Experimental Evaluation of Simple Estimators for Humanoid Robots

T. Flayols[1], A. Del Prete[1], P. Wensing[3], A. Mifsud[1], M. Benallegue[2] and O. Stasse[1]

Abstract— **This paper introduces and evaluates a family of new simple estimators to reconstruct the pose and velocity of the floating base. The estimation of the floating-base state is a critical challenge to whole-body control methods that rely on full-state information in high-rate feedback. Although the kinematics of grounded limbs may be used to estimate the pose and velocity of the body, modelling errors from ground irregularity, foot slip, and structural flexibilities limit the utility of estimation from kinematics alone. These difficulties have motivated the development of sensor fusion methods to augment body-mounted IMUs with kinematic measurements. Existing methods often rely on extended Kalman filtering, which lack convergence guarantees and may present difficulties in tuning. This paper proposes two new simplifications to the floating-base state estimation problem that make use of robust off-the-shelf orientation estimators to bootstrap development. Experiments for in-place balance and walking with the HRP-2 show that the simplifications yield results on par with the accuracy reported in the literature for other methods. As further benefits, the structure of the proposed estimators prevents divergence of the estimates, simplifies tuning, and admits efficient computation. These benefits are envisioned to help accelerate the development of baseline estimators in future humanoids.**

I. INTRODUCTION

Legged robots, including humanoids, have actuated joints, but are not rigidly attached to the environment. As a result, the positions of the actuated joints do not define the pose (i.e. position and orientation) of the robot with respect to the environment. The configuration requires six additional degrees of freedom to be specifed, usually the position and the orientation of a specific link with respect to an inertial coordinate system. This link is known as the *floating base* or the *free-flyer*. The position of every link of the robot can then be reconstructed using the pose of the floating base and the joint positions, which are measured by encoders.

Since no direct measurement of the floating base state is typically available, a fast and reliable state estimator is necessary for the implementation of state-feedback control on legged robots. With exact knowledge of the contact locations, the joint configuration is enough to reconstruct the floating base pose. However, contacts are subject to slipping or tipping on an edge, the environment may be irregular, or the robot may have uncertainty in its geometric model (e.g. because of flexible parts). Therefore the joint configuration is not enough to achieve a reliable state estimation.

Force/Torque (F/T) sensors and inertial measurement units (IMUs) can help to solve this issue. Force measurements can be used to detect contact and detect slip or tipping of the foot. These measurements can also be used to estimate the configuration of flexible elements, such as the ankle flexibility in HRP-2. IMUs typically provide the angular velocity and linear acceleration (biased by gravity) at a particular point of the robot. These IMU measurements are only sufficient to observe the orientation modulo yaw .

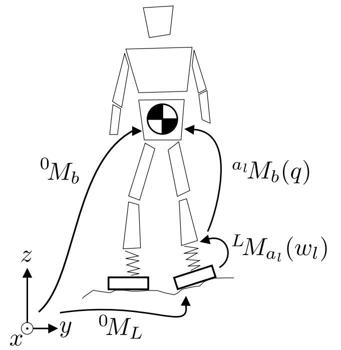

Fig. 1. This paper aims to estimate the pose ${}^0M_b$ and velocity of the floating base with simple estimators to fuse kinematics with IMU data.

Recent approaches have aimed at fusing IMU information, force data, and kinematics measurements for high-fidelity state estimation. This IMU/kinematic fusion was first used to address the problem of odometry [1], [2]. Later, higher-frequency estimators have been developed to estimate velocities and feed closed-loop controllers: they can operate at kinematic level [3], [4], [5], or can be extended to exploit dynamic models [6], [7]. Additional sensors can also be included, such as LIDAR [8]. As a common characteristic of these approaches, the methods solve a fully coupled inference problem to deduce position and orientation pose estimates from available measurements. However, this fully-coupled inference can be computationally costly, may be difficult to tune, and is generally complex to set up and debug.

Simpler estimators have been proposed in combination with balancing controllers, often leading to impressive experimental results [9], [10], [11], [12], [13]. These estimators often achieve their simplicity by decoupling the estimation of orientation and position. This begs the question: To what extent do we need to employ fully-coupled inference over position and orientation in order to enable high-performance control? It is hard to answer this question today because of the limited rigorous validation of these approaches, which have never been properly benchmarked.

The main contribution of this work is to benchmark two simple estimators on in-place balance and walking experiments with the humanoid robot HRP-2. This benchmark also investigates the contribution of the different components

---

[1] CNRS - LAAS, Toulouse, France, `tflayols@laas.fr`
[2] AIST, Tsukuba, Ibaraki, Japan
[3] University of Notre Dame, Indiana, U.S.A.

of the estimators, such as the inclusion of IMU data, and the use of force sensors to detect contact disturbances. We present two variants of simple estimators in Section II and Section III. Section IV shows and discusses the experimental results. Concluding remarks are provided in Section V.

