



HAL
open science

The Kinetics Observer: A Tightly Coupled Estimator for Legged Robots

Arnaud Demont, Mehdi Benallegue, Abdelaziz Benallegue, Pierre Gergondet,
Antonin Dallard, Rafael Cisneros, Masaki Murooka, Fumio Kanehiro

► **To cite this version:**

Arnaud Demont, Mehdi Benallegue, Abdelaziz Benallegue, Pierre Gergondet, Antonin Dallard, et al..
The Kinetics Observer: A Tightly Coupled Estimator for Legged Robots. 2024. hal-04616647v2

HAL Id: hal-04616647

<https://hal.science/hal-04616647v2>

Preprint submitted on 28 Jan 2025

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.

Copyright

The Kinetics Observer: A Tightly Coupled Estimator for Legged Robots

A. Demont, M. Benallegue, A. Benallegue, P. Gergondet, A. Dallard,
R. Cisneros-Limón, M. Murooka, F. Kanehiro

Index Terms—Legged robots, State estimation, Proprioceptive odometry, Humanoid robots, Contacts estimation.

Abstract—In this paper, we propose the “Kinetics Observer”, a novel estimator addressing the challenge of state estimation for legged robots using proprioceptive sensors (encoders, IMU and force/torque sensors). Based on a Multiplicative Extended Kalman Filter, the Kinetics Observer allows the real-time simultaneous estimation of contact and perturbation forces, and of the robot’s kinematics, which are accurate enough to perform proprioceptive odometry. Thanks to a viscoelastic model of the contacts linking their kinematics to the ones of the centroid of the robot, the Kinetics Observer ensures a tight coupling between the whole-body kinematics and dynamics of the robot. This coupling entails a redundancy of the measurements that enhances the robustness and the accuracy of the estimation. This estimator was tested on two humanoid robots walking on flat terrain and on slippery obstacles, and performing non-coplanar multi-contact locomotion.

I. INTRODUCTION

The control of legged robots is a highly challenging topic in robotics. These robots are notably intended to operate in industrial and personal assistance contexts, and their behavior must be reliable and thus robust against failures caused by external disturbances or internal malfunctions. This robustness is crucial not only for the correct execution of the tasks but, more importantly, for ensuring the safety of nearby users. The balance and motion of legged robots are achieved solely through contact interactions with their environment, whether intentional or not, and therefore rely on locally applied forces and torques. This under-actuation implies that, in order to achieve the desired motion of the robot, the appropriate contact forces must be applied.

Furthermore, the dynamics and kinematics of legged robots constrain the admissible robot’s postures and trajectories that allow them to maintain balance. It is therefore essential to know the robot’s posture as precisely as possible and a fortiori its general pose in the environment. This can be done by estimating the kinematics of its floating base, which is the root of its kinematic tree. From these kinematics, one can obtain the pose of any limb and estimate the position of the robot’s center of mass.

A. Demont, M. Benallegue, P. Gergondet, R. Cisneros-Limón, M. Murooka and F. Kanehiro are with the CNRS-AIST JRL (Joint Robotics Laboratory), IRL, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. A. Demont and A. Benallegue are also with Université Paris-Saclay, Gif-sur-Yvette, France, and Laboratoire d’Ingénierie des Systèmes de Versailles, France. A. Dallard is with the CNRS-University of Montpellier, LIRMM, UMR5506, Montpellier, France.

Manuscript received XX XX, XX; revised XX XX, XX.

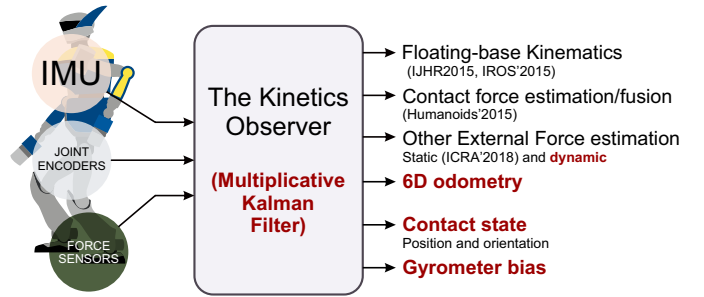


Fig. 1. Summary of the sensors we use and the estimated state. We highlighted in bold red the new features compared to our prior works (IJHR2015 [1], IROS’2015 [2], Humanoids’2015 [3], ICRA’2018 [4]).

Finally, an accurate spatial awareness of the robot is an inevitable step towards autonomy, allowing it to navigate independently through the environment. This explains why by far the most explored estimation field in legged robotics is the pose estimation for the robot. It can be divided into two branches: one focusing on a real-time estimation that is, for instance, necessary for stabilization and high frequency control purposes, and one focusing on minimizing the estimation drifts over long distances (accurate localization and odometry), which generally runs at a lower frequency.

A. Related Work

1) *High-frequency pose estimators and proprioceptive odometry*: High-frequency pose estimators typically rely on proprioceptive sensors, particularly on joints encoders and IMUs due to their high bandwidth. The integration of the IMU measurements is enhanced with the successive positions of the contacts, adding kinematic constraints with no-slip conditions and greatly improving the tilt estimation [5]. This method allows for very accurate proprioception, notably on the tilt estimation, but it is still subject to drifts in the position and the yaw orientation in the world. These drifts can be due to various reasons, such as contact slippage, compliance, uncertain contact detection, etc. To perform odometry, the contact positions can be added to the estimated state within an Extended Kalman Filter. They can then be partially corrected, reducing the drifts [5], [6], [7], [8]. Improvements can be achieved by better estimating the time a contact is firmly set and can be used in the estimation. A common method is the thresholding of the Ground Reaction Force (GRF) [9], [10], [11], but it is notably affected by the perturbations of the GRF during impact and by slippage. Lin et al. [12] implemented

a detection based on a neural network that uses the IMU measurements and the joint encoders of quadruped robots to deal with this bouncing problem. Maravgakis et al. [13] proposed an alternative using IMUs at the end effectors to robustify contact detection in the presence of slippage. Yoon et al. [14] introduced a method to detect and handle dynamic contacts based on the thresholding of the estimated foot velocity and acceleration. Instead of ignoring contact information in the case of dynamic contacts, they increase the covariance associated with the static contact foot assumption. They also enforce their static contact condition with a dedicated cost in their invariant smoother when the conditions are met. Finally, Buchanan et al. [15] addressed the slippage problem by estimating the sliding displacement with a Convolutional Neural Network, which uses the IMU's measurements.

2) *Exteroceptive odometry*: The most accurate odometry results are typically achieved by incorporating additional spatial awareness from exteroceptive information (mainly LIDARs and cameras) [16], notably adding robustness to contact slippage, although they are highly dependent on the non-failure of the exteroceptive source [17]. The most robust solutions [18], [19], [20], [21] use factor graphs to leverage the redundancies in exteroceptive and proprioceptive odometry information and improve their coupling, thereby benefiting from the advantages of both methods. This better explicitation of the coupling between key variables has also been adopted by some proprioceptive methods, for instance to account for the robot's kinematic chain model [22] or the measured contact forces [23], resulting in more accurate and robust estimations. In addressing odometry for legged robots, our approach acknowledges the importance of exteroceptive sensing for achieving the highest levels of accuracy. However, we believe that there is significant potential to further improve proprioceptive odometry itself, and we aim to focus on these improvements before integrating exteroceptive information.

3) *Estimation frameworks for legged robots odometry*: To improve the floating base pose estimation, a more mathematically accurate representation of orientations within the estimation framework has been widely adopted. Although giving decent results, the initially used 3D vectorial representation (e.g., with Euler angles [24], [25]) induces inaccurate additions and uncertainty propagation due to the use of operators defined on \mathbb{R}^3 on orientations belonging to the $SO(3)$ Lie group. Quaternion Extended Kalman Filters [5] addressed this issue by using the quaternion representation of orientations and their multiplication operator for the update. Formalized by Bourmaud et al. for Extended Kalman Filters (EKF) [26], the use of the Lie group properties of $SO(3)$ with its appropriate operators ensures the mathematical consistency of estimators and has been widely adopted [16], [17], [27], [28]. Invariant Extended Kalman Filters (InEKF) [29], [30], [31], [32] used Lie groups to take advantage of the so-called group affine property on nonlinear systems, guaranteeing the invariance of the estimation error on the system variables subject to symmetry. This gives a more robust and accurate linearization and provides local convergence properties of these variables, improving the estimation [31]. Phogat et al. [33] proposed an improvement of the InEKF by bypassing the need to express

the nonlinear error dynamics in the tangent space of the state through its *log* operator, allowing for a faster computation. However, the invariance remains a constraining property that is difficult to scale up to more complex dynamics, especially with multiple couplings between the orientation and the dynamics of other state variables [34]. Overall, the EKF is the predominant filter used for floating base estimation, its main advantage being its superior computational efficiency. Despite its widespread use, we can find alternative filtering methods in the literature. For example, Chauchat et. al [35] extended the use of the group affine property to smoothing frameworks. This work was then applied to legged robot state estimation by Yoon et. al [14]. A more common alternative to the EKF is the Unscented Kalman Filter (UKF) [36], [37]. In [37], Bloesch et al. justify their use of the UKF by its eased handling of correlated noise between the prediction and correction steps. In another paper [38], we proposed the Tilt Estimator based on a kinematically coupled complementary filter, which to our knowledge is the only estimator for humanoid robots with a proof of global convergence.

4) *Biases estimation*: Another way to improve the global estimation is to estimate biases in the model or the sensor measurements along with the kinematics of the floating base. Common biases are related to IMU measurements [37], [39], [40], or to the kinematics of the robot's center of mass [41]. The latter is analogous to a constant external force applied to the robot.

5) *External forces estimation*: While the estimation of biases on the IMU measurements is now mainstream, other variables can be estimated to improve the estimation and the control of the robot, notably the external forces exerted on the robot. These forces can be categorized into two types: disturbance forces and contact forces. Disturbance forces, such as those caused by unanticipated collisions, must be estimated and taken into account to ensure the stability of the robot when interacting with the environment, including interactions with humans [42]. On the other hand, contact forces are exerted at a limb of the robot in a fixed contact with the environment that the robot is aware of. Since contacts are a key element of the control of legged robots, estimating these contact forces is crucial not only for maintaining the robot's balance, but also for accurately generating the desired motions [43]. We first introduce methods for estimating the resultant of the disturbance forces, which are more extensively described by Masuya and Ayusawa [41]. Kaneko et al. [42] used a simple model to estimate the resultant external disturbance force exerted on the center of mass from its measured linear acceleration (assuming that the IMU is located at the center of mass) and the measured force at the feet. More recent methods have allowed to estimate the external torque applied to the robot. To achieve this, Flacco et al. [44] adapted the residual method, already used for fixed-base robots, to floating-base robots equipped with joint torque sensors. The introduced estimators show strong overall performance. However, they address the estimation of the center of mass kinematics / external forces independently of the floating base's kinematics. The latter must therefore be estimated in another estimator running in parallel. Piperakis et al. [45] proposed such a two-

estimator cascade framework that fuses joint encoders, IMU, feet sensitive resistors and visual odometry measurements. Overall, the estimation of contact forces appears to be a less explored topic. We found two estimation methods of these forces, both based on the joint measurements of their robot. Xu et al. [46] combine the kinematic model of the leg of their hexapod robot with a model of the force applied to its tip, to estimate the latter from the output torque of the three motors actuating the leg. Cong et al. [47] use a relationship between the generalized momentum of their robot and the contact force at the newly created contact of their quadruped through the Jacobian matrix of the legs. Nevertheless, we believe that the coupling between the floating-base kinematics and the contact forces is being underexploited. Indeed, we have shown in a previous paper that we could estimate both the contact forces and the floating base kinematics based only on IMU measurements [2]. To conclude on this section, only our previous work [4] seems to propose the simultaneous estimation of the floating base's kinematics along with the external forces estimation, by considering contact and disturbance forces independently, without needing torque sensors. This is because the IMU provides the observability of the resulting wrench from which we can deduct the measured ones to get the remaining external wrench [48].

B. Contributions

To the best of our knowledge, no existing approach deals with odometry and the estimation of contact and external forces, in a single, tightly coupled estimation loop. Our work aims to address this issue in the case of proprioceptive sensors, with the following approach: if available, the contact wrench measurements are not used only for contact detection purposes, but are directly involved in the correction of the estimated kinematics. The floating base's kinematics, the information from the IMU, contact poses and wrenches are all explicitly coupled in our dynamical model, providing highly redundant measurements and a total coherence between our state variables. This rich coupling allows for a better estimation of variables with low observability, such as the yaw angle and the robot's position in the world, and an improved robustness to drifts and uncertainties.

Fig. 1 illustrates the main features of the kinetics observer and the novelties compared to our previous work. In addition, the main contributions of this work to the state of the art are summarized below:

- To the best of our knowledge, the Kinetics Observer is the first estimator to perform proprioceptive odometry while also estimating the external wrench and the contact wrenches applied to the robot, simultaneously and in a tightly coupled manner.
- It is also the only estimator that uses the measurements of the force and torque sensors to estimate the kinematics of the robot.
- It can locally reconstruct the characteristics of the environment, namely the position and orientation of the ground and obstacles, based on all the proprioceptive sensors, including the IMU. The representation of contacts

can also be extended to point contacts and contacts on edges.

- The proposed estimator is designed for all kinds of legged robots. It is notably adaptable to any number of legs, wrench sensors, and IMUs, and their position on the robot is not imposed. In the case of legged robots without force sensors, the Kinetics Observer can still be used to perform a leg-inertial odometry which can also account for contact orientations.
- We designed a specifically constrained covariance matrix for the process noise of the contact positions in the state of an Extended Kalman Filter for proprioceptive odometry. We show that this allows for the relaxation of fixed-contact constraint and thus for a better correction of the contact positions, while preventing their average position from drifting.
- The code of the Kinetics Observer is already available as an open-source framework¹

After giving an overview of the notions necessary for the presentation of this work in Section II, we will define the system for which we developed the Kinetics Observer in Section III. In the Section IV, we will then detail the implementation of the Kinetics Observer before presenting our experimental results in Section V. Finally, we will discuss these results and the future improvements.

II. PRELIMINARIES

A. General notations

- Different notations are used to distinguish estimated ($\hat{\circ}$), predicted ($\bar{\circ}$), and error ($\tilde{\circ}$) variables.
- $\mathbb{I}_{n \times n}$ and $\mathbb{O}_{n \times n}$ are, the $n \times n$ identity and zero square matrices, respectively. If the dimension n is not given, the 3×3 matrix is implicitly referred to.
- Reference frames are represented by uppercase calligraphy variables (e.g., \mathcal{W} for the world frame and \mathcal{B} for the frame of the floating base) or uppercase Greek letters (e.g., Γ for the centroid frame).
- Kinematic variables (position, orientation, velocities and accelerations) are represented using the notation ${}^{\mathcal{B}}\mathbb{O}_{\mathcal{A}}$, which expresses the kinematics of the frame \mathcal{A} in the frame \mathcal{B} . To simplify the notation, we omit the world frame symbol \mathcal{W} whenever it can be inferred from the context: ${}^{\mathcal{W}}\mathbb{O}_{\mathcal{A}} = \mathbb{O}_{\mathcal{A}}$.
- The matrix representation of a rotation \mathbf{R} is used in the developed equations, but to keep rigorous notations, we also use a vector representation Ω when necessary. For example, ${}^{\mathcal{B}}\Omega_{\mathcal{A}}$ refers to the vector representation of the rotation ${}^{\mathcal{B}}\mathbf{R}_{\mathcal{A}}$ (e.g., quaternions, Euler angles, etc.).

The state of our system lies in a high-dimensional Lie group consisting of components from \mathbb{R}^3 and components of the Lie group of 3D rotation matrices $\text{SO}(3)$. The corresponding operators are described below. We also recommend reading [49] for a more in-depth explanation of Lie groups and their operators. In the following we denote \mathbf{E} an element of a

¹<https://github.com/jrl-umi3218/state-observation>.

generic Lie group G and e an element of its Lie Algebra \mathfrak{g} .

