Centroidal State Estimation and Control for Hardware-constrained Humanoid Robots

Grzegorz Ficht and Sven Behnke

Abstract— We introduce novel methods for state estimation, feedforward and feedback control, which specifically target humanoid robots with hardware limitations. Our method combines a five-mass model with approximate dynamics of each mass. It enables acquiring an accurate assessment of the centroidal state and Center of Pressure, even when direct forms of force or contact sensing are unavailable. Upon this, we develop a feedforward scheme that operates on the centroidal state, accounting for insufficient joint tracking capabilities. Finally, we implement feedback mechanisms, which compensate for the lack in Degrees of Freedom that our NimbRo-OP2X robot has. The whole approach allows for reactive stepping to maintain balance despite these limitations, which was verified on hardware during RoboCup 2023, in Bordeaux, France.

I. INTRODUCTION

Robust bipedal locomotion remains a coveted research goal, still posing several challenges despite much work in the field. The difficulties stem not only from the high dimensionality of the problem, underactuation, and complex dynamics, but also originate from practical issues such as robot hardware limitations.

If real-time operation and the gap between models and reality wouldn't be an issue, optimization-based methods could be used to plan motions capable of exploiting complex dynamics [1][2]. Reduced-order models provide tractability by focusing on the core dynamics of the system, e.g. planning and executing Center of Mass (CoM) trajectories given a sequence of Center of Pressure (CoP) locations (footholds). Examples include Linear Quadratic Regulation (LQR) [3] and Model Predictive Control [4]. The simplified dynamics often allow for finding closed-form solutions, such as for stabilizing the Zero Moment Point (ZMP) [5]. The problem can also be further reduced by splitting the CoM dynamics into stable and unstable components, where only the unstable ones need to be handled. The unstable part is usually called the Instantaneous Capture Point [6], or its 3D generalization: The Divergent Component of Motion [7][8]. Single mass models can also be extended with a flywheel, which models the Centroidal Angular Momentum (CAM) generated around the CoM [6]. In recent years, the concept of Centroidal Dynamics [9] is becoming more popular, allowing for expressive upper body movement and CAM regulation [10]. In this regard, there are still open questions, such as finding a meaningful *angular equivalent* to CoM [11] or generating reference CAM trajectories [12]. Our most recent work [13]

All authors are with the Autonomous Intelligent Systems (AIS) Group, Computer Science Institute VI, University of Bonn, Germany. Email: ficht@ais.uni-bonn.de.
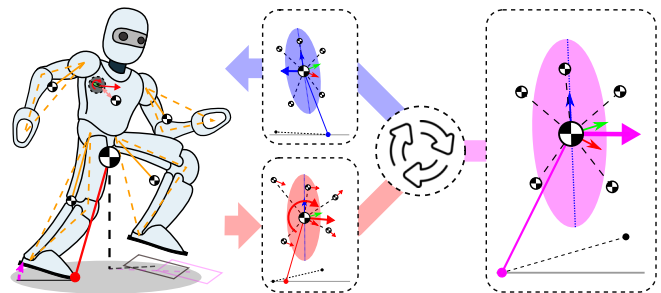

Fig. 1. Centroidal Control of humanoid robots with hardware limitations. The IMU readings and limb dynamics are fused into a centroidal state of the robot. Three of such models form the basis of a feedforward compensator. Reactive stepping feedback is used to counter tilting due to the robot's kinematic limitations.

falls into the CoM-ZMP category, but also tackles the mentioned issues related to CAM, where we propose to use the orientation of the Principal Axes of Inertia for direct CAM regulation.

A common issue of adopting ZMP-based approaches in physical robots is the assumption that the joint trajectories producing stable motions can be accurately tracked. In lower quality hardware, where torque is limited, latencies are significant, and backlash is present, this is not the case. Our humanoid NimbRo-OP2X [14] falls into this category, and—so far—was able to walk only by using hand-crafted gaits with Central Pattern Generation (CPG) [15][16]. To achieve better tracking performance, feedforward techniques are typically applied on the joint level. These methods often require some system and parameter identification and runtime computation of expected loads [17]. The modelling complexity of such schemes can approach extreme levels [18], but learning-based approaches can alleviate the need to deal directly with these intricacies [19]. The robot itself might also not be capable of such dynamic movement, though high-bandwidth Quasi-Direct Drive (QDD) actuator technology currently used in quadrupeds allows for shaping ground reaction forces within milliseconds [20]. Feedforward techniques remain an important component for bipedal walking, at least until humanoids utilizing QDD actuation become more widespread [21].