II. METHOD 1: TWO-STAGE WEIGHTED AVG. (WA)

This section describes a simple estimator that averages different estimates of the floating base pose. In Section II-B, we express two estimates of the floating base pose using leg kinematics and prior knowledge of the contact locations. Section II-B quickly reviews how an IMU can provide an estimate of the robot orientation with respect to the gravity field. In Section II-C we present the design of weighting functions, which are used to average the different estimates in Section II-D. Finally, Section II-E discusses how to compensate for potential foot drifts.

A. Kinematics-based 6D Estimation

When the robot is standing on two grounded feet, knowing the pose of the feet in the world:

$$ {}^0M_L = \begin{bmatrix} {}^0R_L & {}^0p_L \\ 0 & 1 \end{bmatrix} \qquad {}^0M_R = \begin{bmatrix} {}^0R_R & {}^0p_R \\ 0 & 1 \end{bmatrix}, \quad (1) $$

and limb kinematics can provide two estimates of the floating base pose $\widehat{{}^0M_b}^L$ and $\widehat{{}^0M_b}^R$ (see Fig. 1).

In the case of HRP-2, the kinematic chains for each leg contain flexibilities below the ankles, designed to absorb impacts. We model these flexibilities as 6D linear springs, and use the force-torque sensors for each foot to estimate the associated 6D deformations:

$$ w_l = K_l \begin{bmatrix} {}^Lp_{a_l} \\ \text{rpy}({}^LR_{a_l}) \end{bmatrix}, \qquad w_r = K_r \begin{bmatrix} {}^Rp_{a_r} \\ \text{rpy}({}^RR_{a_r}) \end{bmatrix}, $$

where w_l/w_r are the measured 6D wrenches, K_r/K_l are diagonal positive-definite stiffness matrices, ${}^Lp_{a_l}/{}^Rp_{a_r}$ are the translational deformations, ${}^LR_{a_l}/{}^RR_{a_r}$ are the angular deformations, and rpy(.) is a function converting a rotation matrix to roll-pitch-yaw (rpy) angles. The use of rpy angles introduces coupling between the axes, but is not an issue in practice since the maximum deflection is small ($< 10°$).

This flexibility model allows us to reconstruct two estimations of ${}^0M_b$:

$$ \begin{aligned} \widehat{{}^0M_b}^L &= {}^0M_L \, {}^LM_{a_l}(w_l) \, {}^{a_l}M_b(q) \\ \widehat{{}^0M_b}^R &= {}^0M_R \, {}^RM_{a_r}(w_r) \, {}^{a_r}M_b(q), \end{aligned} \quad (2) $$

where ${}^{a_l}M_b(q)$ and ${}^{a_r}M_b(q)$ are transformations between the ankles and the floating base, computed through standard forward kinematics.

B. IMU-based Orientation Estimation

Legged robots (such as HRP-2) are usually equipped with an IMU sensor, composed of a 3D gyroscope and a 3D accelerometer. Work in [14] presented a light-weight complementary filter to estimate orientation with respect to the gravity field from IMU measurements. Remarkably, efficient open-source implementations of this filter are available [14], greatly simplifying deployment on real systems.

Its principle is that orientation can be obtained through angular velocity integration. Then, under the hypothesis that the accelerometer measurement is aligned with gravity on average, orientation can be unbiased through low-bandwidth accelerometer feedback. Only the orientation about the vertical axis cannot be unbiased because the gravity vector measurement does not depend on it. This complementary filter then provides a partial orientation estimation, containing only roll and pitch angles $\widehat{{}^0R_b}^{IMU}$.

C. Weighting Functions

To fuse different estimates, the first proposed simple estimator uses a set of weighting adaptive weighting-factors, denoted with λ. At each instant, these weights are designed to quantify the *validity* of each estimate. Clear parallels can be drawn between these weights and asscociated variances in standard Kalman (least-squares) filtering. The role of the weights here is to capture the fact that a foot may be tilting or slipping, which would invalidate the associated estimate. A simple analysis can identify two main causes of invalid data from a leg kinematic estimate.

1) The foot is in contact with only one edge/corner.
2) The foot is not in contact with the ground.

We use F/T sensor measurements to detect these situations and modify the weights accordingly for sensor fusion.