- $\oplus : G \times G \rightarrow G$ is the sum operator between two elements of a Lie group.
- $\ominus : G \times G \rightarrow G$ is the difference operator between two elements of a Lie group.
- $\exp_G(e) : \mathfrak{g} \rightarrow G$ is the homeomorphism that maps an element e of the Lie Algebra \mathfrak{g} to its associated element E of the Lie group G , such that E is the "integral" of e over the interval $[0, 1]$. The nature of the integral is determined by the inner operator of the Lie group.
- $\log_G(E) : G \rightarrow \mathfrak{g}$ is the homeomorphism that inverts the \exp operator. In other words, it maps an element $E \in G$ to its associated element $e \in \mathfrak{g}$ such that e has the minimal norm and $\exp(e) = E$. This function is usually well defined in the neighborhood of the neutral element of the Lie group and is usually homeomorphic to \mathbb{R}^{n_G} where n_G is the manifold dimension of G .

We note that the Lie Algebra of $SO(3)$ denoted by $\mathfrak{so}(3)$, is the group of skew-symmetric matrices, but to represent elements of $\mathfrak{so}(3)$ in a more readable way we use the vee (\vee) operator (declined below) which maps $\mathfrak{so}(3)$ over \mathbb{R}^3 . In other words,

- $[\cdot]_{\times} : \mathbb{R}^3 \rightarrow \mathfrak{so}(3)$ is the skew-symmetric (or anti-symmetric / cross-product) matrix operator. For $v = (v_x \ v_y \ v_z)^T$,

$$[v]_{\times} = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix}. \quad (1)$$

- $[\cdot]_{\vee} : \mathfrak{so}(3) \rightarrow \mathbb{R}^3$ is the vectorization operator of skew-symmetric matrices:

$$[[v]_{\times}]_{\vee} = v. \quad (2)$$

Table I summarizes the definitions of these operators for \mathbb{R}^3 and $SO(3)$. From this table, we can see that we can omit the notations of the Lie group operators for vectors in \mathbb{R}^3 because their definitions are straightforward. Furthermore, to simplify the notations, we introduce the following two operators that use the homeomorphism between \mathbb{R}^3 and $\mathfrak{so}(3)$ to directly handle vectors instead of skew-symmetric matrices.

- $\text{Log} : SO(3) \rightarrow \mathbb{R}^3$ such that $R \mapsto \text{Log}(R) = [\log(R)]_{\vee}$.
- $\text{Exp} : \mathbb{R}^3 \rightarrow SO(3)$ such that $\omega \mapsto \text{Exp}(\omega) = \exp([\omega]_{\times})$.

B. Frames and variables definition

The Kinetics Observer relies on a number of frames, which we introduce in this section. To help visualize them, they are represented on Fig. 2 along the main state variables.

We first introduce the *centroid frame* denoted Γ , which relies on the centroid definition specific to humanoids [50]. This frame is attached to the robot and is located at the center of mass. The conditions that its orientation must satisfy are that (i) its trajectory in the world frame is twice differentiable and (ii) knowing the kinematics (pose, velocities, and accelerations) of the centroid in the world frame allows to obtain the

kinematics of all the robot limbs, including the floating-base, in the world frame using simple frame transformations. In this paper, we define this orientation as that of the floating base. Our notation for the variables of the centroid frame kinematics in the world frame is defined in Table II. Note that besides the rotations, they are all expressed in the centroid frame and not in the world frame. To make this specificity clearer throughout the paper, we add the subscript l to these variables.

The robot is in contact with the environment. The number of contact points, denoted n_c , is arbitrary and time-varying. A contact frame C_i is attached to the i -th contact point. Its kinematics $\{p_{C_i}, \Omega_{C_i}, \dot{p}_{C_i}, \omega_{C_i}\}$ in the world frame are obtained by forward kinematics from those of the centroid frame. A reaction wrench is applied to the contact point, which is composed of a force and a moment. These can be expressed in the contact frame C_i , and we denote them by ${}^C F_i$ and ${}^C T_i$. A wrench sensor can be associated with the contact. To avoid introducing additional frames, we consider that the sensor's frame is attached to C_i . When referring to the wrench measured by the sensor, we consider that the wrench has been preliminarily expressed in C_i . We consider a viscoelastic contact model, meaning that the contact wrench depends on the deformation of the contact material. For each contact, we thus also consider a *rest* frame $C_{r,i}$. This frame denotes the pose of the contact frame such that the deformation would be null. It can therefore be seen as the local characteristics of the environment at the instant the contact is created. Its linear and angular velocities in the world are zero. Fig. 2 helps to grasp the difference between the frame of the contact C_i and its rest frame $C_{r,i}$. To simplify the notation, we denote the position and orientation of the rest frame $C_{r,i}$ in the world frame $p_{r,i} \triangleq p_{C_{r,i}}$ and $\Omega_{r,i} \triangleq \Omega_{C_{r,i}}$, respectively. We will also refer to them as the contact's rest position and rest orientation, respectively. Note that $p_{r,i}$ and $\Omega_{r,i}$ are always attached to the world frame, since they represent the environment's configuration at the contact point. It is important to note that slippage and moving contacts are *not* neglected in our model. They are simply reflected by a change in the position of the rest frame $C_{r,i}$. In summary, at any moment there is a non-fixed pose $\{p_{r,i}, \Omega_{r,i}\}$ for the i -th contact, so that if the contact is in this pose, there would be no reaction wrench. The dynamics of this model are described in detail in Section III-B3. We define a contact state denoted by $x_{c,i} \triangleq \{p_{r,i}, \Omega_{r,i}, {}^C F_i, {}^C T_i\}$, a vector containing the pose of the contact rest frame $C_{r,i}$ in the world frame and the forces and moments of this contact expressed at the contact frame C_i . Finally, we note that a summary of the notation is available in Appendix A.

III. PROBLEM STATEMENT

The Kinetics Observer estimates the extrinsic state of a humanoid robot in contact with the environment. Below are the system specifications and the requirements for the observer.

We consider that some of the contacts with the robot may be equipped with wrench sensors, and some may not. The sensors may operate at a lower frequency than the control loop, and thus may not provide a value at each iteration. We denote the time-varying number of contact wrench sensors by n_w .

TABLE I
DEFINITION OF THE LIE GROUP ELEMENTS AND THEIR OPERATORS FOR \mathbb{R}^3 AND $SO(3)$

Lie group G	\mathbb{R}^3	$SO(3)$
Lie Algebra \mathfrak{g}	\mathbb{R}^3 (itself)	$\mathfrak{so}(3)$, homeomorphic to \mathbb{R}^3
Neutral element	$\mathbb{O}_{3 \times 1}$	\mathbb{I}
Inverse of elem. \mathbf{E}	$-\mathbf{E}$	\mathbf{E}^T
\oplus	$\mathbf{x}_1 \oplus \mathbf{x}_2 = \mathbf{x}_1 + \mathbf{x}_2$	$\mathbf{E}_1 \oplus \mathbf{E}_2 = \mathbf{E}_1 \cdot \mathbf{E}_2$
\ominus	$\mathbf{x}_1 \ominus \mathbf{x}_2 = \mathbf{x}_1 - \mathbf{x}_2$	$\mathbf{E}_1 \ominus \mathbf{E}_2 = \mathbf{E}_1 \cdot \mathbf{E}_2^T$
$\log_G(\mathbf{E}) : G \mapsto \mathfrak{g}$	$e = \log_G(\mathbf{E}) = \mathbf{E}$	$e = \log(\mathbf{E}) \triangleq \sum_{k=1}^{\infty} (-1)^{k+1} \frac{(\mathbf{E} - \mathbf{I}_n)^k}{k} = \frac{\arccos\left(\frac{\text{Tr}(\mathbf{E}) - 1}{2}\right)}{\sqrt{4 - (\text{Tr}(\mathbf{E}) - 1)^2}} (\mathbf{E} - \mathbf{E}^T)$
$\exp_G(e) : \mathfrak{g} \mapsto G$	$\mathbf{E} = \exp_G(e) = \int_0^1 e \cdot dt = e$	$\mathbf{E} = \exp(e) \triangleq \sum_{n=0}^{\infty} \frac{e^n}{n!} = \mathbb{I} + \frac{\sin\ e\ _V}{\ e\ _V} e + \frac{1 - \cos\ e\ _V}{\ e\ _V^2} e^2$

TABLE II
NOTATIONS FOR THE KINEMATIC VARIABLES OF THE CENTROID FRAME IN THE WORLD FRAME

	Notation	Definition
Position	\mathbf{p}_l	$\mathbf{R}_\Gamma^T \mathbf{p}_\Gamma$
Orientation	\mathbf{R}	\mathbf{R}_Γ
	Ω	Ω_Γ
Lin. Velocity	\mathbf{v}_l	$\mathbf{R}_\Gamma^T \dot{\mathbf{p}}_\Gamma$
Ang. Velocity	ω_l	$\mathbf{R}_\Gamma^T \dot{\omega}_\Gamma$
Lin. Acceleration	\mathbf{a}_l	$\mathbf{R}_\Gamma^T \ddot{\mathbf{p}}_\Gamma$
Ang. Acceleration	$\dot{\omega}_l$	$\frac{d}{dt} \omega_l = \mathbf{R}_\Gamma^T \dot{\omega}_\Gamma$

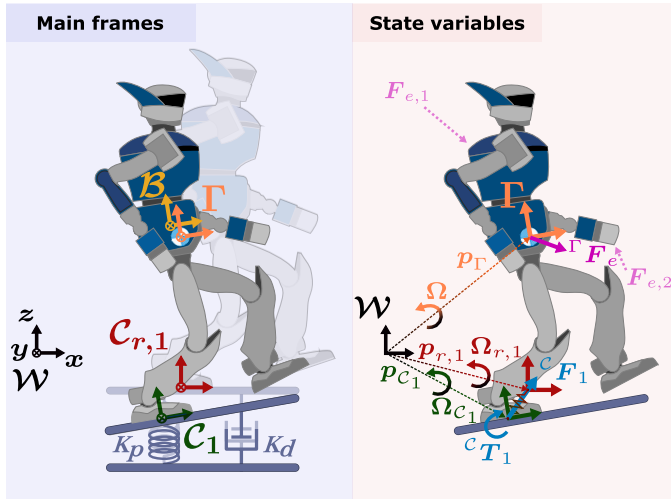


Fig. 2. **Left:** Main frames introduced in the Kinetics Observer. \mathcal{W} : world frame, Γ : centroid frame, \mathcal{B} : floating base's frame, \mathcal{C}_i : i -th contact's frame, $\mathcal{C}_{r,i}$: i -th contact's rest frame. The transparent robot corresponds to the state of the robot at the instant the contact is created and no deformation has occurred yet, while the non-transparent one represents it after the deformation resulted from the contact. This is a virtual case with unrealistic and exaggerated deformations, in order to make the frames clearer. **Right:** Main state variables involved in the Kinetics Observer. For more clarity, the forces and torques are represented using their notation in the state vector.

The robot may also be equipped with none to several IMUs, consisting of an accelerometer and a gyrometer. We denote n_I as the number of available IMU signals. The number of signals delivered may vary in time, for example, if the sensors have different sampling frequencies. The position of the IMUs on the robot is arbitrary. The observer should then estimate the following state components:

- 1) The kinematics of the centroid frame in the world frame $\{\mathbf{p}_l, \Omega, \mathbf{v}_l, \omega_l\}$ together with the predictions of the linear and angular acceleration, \mathbf{a}_l and $\dot{\omega}_l$, respectively. Thanks to frame transformations, this allows the framework to estimate the kinematics of any limb of the robot.
- 2) A contact state $\mathbf{x}_{c,i} \triangleq \{\mathbf{p}_{r,i}, \Omega_{r,i}, {}^c\mathbf{F}_i, {}^c\mathbf{T}_i\}$ for each of the current n_c contacts. Our state vector thus has a dynamic size that changes as the robot creates or breaks contacts with the environment.
- 3) The bias $\mathbf{b}_{g,j}$ that alters the signals of each gyrometer.
- 4) Finally, other external forces and torques $\{\Gamma \mathbf{F}_e, \Gamma \mathbf{T}_e\}$ that are not associated with our model of contacts, expressed in the centroid frame.

The relationships between the different variables described in this section are summarized in Fig. 4 in Section IV.

A. Vector state definition

Our state vector is defined as follows:

$$\mathbf{x} \triangleq \left(\mathbf{p}_l, \Omega, \mathbf{v}_l, \omega_l, \{\mathbf{b}_{g,j}\}_{j=0}^{n_I}, \Gamma \mathbf{F}_e, \Gamma \mathbf{T}_e, \{\mathbf{x}_{c,i}\}_{i=0}^{n_c} \right)^T. \quad (3)$$

It includes components from \mathbb{R}^3 and components of $SO(3)$, defining a Lie group G_x as the state space.

\mathbf{p}_l , \mathbf{v}_l , and ω_l are expressed in the centroid frame. These variables are dynamically linked to each other. Their relations are modeled with a discrete-time state-transition function that allows to predict the future state \mathbf{x}_{k+1} of the system based on the current state \mathbf{x}_k and the system inputs \mathbf{u}_k . In other words,

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k). \quad (4)$$

The measurements vector, defined in $\mathbb{R}^{6(n_I+n_w)}$, is the following:

$$\mathbf{y} \triangleq \left(\{\mathbf{y}_{a,j}, \mathbf{y}_{g,j}\}_{j=0}^{n_I}, \{\mathbf{y}_{F,i}, \mathbf{y}_{T,i}\}_{i=0}^{n_w} \right)^T, \quad (5)$$

with $\mathbf{y}_{a,j}$ and $\mathbf{y}_{g,j}$ the accelerometer and gyrometer measurements of the j -th IMU, respectively. Similarly, $\mathbf{y}_{F,i}$ and $\mathbf{y}_{T,i}$

are the force and torque measurement of the i -th contact's wrench sensor. Since the number of IMU signals n_I and wrench sensor signals n_w can vary over time, the size of the measurement vector will evolve accordingly. The measurements can also be predicted using a model:

$$\mathbf{y}_k = g(\mathbf{x}_k, \mathbf{u}_k). \quad (6)$$

The input \mathbf{u} is composed of dynamic variables of the system and information about the considered contacts and IMUs:

$$\mathbf{u} \triangleq \left(\begin{matrix} \Gamma \mathbf{I}, \Gamma \dot{\mathbf{I}}, \Gamma \boldsymbol{\sigma}, \Gamma \dot{\boldsymbol{\sigma}}, \Gamma \mathbf{F}_{res}, \Gamma \mathbf{T}_{res}, \{\Xi_i\}_{i=0}^{n_c}, \{\Psi_j\}_{j=0}^{n_I} \end{matrix} \right)^T, \quad (7)$$

with $\Gamma \mathbf{I}$, $\Gamma \dot{\mathbf{I}}$, $\Gamma \boldsymbol{\sigma}$ and $\Gamma \dot{\boldsymbol{\sigma}}$ being the total inertia matrix, the total angular momentum of the robot expressed at its centroid, and their derivatives. $\{\Gamma \mathbf{F}_{res}, \Gamma \mathbf{T}_{res}\}$ is the resultant wrench measured by the sensors not associated with currently set contacts, expressed in the centroid frame. This allows us to take advantage of all force sensor measurements, especially those mounted on limbs that interact with the environment but not through contacts. For example, this would allow us to consider the wrench applied to the the robot's hand in the case of interaction with humans or objects. This wrench also helps to ensure the continuity of the total wrench exerted on the robot even when creating or breaking contacts, since when a contact is broken, the associated wrench measurement would simply be added to this input, and vice versa.