Another requirement for ZMP control is to actually sense the CoP, which is done using Force-Torque (FT) sensors. FT sensors are versatile in the sense that they measure not only if the limb is in contact with the environment, but what wrench is acting on the system through it. However, they suffer from noise and creep due to the constant forces they are subjected to. In systems like ours, without any

direct form of contact and force sensing, estimating the external wrench working on the it and assuring balance is not trivial. Current non-FT ZMP estimators still rely on having Force Sensitive Resistors (FSRs), or similarly to our previous work [13] assume the measured torso acceleration approximates the CoM acceleration [22]. In this paper, we show that a higher-accuracy estimate can be achieved by including limb dynamics, even without foot sensors altogether. Aside from restricted sensing options, our system is also limited in its kinematic structure. Due to a double 4-bar linkage parallel kinematics design, the legs have only 5 Degrees of Freedom (DoF) and leave the torso-foot pitch angle locked to be orthogonal. This makes the robot prone to tilting, and poses an additional challenge to the balance controller.

The contributions of this paper can be split into three main components, (see Fig. 1):

- an accurate centroidal state estimator that uses a five-mass model incorporating limb dynamics;
- a simple, yet effective feedforward scheme, compensating for state tracking inaccuracies by operating on the centroidal state; and
- hardware-focused correction mechanisms for a robot with kinematic restrictions, extending previous work [13].

Ultimately, we show that even limited hardware can achieve relatively high performance in bipedal walking with adequate state estimation and control.

II. FIVE-MASS MODEL

The following section summarizes our five-mass representation, describing how single mass kinematics and dynamics relate to joint angles and overall system properties.

A. Limb CoM

The Center of Mass (CoM) for each limb is described using a triangle parameterization, denoted by $P = (p_s, p_l) \in [0,1] \subset \mathbb{R}$. The *side distribution* p_s divides the lower link of the limb, forming a vector between the division point and the limb's origin. The *length distribution* p_l represents the ratio of the centroid's distance to said vectors length. For example, a triangle with uniform density has $p_s = \frac{1}{2}$, $p_l = \frac{2}{3}$, and the origin to CoM vector lies on a median (see Fig. 2).

Both the kinematics and dynamics of a limb's mass $\mathbf{m}_*$ are predominantly determined by the limb extension and rotation. The weight of the end effector (hand, foot) is included with adequate P values, but its orientation is omitted due to the negligible effect it has on $\mathbf{m}_*$. Hence, only the first n limb joints $\mathbf{q}_*$ are necessary to obtain $\mathbf{m}_*$; with the limb transform $\mathbf{T}_*$ tying these values through P:

$$\mathbf{m}_* = \mathbf{o}_* + \mathbf{T}_* ,$$

$$\mathbf{T}_* = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \mathbf{R}_O \begin{bmatrix} -a p_l p_s \sin(q_n) \\ 0 \\ -(a p_l p_s \cos(q_n) + c p_l) \end{bmatrix} , \quad (1)$$

where q_n is the bending joint angle (knee, elbow), a, c are triangle sides, and $\mathbf{R}_O$ is a matrix obtained from a sequence
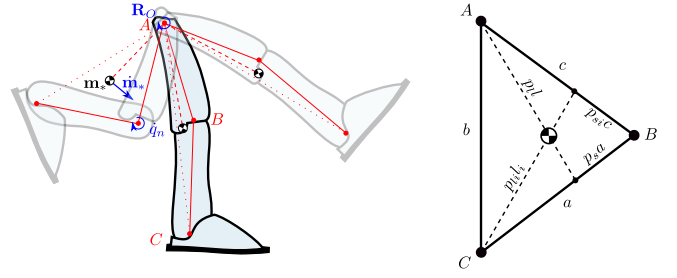

Fig. 2. Limb kinematics and dynamics in relation to mass positioning. Using a triangle mass distribution—parametrized in P—a direct relation between limb mass movement and joint angles is achieved.

of $n-1$ rotations performed at the limb origin $\mathbf{o}_*$ (hip, shoulder). We can then compute the limb mass Jacobian $\mathbf{J}_*$:

$$\mathbf{J}_* = \begin{bmatrix} \frac{\partial T_x}{\partial q_1} \cdots \frac{\partial T_x}{\partial q_n} \\ \frac{\partial T_y}{\partial q_1} \cdots \frac{\partial T_x}{\partial q_n} \\ \frac{\partial T_z}{\partial q_1} \cdots \frac{\partial T_x}{\partial q_1} \end{bmatrix} ; \quad \dot{\mathbf{m}}_* = \mathbf{J}_* \dot{\mathbf{q}}_* \quad (2)$$

tying the limb joint velocities $\dot{\mathbf{q}}$ to its CoM velocity $\dot{\mathbf{m}}_*$, which will play an important role in Section III.