1) Situation 1: This situation can be detected through the distance between the foot (local) Zero Moment Point (ZMP), (denoted (z_x, z_y) in foot frame), and the edges of the sole. Under an assumption of an additive zero-mean Gaussian noise on ZMP measurements, distributed as $\mathcal{N}(0, \sigma_{\varepsilon z}^2)$, the probability that the ZMP lies inside the sole is:

$$ \begin{aligned} P_{zmp} = P(x_{min} < z_x < x_{max}) P(y_{min} < z_y < y_{max}) = \\ \int_{x_{min}}^{x_{max}} f_{z_x}(x)\,\mathrm{d}x \int_{y_{min}}^{y_{max}} f_{z_y}(y)\,\mathrm{d}y, \end{aligned} $$

where f_{z_x}/f_{z_y} are the marginal probability density functions of the random variable z, and $x_{min}, x_{max}, y_{min}, y_{max}$ denote the boundaries of the rectangular foot sole. Extension to non-rectangular contact surfaces is possible using an approximation in [15]. Under the reasonable assumption that $3\sigma_{\varepsilon z} < \min(y_{max} - y_{min}, x_{max} - x_{min})$, then the probability P_{zmp} varies between 0.25 and 1 as long as the ZMP measurement is inside the sole. To get a valid weighting, we scale it between 0 and 1:

$$ \lambda_z = \min\left(0, \frac{4}{3}(P_{zmp} - 0.25)\right) $$

2) Situation 2: This situation can be detected through the norm of the forces acting on the foot. The same method presented for Situation 1 can be adapted, leading to a weight depending on the standard deviation of the normal force measurements $\sigma_{\varepsilon_{Force}}$:

$$ \lambda_f = \min(0, 2(P(f_z > f_{min}) - 0.5)), $$

where f_{min} is the minimum normal force to be in contact.

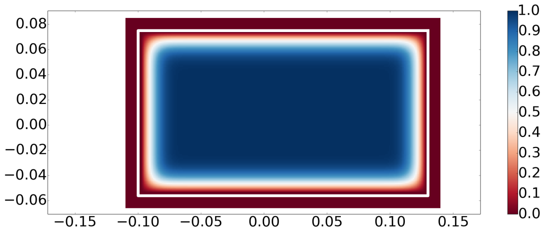

Fig. 2. Weight λ_z as a function of ZMP location with $\sigma_{\varepsilon_z} = 1cm$. The white rectangle depicts the foot limits of HRP-2.

Finally, for each foot, we obtain a weighting function that depends only on the corresponding wrench:

$$\lambda^R(w_r) = \lambda_z^R(w_r)\,\lambda_f^R(w_r), \quad \lambda^L(w_l) = \lambda_z^L(w_l)\,\lambda_f^L(w_l)$$

D. *Fusing Estimates*

This section describes a fusion process to obtain estimates for the 6D pose given by x, y, z, roll, pitch, yaw. We choose to fuse in rpy space because of the sparsity of the IMU-based estimate (roll and pitch only). Fusion is performed in two stages. First, we fuse the orientations computed from the IMU and the kinematics. Then, assuming a fixed orientation, we compute the base position that minimizes the distance to the known foot positions.

We prefer this two-step approach to a straightforward one-step 6D fusion because the latter may result in large deviations of the foot positions. Intuitively, this is due to the fact that small variations in the base orientation result in large variations of the foot positions, due to the long lever arm. Taking into account the base orientation (estimated in the first stage) allows us to find (in the second stage) the base position that makes the foot positions match (in the least-squares sense) the *known* foot positions ${}^0p_L, {}^0p_R$.

1) Orientation average: The orientations from the leg kinematics are extracted from (2), and are denoted $\widehat{{}^0R_b}^L$ and $\widehat{{}^0R_b}^R$. The weighted average estimation of the base orientation is computed as through weighting matricies:

$$\Lambda_*^{IMU}\,\mathrm{rpy}\left(\widehat{{}^0R_b}^{IMU}\right) + \Lambda_*^{L}\,\mathrm{rpy}\left(\widehat{{}^0R_b}^{L}\right) + \Lambda_*^{R}\,\mathrm{rpy}\left(\widehat{{}^0R_b}^{R}\right), \quad (3)$$

where

$$\begin{aligned}\lambda^{norm} &= (\lambda^{IMU} + \lambda^L + \lambda^R)^{-1}\\ \Lambda_*^{IMU} &= \lambda^{IMU}\mathrm{diag}(\begin{bmatrix}\lambda^{norm} & \lambda^{norm} & 0\end{bmatrix})\\ \Lambda_*^{L} &= \lambda^{L}\mathrm{diag}(\begin{bmatrix}\lambda^{norm} & \lambda^{norm} & (\lambda^L+\lambda^R)^{-1}\end{bmatrix})\\ \Lambda_*^{R} &= \lambda^{R}\mathrm{diag}(\begin{bmatrix}\lambda^{norm} & \lambda^{norm} & (\lambda^L+\lambda^R)^{-1}\end{bmatrix})\end{aligned}$$

2) Position average: Knowing the foot positions and the orientation of the base, its position is computed as:

$$\widehat{{}^0p_b} = \frac{\lambda^L({}^0p_L - \widehat{{}^0R_b}\,{}^bp_L) + \lambda^R({}^0p_R - \widehat{{}^0R_b}\,{}^bp_R)}{\lambda^L + \lambda^R}, \quad (4)$$

where ${}^0p_L, {}^bp_R, {}^0p_R$ and ${}^bp_L$ are 3D translations extracted from (2).