$\Xi_i \triangleq \{\check{\mathbf{p}}_{r,i}, \check{\boldsymbol{\Omega}}_{r,i}, \Gamma \mathbf{p}_{C_i}, \Gamma \boldsymbol{\Omega}_{C_i}, \Gamma \dot{\mathbf{p}}_{C_i}, \Gamma \boldsymbol{\omega}_{C_i}\}$ correspond to the input variables associated with each contact i . $\{\Gamma \mathbf{p}_{C_i}, \Gamma \boldsymbol{\Omega}_{C_i}, \Gamma \dot{\mathbf{p}}_{C_i}, \Gamma \boldsymbol{\omega}_{C_i}\}$ are the kinematics of the frame of the i -th contact in the centroid frame and $\{\check{\mathbf{p}}_{r,i}, \check{\boldsymbol{\Omega}}_{r,i}\}$ is the initial value of its rest pose in the world frame at the time it is created, used to initialize the contact in the state vector. This value is an 'impulsional input', meaning that it is given only at the time a new contact is set with the environment. $\Psi_j \triangleq \{\Gamma \mathbf{p}_{S_j}, \Gamma \boldsymbol{\Omega}_{S_j}, \Gamma \dot{\mathbf{p}}_{S_j}, \Gamma \boldsymbol{\omega}_{S_j}, \Gamma \ddot{\mathbf{p}}_{S_j}\}$ are the input variables associated with each IMU j . They correspond respectively to the pose, the velocities, and the linear acceleration of the j -th IMU's frame S_j in the centroid frame. Since the latter acceleration depends on the joint accelerations, which are not measurable, we compute it from the reference joint accelerations used to control the robot. The other variables can be obtained from the joint encoders and their time-derivative.

B. State transition model

1) *Kinematics of the centroid frame:* The prediction of the kinematic variables of the centroid frame is done by the discrete integration of the current ones. The latter include the linear and angular accelerations, $\mathbf{a}_{l,k}$ and $\dot{\boldsymbol{\omega}}_{l,k}$ respectively, which are considered constant over the integration interval δ_T . This integration is allowed by the Lie group properties

of SE(3):

$$\begin{cases} \mathbf{p}_{l,k+1} = \text{Exp}(\boldsymbol{\theta}_k)^T \mathbf{p}_{l,k} + \mathbf{V}_k^T \left(\delta_T \mathbf{v}_{l,k} + \frac{\delta_T^2}{2} \mathbf{a}_{l,k} \right), & (8) \\ \mathbf{R}_{k+1} = \mathbf{R}_k \text{Exp}(\boldsymbol{\theta}_k), & (9) \\ \mathbf{v}_{l,k+1} = \mathbf{v}_{l,k} + \delta_T \left(-[\boldsymbol{\omega}_{l,k}]_{\times} \mathbf{v}_{l,k} + \mathbf{a}_{l,k} \right), & (10) \\ \boldsymbol{\omega}_{l,k+1} = \boldsymbol{\omega}_{l,k} + \delta_T \dot{\boldsymbol{\omega}}_{l,k}, & (11) \end{cases}$$

With $\boldsymbol{\theta}_k = \delta_T \boldsymbol{\omega}_{l,k} + \frac{\delta_T^2}{2} \dot{\boldsymbol{\omega}}_{l,k}$. The matrix $\mathbf{V}(\boldsymbol{\theta})$ is the one defined in the exponential map of the SE(3) Lie group [51].

$$\mathbf{V}(\boldsymbol{\theta}) = \mathbb{I} + \frac{(1 - \cos(\|\boldsymbol{\theta}\|))}{\|\boldsymbol{\theta}\|^2} [\boldsymbol{\theta}]_{\times} + \frac{\|\boldsymbol{\theta}\| - \sin(\|\boldsymbol{\theta}\|)}{\|\boldsymbol{\theta}\|^3} [\boldsymbol{\theta}]_{\times}^2 \quad (12)$$

This term allows to integrate the linear velocity expressed in the centroid frame into the position while taking into account the evolution of the orientation over the integration integral. The accelerations $\mathbf{a}_{l,k} = f_a(\mathbf{x}_k, \mathbf{u}_k)$ and $\dot{\boldsymbol{\omega}}_{l,k} = f_{\dot{\boldsymbol{\omega}}}(\mathbf{x}_k, \mathbf{u}_k)$, are obtained from Newton-Euler's equations (14) and (16). They are thus expressed as functions of the state, in particular of the kinematics of the centroid and of the external and contact wrenches. By integrating these accelerations to obtain the new state kinematics, we obtain a very tight coupling between our state kinematics and the wrenches.

This method differs from the usual state-of-the-art ones in that we don't directly use the angular velocity and linear acceleration measured by an IMU input to integrate the kinematics. Also, we ensure the coupling directly in the modeling of our system.

a) *Newton's equations for multi-body systems:* Considering the robot as a rigid body, the linear acceleration of the centroid frame in the world frame can be expressed from the forces \mathbf{F} applied to this point by using the Newton's relation

$$\mathbf{F} = m \ddot{\mathbf{p}}, \quad (13)$$

where m is the total mass of the robot, and $\ddot{\mathbf{p}}$ is the linear acceleration of the centroid frame in the world frame. Note that considering the centroid frame allows us to eliminate the inertial effects due to the distance of a point from the center of mass of a moving object, drastically simplifying the expressions.

We can then write the linear acceleration of the centroid frame in the world frame, expressed in the centroid frame:

$$\mathbf{a}_l = \mathbf{R}^T \ddot{\mathbf{p}} = \frac{\Gamma \mathbf{F}_{res} + \Gamma \mathbf{F}_e + \sum_{i=0}^{n_c} \Gamma \mathbf{R}_{C_i} {}^C \mathbf{F}_i}{m} - g_0 \mathbf{R}^T \mathbf{e}_z, \quad (14)$$

where $\Gamma \mathbf{F}_e$ corresponds to the estimated unmodeled external force applied to the centroid and $\Gamma \mathbf{F}_{res}$ is the resulting force measured by the sensors not associated with contacts, expressed at the centroid. ${}^C \mathbf{F}_i$ is the estimated force at the contact i , and $\Gamma \mathbf{R}_{C_i}$ is the input orientation of the contact in the centroid frame. Also, g_0 is the gravitational acceleration constant.

b) *Euler's equations for a multi-body system:* The rotational dynamics can be expressed by Euler's relation:

$$\mathbf{T} = \frac{d}{dt} \left(\sum_{b=1}^{n_b} \left(\mathbf{R}_b \mathbf{I}_b \mathbf{R}_b^T \boldsymbol{\omega}_b + m_b [c_b]_{\times} \dot{c}_b \right) \right), \quad (15)$$

where \mathbf{T} is the sum of the external torques/moments applied to the system expressed at the robot's center of mass, and n_b is the number of bodies composing the robot. \mathbf{R}_b and $\boldsymbol{\omega}_b$ are the orientation and the angular velocity in the world frame of the body b . m_b and \mathbf{I}_b are its mass and its inertia matrix expressed in its frame, and \mathbf{c}_b is the translation vector from the robot's center of mass to the center of mass of the body b , expressed in the world frame. This expression can be developed to give the following angular acceleration of the centroid frame in the world frame, expressed in the centroid frame:

$$\begin{aligned} \dot{\boldsymbol{\omega}}_l &= \mathbf{R}^T \dot{\boldsymbol{\omega}} \\ &= {}^\Gamma \mathbf{I}^{-1} \left({}^\Gamma \mathbf{T}_{res} + {}^\Gamma \mathbf{T}_c + {}^\Gamma \mathbf{T}_e - {}^\Gamma \dot{\mathbf{I}} \boldsymbol{\omega}_l - {}^\Gamma \dot{\boldsymbol{\sigma}} - [{}^\Gamma \boldsymbol{\omega}_l]_\times ({}^\Gamma \mathbf{I} \boldsymbol{\omega}_l + {}^\Gamma \boldsymbol{\sigma}) \right), \end{aligned} \quad (16)$$

where ${}^\Gamma \mathbf{I}$ and ${}^\Gamma \boldsymbol{\sigma}$ are the input inertia matrix and angular momentum of the multi-body robot expressed in the centroid frame, with their respective derivatives ${}^\Gamma \dot{\mathbf{I}}$ and ${}^\Gamma \dot{\boldsymbol{\sigma}}$. ${}^\Gamma \mathbf{T}_e$ is our estimate of the unmodeled external torque applied on the robot and expressed in the centroid frame, and ${}^\Gamma \mathbf{T}_{res}$ is the resulting torque measured by the sensors not associated with contacts, expressed in the centroid frame. Finally, ${}^\Gamma \mathbf{T}_c$ is the total contact torque, expressed in the centroid frame and defined by

$${}^\Gamma \mathbf{T}_c = \sum_{i=0}^{n_c} \left({}^\Gamma \mathbf{R}_{C_i} {}^C \mathbf{T}_i + [{}^\Gamma \mathbf{p}_{C_i}]_\times {}^\Gamma \mathbf{R}_{C_i} {}^C \mathbf{F}_i \right), \quad (17)$$

with $\{{}^C \mathbf{F}_i, {}^C \mathbf{T}_i\}$ the estimated wrench at the contact i and $\{{}^\Gamma \mathbf{p}_{C_i}, {}^\Gamma \mathbf{R}_{C_i}\}$ the input pose of the contact i in the centroid frame.

2) Gyrometer bias, external wrench and contacts rest pose:

The bias on each gyrometer is assumed to be unpredictable and subject to small variations over short periods of time. Similarly, the external wrench and the rest poses of the contacts are assumed to have slow variations over time. Therefore, the state transition model is considered constant for all these variables. This constant prediction is corrected at the sensor-update phase of the estimation.

3) Viscoelastic model of the contacts :

We cannot rely solely on the force measurements to predict the acceleration of our system. This is firstly because the measurements may not be available, but secondly and more importantly, relying only on force estimates to predict the accelerations leads to unbounded drifts in the absolute position. This is due to the uncertainties in the sensor measurements and the robot models. Nevertheless, we know that the robot has relatively reliable anchors in the environment: the contacts. However, integrating kinematically this information would conflict with the Newton-Euler dynamics. The correction of the drifts must, therefore, be applied through a wrench at the contact. To this end, we predict the contact reaction wrench using the viscoelastic model of the contacts, comparable to the one we defined in [2], which links the reaction wrench to the estimated contact poses. A contact is modeled as a {spring + damper} system between its current estimated frame \mathcal{C}_i and its rest frame $\mathcal{C}_{r,i}$, which corresponds to the pose of the environment-attached frame before the contact. The following simplified example illustrates this contact model. At the exact instant the body touches the environment, there is no force and no deformation yet and thus $\mathcal{C}_{r,i} = \mathcal{C}_i$. Afterwards, $\mathcal{C}_{r,i}$ remains constant while

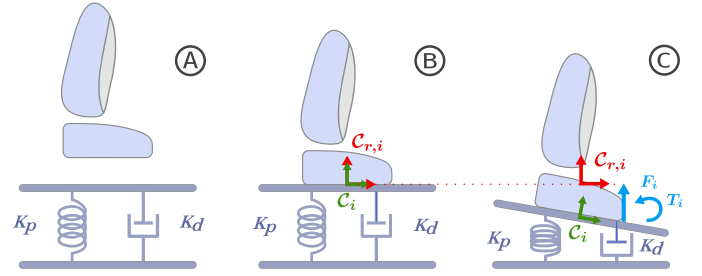


Fig. 3. Viscoelastic model of contacts. (A) Foot before the contact with the ground. (B) Creation of the contact. No force is applied, the current contact frame matches the rest frame. (C) Generation of a reaction wrench due to the deformation.

\mathcal{C}_i moves due to the deformation of the environment, which generates reaction forces². This is illustrated in Fig. 3.

The rest kinematics $\{\mathbf{p}_{r,i}, \mathbf{R}_{r,i}, \mathbf{0}, \mathbf{0}\}$ of the contact in the world are composed of the rest pose of the contact, which is part of our state vector, and of zero linear and angular velocities. The current kinematics $\{\mathbf{p}_{C_i}, \mathbf{R}_{C_i}, \dot{\mathbf{p}}_{C_i}, \boldsymbol{\omega}_{C_i}\}$ of the contact's frame \mathcal{C}_i in the world frame are obtained by forward kinematics from the current estimate of the centroid frame's kinematics in the world frame:

$$\begin{cases} \mathbf{p}_{C_i} = \mathbf{R} ({}^\Gamma \mathbf{p}_{C_i} + \mathbf{p}_l) & (18) \\ \mathbf{R}_{C_i} = \mathbf{R} {}^\Gamma \mathbf{R}_{C_i} & (19) \end{cases}$$

$$\begin{cases} \dot{\mathbf{p}}_{C_i} = \mathbf{R} ({}^\Gamma \dot{\mathbf{p}}_{C_i} + [{}^\Gamma \boldsymbol{\omega}_l]_\times {}^\Gamma \mathbf{p}_{C_i} + \mathbf{v}_l) & (20) \\ \boldsymbol{\omega}_{C_i} = \mathbf{R} ({}^\Gamma \boldsymbol{\omega}_{C_i} + \boldsymbol{\omega}_l) & (21) \end{cases}$$

with $\{{}^\Gamma \mathbf{p}_{C_i}, {}^\Gamma \mathbf{R}_{C_i}\}$ and $\{{}^\Gamma \dot{\mathbf{p}}_{C_i}, {}^\Gamma \boldsymbol{\omega}_{C_i}\}$ the input pose and velocity of the contact in the centroid frame, respectively. As a reminder, $\{\mathbf{p}_l, \mathbf{R}\}$ and $\{\mathbf{v}_l, \boldsymbol{\omega}_l\}$ are the state pose and velocity of the centroid frame in the world frame, expressed in the centroid frame. The discrepancy between the current and the rest kinematics can be divided into a linear part $\{\tilde{\mathbf{p}}_i, \tilde{\mathbf{v}}_i\}$ and an angular part $\{\tilde{\mathbf{R}}_i, \tilde{\boldsymbol{\omega}}_i\}$:

$$\begin{cases} \tilde{\mathbf{p}}_i = \mathbf{p}_{C_i} - \mathbf{p}_{r,i} & (22) \\ \tilde{\mathbf{v}}_i = \dot{\mathbf{p}}_{C_i} - \mathbf{0} = \dot{\mathbf{p}}_{C_i} & (23) \end{cases}$$

$$\begin{cases} \tilde{\mathbf{R}}_i = \mathbf{R}_{C_i} \mathbf{R}_{r,i}^T & (24) \\ \tilde{\boldsymbol{\omega}}_i = \boldsymbol{\omega}_{C_i} - \mathbf{0} = \boldsymbol{\omega}_{C_i}. & (25) \end{cases}$$

Using the viscoelastic model of the contacts, the linear discrepancy yields a contact force and the angular discrepancy results in a contact torque. The contact reaction wrench expressed in the contact's frame \mathcal{C}_i is thus:

$${}^C \mathbf{F}_i = -\mathbf{R}_{C_i}^T (K_{p,t} \tilde{\mathbf{p}}_i + K_{d,t} \tilde{\mathbf{v}}_i) \quad (26)$$

$${}^C \mathbf{T}_i = -\mathbf{R}_{C_i}^T \left(\frac{1}{2} K_{p,r} [\tilde{\mathbf{R}}_i - \tilde{\mathbf{R}}_i^T]_{\vee} + K_{d,r} \tilde{\boldsymbol{\omega}}_i \right) \quad (27)$$

where $K_{p,t}$ and $K_{d,t}$ are the 3×3 SPD matrices corresponding to the linear stiffness and damping of the contact.

²In this simplified example, we consider the viscoelastic model represents only the environment's flexibility. This simplification intends to make it easier to apprehend, and ignores the structural flexibility of the robot which also contributes to the deformation. Moreover, the presence of damping may produce discontinuous forces at impact.

Similarly, $\mathbf{K}_{p,r}$ and $\mathbf{K}_{d,r}$ correspond to the angular stiffness and damping of the contact. Note that the expression of the contact torque relies on the property that for a rotation matrix \mathbf{R} we have

$$\frac{1}{2} \left[\mathbf{R} - \mathbf{R}^T \right]_{\vee} = \frac{\sin(\|\text{Log}(\mathbf{R})\|)}{\|\text{Log}(\mathbf{R})\|} \text{Log}(\mathbf{R}), \quad (28)$$

which approximates $\text{Log}(\mathbf{R})$ (the equivalent rotation vector) for small angles of \mathbf{R} , using the *vee* operator defined in Section II.