B. System CoM and Inertia

The root of the robot is defined as a floating 6 DoF base frame $\boldsymbol{F}_B$ attached to the torso with an orientation of $\mathbf{R}_t$ and its position centered between the hip joint origins $\mathbf{h}_m$. The head—due to its limited range of motion—is considered as a part of the torso, with their combined mass position $\mathbf{m}_t$ set at a configurable offset from $\boldsymbol{F}_B$. This leaves its movement unconstrained, which is advantageous for visual tracking algorithms. The robot's limbs originate at an offset $\mathbf{o}_*$ to the torso and their joint configuration results in the (left and right) leg $\mathbf{m}_{ll}$, $\mathbf{m}_{rl}$ and arm $\mathbf{m}_{la}$, $\mathbf{m}_{ra}$ CoM (Section II-A). Combining these five masses $\mathbf{m}_i$ and accounting for their physical weight w_i we achieve the overall CoM position $\mathbf{m}_c$:

$$\mathbf{m}_c = \frac{\sum_i (\mathbf{m}_i w_i)}{\sum_i w_i} ; \quad i \in \{t, ll, rl, la, ra\}. \quad (3)$$

The five masses also shape the robot's inertia $\mathbf{I}_R$, which we partition into the principal axis form using the moments $\mathbf{I}_{PA}$ and orientation $\mathbf{R}_I$:

$$\mathbf{I}_R = \mathbf{R}_I \mathbf{I}_{PA} \mathbf{R}_I^\top = \mathbf{R}_I \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \mathbf{R}_I^\top. \quad (4)$$

The leg masses form a dumbbell which rotates around their aggregate mass $\mathbf{m}_l$ named the *lower body* mass. This is repeated for the *upper body*, where the arm masses are combined with half of the torso mass each into $\mathbf{m}_{lu}$, $\mathbf{m}_{ru}$ and rotate around $\mathbf{m}_u$. The vector from $\mathbf{m}_u$ to $\mathbf{m}_l$ passes through $\mathbf{m}_c$ and its normal sets the *tilt* z-axis $\mathbf{z}_I$ of the orientation $\mathbf{R}_I$. The definition is complete with the *yawing* component ψ_I, computed from weighted yaw angles of the upper ψ_u and lower dumbbells ψ_l, around $\mathbf{z}_I$:

$$\psi_I = \frac{\psi_u w_u + \psi_l w_l}{w_u + w_l} , \quad (5)$$

$$w_u = w_t + w_{ra} + w_{la}, \qquad w_l = w_{ll} + w_{rl}.$$

The computation of $\mathbf{R}_I$ from $\mathbf{z}_I$ and ψ_I follows the intermediate tilt representation from [23].

Our five-mass representation describes the physical properties of the system with high fidelity across a wide range of configurations. The tunable mass parameters can account for even large modelling errors and be fitted to the actual robot, greatly simplifying the hardware transfer. As the mapping between our five-mass representation and kinematics is unique, we developed an inertia-shaping whole-body inverse kinematics technique in [24]. By providing the foot-relative desired centroidal properties $\mathbf{m}_c$, $\mathbf{R}_I$, and $\mathbf{I}_{PA}$ over time, we achieve simultaneous control over linear and angular momenta in a uniform fashion. The splitting of the inertia tensor into the Principal Axes form and the defined, continuous behavior of $\mathbf{R}_I$ as an *angular counterpart* to the CoM is what sets our approach apart from typically used methods, e.g. the CoM Jacobian.

III. State Estimation

The robot's state can be captured through various quantities, some of which can be observed directly with physical sensors, others indirectly using soft sensors. The Inertial Measurement Unit (IMU) is crucial to our approach. It is a versatile sensor providing not only the floating base (torso) orientation $\mathbf{R}_t$ [25] [26] [27], but also the external forces acting on it through accelerations. The pose of the robot is reconstructed by measuring the joint angles $\mathbf{q}$ with encoders, which also measure $\dot{\mathbf{q}}$. Next, foot frames $\boldsymbol{F}_{FR}$, $\boldsymbol{F}_{FL}$ are obtained, where the lower one is assumed to be the supporting leg. To prevent unwanted toggling of support exchanges, a threshold on the height difference is set. With the whole pose reconstructed, we compute the centroidal properties $\mathbf{m}_c$ and $\mathbf{I}_R$.