3) Velocity average: Two estimates of base velocity can be computed via numerical differentiation of (2). Since this operation amplifies noise, a delayed differentiation is provided by a Savitzky-Golay filter here, whose parameters need to be adapted to the sensor noises. Then these two velocities are averaged with the same weights used in (4).

E. *Foot Drift Compensation*

Since multiple measurements are averaged, any bias in these measurements will result in bias for the weighted average. To overcome this issue, we propose a low-gain feedback method to remove bias by slowly adapting the assumed pose of each foot. We estimate the kinematics bias by computing the difference between the current foot pose and the one according to the current base estimation:

$$M_{drift}^L = {}^0M_L^{-1}\,\widehat{{}^0M_b}\,{}^bM_L, \qquad M_{drift}^R = {}^0M_R^{-1}\,\widehat{{}^0M_b}\,{}^bM_R$$

At each cycle, a part of this drift is compensated for[1]:

$${}^0M_L := {}^0M_L\mathrm{SE3interp}(\alpha_L, I, M_{drift}^L)$$

$${}^0M_R := {}^0M_R\mathrm{SE3interp}(\alpha_R, I, M_{drift}^R),$$

where SE3interp is a function that interpolates between two SE3 elements, I is the 4x4 identity matrix, and α_L/α_R are the convergence rates. These variables depend on the validity of the opposite foot contact because drifts are most likely to occur on the least trusted foot:

$$\alpha_L = \lambda^R\alpha, \qquad \alpha_R = \lambda^L\alpha,$$

where the constant value α is the maximum convergence rate, which needs to be adapted to the application.

III. SECOND METHOD: A LINEAR KALMAN FILTER FOR POSITION AND VELOCITY

The second method considered here uses an adaptive weighting through the application of Kalman Filtering to estimate the position and velocity of the floating base. Previous state estimation approaches have used Extended Kalman Filters (EKFs) to good effect in the estimation of the floating base state for humanoids and quadrupeds [3], [4], [5]. While EKF filters may perform well in many cases, they lack guarantees on the stability of their error dynamics, and can be difficult to tune in practice. As a key observation, this section describes how floating-base state estimation can be decomposed into a cascade of orientation and position filters under a mild assumption. With this observation, the estimation of the base position and velocity turns into a linear problem, which we tackle with a Kalman filter.

---

[1] This foot pose update can generate drift in unobservable components (yaw and the 3D position). If absolute positions are needed, other sensors have to be used. However, this is often not an issue, since whole-body goals in control are often defined relative to contact positions and the gravity field.

A. Process Model

As in [3], the state of the position and velocity estimate, denoted x, is chosen to include the positions of the feet.

$$x = \left[{}^0p_b^T,\ {}^0v_b^T,\ {}^0p_L^T,\ {}^0p_R^T\right]^T \tag{5}$$

Given a reading of the accelerometer a_n at timestep n, the base position and velocity evolve as:

$$x_{n+1} = \begin{bmatrix} {}^0p_{b,n} + {}^0v_{b,n}\,\Delta t + \frac{1}{2}\,[{}^0R_{b,n}\,a_n + g](\Delta t)^2 \\ {}^0v_{b,n} + [{}^0R_{b,n}\,a_n + g]\Delta t \\ {}^0p_{L,n} \\ {}^0p_{R,n} \end{bmatrix}, \tag{6}$$

where we assumed that the feet stay stationary.

In general, if estimation of ${}^0R_b$ is pursued alongside x, state estimation must take place over a nonlinear discrete time process. However, as an approximation, if the estimation of ${}^0R_b$ is decoupled from x, then the process model (6) becomes linear time invariant through viewing ${}^0R_{b,n}\,a_n + g$ as a time-varying control input $u_n := {}^0R_{b,n}\,a_n + g$.

For estimating ${}^0R_b$ we rely on an IMU-based orientation estimation (similar to the one described in Section II-B). In particular, we use the second version of the estimator presented in [14]. The original filter was designed to be used with a magnetometer. Here, we replaced the magnetometer information with an average of the base orientation estimates obtained through kinematics.

