pp. 44–54, 2015.
- [23] K. Masuya and T. Sugihara, “Dead reckoning for biped robots that suffers less from foot contact condition based on anchoring pivot estimation,” *Advanced Robotics*, vol. 29, no. 12, pp. 785–799, 2015.
- [24] T.-Y. Lin, R. Zhang, J. Yu, and M. Ghaffari, “Legged robot state estimation using invariant kalman filtering and learned contact events,” in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 1057–1066.
- [25] M. Maravgakis, D.-E. Argiropoulos, S. Piperakis, and P. Trahanias, “Probabilistic contact state estimation for legged robots using inertial information,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 163–12 169.
- [26] Z. Yoon, J.-H. Kim, and H.-W. Park, “Invariant smoother for legged robot state estimation with dynamic contact event information,” *IEEE Transactions on Robotics*, 2023.
- [27] R. Hartley, M. Ghaffari, R. M. Eustice, and J. W. Grizzle, “Contact-aided invariant extended kalman filtering for robot state estimation,” *The International Journal of Robotics Research*, vol. 39, no. 4, pp. 402–430, 2020.
- [28] A. Demont, M. Benallegue, A. Benallegue, P. Gergondet, A. Dallard, R. Cisneros, M. Murooka, and F. Kanehiro, “The Kinetics Observer: A Tightly Coupled Estimator for Legged Robots,” Dec. 2024, working paper or preprint.
- [29] R. Grandia, E. Knoop, M. Hopkins, G. Wiedebach, J. Bishop, S. Pickles, D. Müller, and M. Bächer, “Design and control of a bipedal robotic character,” in *Robotics: Science and Systems XX*, ser. RSS2024. Robotics: Science and Systems Foundation, Jul. 2024.
- [30] M. Tsuru, A. Escande, I. Kumagai, M. Murooka, and K. Harada, “Online multi-contact motion replanning for humanoid robots with semantic 3d voxel mapping: Exocetomap,” *Sensors*, vol. 23, no. 21, 2023.
- [31] M. Benallegue, A. Benallegue, R. Cisneros, and Y. Chitour, “Velocity-aided imu-based tilt and attitude estimation,” *IEEE Transactions on Automatic Control*, vol. 68, no. 10, pp. 5823–5836, 2023.
- [32] M. Benallegue, G. Lorthioir, A. Dallard, R. Cisneros-Limón, I. Kumagai, M. Morisawa, H. Kaminaga, M. Murooka, A. Andre, P. Gergondet, K. Kaneko, G. Caron, F. Kanehiro, A. Kheddar, S. Yukizaki, J. Karasuyama, J. Murakami, and M. Kamon, “Humanoid robot hrp friends: Seamless combination of autonomous and tele-operated tasks in a nursing context,” *IEEE Robotics and Automation Magazine*, vol. 32, no. 1, pp. 79–90, March 2025.
- [33] K. Kaneko, H. Kaminaga, T. Sakaguchi, S. Kajita, M. Morisawa, I. Kumagai, and F. Kanehiro, “Humanoid robot hrp-5p: An electrically actuated humanoid robot with high-power and wide-range joints,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1431–1438, April 2019.
- [34] Z. Zhang and D. Scaramuzza, “A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2018, pp. 7244–7251.



Arnaud Demont received the M.S. degree in mechanical engineering from the National Institute of Applied Sciences of Lyon, France, and a second M.S. degree in automation and robotics in intelligent systems from the University of Technology of Compiègne, France, in 2021 and 2023 respectively. He is currently pursuing the PhD degree of the Université Paris-Saclay, France, within the CNRS-AIST Joint Robotics Laboratory in Tsukuba, Japan. His research interests include state estimation for legged robots, multi-sensor fusion, and mobile robot perception and autonomous navigation.



Mehdi Benallegue obtained his engineering degree from the National Institute of Computer Science (INI) in Algeria, in 2007. He earned his MSc from the University of Paris 7, France, in 2008, and a Ph.D. from the University of Montpellier 2, France, in 2011. His research took him to the CNRS AIST Joint Robotics Laboratory (JRL), Japan, and to INRIA Grenoble, France. He also worked as a postdoctoral researcher at the Collège de France and at LAAS CNRS, France. Currently, he is a Senior Researcher with JRL, Japan. His research interests include robot estimation and control, legged locomotion, biomechanics, neuroscience, and computational geometry.



Thomas Duvinage received the M.S. degree in computer science with a specialization in robotics and computer vision from the University of Technology of Belfort-Montbéliard, France. He also holds a second M.S. degree in industrial and technology entrepreneurship from the same university. Since 2024, he has been serving as a CNRS delegate research engineer at the CNRS-AIST Joint Robotics Laboratory in Tsukuba, Japan. His work focuses on software integration for robotic systems, with research interests including robotics software architecture, system integration, and real-time robotic applications.



Prof. Abdelaziz Benallegue received the B.S. degree in electronics engineering from Algiers National Polytechnic School, Algeria in 1986 and both the M.S. and Ph.D. degrees in automatic control and robotics from University of Pierre and Marie Curie, Paris 6 (currently Sorbonne University), France in 1987 and 1991 respectively. He was Associate professor in Automatic Control and Robotics at the University Pierre et Marie Curie, Paris 6 (currently Sorbonne University) from 1992 to 2002. In September 2002, he joined the University of Versailles St Quentin as full Professor assigned. He was a CNRS delegate at JRL-AIST, Japan for three years, between 2016 and 2022. His research activities are mainly related to linear and non-linear estimation and control theory (adaptive control, robust control, neural learning control, observers, multi-sensor fusion, etc.) with applications in robotics (humanoid robots, aerial robots, manipulator robots, etc.).

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.