A. Linear Motion

Considering the system as a point mass simplifies analyzing the forces acting on it, which can be directly inferred through accelerations. In previous work [13], we used a Kalman Filter (KF) to combine $\mathbf{m}_c$ and sensed torso acceleration $\ddot{\mathbf{r}}_s$, with the assumption that $\ddot{\mathbf{r}}_s \approx \ddot{\mathbf{m}}_t \approx \ddot{\mathbf{m}}_c$. The torso acceleration is rotated from its own frame $^T\ddot{\mathbf{m}}_t$, into a global $\ddot{\mathbf{m}}_t$, with gravity $\mathbf{g}$ subtracted:

$$\ddot{\mathbf{m}}_t = {}^T\ddot{\mathbf{m}}_t \mathbf{R}_t - \mathbf{g}. \tag{6}$$

This provides fast and sufficiently precise linear motion state estimates, when the CAM is close to zero. This simple method is insufficient, though, when the torso is rotating or when walking at higher speeds—due to significant limb swinging. By addressing these two issues, we can reduce the reliance on this assumption. Firstly, the IMU sensor mounting $\mathbf{r}_s$ rarely coincides with $\mathbf{m}_t$. Thus, angular accelerations of the torso $\ddot{\boldsymbol{\theta}}_t$ contribute a tangential component to the linear $^T\ddot{\mathbf{r}}_s$ measurement:

$$^T\ddot{\mathbf{r}}_s = {}^T\ddot{\mathbf{m}}_t + \ddot{\boldsymbol{\theta}}_t \times (\mathbf{r}_s - \mathbf{m}_t). \tag{7}$$

To obtain $\ddot{\boldsymbol{\theta}}_t$, we simply perform finite numerical differentiation on the gyro measurements. The amplified noise from

the differentiation is handled by increasing the corresponding measurement covariance matrix values of the KF. By rearranging and incorporating (7) into (6), we remove the $\ddot{\mathbf{r}}_s \approx \ddot{\mathbf{m}}_t$ assumption and achieve a purely linear $\ddot{\mathbf{m}}_t$.

The second issue refers to the correlation between $\ddot{\mathbf{m}}_t$ and $\ddot{\mathbf{m}}_c$, in light of the mass distribution. The torso is the root link anchoring the limbs and their movement will generate reaction forces around their origin. These forces displace the torso, similarly to external ones which the IMU detects through $\ddot{\mathbf{r}}_s$ and $\ddot{\boldsymbol{\theta}}_t$ as disturbances. We incorporate limb dynamics by considering each limb's *global* acceleration $\ddot{\mathbf{m}}_*$—a combination of the origin's absolute and origin-relative mass accelerations:

$$\ddot{\mathbf{m}}_* = \ddot{\mathbf{m}}_t + \ddot{\boldsymbol{\theta}}_t \times (\mathbf{o}_* - \mathbf{m}_t) + {}^T\ddot{\mathbf{m}}_*\mathbf{R}_t. \tag{8}$$

The total mass acceleration $\ddot{\mathbf{m}}_c$ is then achieved by weighing the five-mass accelerations similarly to (3). This is then supplied into the KF instead of $\ddot{\mathbf{m}}_t$, thus removing the $\ddot{\mathbf{m}}_t \approx \ddot{\mathbf{m}}_c$ assumption. The origin-relative limb mass accelerations $^T\ddot{\mathbf{m}}_*$ were obtained through differentiation of $^T\dot{\mathbf{m}}_*$ from (2).

B. Angular Motion

Concurrent to linear motion, the contribution of angular motion is equally important in shaping the system dynamics. Control over it necessitates estimating the angular state. We have briefly introduced it in [13], with the state being reconstructed purely from Euler angles $\boldsymbol{\theta}_I$ encoded in the inertia orientation $\mathbf{R}_I$. There was a significant delay associated with this approach though, which rendered the estimates inoperable. Specifically, before the state of angular accelerations finally converged, the real accelerations could have already significantly changed. Similarly to the linear motion estimation, we can improve on this by combining the inertia angles $\boldsymbol{\theta}_I$ with the angular accelerations of the five masses around $\mathbf{m}_c$ (denoted as $^c\ddot{\boldsymbol{\theta}}_i$) in the measurement model of the KF. For each of the masses $\ddot{\mathbf{m}}_i$, it is important to consider that CAM is generated only if they differ from $\ddot{\mathbf{m}}_c$ [28]. The accelerations of the five masses $^c\ddot{\boldsymbol{\theta}}_i$ are then:

$$^c\ddot{\boldsymbol{\theta}}_i = \frac{(\mathbf{m}_i - \mathbf{m}_c) \times (\ddot{\mathbf{m}}_i - \ddot{\mathbf{m}}_c)}{\|(\mathbf{m}_i - \mathbf{m}_c)\|^2}. \tag{9}$$