B. Measurement Model

To estimate x, kinematic measurements of the IMU position are provided relative to the feet. In addition, the heights of the feet (z_L and z_R, assumed 0 here), are provided as pseduo-measurements:

$$y = \begin{bmatrix} {}^0R_b\,{}^bp_L(q_L) \\ {}^0R_b\,{}^bp_R(q_R) \\ z_L \\ z_R \end{bmatrix} \tag{7}$$

Through the decoupled estimation of ${}^0R_b$, its estimate can be incorporated into the measurement y. This approach results in a linear relationship between the estimated measurement $\hat{y}$ and estimated state $\hat{x}$:

$$\hat{y} = \begin{bmatrix} {}^0\hat{p}_b - {}^0\hat{p}_L \\ {}^0\hat{p}_b - {}^0\hat{p}_R \\ \hat{z}_L \\ \hat{z}_R \end{bmatrix} := C\hat{x} \tag{8}$$

C. Weighted Noise Model

Under the previous decoupling assumptions, the estimation of x can be written as a standard linear Kalman filter over a stochastic process:

$$x_{n+1} = Ax_n + Bu_n + \mathcal{N}(0, Q_n) \tag{9}$$

$$y_{n+1} = Cx_{n+1} + \mathcal{N}(0, R_{n+1}) \tag{10}$$

for matrices A and B that satisfy (6), with Q_n and R_n the covariance of the process noise and measurement noise respectively. To provide rough agreement with the operation of the WA filter, the measurement noise is designed as:

$$R_n = \text{blockDiag}\left(\sigma_{p_L} I_3,\ \sigma_{p_R} I_3,\ \sigma_{z_L},\ \sigma_{z_R}\right)^2 \tag{11}$$

with σ_{p_L} and σz_L scheduled as:

$$\sigma^2_{p_L} = \sigma^2_{kin}/\lambda^L \qquad \sigma^2_{z_L} = \sigma^2_h/\lambda^L \tag{12}$$

from some fixed nominal standard deviations σ_{kin} and σ_h. Analogous scaling is adopted for the right foot noise model. The process noise is modeled as:

$$Q_n = \text{blockDiag}\left(0_3,\ \sigma_a I_3 \Delta t,\ \sigma_{\dot{p}_L} I_3 \Delta t,\ \sigma_{\dot{p}_R} I_3 \Delta t\right)^2 \tag{13}$$

where σ_a is the fixed RMS error of the accelerometer, and $\sigma_{\dot{p}_L}$, is again an adaptively-scaled standard deviation for the evolution of the foot position. This noise model allows the filter to automatically capture the evolution of the foot position during leg swing in walking. To provide more flexibility to this process, the noise model was exaggerated through an inverse logistic sigmoid:

$$\sigma^2_{\dot{p}_L} = \sigma^2_{\dot{p}}(1 + \mathrm{e}^{\gamma(0.5-\lambda_L)}),$$

with $\sigma^2_{\dot{p}}$ a nominal variance for the foot evolution velocity. Standard Kalman filter update equations were then used to construct position and velocity estimates.

IV. RESULTS

The main goal of this section is to compare the WA estimator (Section II) and the KF estimator (Section III) in terms of estimation accuracy and computation time. Moreover, we want to assess the contribution of the weighting functions (Section II-C) and the IMU to the final estimation quality. We carried out three experiments with the humanoid HRP-2 robot, which are described in Section IV-D, IV-E and IV-F.

A. Ground Truth

As ground truth we used a motion capture system (by Motion Analysis) to get 100 Hz measurements of the position and orientation of the robot base. We computed numerical derivatives of the position measurements to get a ground-truth velocity. The lower accuracy of the orientation measurements prevented us from computing the angular velocity.

B. Ankle flexibilities identification

Using the data collected in experiment 1, the 6D stiffness of HRP-2 ankles has been identified using motion capture and force-torque sensor. The identified stiffness values are:

$$K_l = K_r = \text{diag}(4034, 23770, 239018, 707, 502, 936)$$

We used these values for all the experiments.[2]

C. Metrics

When assessing the accuracy of an estimator, we think it is important to capture both the nominal behavior and the worst-case behavior. For this reason, we present both the Root Mean Square Error (RMSE) and the maximum error. Moreover, we present separately the errors for position, orientation (based on rpy) and linear velocity.

---

[2] Units are rad/Nm for first 3 linear stiffnesses and m/N for the last 3 angular stiffnesses.

TABLE I

ROOT MEAN SQUARE ERROR (RMSE) AND MAXIMUM ERROR FOR THE BASE 3D POSITION, ORIENTATION AND VELOCITY. THE WA COLUMNS SHOW THE 2-STAGE WEIGHTED AVERAGE WITH OPTIONAL FIXED WEIGHTS. THE KF COLUMN SHOW THE RESULTS OF THE KALMAN FILTER WITH WEIGHTED COVARIANCES.