In the proposed estimator, we use this model for contact wrench prediction. Therefore, the contact wrench state transition depends only on the kinematic part of the current state. This representation of the contacts allows us also to create a coupling between the centroid frame's kinematics and the rest pose of the contacts. A difference between the predicted and measured wrenches could be either caused by a slip of the contacts or by an estimation error in the centroid kinematics. The estimator has to use the measurements to update the state accordingly, which corrects partially the drifts (due to slippage) and the robot kinematics estimation error.

In summary, the viscoelastic representation of contacts has the following strengths:

- It ensures non-divergence of the kinematics of the centroid frame from the contacts by associating their difference with a proportional reaction. It also provides a coupling between the centroid kinematics, the forces and the contact position.
- In the absence of wrench sensors on the contacts, it allows us to estimate the wrench based on the kinematics.
- By including the pose of the rest frame $\mathcal{C}_{r,i}$ in the state, we can correct it and thus cope with slippage, moving contacts and terrain unevenness.
- It considers contact orientation, not just a position, as done for point contacts. Nevertheless, point contacts can still be represented by simply setting the angular stiffness $\mathbf{K}_{p,r}$ and damping $\mathbf{K}_{d,r}$ to zero.

C. Measurements model

The measurements can be predicted using the current states and inputs $\mathbf{y}_k = g(\mathbf{x}_k, \mathbf{u}_k)$. The estimated measurements of the wrench sensor at each contact i correspond to the associated contact wrench in the state vector:

$$\mathbf{y}_{F,i} = {}^c \mathbf{F}_i, \quad (29)$$

$$\mathbf{y}_{T,i} = {}^c \mathbf{T}_i. \quad (30)$$

We predict the biased gyrometer and the accelerometer measurements of each IMU j using forward kinematics:

$$\mathbf{y}_{g,j} = {}^\Gamma \mathbf{R}_{S_j}^T (\boldsymbol{\omega}_l + {}^\Gamma \boldsymbol{\omega}_{S_j}) + \mathbf{b}_{g,j}, \quad (31)$$

$$\begin{aligned} \mathbf{y}_{a,j} = & {}^\Gamma \mathbf{R}_{S_j}^T \left(\mathbf{a}_l + {}^\Gamma \ddot{\mathbf{p}}_{S_j} + \mathbf{R}^T g_0 \mathbf{e}_z \right) \\ & + {}^\Gamma \mathbf{R}_{S_j}^T \left(\left([\boldsymbol{\omega}_l]_{\times}^2 + [\dot{\boldsymbol{\omega}}_l]_{\times} \right) {}^\Gamma \mathbf{p}_{S_j} + 2 [\boldsymbol{\omega}_l]_{\times} {}^\Gamma \dot{\mathbf{p}}_{S_j} \right), \end{aligned} \quad (32)$$

where $\boldsymbol{\omega}_l$ is the state angular velocity of the centroid frame in the world frame, expressed in the centroid frame, and $\mathbf{b}_{g,j}$ is the estimated bias on the gyrometer measurement from the

IMU. $\left\{ {}^\Gamma \mathbf{p}_{S_j}, {}^\Gamma \mathbf{R}_{S_j}, {}^\Gamma \dot{\mathbf{p}}_{S_j}, {}^\Gamma \boldsymbol{\omega}_{S_j}, {}^\Gamma \ddot{\mathbf{p}}_{S_j} \right\}$ are the pose, linear and angular velocities, and linear acceleration of the IMU j in the centroid frame, available in \mathbf{u}_k . The linear and angular accelerations $\{\mathbf{a}_l, \dot{\boldsymbol{\omega}}_l\}$ of the centroid frame in the world frame, expressed in the world frame, are obtained from (14) and (16), respectively. Based on these relations, the prediction of the accelerometer measurement is therefore a function of the total wrench exerted on the robot, which is estimated in our state vector. Our accelerometer is thus comparable to an additional total force sensor, which embraces the principle of estimating the *kinetics* of the robot.

IV. THE KINETICS OBSERVER

The Kinetics Observer is a global estimator designed to simultaneously estimate the variables describing the interactions with the environment. At its core, is a Multiplicative Extended Kalman Filter (MEKF), where the term multiplicative refers to the use of matrix Lie groups within an EKF. The structure of the Kinetics Observer, along with the main variables it relies on, is summarized in Fig. 4.

Since our state space G_x is a Lie group, we can express interdependence of the state variables by a single tangent space. This gives us a mathematically consistent model, especially for the propagation of the state covariance.

The Kalman Filter implementation consists of two main steps: the prediction and the update/innovation. During the prediction step, the state-transition model is applied to the current estimate $\hat{\mathbf{x}}_k$ of the state and the current system inputs \mathbf{u}_k to predict the future state $\bar{\mathbf{x}}_{k+1|k}$, and the measurement model is used to predict the measurements $\bar{\mathbf{y}}_{k+1|k}$ of the sensors in that predicted state:

$$\mathbf{A} = \left(\frac{\partial f}{\partial \mathbf{x}} (\mathbf{x}, \mathbf{u}) \right)_{\mathbf{x}=\hat{\mathbf{x}}_k, \mathbf{u}=\mathbf{u}_k}, \quad (33)$$

$$\bar{\mathbf{x}}_{k+1|k} = f(\hat{\mathbf{x}}_k, \mathbf{u}_k), \quad (34)$$

$$\bar{\mathbf{y}}_{k+1|k} = g(\bar{\mathbf{x}}_{k+1|k}, \mathbf{u}_k), \quad (35)$$

$$\mathbf{P}_{k+1|k} = \mathbf{A} \mathbf{P}_{k|k} \mathbf{A}^T + \mathbf{Q}, \quad (36)$$

$$\mathbf{C} = \left(\frac{\partial g}{\partial \mathbf{x}} (\mathbf{x}, \mathbf{u}) \right)_{\mathbf{x}=\hat{\mathbf{x}}_k, \mathbf{u}=\mathbf{u}_k}. \quad (37)$$

The matrices \mathbf{A} and \mathbf{C} are the state-transition and the measurement matrices of the system, respectively, resulting from the linearization of the model. This linearization can be expressed analytically or computed by finite differences. The first method is more tedious to implement, but it allows for a much faster computation and thus to run the estimator in real-time despite the large number of variables to estimate. \mathbf{Q} and \mathbf{R} correspond to the covariance matrices of the Gaussian noise associated with the state-transition and measurement models. For simpler formulations, we will refer to the matrix \mathbf{Q} as the *process noise covariance matrix*, and the diagonal term associated with each of the state variables as its *process noise variance*. Similarly, we will now refer to the matrix \mathbf{R} as the *measurement noise covariance matrix*, and its diagonal terms as the *measurement noise variances*. \mathbf{P} is the covariance matrix associated with the state estimate, which we will call the *state covariance matrix*.

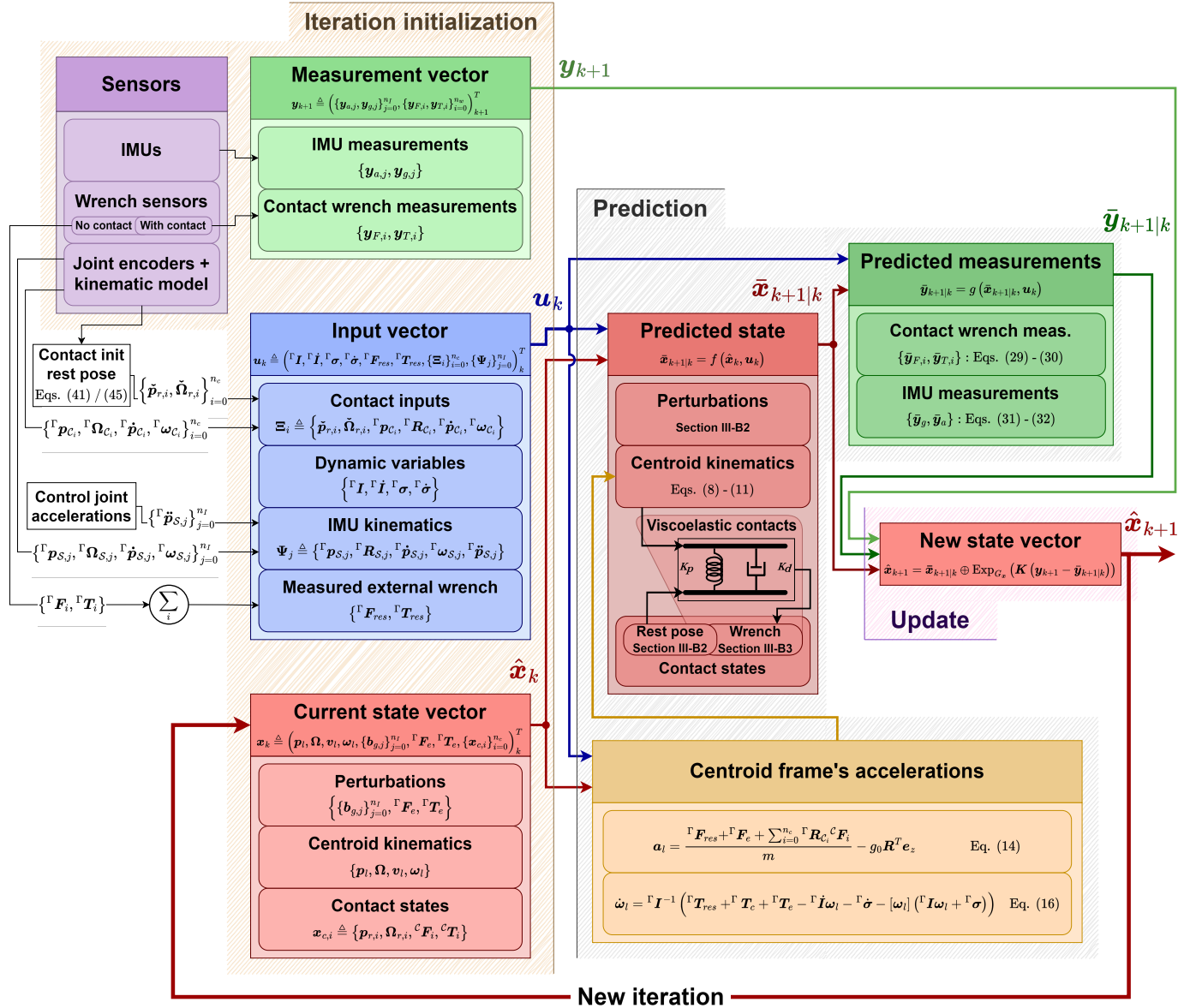


Fig. 4. Summary of the structure of the Kinetics Observer. We present the main elements that make up the Kalman Filter with a focus on the dynamical model and the origin of each variable involved.

The update step performs the innovation, which weighs the contribution of the predicted state and the measurements thanks to the Kalman gain K to obtain the corrected estimated state \hat{x}_{k+1} , and computes the newly obtained covariance on the state estimate for the next iteration of the MEKF:

$$K = P_{k+1|k} C^T (C P_{k+1|k} C^T + R)^{-1}, \quad (38)$$

$$\hat{x}_{k+1} = \bar{x}_{k+1|k} \oplus \text{Exp}_{G_x} \left(K (y_{k+1} - \bar{y}_{k+1|k}) \right), \quad (39)$$

$$P_{k+1|k+1} = (\mathbb{I} - KC) P_{k+1|k} (\mathbb{I} - KC)^T + KRK^T \quad (40)$$

where \oplus is the Lie group operator defined in Section II. As a reminder, $\text{Exp}_{G_x}(\cdot)$ is the operator that converts the rotation components of the state's tangent vector to rotation matrices.

A. Kinetics Observer's odometry

In order to perform odometry, the Kinetics Observer requires the 'impulsional' input $\{\tilde{p}_{r,i}, \tilde{\Omega}_{r,i}\}$, which corresponds to the initial guess on the rest pose $\{p_{r,i}, \Omega_{r,i}\}$ in the world of the successive contacts, which is used to initialize them in the state vector. As a reminder, the contact's rest pose is the pose of the contact's rest frame, which was the frame attached to the contact when the contact wrench was zero. Since a wrench is already applied to the contact when it is detected, its current pose $\{p_{c,i}, R_{c,i}\}$ obtained through forward kinematics with (18) and (19) does not match its rest pose anymore due to the deformation of the contact material (Fig. 3). To recover the rest pose, it is necessary to identify and compensate for this deformation. We get this deformation initial value by inverting the viscoelastic model from the current wrench measurement $\{y_{F,i}, y_{T,i}\}$. In other words, from (26) we

obtain the translational deformation and add it to \mathbf{p}_{C_i} to get the initial guess on the rest position:

$$\check{\mathbf{p}}_{r,i} = \mathbf{p}_{C_i} + \mathbf{R}_{C_i} \mathbf{K}_{p,t}^{-1} \left(\mathbf{y}_{F,i} + \mathbf{R}_{C_i}^T \mathbf{K}_{d,t} \dot{\mathbf{p}}_{C_i} \right). \quad (41)$$

Next, we will explain how we obtain the orientation discrepancy from (27). Let us define the term $\mathbf{D}_i = \tilde{\mathbf{R}}_i - \check{\mathbf{R}}_i^T$, where $\check{\mathbf{R}}_i = \mathbf{R}_{C_i} \check{\mathbf{R}}_{r,i}^T$ is the discrepancy between the current and the rest orientation of the contacts as defined in (24). From (27), we have:

$$[\mathbf{D}_i]_{\vee} = -2\mathbf{R}_{C_i} \mathbf{K}_{p,r}^{-1} \left(\mathbf{y}_{T,i} + \mathbf{R}_{C_i}^T \mathbf{K}_{d,r} \boldsymbol{\omega}_{C_i} \right). \quad (42)$$

\mathbf{D}_i also respects the property (28), which gives

$$\phi_i = \arcsin \left(\frac{\|[\mathbf{D}_i]_{\vee}\|}{2} \right), \quad (43)$$

$$\mathbf{u}_i = \frac{[\mathbf{D}_i]_{\vee}}{\|[\mathbf{D}_i]_{\vee}\|}, \quad (44)$$

where ϕ_i and \mathbf{u}_i are the rotation axis and angle corresponding to $\tilde{\mathbf{R}}_i$, allowing to compute this latter matrix and to get the contact rest orientation:

$$\check{\mathbf{R}}_{r,i} = \tilde{\mathbf{R}}_i^T \mathbf{R}_{C_i}. \quad (45)$$

$\{\check{\mathbf{p}}_{r,i}, \check{\boldsymbol{\Omega}}_{r,i}\}$ is finally used to initialize the rest pose of the contact in the state vector. Note that since the position along the vertical axis z is not observable, small displacements resulting from the IMU integration and the viscoelastic behavior of the contacts may accumulate at each step, and cannot be corrected by the observer. This would lead to significant drifts over long walks. To solve this problem when walking on a flat ground, one can constrain the robot to remain on the ground by initializing the rest position along the vertical axis and the associated initial state variance of all newly set contacts to zero.

Finally, it is important to note that this estimator can be used without odometry. This is useful when we don't want the robot estimate to drift from the reference plan, but we still need to observe the local state of the robot, for example to control balance and locomotion. In such a case it is sufficient to set

$$\{\check{\mathbf{p}}_{r,i}, \check{\boldsymbol{\Omega}}_{r,i}\} = \{\mathbf{p}_{c,i}^*, \boldsymbol{\Omega}_{c,i}^*\}, \quad (46)$$

where $\mathbf{p}_{c,i}^*$ and $\boldsymbol{\Omega}_{c,i}^*$ are the reference contact position and orientation provided by the contact planner.

B. Covariance tuning of the Kinetics Observer

The main challenges in implementing the Kinetics Observer are tuning the covariances involved in the Kalman filter and identifying the stiffness and damping of the viscoelastic model. Here we provide some insight into the process. The covariance matrices are constructed by combining the covariance submatrices corresponding to each state and measurement variable along the three axes. These submatrices are considered diagonal, meaning that we define only the variance on each variable and assume that they are independent of each other. The measurement variances are obtained from noise models on the sensors.