Again, simply weighing these angular accelerations as in (3) yields the total system centroidal acceleration $^c\ddot{\boldsymbol{\theta}}_I$. Alternatively, the angular motion could be calculated using velocities. As $\dot{\boldsymbol{\theta}}_t$ along with all limb velocities $\dot{\mathbf{m}}_*$ are readily available, the only requirement would be to estimate $\dot{\mathbf{m}}_t$. We decided against this as it would require chaining filters, potentially introducing stability issues.

IV. Control

The following section describes all the control components specific to the presented approach. As the methods presented are an extension of the authors previous work [13], it is recommended to use it as reference for a better overview.

A. Reference Trajectories

The open-loop trajectory generation scheme is similar to the one previously presented in the DCC [13], albeit with modifications and improvements. The Center of Mass $\mathbf{m}_c$ and its dynamics resolve with time t from two planar ($\mathbf{m}_{c,x}, \mathbf{m}_{c,y}$) Linear Inverted Pendulum Models (LIPM) with a height of $\mathbf{m}_{c,z} = h$ and Center of Pressure (CoP) $\mathbf{r}_p$ [29]:

$$\ddot{\mathbf{m}}_{c,x} = \omega^2(\mathbf{m}_{c,x} - \mathbf{r}_{p,x}), \quad \omega = \sqrt{\mathbf{g}_z/h},$$

$$\mathbf{m}_{c,x}(t) = \mathbf{m}_{c,x_0}\cosh(\omega t) + \dot{\mathbf{m}}_{c,x_0}\frac{1}{\omega}\sinh(\omega t), \quad (10)$$

$$\dot{\mathbf{m}}_{c,x}(t) = \mathbf{m}_{c,x_0}\omega\sinh(\omega t) + \dot{\mathbf{m}}_{c,x_0}\cosh(\omega t).$$

Walking velocity $\mathbf{v}_g$ is controlled by setting a constant gait frequency f_g and altering the stride length. Step placement $\mathbf{s}$ is initially selected to satisfy the symmetry constraint, but later refined to maintain continuous trajectories [30]. Previously, the swing foot trajectory would linearly progress from the last foothold to next one according to the step progression μ. With the accumulated joint compliance and tracking errors, often the swing foot would contact the ground prematurely and still try to move. This degraded performance, introduced instabilities, and was harmful to the hardware. We alter the trajectory progression to μ_a with a customizable sigmoid $\sigma(\mu, p, m)$:

$$\mu_a = \frac{\sigma(\mu) - \sigma(0)}{\sigma(1) - \sigma(0)}, \quad \sigma(\mu, p, m) = \frac{1}{1 + e^{p(m+0.5-\mu)}}, \quad (11)$$

$$\mu \in [0, 1), \qquad \mu[n+1] = \mu[n] + f_g\Delta t.$$

While $m \in (-1, 1)$ shifts the midpoint of the output, p adjusts the steepness of the sigmoid: $p \to 0$ results in a linear progression and $p \to \infty$ in a step function. The combination of these parameters allows for a speed-up in the progression and a settling period at the beginning of a step, its end, or both.

To finalize the trajectory, the inertial properties $\mathbf{I}_R$ need to be set. The principal moments $\mathbf{I}_{PA}$ are kept constant from a desired halt-pose, which are then usually adjusted by the whole body pose generator to satisfy other constraints ($\mathbf{m}_c$, $\mathbf{R}_I$). The robot's inertia orientation $\mathbf{R}_I$ is set by combining a desired tilt z-axis $\mathbf{z}_I$ and yaw angle ψ_I (aligned with the step yaw progression). To prevent tilting moments around the CoM, $\mathbf{z}_I$ is aligned with the robot tilt axis, giving a *zero CAM heuristic*. In [24], the robot tilt axis was based purely on averaged foot frame position vectors $\mathbf{r}_{FR}$, $\mathbf{r}_{FL}$ pointed towards the CoM. With a kinematically locked torso, shaping the inertia is severely limited and might force the arms to overcompensate with undesired configurations or rapid movement. We circumvent that by including $^T\mathbf{m}_t$ and mass distribution in the calculation, thus finalizing $\mathbf{R}_I$:

$$\mathbf{z}_I = \frac{w_u}{\Sigma w_i}(^T\mathbf{m}_t) - \frac{w_l}{2\Sigma w_i}(\mathbf{r}_{FL} + \mathbf{r}_{FR}). \quad (12)$$