		WA $\lambda^{IMU}=0$ $\lambda^L=\lambda^R=1$		WA $\lambda^{IMU}=0$		WA $\lambda^{IMU}=1$ $\lambda^L=\lambda^R=1$		WA $\lambda^{IMU}=1$		KF	
		RMSE	Max	RMSE	Max	RMSE	Max	RMSE	Max	RMSE	Max
Pushes	T $[mm]$	4.2	19.2	3.7	19.4	3.3	12.5	3.0	11.1	3.0	8.5
	R $[deg]$	0.5	1.4	0.5	1.4	0.5	1.4	0.5	1.4	0.5	1.5
	V $[mm/s]$	24.4	87.9	24.2	83.4	24.4	87.9	24.2	83.4	14.2	49.5
CoM Sinusoid	T $[mm]$	6.8	19.7	4.4	8.7	5.0	12.5	4.1	7.3	5.7	11.1
	R $[deg]$	0.3	0.8	0.3	0.8	0.3	0.8	0.4	0.8	0.5	0.9
	V $[mm/s]$	36.7	191.0	28.1	85.2	36.7	191.0	28.1	85.2	18.5	103.6
Walking	T $[mm]$	39.7	68.6	37.4	68.2	36.9	65.0	34.9	63.9	26.0	50.4
	R $[deg]$	0.8	1.9	0.8	2.0	0.8	2.0	0.8	2.0	0.9	2.0
	V $[mm/s]$	208.8	467.1	107.8	313.4	208.8	467.1	107.8	313.4	82.2	298.0

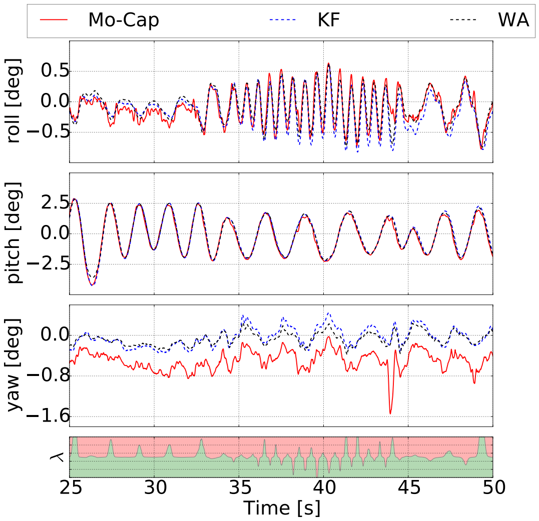

Fig. 3. Base orientation during experiment 1. The bottom plot shows the normalized weighting function $\lambda = \lambda_L/(\lambda_L + \lambda_R)$.

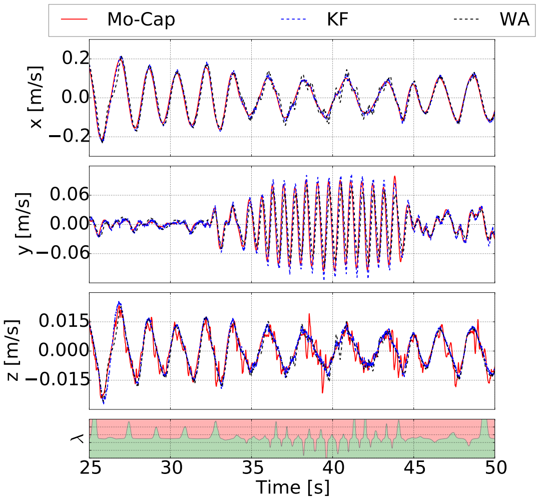

Fig. 4. Base velocity during experiment 1. The bottom plot shows the normalized weighting function $\lambda = \lambda_L/(\lambda_L + \lambda_R)$.

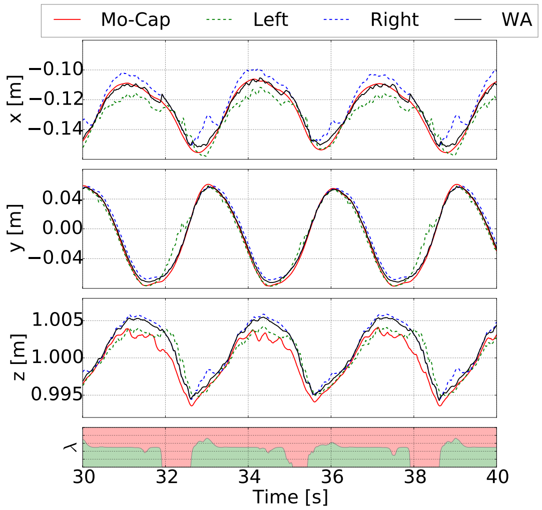

Fig. 5. Base position during experiment 2. This figure highlights the importance of the weighting function $\lambda = \lambda_L/(\lambda_L + \lambda_R)$, which make the WA estimate tend more towards the right-foot estimate when λ decreases.