In this paragraph, we want to highlight the role of the covariances associated with contacts, since they play a crucial role in the behavior of our estimator, especially of our odometry. Taking into account the expected discrepancy between our model and the reality, we have determined the initial variance associated with the contact rest pose. This allows for the correction of the rest pose upon the creation of the contact. The process noise covariance submatrix associated with the contact rest poses is designed to respect specific constraints, as we further develop in Section IV-C. The state-transition model associated with the contact wrenches is our viscoelastic model defined in Section III-B3. Since this model analytically relates the contact wrench to the current state variables, its corresponding process noise variances directly correspond to the confidence in the viscoelastic model. While the viscoelastic model is a useful approximation, it is evident that the wrench sensors are more reliable, which explains why the process noise variances on the contact wrenches are higher than their measurement noise variances. All the parameters that we tuned for our experiments are summarized up in Table III. Note that the parameters used are the same for both robots involved in our experiments (HRP-5P [52] and RHP Friends [53]) in three very different scenarios to show the robustness of the estimation with respect to these parameters.

Finally, as an important note, the covariances of new contact poses are initialized with a fixed value even in odometry mode. This is not consistent with reality since the position of every step is supposed to be more uncertain than the previous one, and thus, the covariances should add up. This imprecise choice has been made to prevent the covariance from building up and causing instability problems during long experiments. However, this care can be dropped if an additional absolute pose measurement (e.g., GPS or SLAM) is added to the estimator, since in this case the system would be observable, and the covariance would remain bounded.

C. Constrained process noise covariance sub-matrix for the contact poses

The initial variance on the contact's rest pose in the state covariance matrix upon its creation allows for its immediate correction by the Kalman Filter. However, the possibility of this correction decreases as the uncertainty on these variables converges, and is thus insufficient in the presence of slippage even shortly after the contact is created. Associating a process noise covariance to our model of fixed contact rest poses (Section III-B2) would help address this issue. However, this would allow these poses to drift freely over time, hindering the main anchor purpose of the contacts. For this reason, in the case of simultaneous multiple contacts, we introduce a specially constrained process noise covariance matrix for the sub-state of contact rest positions and another for the sub-state of contact rest orientations. These matrices are designed so that the process noise on the *average* rest position and yaw of the contacts (the unobservable part of the rest pose) is zero. Thus, they allow for the correction of the contact's rest pose over its entire lifetime (in the presence of other contacts), while ensuring that the process noise cannot cause the average rest

TABLE III
TUNED PARAMETERS OF THE KINETICS OBSERVER (UNITS OMITTED)

	Initial state variances	Process noise variances
Position \mathbf{p}_l	\mathbb{O}	$10^{-10}.\mathbb{I}$
Orientation \mathbf{R}	$\text{diag}(1.10^{-2}, 1.10^{-2}, 0)$ ⁽¹⁾	$10^{-12}.\mathbb{I}$
Lin. Velocity \mathbf{v}_l	\mathbb{O}	$10^{-10}.\mathbb{I}$
Ang. Velocity $\boldsymbol{\omega}_l$	\mathbb{O}	$10^{-12}.\mathbb{I}$
Gyrometer bias $\mathbf{b}_{g,j}$	$10^{-8}.\mathbb{I}$	$10^{-18}.\mathbb{I}$
Unmodeled force \mathbf{F}_e	\mathbb{O}	$9.10^{-2}.\mathbb{I}$
Unmodeled torque \mathbf{T}_e	\mathbb{O}	$5.10^{-2}.\mathbb{I}$
Contact rest pos. $\mathbf{p}_{r,i}$	$10^{-6}.\mathbb{I}$	$1.10^{-10}.\mathbb{I}$
Contact rest ori. $\mathbf{R}_{r,i}$	$10^{-6}.\mathbb{I}$	$\text{diag}(0, 0, 1.10^{-8})$
Contact force ${}^C\mathbf{F}_i$	$400.\mathbb{I}$	$100.\mathbb{I}$
Contact torque ${}^C\mathbf{T}_i$	$360.\mathbb{I}$	$25.\mathbb{I}$
Measurement noise variances (HRP-5P RHP Friends)		
Gyrometer	$25.10^{-8}.\mathbb{I}$	$16.10^{-6}.\mathbb{I}$
Accelerometer	$25.10^{-4}.\mathbb{I}$	$36.10^{-4}.\mathbb{I}$
Force sensors	$1.\mathbb{I}$	$1.\mathbb{I}$
Torque sensors	$9.10^{-4}.\mathbb{I}$	$1.10^{-2}.\mathbb{I}$
Contact flexibilities (HRP-5P RHP Friends)		
Linear stiffness $\mathbf{K}_{p,t}$ [N/m]	$3.10^5.\mathbb{I}$	$3.10^4.\mathbb{I}$
Linear damping $\mathbf{K}_{d,t}$ [N.s/m]	$150.\mathbb{I}$	$150.\mathbb{I}$
Angular stiffness $\mathbf{K}_{p,r}$ [N.m/rad]	$1000.\mathbb{I}$	$727.\mathbb{I}$
Angular damping $\mathbf{K}_{d,r}$ [N.m.s/rad]	$17.\mathbb{I}$	$17.\mathbb{I}$

(1) $\text{diag}()$ is the operator that transforms a \mathbb{R}^n vector into a $\mathbb{R}^{n \times n}$ diagonal matrix whose diagonal terms correspond to the vector components.

pose to drift. We will refer to this feature as the *constrained contact process covariances*.

Let us first consider the substate consisting only of the rest positions \mathbf{X} of the n contacts currently set with the environment:

$$\begin{pmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_n \end{pmatrix}_{k+1} = \begin{pmatrix} \mathbf{X}_1 \\ \vdots \\ \mathbf{X}_n \end{pmatrix}_k + \mathbf{M}\mathbf{v}, \quad (47)$$

with $\mathbf{M}\mathbf{v}$ the noise on our state-transition model. \mathbf{v} is of dimension $1 \times 3n$ and of rank $1 \times 3(n-1)$ with n the number of contacts and \mathbf{M} of dimension $3n \times 3n$. We want the covariance of the sum of this noise over each axis to be zero. This constraint imposes a constant average position of the contacts in our state-transition model. This constraint can be written as

$$\text{cov}(\mathbf{S}^T \mathbf{M}\mathbf{v}) = \mathbf{0}, \quad (48)$$

with \mathbf{S}^T the matrix that computes this sum by multiplication:

$$\mathbf{S}^T = \begin{pmatrix} \mathbb{I}_{3 \times 3} & \cdots & \mathbb{I}_{3 \times 3} \end{pmatrix} \in \mathbb{R}^{3 \times 3n}. \quad (49)$$

We can write

$$\text{cov}(\mathbf{S}^T \mathbf{M}\mathbf{v}) = \mathbf{S}^T \text{cov}(\mathbf{M}\mathbf{v}) \mathbf{S}, \quad (50)$$

and

$$\text{cov}(\mathbf{M}\mathbf{v}) = \mathbf{M} \text{cov}(\mathbf{v}) \mathbf{M}^T. \quad (51)$$

From (50) and (51), we obtain

$$\text{cov}(\mathbf{S}^T \mathbf{M}\mathbf{v}) = \mathbf{S}^T \mathbf{M} \text{cov}(\mathbf{v}) \mathbf{M}^T \mathbf{S} = \mathbf{0}. \quad (52)$$

This must be true for any vector \mathbf{v} , which is the case only if $\mathbf{S}^T \mathbf{M} = \mathbf{0}$. We can now express the matrix \mathbf{M} ,

$$\mathbf{M} = \mathbb{I}_{3n \times 3n} - \left(\mathbf{S}^T\right)^\dagger \mathbf{S}^T, \quad (53)$$

with $\left(\mathbf{S}^T\right)^\dagger = \frac{1}{n} \mathbf{S} \in \mathbb{R}^{3n \times 3}$. Note that the matrix \mathbf{M} is symmetric and can be computed offline, since it depends solely on the number of current contacts. From (51), we finally obtain the process noise covariance matrix on the state-transition model of our substate:

$$\text{cov}(\mathbf{M}\mathbf{v}) = \begin{pmatrix} \text{cov}(\mathbf{X}_1, \mathbf{X}_1) & \cdots & \text{cov}(\mathbf{X}_1, \mathbf{X}_n) \\ \vdots & \ddots & \vdots \\ \text{cov}(\mathbf{X}_n, \mathbf{X}_1) & \cdots & \text{cov}(\mathbf{X}_n, \mathbf{X}_n) \end{pmatrix}. \quad (54)$$

We then use it to replace the corresponding parts of the process noise covariance matrix \mathbf{Q} on each iteration (if necessary) before its use in (36).

We apply the same methodology to the rest orientation of the contacts, but this time we add the constrained process noise covariance only to the rest yaw, since this variable is not observable (the process noise covariance associated with the rest roll and pitch is set to zero). Since the focus is on a single axis, only the last row of the \mathbf{S}^T matrix is non-zero in this case. We note that this feature is independent of the viscoelastic model of contacts and thus can be used in any Kalman Filter performing legged odometry.

V. EXPERIMENTAL RESULTS

The Kinetics Observer has been tested in three scenarios on two different humanoid robots: RHP Friends and HRP-5P. Both robots are equipped with wrench sensors on the contact limbs and an IMU located in the upper body. In all the scenarios, the robots were controlled via the `mc_rtc` framework³. The first scenario (Section V-A), using the robot RHP Friends, evaluates the odometry on flat ground. The second scenario (Section V-B) extends the first scenario by adding slippery obstacles to the robot's path, in order to evaluate its performance in presence of slipping contacts. The third scenario (Section V-C), using the robot HRP-5P, evaluates the odometry performed during multi-contact motions. This scenario is also used in Section V-D to show how we can use this estimator to get information on the environment, and in Section V-E to evaluate the external wrench estimation performed by the Kinetics Observer.

Our estimation is compared across all scenarios with the right-invariant EKF (RI-EKF) proposed by Hartley et al. [7].

³https://jrl-umi3218.github.io/mc_rtc/

The Kinetics Observer and the RI-EKF were tuned using identical covariances for the sensor measurement noise and the process noise associated with the gyrometer bias. This ensures that both methods utilize the sensor data equally, allowing for a fair comparison of their performance. To demonstrate the improvements achieved by the *constrained contact process covariances* feature introduced in Section IV-C, we add to the comparison a variant of the Kinetics Observer that does not implement it. Using KO as a diminutive for the Kinetics Observer, this variant is referred to as KO-ZPC (Kinetics Observer with Zero Process Covariance). The Kinetics Observer is able to estimate the robot’s pose even in the absence of contact force sensor measurements. To prove this, we add the evaluation of the trajectory estimated in this case to the compared results. The information redundancy coming from the force sensor being lost, we removed the external force from the state and reduced the covariance of the gyrometer bias estimation. All other parameters were kept identical⁴. To the trajectory plots, we also include the robot-modeled trajectory, resulting from the integration of the commands at the output of the controller. This trajectory assumes perfect tracking of the references, and can thus be compared with the ground truth trajectory in order to assess the discrepancy between our ideal model of the system {robot + environment} and the actual system. We refer to this modeled trajectory as the “control” trajectory. Note that none of the estimators has access to the control trajectory; they rely solely on measurements. The estimated trajectories are compared to ground truth data obtained by motion capture (Opti Track system with 16 PrimeX 13 cameras). Finally, we note that the Kinetics Observer and the RI-EKF are designed to receive the contact states from any contact detection framework. For a fair comparison, all the estimators receive the same information from a Schmitt Trigger on the forces measured by the sensors. We set the lower threshold of the Schmitt Trigger to 10% of the weight and its upper threshold to 15% of the robot’s weight.

Note that the Kinetics Observer estimates the pose of the centroid frame, while the RI-EKF estimates the pose of the IMU and the motion capture measures the pose of the floating base in the world. Thus, we apply a simple transformation on the estimated trajectories to obtain trajectory estimates of the floating base. These estimates are evaluated against the ground truth using the Absolute Trajectory Error (ATE) and the Relative Error (RE) from [54]. Lateral (along the x and y axes) and vertical (along the z axis) translations are evaluated independently, as their errors originate from different sources and they can represent distinct targets for odometries. Indeed, the accurate estimation of lateral translations relies on both accurate estimates of local displacements and the robot’s orientation in the world relative to the vertical axis (yaw angle). In contrast, the quality of vertical translation estimation depends on accurate roll and pitch angle estimates, as well as the estimator’s ability to account for potential errors in new contact heights, notably due to material deformations. Accord-

ingly, the errors in yaw, and in roll and pitch angle estimations are also distinguished. We use the formulation of the Absolute Trajectory Error (ATE) for “Visual(-inertial)” systems. This metrics first realigns the estimated trajectory with the ground truth in order to minimize the squared error between both trajectories overall. It thus helps to evaluate how closely the estimated trajectory matched the actual one overall. Since they are observable, the roll and pitch angles remain unchanged, the realignment will thus shift the trajectory up/down to minimize the error along the vertical axis. This explains why its mean is zero in the results we present. The standard deviation of this error will thus help to evaluate the good tracking of the robot’s elevation. Similarly, the yaw is part of the minimization goals for the translation part, so we will focus on its standard deviation.

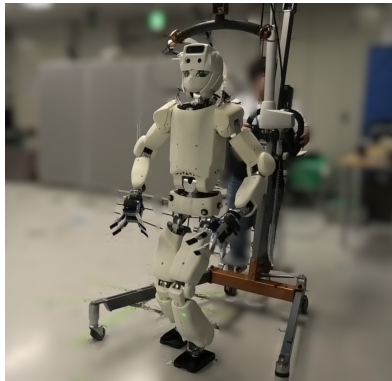
The linear velocity estimation performed by the compared estimators is also evaluated. The velocities are expressed in the IMU’s frame such that they are independent of the robot’s orientation. To avoid confusion, the velocity error is denoted as ^SVel , consistent with the notation S for the IMU frame used in Section III-A. The ground truth velocity is obtained by finite differences from the ground truth position, filtered with a zero-phase low pass filter.

A. Odometry on flat ground

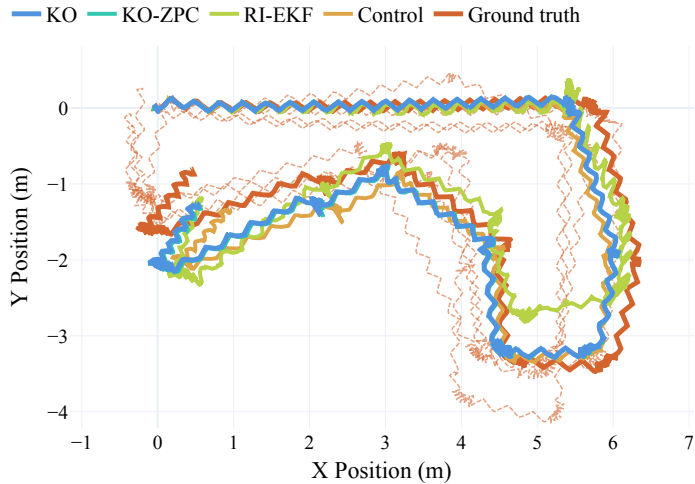
The test of the odometry on flat ground was performed on the humanoid robot RHP Friends (Fig. 5a) over five sequences, using a LIPM walking controller with reference footstep generation. The experiment involved making the robot walk and turn over about 18 meters (approximately 200 steps). Fig. 5b shows the trajectories followed by the robot over all the trials. First, we note that in each experiment, the estimators all consistently underestimated the forward translations, as can be seen in Figure 5b. This is also noticeable in the control trajectory, suggesting that this drift is not due to measurements, but rather to the geometric model of the robot involved in the direct kinematics.