B. Feedforward Compensation

To account for known disturbances, we apply a feed-forward control scheme. Unlike typically used methods which offset the position or provide expected torques to
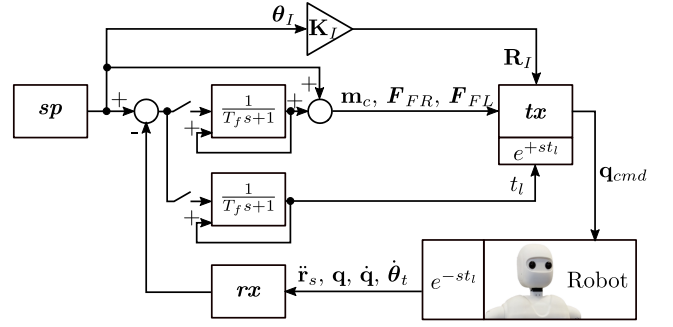

Fig. 3. Scheme of the feedforward compensation. The *sp* model provides nominal setpoint trajectories, which get transmitted to the *tx* model. Once the Lowpass Filters get enabled, they will start comparing the *sp* values to the received ones in *rx*, which will make the *tx* diverge not only in values, but also forward it in time. This in turn brings *rx* closer to *sp*.

controllers in joint space, our approach focuses on the quantities relevant for the centroidal dynamics. While walking, a leg can either be supporting or swinging, which is tightly connected to the loads it is subjected to and thus how fast it can move. Factors such as external disturbances, joint friction, backlash, compliance, torque limitations, and sensorimotor delays can all contribute to poor state tracking. In the centroidal space, we identified and correct for the following cases:

- **CoM to foot vector:** Both with respect to the support and swing foot, we track the x, y values of $\mathbf{m}_c - \mathbf{r}_{FL/FR}$ to assure that steady-state offsets $\mathbf{r}_{sup}^{ff}$, $\mathbf{r}_{sw}^{ff}$ are generated and compensate for the torque necessary to carry the robot's mass, or place the swing foot in the desired position.
- **CoM and step height:** Similarly to the x, y displacement, from $\mathbf{m}_c, z - \mathbf{r}_{FL/FR,z}$ we infer if either of the feet need to be pushed down or pulled up to accommodate for carrying the respective body or leg weight.
- **CoM velocity:** Due to dynamically changing loads within the gait cycle as well as wrongly placed feet, the CoM might accumulate a velocity error $\dot{\mathbf{m}}_c$ that needs to be taken into consideration by applying $\mathbf{r}^v$.
- **Swing foot orientation:** The orientation of the swing foot $\mathbf{R}_{FR/FL}$ is equally important to its placement, as it allows for a smooth support exchange to take place.
- **Support exchange event:** If the moment when the foot hits the ground varies from the expected one, then there exists a sensorimotor delay (latency) to account for. We quantify it by measuring the time difference t_l.
- **Inertia orientation:** By measuring the Euler angles $\boldsymbol{\theta}_I$ of $\mathbf{R}_I$, it is easily observable if the upper body has sufficient torque to move as requested. Gains $\mathbf{K}_I$ are applied to the set angles of the inertia rotation to either enforce more or decrease the rotation of the upper body.

The feedforward compensation scheme consists of three model states: *rx*, *sp*, *tx* and a block of Lowpass Filters (LPF), as shown in Fig. 3. During walking, robot measurements of the above-mentioned quantities are put into *rx* and compared against setpoint values *sp* with the result placed in the

respective value LPF. Due to the symmetry in the lateral movement, all y values and the roll foot angle are unified with a multiplication by the leg sign ι: $\{L = -1, R = 1\}$. Toggling the filter input switch effectively enables/disables the feedforward term estimation process. Once the filters have been enabled, their output value will slowly approach the necessary compensation value. At the same time, the corrective terms are applied to the *tx* model and transmitted to the robot, decreasing the *sp − rx* errors. Before applying the corrections, the *tx* trajectories are forwarded in time by the estimated t_l, which essentially predicts away the latency. The feedback loop in the LPF effectively turns them into integrators, keeping their value even if their input is disabled. The LPFs time constant T_f approximately describes the necessary time for the feedforward terms to settle; we set it to a duration of several steps. The compensation variables are estimated at zero and nonzero walking velocities. Based on the desired walking speed, they are linearly interpolated for uniform operation.