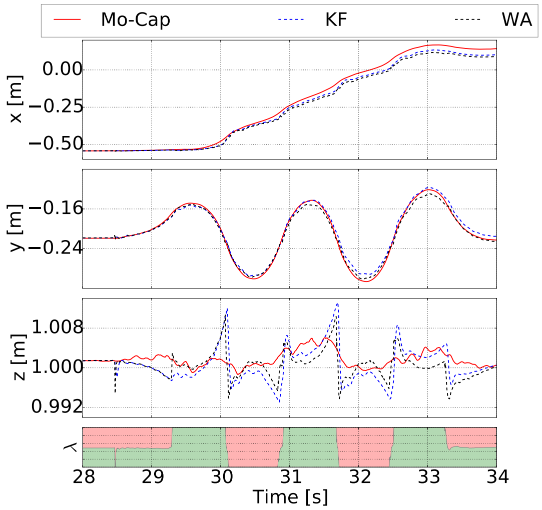

Fig. 6. Base position during experiment 3. The bottom plot shows the normalized weighting function $\lambda = \lambda_L/(\lambda_L + \lambda_R)$. Both estimators (KF and WA) exhibit a drift of about 10% of the traveled distance along the x axis.

D. Experiment 1: Pushes

In the first test the HRP-2 robot was controlled with high-gain position control while standing with its feet on flat ground. The reference joint angles were constant, so the motion of the robot was only due to the motion of the ankle flexibilities, caused by external pushes in different directions.

E. Experiment 2: CoM Sinusoid

In the second test the robot joints were torque controlled, and its feet were on flat ground (as for Experiment 1). We used a state-of-the-art task-space inverse dynamics controller to track a sinusoidal trajectory along the y axis (i.e. lateral motion) with the robot Center of Mass (CoM). The peak-to-peak amplitude of the sinusoid was 14 cm, and the peak-to-peak time was 1.5 s.

F. Experiment 3: Walking

In the third test we made HRP-2 walk on a flat terrain using high-gain position control. The robot walked three steps before coming to rest. The small number of steps was necessary for full coverage with our motion-capture system.

G. Feet update policy

The ground truth provided by the motion capture system gives an absolute pose of the robot, whereas the estimators have inherent bias in unobservable components as discussed in Section II-E. To limit the effect of this drift in error metric, we have set the update rate $\alpha = 0$ for experiment 1 and 2, during which the feet did not move. However, to ensure that the feet placements lead to close estimates, they are fully updated at the estimator initialization.

On the contrary, for the walking experiment, foot positions are updated with a high rate $\alpha = 0.8$ to track the swing foot motion before landing. A modification of the update phase is performed to ensure no drift along the z axis. Note that no information from the controller is used to determine which foot is in contact.

H. Computation Times

As expected, both estimators proved to be extremely computationally efficient. For the WA, most of the computation time is spent in the computation of the robot kinematics, leading to an average time of 8 μs. For the KF, besides computing the robot kinematics, the most expensive operation are the matrix decompositions and multiplications performed in the innovation step. The overall average computation time of the KF approach was about 16 μs. In conclusion, both estimators are suitable for implementation on a fast control loop, which is a key requisite for application on a real robot.

I. Discussion

Table IV-C summarizes the results of the three experiments. Overall, both estimators achieved small estimation errors for the position (below 1 cm) and orientation (below 1°) of the robot base in the first two tests. The larger position errors shown in the last test are due to the unobservability of the position, which leads to a drift, mainly along the x-axis (see Fig. 6). This drift is approximately equal to 10% of the traveled distance and is comparable to the drift exhibited by state-of-the-art EKF-based estimators [3]. The position drift is also the likely cause of the larger position errors of the KF approach for the second test. The WA approach was not subject to drift in the first two tests because the foot update rate had been set to zero (as already discussed in Section IV-G).

Despite the unobservability of the yaw angle, the orientation errors for the last test are comparable to the ones of the first two tests. We expect this to be no longer the case during longer walking experiments.

Regarding the velocity estimation, the KF largely outperformed the WA approach. This is likely due to the reduced delay of the KF velocity estimation, achieved thanks to the use of the accelerometer measurements.

Overall, introducing the weighting functions has lead to better results. This is especially true for the second experiment (see Fig. 5), in which the left foot was often in contact only with one edge. This made the corresponding weight decrease, thus the estimator relied more on the information coming from the right leg kinematics.

The same can be said for the introduction of the IMU data. The improvements due to the IMU data are negligible for the orientation, but they are significant for the position. This is due to the fact that small changes in the base orientation can lead to large changes in the positions of the feet. Thanks to the 2-stage approach, the slight improvements in orientation estimation lead to improvements up to 30% for the position estimation.

In conclusion, the estimation accuracy achieved by the presented simple estimators is comparable to the accuracies reported in the literature by EKF-based estimators. This suggests that these simple estimators could be sufficient to perform feedback control of humanoid robots. The KF estimator provided a much better velocity estimation than the WA estimator (at the expense of a slightly larger computation time), so we think it should be preferred if a velocity feedback is necessary.