Figure 6 shows the pose and linear velocity of RHP Friends estimated by the Kinetics Observer and the RI-EKF over the first walk, compared to their ground truth obtained by motion capture. We focus first on the estimate of translation along the vertical axis z , which is not observable by proprioceptive odometries. The Kinetics Observer was able to track the vertical position with a satisfactory drift of about 5 cm. Meanwhile, we see that at about 140 seconds after the start, the RI-EKF’s estimate of this position suddenly and erroneously decreased, to then increase greatly at about 190 seconds after the start. This temporally coincides with the positive and then negative discrepancy between the estimated pitch angle and the actual one. We verify that this pitch error is indeed the cause of the descent with the following approach. Between 140 seconds and 180 seconds, the average error between the pitch estimated by the RI-EKF and the ground truth is 1° . During this interval, the robot walked 7.4 meters along the x and y axes. The error in elevation Δ_z that angular error causes over this distance ($\Delta_z = \tan(1) \times 7.4$), would be over -13 cm, which roughly corresponds to the -19 cm drift

⁴Note that there was no adaptation of this force-sensor-free variant besides removing external forces and reducing bias variability. This means that we may expect better results with extensive tuning, especially in terms of covariance of the viscoelastic model.



(a)



(b)

Fig. 5. (a) RHP-Friends. (b) Top view of the ground truth and estimated trajectories of the robot during the walk on flat ground. The robot starts at the point (0,0) and walks towards the x direction. The trajectory of the first trial is shown with opaque lines, while the trajectories of the subsequent trials are transparent. The ground truth trajectories of the subsequent trajectories are also shown with dashed lines to highlight the proximity between all trajectories.

we can see in Figure 6. This drift also partly explains the underestimated forward translations of RI-EKF. Interestingly, this phenomenon occurred in the five trials of this experiment.

The second variable we focus on is the yaw, which is also unobservable and significantly influences the quality of the lateral translation estimation. We observe, notably in the focused areas, that although the trajectory estimates from both estimators drifted progressively, the Kinetics Observer demonstrated a better tracking of the yaw angle. Finally, we note that both estimators were able to satisfactorily estimate the robot’s linear velocity. However, the Kinetics Observer tended to overestimate this velocity, particularly along the forward axis x (we remind that the velocity is expressed in the frame of the IMU attached to the robot). Table IV provides more insight into the estimation results with the Absolute Trajectory Error (ATE) and the Relative Error (RE) computed over the five trials. The results of the 1 meter Relative Error in yaw corroborates our earlier observation that the Kinetics Observer was able to track the yaw more accurately than the RI-EKF, with an average error of 0.4° per meter walked for the Kinetics Observer versus 0.9° per meter. The standard deviation of the Absolute Trajectory Error ATE_{yaw} is nearly 6 times lower for the Kinetics Observer, demonstrating its better closeness over the entire trajectory. The Kinetics Observer was also able to estimate the lateral translations better with an average error of 2.5 cm over 1 meter versus 4,6 cm. Also, the significant drift on the estimated vertical position is reflected in the standard deviation of the ATE_z computed for the RI-EKF, which is 10 times higher than that computed for the Kinetics Observer. Finally, the error statistics for the lateral linear velocity estimation indicate that both estimators achieve an accuracy on the order of centimeters per second. However, as expected from the previous observations, the RI-EKF showed better accuracy compared to the Kinetics Observer.

B. Odometry with slippage

The purpose of this scenario is to assess the robustness of our estimation against slipping contacts. To this end, tiles with dimensions $7.5 (l) \times 7.5 (w) \times 1 (h)$ cm, which have low friction coefficients with both the robot’s feet and the ground, are placed along the robot’s path, as shown in Fig. 7a. Over three experiments and over a distance of about 5 meters, the robot walks and turns on the tiles, causing slippage in translation and rotation.

Looking at the evaluation metrics in Table V, we first note that as observed in the regular flat odometry in Section V-A, the Kinetics Observer was able to track the robot’s position and yaw more accurately than the RI-EKF. We also observe that although the estimation error on the position and the yaw increased when adding slippery contacts for both estimators, the estimation made by the Kinetics Observer was less degraded than that of the RI-EKF. Indeed, the average Relative Error on the lateral translation estimate over a 1 m translation increased by about 1 cm for the Kinetics Observer, against almost 2 cm for the RI-EKF. The small standard deviation of this error also proves the consistency of the translation estimate. Similarly, the mean Relative Error on the yaw estimate increased only slightly (0.1°) for the Kinetics Observer, while the estimation error for the RI-EKF increased by more than 1° , reaching a significant drift of 2.3° per meter walked. These results highlight the increased robustness of the proposed method to slipping contacts.

Note that we did not increase the slippage further because we firstly needed the robot to keep balance which would have been challenging, and secondly the drift caused by more significant slippage is not observable and would have increased the resulting variability across experiments.

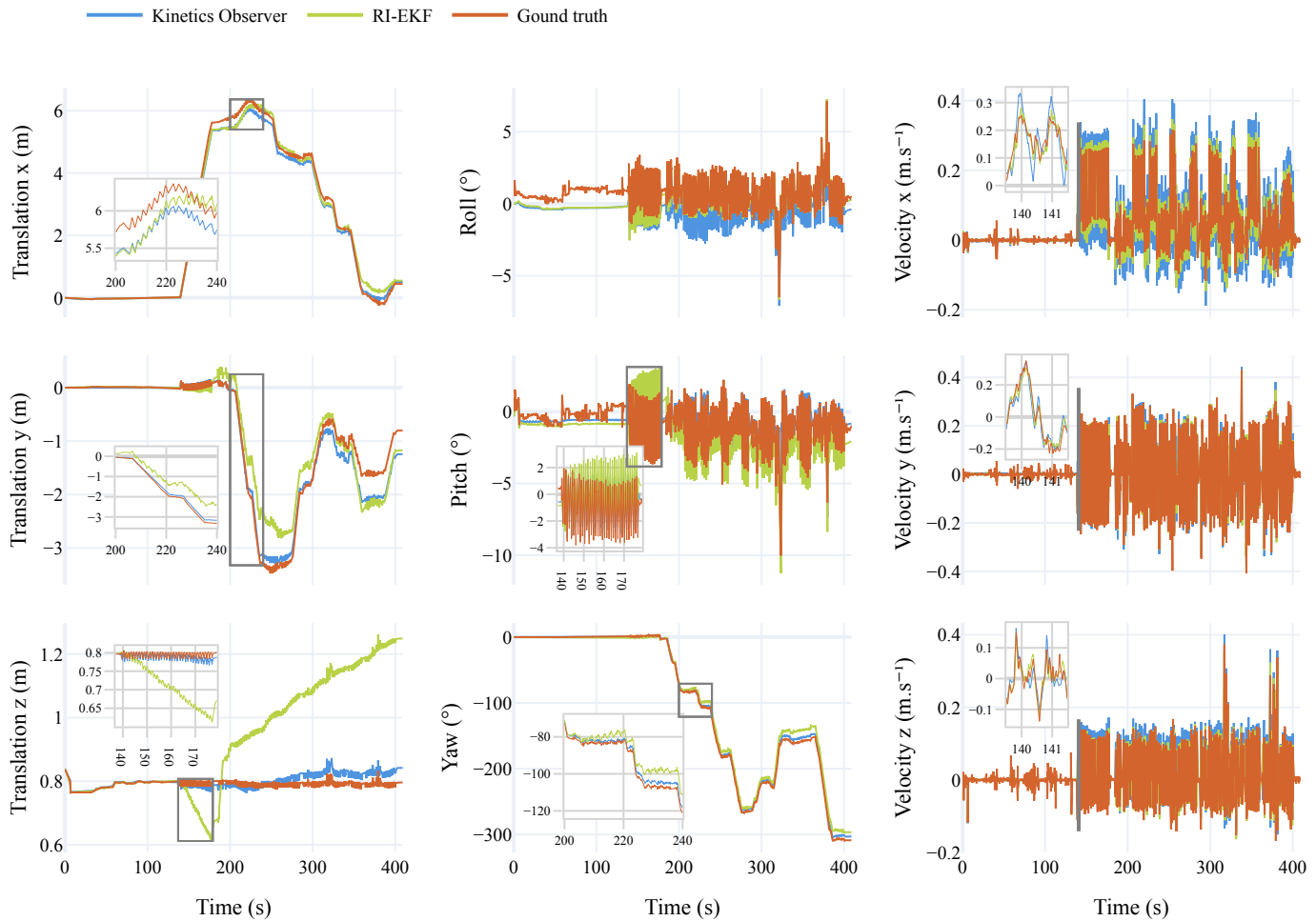


Fig. 6. Pose and linear velocity of RHP-Friends estimated by the Kinetics Observer and the RI-EKF during the first walk on flat ground, compared to the ground truth obtained by motion capture.

C. Multi-contact motion with tilted obstacles

To evaluate the contact pose estimation and the odometry performance of the Kinetics Observer over multi-contact motion, it was also tested within a non-coplanar contact scenario using a multi-contact controller⁵. This trajectory involves stepping, and pushing with the left hand on oriented and elevated obstacles, over a distance of about 2 meters. This experiment was performed four times on the humanoid robot HRP-5P.

In this experiment as well, the Kinetics Observer was able to estimate the pose more accurately. But this experiment is particularly interesting because it showcases the improvements brought by the *constrained contact process covariances* feature implemented in the Kinetics Observer, and introduced in Section IV-C. As a reminder, in the case of simultaneous multiple contacts, the covariance of the process noise on the contact rest poses is no longer zero (to avoid free drift), which allows for the correction of the contacts over their entire lifetime, while respecting a constraint such that the average pose of the robot cannot drift due to this process noise. During the experiments of flat odometry with slippage (Section V-B), the effects of this feature could not be highlighted. Indeed,

the difference between the Kinetics Observer and its variant KO-ZPC, which does not implement this feature, was not significant. This is actually normal, since this experiment only involved contact on both feet, and the duration of simultaneous contact was very brief—approximately half a second on average. During this short period, only a negligible amount of process noise covariance on the contact rest poses could be introduced into the Kinetics Observer, as it is only added in the case of simultaneous contacts. Thus, both estimators were performing a similar correction due to their identical initial state variance on the contact rest pose. Since this experiment involves multiple simultaneous contacts for much longer durations, the significant improvements brought by this feature are clearly observed. This is presented in Fig. 9, which shows the evolution of the foot’s yaw in the world and its estimates, at the beginning of the experiment.

For each estimate and the ground truth, this yaw is obtained by forward kinematics from the pose of the floating base, allowing to visualize how accurately the robot’s yaw estimate reflects the actual behavior of the contacts. About 2.5 seconds after the start of the experiment, it can be observed that the yaw estimates of both the KO-ZPC and the RI-EKF start to decrease and follow a similar pattern that quickly differs

⁵<https://github.com/isri-aist/MultiContactController>

TABLE IV
MEAN AND STANDARD DEVIATION (IN PARENTHESES) OF ERRORS COMPUTED DURING WALK ON FLAT GROUND. THE 1 M RELATIVE ERROR IS REPRESENTED. THE BEST RESULTS FOR EACH METRIC ARE HIGHLIGHTED IN BOLD.

	Translation [m]				Orientation [°]				Linear velocity [m.s ⁻¹]	
	Lateral { <i>x</i> , <i>y</i> }		Vertical <i>z</i>		{roll, pitch}		yaw		Lateral { <i>x</i> , <i>y</i> }	Vertical <i>z</i>
	ATE _{<i>x</i>,<i>y</i>}	RE _{<i>x</i>,<i>y</i>}	ATE _{<i>z</i>}	RE _{<i>z</i>}	ATE _{<i>r</i>,<i>p</i>}	RE _{<i>r</i>,<i>p</i>}	ATE _{yaw}	RE _{yaw}	^S Vel _{<i>x</i>,<i>y</i>}	^S Vel _{<i>z</i>}
Kinetics Observer	0.2008 (0.0832)	0.0250 (0.0184)	7.1e-16 (0.0178)	0.0148 (0.0442)	1.0859 (0.4202)	0.9966 (1.9868)	0.5183 (2.2926)	0.3980 (0.4276)	0.0234 (0.0269)	0.0011 (0.0180)
Kinetics Observer without wrench meas	0.5680 (0.3696)	0.0210 (0.0166)	-1.7e-14 (0.1658)	0.0226 (0.0460)	1.1225 (0.4718)	0.9884 (1.9665)	4.5233 (9.8563)	0.5732 (0.5391)	0.0223 (0.0241)	-0.0013 (0.0220)
KO-ZPC	0.2034 (0.0876)	0.0258 (0.0185)	3.3e-15 (0.0120)	0.0138 (0.0442)	0.9921 (0.4236)	1.0334 (1.9814)	0.4969 (2.3089)	0.4185 (0.4522)	0.0240 (0.0266)	0.0007 (0.0190)
RI-EKF [7]	0.2597 (0.1146)	0.0462 (0.0562)	-2.4e-15 (0.1781)	0.0215 (0.0507)	1.2156 (0.3989)	1.0035 (1.8407)	-1.0298 (6.1223)	0.9601 (0.9159)	0.0134 (0.0147)	0.0029 (0.0124)

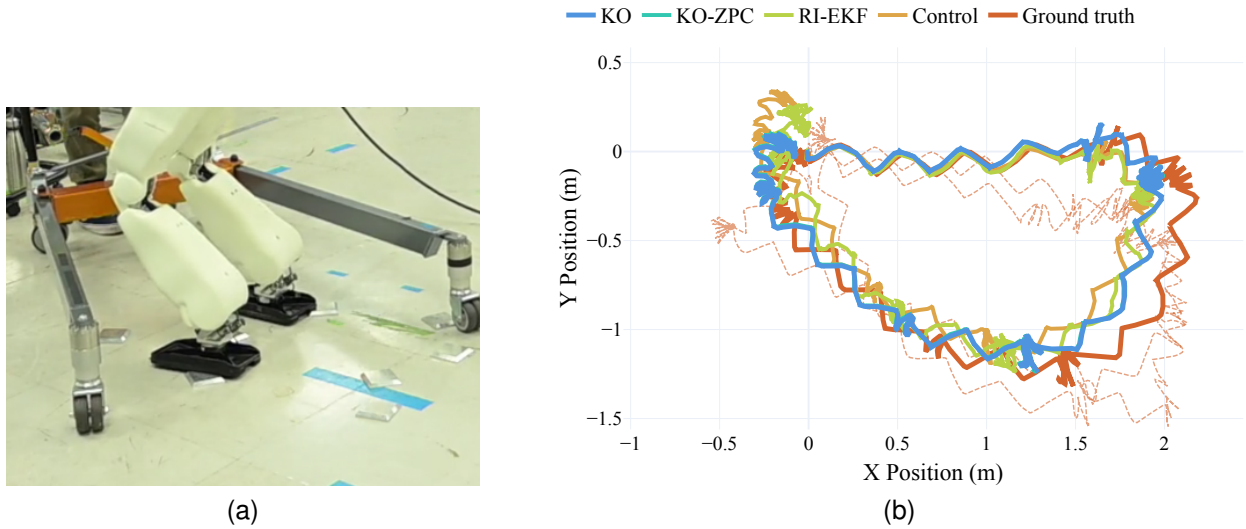
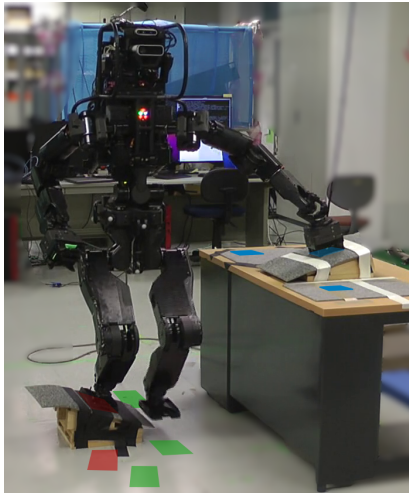


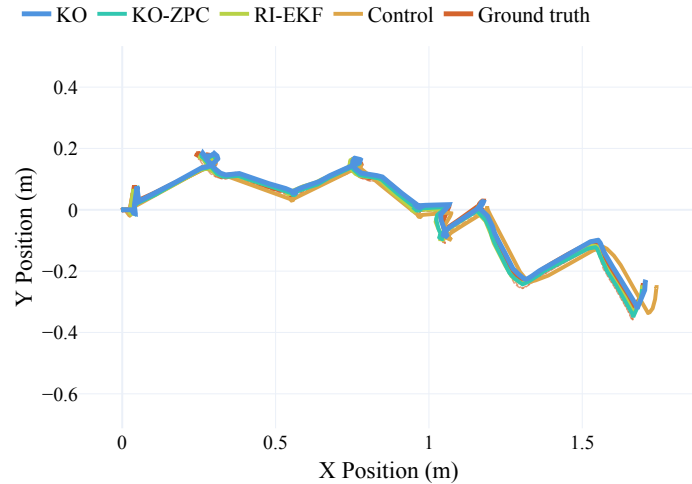
Fig. 7. (a) Slippery tiles RHP-Friends walked over during the experiment. (b) Top view of the ground truth and estimated trajectories of the robot during the walk with slippage. The robot starts at the point (0,0) and walks towards the *x* direction. The trajectory of the first trial is shown with opaque lines, while the trajectories of the subsequent trials are transparent. The ground truth trajectories of the subsequent trajectories are also shown with dashed lines to highlight the proximity between all trajectories.