The usage of the feedforward term depends on the controlled quantity. Unlike the straightforward application of the leg height and foot angles, the x, y placement of the feet is less trivial. The trajectories need to be continuous between steps, meaning that they need to tolerate coordinate frame changes and smoothly fade out. The constant support $\mathbf{r}_{sup}^c$ and swing $\mathbf{r}_{sw}^c$ offsets, and combined with the velocity offset $\mathbf{r}^v$ into complete feedforward terms $\mathbf{r}_{sup}^{ff}$, $\mathbf{r}_{sw}^{ff}$ applied to the feet:

$$
\begin{aligned}
\mathbf{r}^v &= \left[\frac{\mu - 0.5}{f_g} v_x \,,\, 0 \,,\, 0 \right], \\
\mathbf{r}_{sup}^{ff} &= \mathbf{r}_{sup}^c - \mathbf{r}^v, \\
\mathbf{r}_{sw}^{ff} &= \mathbf{r}_{sw}^c + \mathbf{r}_{sup}^c + \mathbf{r}^v,
\end{aligned}
\tag{13}
$$

where v_x is the estimated sagittal velocity offset. In the lateral direction, the foot placement was sufficient in achieving the desired velocity. As the swing foot can move more freely and is also carried by the support foot, we include $\mathbf{r}_{sup}^c$ in its applied feedforward vector $\mathbf{r}_{sw}^{ff}$. Finally, the feedforward terms are applied to the left and right foot, depending on the supporting leg ι_{tx}.

C. CoM Feedback Control

Feedback control is assured with a CoM-ZMP controller initially proposed by Choi et al. [31] and extended by us in [13]. The controller realizes input-to-state stability by adjusting $\dot{\mathbf{m}}_c$ to steer $\mathbf{m}_c$ back to follow the nominal trajectories in the presence of errors, while considering the CoP position $\mathbf{r}_p$. We modify our controller further by using the Centroidal Moment Pivot (CMP) instead of ZMP, and include $\mathbf{m}_c$ rate limiters to avoid large, abrupt CoM changes.

D. Tilt-compensating Step Feedback

By locking the foot pitch DoF, our robot is prone to tilting. Sufficiently large disturbances that are not naturally damped or dispersed by the feedback controller will create a torque in the sagittal direction and make the robot pivot about the edge of its feet. Depending on the combination of tilt and walking
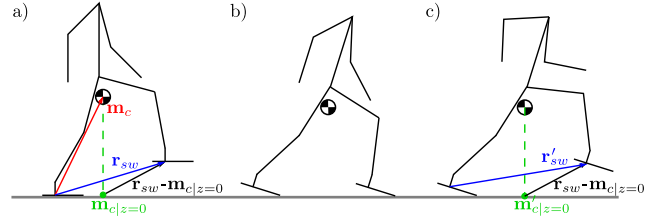

Fig. 4. Visual representation of the tilt compensation mechanism. a) Nominal trajectory execution. b) Preemptive touchdown due to unexpected tilt. c) Corrective mechanism assuring approximate desired touchdown is achieved.

direction, the foot will either prematurely hit the ground at the wrong position or stay mid-air without performing a support exchange. Both of these situations result in quick loss of stability. To account for this, we transform the swing foot trajectory and aim for the swing foot to contact the ground at the correct x-distance from the CoM. We do so by: taking the nominal swing foot distance to the zero-tilt ground-projected CoM: $\mathbf{r}_{sw} - \mathbf{m}_{c|z=0}$, applying it to the tilt-rotated nominal CoM projection $\mathbf{m}'_{c|z=0}$, unrotating the resulting $\mathbf{r}'_{sw}$ and setting it as the commanded swing foot placement (see Fig. 4). As a result, the step should place the CoM onto a trajectory close to the nominal one. In combination with the CoM controller, this is sufficient for the robot to tolerate moderate disturbances. For greater disruptions in the gait, we not only transform the foot trajectory but also recalculate the desired step placement. We do so by plugging the current state $\mathbf{m}_c$, $\dot{\mathbf{m}}_c$ and the remaining step time $\frac{1-\mu}{f_g}$ into the LIPM equations (10) to predict the End Of Step (EOS) state $\mathbf{m}_c^{eos}$ and compute its expected EOS error $\mathbf{e}_c^{eos}$. Instead of purely using the prediction, we linearly interpolate between the EOS error and the current CoM tracking error $\mathbf{e}_c = \mathbf{m}_c^{ref} - \mathbf{m}_c$ according to time passed. Using the symmetry assumption [30] we offset the next nominal step:

$$
\mathbf{s}\,[k+1]^{new} = \mathbf{s}\,[k+1] + 2(1-\mu)\mathbf{e}_c + 2\mu\mathbf{e}_c^{eos}. \tag{14}
$$

This assures that predictions made directly after a support exchange (when the foot is still aligning to the ground and CMP might fluctuate) do not falsely steer the gait. Naturally, once these errors reduce so does the commanded velocity and the robot returns to the nominal walking velocity.