V. CONCLUSION

This paper presented two simple state estimators for the floating-base pose and velocity, and benchmarked them through in-place balancing and walking experiments with the humanoid robot HRP-2. Both estimators are based on off-the-shelf IMU-based orientation estimators, which are used to estimate the orientation of the robot base. Once the base orientation is fixed, two approaches are presented to estimate its position and velocity. Both methods use weighting functions based on the F/T measurements of the contact forces, which help detect violations of the contact assumptions. The presented results show the importance of the IMU and F/T sensor measurements in improving the kinematics-based estimation. Despite their simplicity compared to EKF-based methods, both estimators performed well in the tested scenarios. For in-place balancing, average position errors are well below 1 cm, while orientation errors are below

1°. For velocity estimation, the second approach greatly outperformed the first one, showing average errors about 40% smaller. During walking, both estimators exhibited a drift in the unobservable components (i.e. position and yaw angle), which is comparable to the one reported for EKF-based methods [3]. Thanks to their simplicity, both estimators reported extremely fast computation times (below $20 \mu s$), which makes them suitable for high-frequency feedback control.

While accuracy has been our main criterion to evaluate the quality of state reconstructions, it is not the only criterion allowing to predict the performance of an estimator in closed-loop control. Other parameters such as stability, global convergence rate and robustness to sensor noise and modeling errors can deeply affect the global stability of the control, especially for highly-dynamic motions or for high-gain control. We plan to investigate these criteria in our future work. Moreover, we would like to expand our benchmarks including one of the EKF-based methods, to quantify the performance loss potentially induced by the orientation-position decoupling.

REFERENCES

[1] B. Gassmann, F. Zacharias, J. M. Zollner, and R. Dillmann, "Localization of Walking Robots," in *IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 1471–1476.

[2] M. Reinstein and M. Hoffmann, "Dead reckoning in a dynamic quadruped robot: Inertial navigation system aided by a legged odometer," in *IEEE Int. Conf. on Robotics and Automation*, May 2011, pp. 617–624.

[3] M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C. David Remy, and R. Siegwart, "State Estimation for Legged Robots - Consistent Fusion of Leg Kinematics and," in *Robotics: Science and Systems (RSS)*, 2012.

[4] N. Rotella, M. Bloesch, L. Righetti, and S. Schaal, "State Estimation for a Humanoid Robot," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2014.

[5] M. Benallegue and F. Lamiraux, "Estimation and stabilization of humanoid flexibility deformation using only inertial measurement units and contact information," *International Journal of Humanoid Robotics*, vol. 13, no. 3, pp. 1 550 025:1–20, 2015.

[6] M. Benallegue, A. Mifsud, and F. Lamiraux, "Fusion of force-torque sensors, inertial measurements units and proprioception for a humanoid kinematics-dynamics observation," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2015.

[7] N. Rotella, A. Herzog, S. Schaal, and L. Righetti, "Humanoid momentum estimation using sensed contact wrenches," *IEEE-RAS International Conference on Humanoid Robots*, pp. 556–563, 2015.

[8] M. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and LIDAR sensing," in *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-Febru, nov 2015, pp. 112–119.

[9] H.-w. Park, P. M. Wensing, and S. Kim, "Online planning for autonomous running jumps over obstacles in high-speed quadrupeds," in *Robotics: Science and Systems*, 2015.

[10] S. Faraji, L. Colasanto, and A. J. Ijspeert, "Practical considerations in using inverse dynamics on a humanoid robot: Torque tracking, sensor fusion and Cartesian control laws," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2015, pp. 1619–1626.

[11] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-conrolled humanoid robots in multi-contact scenarios," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016.

[12] M. Johnson, B. Shrewsbury, S. Bertrand, T. Wu, D. Duran, M. Floyd, P. Abeles, D. Stephen, N. Mertins, A. Lesman, J. Carff, W. Rifenburgh, P. Kaveti, W. Straatman, J. Smith, M. Griffioen, B. Layton, T. de Boer, T. Koolen, P. Neuhaus, and J. Pratt, "Team IHMC's Lessons Learned from the DARPA Robotics Challenge Trials," *Journal of Field Robotics*, vol. 32, no. 2, pp. 192–208, 2015.

[13] B. Stephens and C. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in *IEEE-RAS International Conference on Humanoid Robots*, Dec 2010, pp. 52–59.

[14] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," Tech. Rep., 2010.

[15] A. Del Prete and N. Mansard, "Robustness to Joint-Torque Tracking Errors in Task-Space Inverse Dynamics," *IEEE Transaction on Robotics*, vol. 32, no. 5, pp. 1091 – 1105, 2016.