TABLE V
MEAN AND STANDARD DEVIATION (IN PARENTHESES) OF ERRORS COMPUTED DURING WALK WITH SLIPPAGE. THE 1 M RELATIVE ERROR IS REPRESENTED. THE BEST RESULTS FOR EACH METRIC ARE HIGHLIGHTED IN BOLD.

	Translation [m]				Orientation [°]				Linear velocity [m.s ⁻¹]	
	Lateral { <i>x</i> , <i>y</i> }		Vertical <i>z</i>		{roll, pitch}		yaw		Lateral { <i>x</i> , <i>y</i> }	Vertical <i>z</i>
	ATE _{<i>x</i>,<i>y</i>}	RE _{<i>x</i>,<i>y</i>}	ATE _{<i>z</i>}	RE _{<i>z</i>}	ATE _{<i>r</i>,<i>p</i>}	RE _{<i>r</i>,<i>p</i>}	ATE _{yaw}	RE _{yaw}	^S Vel _{<i>x</i>,<i>y</i>}	^S Vel _{<i>z</i>}
Kinetics Observer	0.0606 (0.0331)	0.0341 (0.0121)	-4.1e-16 (0.0322)	0.0108 (0.0086)	1.1223 (0.9255)	1.3287 (1.1682)	0.7865 (1.0020)	0.5644 (0.5414)	0.0363 (0.0364)	0.0005 (0.0257)
Kinetics Observer without wrench meas	0.0690 (0.0362)	0.0267 (0.0132)	7.1e-17 (0.1091)	0.0250 (0.0087)	1.0010 (0.8540)	1.2516 (1.1596)	0.4235 (4.3569)	0.8940 (0.7064)	0.0343 (0.0361)	0.0038 (0.0214)
KO-ZPC	0.0626 (0.0342)	0.0345 (0.0124)	1.8e-16 (0.0292)	0.0100 (0.0089)	1.1425 (0.9170)	1.3489 (1.1793)	0.8233 (1.1007)	0.5836 (0.5611)	0.0366 (0.0363)	-0.0002 (0.0267)
RI-EKF [7]	0.1134 (0.0498)	0.0645 (0.0451)	2.4e-15 (0.0573)	0.0104 (0.0130)	0.8973 (0.8022)	1.2623 (1.0998)	-0.1044 (5.8003)	2.8321 (2.2962)	0.0227 (0.0305)	0.0021 (0.0188)



(a)



(b)

Fig. 8. (a) HRP-5P walking over tilted obstacles during the multi-contact experiment. The successive imprints of the main contacts of the motion are represented by colored trapezoids. In red: Right foot. In green: Left foot. In blue: Left hand. (b) Top view of the ground truth and estimated trajectories of the robot during the multi-contact motion. The robot starts at the point (0,0) and walks towards the x direction. The trajectory of the first trial is shown with opaque lines, while the trajectories of the subsequent trials are transparent. The ground truth trajectories of the subsequent trajectories are also shown with dashed lines to highlight the proximity between all trajectories.

TABLE VI

MEAN AND STANDARD DEVIATION (IN PARENTHESES) OF ERRORS COMPUTED DURING MULTI-CONTACT MOTIONS. THE 0.5 M RELATIVE ERROR IS REPRESENTED. THE BEST RESULTS FOR EACH METRIC ARE HIGHLIGHTED IN BOLD.

	Translation [m]				Orientation [°]				Linear velocity [m.s ⁻¹]	
	Lateral { x, y }		Vertical z		{roll, pitch}		yaw		Lateral { x, y }	Vertical z
	ATE _{x, y}	RE _{x, y}	ATE _{z}	RE _{z}	ATE _{r, p}	RE _{r, p}	ATE _{yaw}	RE _{yaw}	^S Vel _{x, y}	^S Vel _{z}
Kinetics Observer	0.0043 (0.0029)	0.0089 (0.0049)	3.9e-16 (0.0038)	-0.0005 (0.0029)	0.4885 (0.1788)	0.2916 (0.1837)	-0.8007 (0.3593)	0.1419 (0.1228)	0.0110 (0.0120)	-2.5e-05 (0.0060)
Kinetics Observer without wrench meas	0.0044 (0.0029)	0.0098 (0.0052)	4.2e-15 (0.0027)	-0.0008 (0.0033)	0.6419 (0.3157)	0.3828 (0.2074)	-0.9644 (0.5703)	0.3084 (0.2108)	0.0111 (0.0115)	7.9e-05 (0.0066)
KO-ZPC	0.0067 (0.0042)	0.0100 (0.0061)	2.5e-15 (0.0031)	-0.0018 (0.0037)	0.8125 (0.4741)	0.5343 (0.2440)	-0.7980 (0.8842)	0.5818 (0.3063)	0.0111 (0.0119)	-7.6e-05 (0.0061)
RI-EKF [7]	0.0062 (0.0032)	0.0193 (0.0213)	2.2e-16 (0.0038)	0.0005 (0.0111)	0.6880 (0.2619)	0.9278 (1.4890)	-1.2823 (0.6982)	0.5916 (0.8105)	0.0094 (0.0087)	0.0008 (0.0081)

from the actual one. This mis-estimation is actually due to the slippage of the left foot and the right foot, that occurred at that time along the x axis, causing a change in their relative position to each other. In the absence of covariance on the process noise for the contact positions, both the KO-ZPC and the RI-EKF were unable to correct them, and thus explained the change in relative position as a rotation of the floating base around the yaw axis. We observe similar behaviors when the left hand is in contact and the left foot is repositioned about 10 and 12 seconds after the start of the experiment, respectively. The information coming from the additional contact seems to improve the estimation, but as the state covariance on the contact pose converges again, the quality of the estimation decreases again. On the contrary, we observe that the Kinetics Observer succeeded in accurately tracking the yaw of the contact over the entire motion. Thanks to the *constrained contact process covariances*, it was able to correct the rest

pose of the contacts upon slippage over their entire lifetime, thus preventing the estimate of the yaw of the floating base from being erroneously corrected.

This experiment highlights also that our estimator seamlessly handles any number of contacts without any structural changes.

D. Environment reconstruction through contacts

As introduced in Section III-B3, the Kinetics Observer is able to reconstruct the local characteristics of the environment it is in contact with through the contact rest pose estimation. The latter is more accurate than the orientation obtained by forward kinematics from the orientation of the floating base, since it is affected by the robot's internal flexibilities. To evaluate this reconstruction, we virtually highly degraded the initial orientation $\Omega_{r,i}$ of the contacts (excluding the first two contacts to allow the Kalman Filter to converge) at their

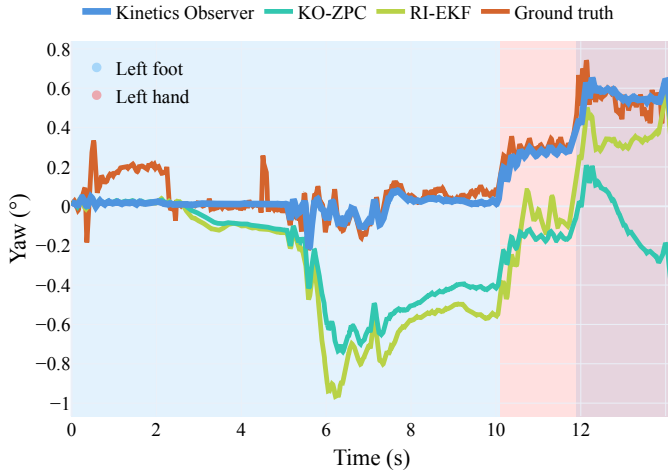


Fig. 9. Yaw of the right foot of HRP-5P at the beginning of the multi-contact experiment. The colored areas represent the times that the left foot and / or the right foot are in contact with the environment.

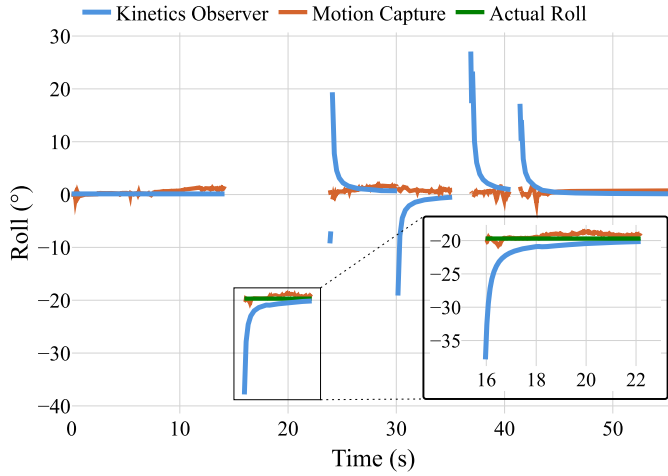


Fig. 10. Right foot roll estimated by the Kinetics Observer during the multi-contact experiment, compared to the measured ground truth and the roll obtained by forward kinematics from the ground truth pose of the floating base (motion capture). The plot is discontinuous since the rest pose is not associated with the foot itself, but only exists when the foot is in contact with the environment. The contact happens to be detected over very short periods of time due to the bouncing of the foot when it touches the ground.

creation during the multi-contact experiment. The goal is to show that the Kinetics Observer is able to correct this error and accurately estimate the orientation of the contact. The degraded orientation is obtained by generating a 30° rotation around a random axis, and applying this rotation to the initial orientation. Fig. 10 shows the estimation results of the roll angle of the right foot during the walk. During the motion, the right foot steps on the flat ground, and on an inclined obstacle. Using a precision instrument, we measured this tilt to be -19.7° , which we use as a ground truth. We also compare our estimate to the roll obtained by forward kinematics from the ground truth orientation of the floating base, which we use as the best estimate we could get from a floating base estimator. We observe that despite the considerable initial error, the estimated roll quickly converges within a range of

TABLE VII
MEAN ERROR AND STANDARD DEVIATION OF THE ESTIMATED EXTERNAL WRENCH FROM THE GROUND TRUTH.

	$\Gamma F_{e,x}$	$\Gamma F_{e,y}$	$\Gamma F_{e,z}$	$\Gamma T_{e,x}$	$\Gamma T_{e,y}$	$\Gamma T_{e,z}$
Mean Absolute Error	-17 N	15 N	15 N	13 N.m	9 N.m	8 N.m
Standard Deviation	11 N	12 N	19 N	9 N.m	5 N.m	6 N.m

$\pm 1^\circ$ when walking on the flat ground. More interestingly, it converges to an angle of -20.1° when walking on the tilted obstacle, which is only 0.4° error to the ground truth. The roll obtained by forward kinematics from the ground truth of the floating base’s pose remains within a range of $[-20.7^\circ, -18.9^\circ]$ over the entire step which is generally a worse error than our estimation, and might be caused by flexibilities. Note that we didn’t modify the tuning of the Kalman Filter’s covariances which greatly underestimate the introduced initial error.

E. External wrench estimation

This section aims to evaluate the estimation of the external wrench applied to the robot, made by the Kinetics Observer. For this purpose, the scenario used to evaluate the odometry over multi-contact motions (Section V-C) was adapted through the following change. The Kinetics Observer was made blind to all the information coming from the robot’s left hand. This contact is not detected and therefore its state is not estimated. Accordingly, the measurements of its force sensor are not used. Doing so, the contact at the left hand becomes an external disturbance for the Kinetics Observer. We then evaluate how closely this external wrench is estimated, using the force sensor measurements as a ground truth. Fig.11 shows the wrench estimates on the axes with the highest variations. We can see that the estimator can provide an accurate and reactive estimation of the left hand’s wrench. However, we also observe an offset of the order of $10\text{ N} \sim 20\text{ N}$ (N.m for the torque estimates) between the estimates and the ground truth. This offset shows that this estimate also serves as a slack variable to compensate for modeling errors and uncertainties in our state-transition and measurement models (e.g., an error in the mass or in the CoM position). This slack variable cannot be separated from the estimation of actual external unmodeled perturbations. We observe this offset also in the mean absolute error values in Table VII. Despite this offset, the Kinetics Observer is able to estimate the forces relatively well: for wrenches which can reach more than 150N and 100N.m, respectively, we get an average error of about 15N for forces and about 10N.m for torques. The reasonable standard deviation of the error from its mean also reflects the consistency of our estimate.

The estimation of external wrenches was already partially addressed in a previous paper [4], but only for static cases. Here, the Kinetics Observer was able to estimate them during dynamic motions.

F. Computation speed evaluation

The computation speed was evaluated on a laptop with an Intel Core i7, 7th Generation CPU, and 16 GB of RAM. During the

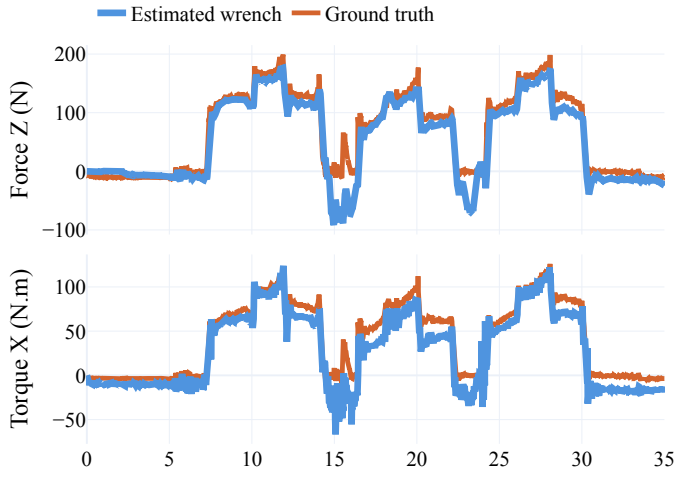


Fig. 11. Estimated external force along the z axis and torque along the x axis compared to the ground truth.

multi-contact experiment with HRP-5P, each iteration of the Kinetics Observer ran under 0.45 ms, which makes it usable for real-time feedback in most control loops.

VI. CONCLUSIONS

In this paper, we presented the Kinetics Observer, a framework able to estimate accurately, simultaneously, and with tight coupling, the kinematics of the robot, the contact location, and external forces applied on the robot, with the ability to perform precise real-time proprioceptive odometry. Thanks to the tight coupling, the estimator can correct the contact location estimation even during the contact. Overall, this estimator exploits all available data and models in a single loop, producing a unique state consistent with these models, the measurements, and the respective beliefs we put in them.

The only estimate showing a relatively lower accuracy than the SoA was the forward velocity during walking, still being within cms/s of error. This could be improved by more extensive tuning of the Covariances, but it is interesting to note that improving the estimation of a variable often results in decreasing accuracy for another one. This is because of the modeling “tension” that we create by integrating all these models: the kinematic, the dynamic and the viscoelastic one, within a single prediction loop.

Of course, this work requires further improvements, including to: (i) study ways to better identify environment stiffness and damping, (ii) to include additional proprioceptive sensors such as joint torque sensors or contact located IMUs, (iii) to simplify the tuning of the Kalman Filter, (iv) to allow for the addition of exteroceptive sensors to ensure observability, and (v) to improve contact and slippage detection.

APPENDIX A

FIRST APPENDIX : SUMMARY OF NOTATION

General notations:

- \hat{o} : estimated variable, \bar{o} : predicted variable, \tilde{o} : error variable.
- n_c : number of current contacts.
- n_w : number of contact wrench sensors.
- n_I : number of IMUs.

- n_b : number of bodies composing the robot.

Main frames 2:

- \mathcal{W} : world frame.
- \mathcal{B} : frame of the floating base.
- Γ : centroid frame.
- \mathcal{C}_i : frame of the i -th contact.
- $\mathcal{C}_{r,i}$: rest frame of the i -th contact.

State variables III-A:

- p_l : position of the centroid frame in the world frame, expressed in the centroid frame.
- Ω : vector representation (e.g. quaternion) of the orientation of the centroid frame in the world frame.
- v_l : linear velocity of the centroid frame in the world frame, expressed in the centroid frame.
- ω_l : angular of the centroid frame in the world frame, expressed in the centroid frame.
- $b_{g,j}$: bias on the j -th gyrometer’s measurement.
- ${}^\Gamma F_e$: unmodeled external force exerted on the robot, expressed in the centroid frame.
- ${}^\Gamma T_e$: unmodeled external torque exerted and expressed in the centroid frame.
- $x_{c,i}$: contact variable regrouping $\{p_{r,i}, \Omega_{r,i}, {}^C F_i, {}^C T_i\}$.
- $p_{r,i}$: i -th contact rest position.
- $\Omega_{r,i}$: i -th contact rest orientation.
- ${}^C F_i$: i -th contact force.
- ${}^C T_i$: i -th contact torque.

Centroid frame’s accelerations II:

- a_l : linear acceleration of the centroid frame in the world frame, expressed in the centroid frame.
- $\dot{\omega}_l$: angular acceleration of the centroid frame in the world frame, expressed in the centroid frame.

Measurements:

- $y_{a,j}$: measurement of the j -th accelerometer.
- $y_{g,j}$: measurement of the j -th gyrometer.
- $y_{F,i}$: measurement of the force sensor at the i -th contact.
- $y_{T,i}$: measurement of the torque sensor at the i -th contact.

Inputs:

- $\{{}^\Gamma I, {}^\Gamma \dot{I}\}$: total inertia matrix of the robot expressed in the centroid frame and its derivative.
- $\{{}^\Gamma \sigma, {}^\Gamma \dot{\sigma}\}$: total angular momentum of the robot expressed in the centroid frame and its derivative.
- $\{{}^\Gamma F_{res}, {}^\Gamma T_{res}\}$: resultant wrench measured by the sensors not associated with currently set contacts, expressed in the centroid frame.
- Ξ_i : input variables related to the contact i .
- $\{\tilde{p}_{r,i}, \tilde{\Omega}_{r,i}\}$: initial guess on the rest pose of the newly created contact i in the world frame.
- $\{{}^\Gamma p_{C_i}, {}^\Gamma \Omega_{C_i}, {}^\Gamma \dot{p}_{C_i}, {}^\Gamma \omega_{C_i}\}$: kinematics of the i -th contact frame in the centroid frame.
- Ψ_j : input variables related to the j -th IMU.
- $\{{}^\Gamma p_{S_j}, {}^\Gamma \Omega_{S_j}, {}^\Gamma \dot{p}_{S_j}, {}^\Gamma \omega_{S_j}, {}^\Gamma \ddot{p}_{S_j}\}$: kinematics of the j -th IMU frame in the centroid frame.
- ${}^\Gamma \ddot{p}_{S_j}$: linear acceleration of the j -th IMU in the centroid frame.

Section III-A: Vector state definition:

- S_j : frame of the j -th IMU.

Section III-B3: Viscoelastic model:

- $\{p_{C_i}, R_{C_i}, \dot{p}_{C_i}, \omega_{C_i}\}$: kinematics of the i -th contact’s frame in the world frame obtained by forward kinematics from the centroid frame.
- $\{K_{p,t}, K_{d,t}\}$: linear stiffness and damping of the contacts.
- $\{K_{p,r}, K_{d,r}\}$: angular stiffness and damping of the contacts.

REFERENCES

- [1] M. Benallegue and F. Lamiraux, “Estimation and stabilization of humanoid flexibility deformation using only inertial measurement units and contact information,” *Int. J. Humanoid Robotics*, vol. 12, no. 3, pp. 1 550 025:1–1 550 025:20, 2015.

- [2] A. Mifsud, M. Benallegue, and F. Lamiroux, "Estimation of contact forces and floating base kinematics of a humanoid robot using only inertial measurement units," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3374–3379.
- [3] M. Benallegue, A. Mifsud, and F. Lamiroux, "Fusion of force-torque sensors, inertial measurements units and proprioception for a humanoid kinematics-dynamics observation," in *15th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2015, Seoul, South Korea, November 3-5, 2015*. IEEE, 2015, pp. 664–669.
- [4] M. Benallegue, P. Gergondet, H. Audrerr, A. Mifsud, M. Morisawa, F. Lamiroux, A. Kheddar, and F. Kanehiro, "Model-based external force/moment estimation for humanoid robots with no torque measurement," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3122–3129.
- [5] 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.
- [6] R. Hartley, J. Mangelson, L. Gan, M. G. Jadidi, J. M. Walls, R. M. Eustice, and J. W. Grizzle, "Legged robot state-estimation through combined forward kinematic and preintegrated contact factors," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–8, 2017.
- [7] 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.
- [8] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, "State estimation for a humanoid robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 952–958.
- [9] S. Ito and H. Kawasaki, "A standing posture control based on ground reaction force," in *Proceedings. 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2000)(Cat. No. 00CH37113)*, vol. 2. IEEE, 2000, pp. 1340–1345.
- [10] M. Focchi, V. Barasuol, I. Havoutis, J. Buchli, C. Semini, and D. G. Caldwell, "Local reflex generation for obstacle negotiation in quadrupedal locomotion," in *Nature-Inspired Mobile Robotics*, 2013.
- [11] X. A. Wu, T. M. Huh, R. Mukherjee, and M. Cutkosky, "Integrated ground reaction force sensing and terrain classification for small legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1125–1132, 2016.
- [12] 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, 2022, pp. 1057–1066.
- [13] 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.
- [14] 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.
- [15] R. Buchanan, M. Camurri, F. Dellaert, and M. Fallon, "Learning inertial odometry for dynamic legged robot state estimation," 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, 2022, pp. 1575–1584.
- [16] M. Camurri, M. Ramezani, S. Nobili, and M. F. Fallon, "Pronto: A multi-sensor state estimator for legged robots in real-world scenarios," *Frontiers in Robotics and AI*, vol. 7, 2020.
- [17] Y. Kim, B. Yu, E. M. Lee, J.-h. Kim, H.-w. Park, and H. Myung, "Step: State estimator for legged robots using a preintegrated foot velocity factor," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4456–4463, 2022.
- [18] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.
- [19] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE transactions on robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [20] 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.
- [21] S. Yang, Z. Zhang, Z. Fu, and Z. Manchester, "Cerberus: Low-drift visual-inertial-leg odometry for agile locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4193–4199.
- [22] V. Agrawal, S. Bertrand, R. Griffin, and F. Dellaert, "Proprioceptive state estimation of legged robots with kinematic chain modeling," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 178–185.
- [23] M. Fourmy, T. Flayols, P.-A. Léziart, N. Mansard, and J. Solà, "Contact forces preintegration for estimation in legged robotics using factor graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1372–1378.
- [24] P.-C. Lin, H. Komsuoglu, and D. E. Koditschek, "Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 932–943, 2006.
- [25] P. Setoodeh, A. Khayatian, and E. Frajeh, "Attitude estimation by separate-bias Kalman filter-based data fusion," *The Journal of Navigation*, vol. 57, no. 2, p. 261–273, 2004.
- [26] G. Bourmaud, R. Mégret, A. Giremus, and Y. Berthoumieu, "Discrete Extended Kalman Filter on Lie groups," in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013, pp. 1–5.
- [27] P. Ramadoss, G. Romualdi, S. Dafarra, F. J. Andrade Chavez, S. Traversaro, and D. Pucci, "Diligent-kio: A proprioceptive base estimator for humanoid robots using extended kalman filtering on matrix lie groups," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2904–2910.
- [28] S. Teng, M. W. Mueller, and K. Sreenath, "Legged robot state estimation in slippery environments using invariant extended kalman filter with velocity update," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 3104–3110.
- [29] A. Barrau and S. Bonnabel, "Invariant Kalman Filtering," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, 2018.
- [30] —, "The Invariant Extended Kalman Filter as a stable observer," *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1797–1812, 2017.
- [31] S. Bonnabel, P. Martin, and P. Rouchon, "Symmetry-preserving observers," *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2514–2526, 2008.
- [32] S. Bonnabel, "Left-invariant extended kalman filter and attitude estimation," in *2007 46th IEEE Conference on Decision and Control*, 2007, pp. 1027–1032.
- [33] K. S. Phogat and D. E. Chang, "Invariant Extended Kalman Filter on matrix Lie groups," *Automatica*, vol. 114, p. 108812, 2020.
- [34] P. Ramadoss, "State estimation for human motion and humanoid locomotion," 2022.
- [35] P. Chauchat, A. Barrau, and S. Bonnabel, "Invariant smoothing on lie groups," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1703–1710.
- [36] S. Nousias and E. Papadopoulos, "Quadruped robot roll and pitch estimation using an Unscented Kalman Filter," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 731–736.
- [37] M. Bloesch, C. Gehring, P. Fankhauser, M. Hutter, M. A. Hoepflinger, and R. Siegwart, "State estimation for legged robots on unstable and slippery terrain," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 6058–6064.
- [38] 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, 2020.
- [39] S. Yang, H. Kumar, Z. Gu, X. Zhang, M. Travers, and H. Choset, "State estimation for legged robots using contact-centric leg odometry," *arXiv preprint arXiv:1911.05176*, 2019.
- [40] P. Wang, J. Liu, F. Zha, W. Guo, X. Wang, M. Li, and L. Sun, "A velocity estimation algorithm for legged robot," *Advances in Mechanical Engineering*, vol. 9, no. 12, p. 1687814017732736, 2017.
- [41] K. Masuya and K. Ayusawa, "A review of state estimation of humanoid robot targeting the center of mass, base kinematics, and external wrench," *Advanced Robotics*, vol. 34, pp. 1380 – 1389, 2020.
- [42] K. Kaneko, F. Kanehiro, M. Morisawa, E. Yoshida, and J.-P. Laumond, "Disturbance observer that estimates external force acting on humanoid robots," in *2012 12th IEEE International Workshop on Advanced Motion Control (AMC)*, 2012, pp. 1–6.
- [43] S. Lengagne, J. Vaillant, E. Yoshida, and A. Kheddar, "Generation of whole-body optimal dynamic multi-contact motions," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1104–1119, 2013.
- [44] F. Flacco, A. Paolillo, and A. Kheddar, "Residual-based contacts estimation for humanoid robots," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 409–415.
- [45] S. Piperakis, M. Koskinopoulou, and P. Trahanias, "Nonlinear state estimation for humanoid robot walking," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3347–3354, 2018.

[46] Y. Xu, F. Gao, Y. Pan, and X. Chai, "Method for six-legged robot stepping on obstacles by indirect force estimation," *Chinese Journal of Mechanical Engineering*, vol. 29, no. 4, pp. 669–679, 2016.

[47] Z. Cong, A. Honglei, C. Wu, L. Lang, Q. Wei, and M. Hongxu, "Contact force estimation method of legged-robot and its application in impedance control," *IEEE Access*, vol. 8, pp. 161 175–161 187, 2020.

[48] A. Mifsud, "Estimation et stabilisation de l'état d'un robot humanoïde compliant," Theses, Institut National Polytechnique de Toulouse - INPT, 2017.

[49] T. D. Barfoot, *State Estimation for Robotics*, 1st ed. USA: Cambridge University Press, 2017.

[50] D. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, 2013.

[51] J. L. Blanco-Claraco, "A tutorial on se(3) transformation parameterizations and on-manifold optimization," *CoRR*, vol. abs/2103.15980, 2021.

[52] 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, 2019.

[53] Y. Kakiuchi, M. Kamon, N. Shimomura, S. Yukizaki, N. Takasugi, S. Nozawa, K. Okada, and M. Inaba, "Development of life-sized humanoid robot platform with robustness for falling down, long time working and error occurrence," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 689–696.

[54] 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)*, 2018, pp. 7244–7251.



Arnaud Demont received the M.S. degree in mechanical engineering with a specialization in mechatronics and systems 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 CRNS-AIST Joint Robotics Laboratory in Tsukuba, Japan. His research interests include state estimation for legged robots (in particular humanoid robots), multi-sensor fusion, and mobile robot perception and autonomous navigation.



Mehdi Benallegue holds an engineering degree from the National Institute of Computer Science (INI) in Algeria, obtained in 2007. He earned a master's degree 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 Franco-Japanese Robotics Laboratory in Tsukuba, Japan, and to INRIA Grenoble, France. He also worked as a postdoctoral researcher at the Collège de France and at LAAS CNRS in Toulouse, France. Currently, he is a Research Associate with

CNRS AIST Joint robotics Laboratory in Tsukuba, Japan. His research interests include robot estimation and control, legged locomotion, biomechanics, neuroscience, and computational geometry.



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.).



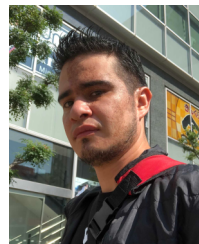
Pierre Gergondet received the MS degree in 2010 from Ecole Centrale de Paris with a speciality in embedded systems. He then received the Ph.D. degree from the University of Montpellier in 2014. His Ph.D. research was conducted on controlling humanoid robot using brain-computer interfaces at the CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/RL in Tsukuba Japan. He continued to work in JRL as a CNRS Research Engineer leading the software developments of the multi-contact real time framework: `mc_rtc`. Between 2019 and 2022, he joined the Beijing Advanced Innovation Center for Intelligent Robots and Systems (BAICIRS) at the Beijing Institute of Technology (BIT) as a special associate researcher, he has since resumed his position at JRL. His current research interests include humanoid robots, control software for robotics and robotics applications.



Antonin Dallard received an Engineering Master degree in mechanical and Industrial from Arts Et Métiers Institute of Technology.

In 2020, he started a Ph.D on the topic of Humanoid robot teleoperation and locomotion at the CNRS-University of Montpellier,

LIRMM in France and at the CNRS-AIST Joint Robotics Laboratory, Tsukuba in Japan.



Rafael Cisneros-Limón received the B.Eng. degree in electronics and computers from the University of the Americas - Puebla (UDLA-P), Puebla, Mexico, in 2006, the M.Sc. degree in automatic control from the Center of Research and Advanced Studies, National Polytechnic Institute (CINVESTAV-IPN), Mexico City, Mexico, in 2009, and the Ph.D. degree in intelligent interaction technologies from the University of Tsukuba, Tsukuba, Japan, in 2015. Since then, he has been with the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan, from 2015 to 2018 as a Postdoc and, since 2018, as a Researcher. He is currently a member of CNRS-AIST Joint Robotics Laboratory (JRL), IRL, AIST. His research interests include torque control, whole-body multi-contact motion control of humanoid robots, multibody collision dynamics, teleoperation, and tactile feedback.



Masaki Murooka received the BE, MS, and PhD degree in information science and technology from The University of Tokyo, Japan, in 2013, 2015, and 2018, respectively. He was a project assistant professor in the Department of Mechano-Informatics at The University of Tokyo from 2018 to 2020. He joined the CNRS-AIST Joint Robotics Laboratory in the National Institute of Advanced Industrial Science and Technology (AIST) in 2020 as a researcher. His research interest includes the motion planning and control of humanoid robots.



Fumio Kanehiro received the BE, ME, and PhD in engineering from The University of Tokyo, Japan, in 1994, 1996, and 1999, respectively. He was a Research Fellow of the Japan Society for the Promotion of Science in 1998-1999. In 2000, he joined the Electrotechnical Laboratory, Agency of Industrial Science and Technology (AIST-MITI), later reorganized as National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. From April 2007, he was a visiting researcher at the LAAS-

CNRS for one year and three months. He is currently Director of CNRS-AIST JRL (Joint Robotics Laboratory), IRL, AIST. His research interests include the software platform development and whole body motion planning of the humanoid robot.