V. Experimental Results

The proposed approach is verified with a 135cm tall, NimbRo-OP2X humanoid [14] with 18 Degrees of Freedom. For ground-truth comparisons of estimates and quantitative results, we use a full MuJoCo simulation of the robot that also models the parallel linkages. The whole motion generation, estimation and control framework is running on-board the robot's computer, with a frequency of 100 Hz.

A. State Estimation

To display the contribution of the limb dynamics in state estimation accuracy, the robot is tasked to slightly pitch its inertia $\mathbf{R}_I$ backwards while keeping $\mathbf{m}_c$ still. As the pitch
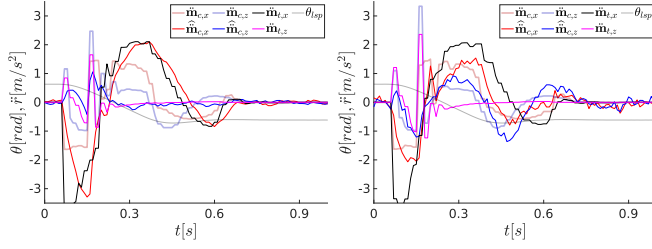
Fig. 5. Comparison of a purely IMU-based estimator (left), and one which includes limb dynamics (right). The latter allows to capture the actual CoM dynamics, instead of the torso's. During both experiments, the robot is shifting its torso back and rotating fully extended arms forward. Left shoulder pitch angle, ground-truth torso and CoM accelerations along with their estimates are shown.
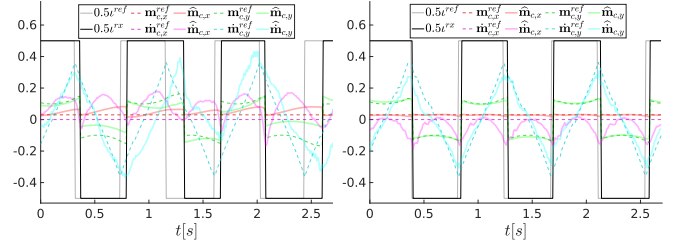

Fig. 6. Effect of the feedforward compensation on the robots CoM state. Without the compensation (left), the robot is unable to maintain the desired nominal trajectories and keeps walking forward. With each step taken, the feet are placed at the wrong time and place. This consequently places the CoM on an undesired trajectory, disallowing the robot to get into a cyclic gait. After estimating and applying the feedforward terms (right), the feet are offset to correct for the errors, allowing the robot to follow the nominal state.

angle of the torso is kinematically locked, the pose generator shifts the hip backward and rotates the fully extended arms towards the front to satisfy the constraints. The result of this experiment comparing the ground-truth and estimates of relevant variables, is shown in Fig. 5. As shown in [22], assuming that the IMU and CoM accelerations are synonymous is currently the most relevant approach when no force or torque measurements are available. Our previous estimator falls into this category and will serve as a baseline for further comparisons. Although it reacts swiftly to the movement, it does not capture $\ddot{\mathbf{m}}_c$ well. It is also quite natural that it barely perceives any changes in $\ddot{\mathbf{m}}_{c,z}$, as the torso just translates its position. The arms by rotating around their shoulder joints, change their mass position on the z-axis over time. This slightly shifts the aggregate mass of the robot in the z-direction, which is clearly observable with the presented approach. The newly included limb accelerations accurately offset the torso measurements and provide estimates visibly closer to $\ddot{\mathbf{m}}_c$.

We also verify the estimator in a more dynamic setting, where the simulated robot is made to walk forward open-loop at a constant velocity for 10 seconds. During this, the ground-truth and estimate values were gathered for the 3D CoM velocities and accelerations. From these, we computed the median $\mathbb{M}(e)$, standard deviation $\text{SD}(e)$ and average absolute $\frac{\Sigma|e|}{T}$ of the estimation errors. The results of these calculations were placed in Table I, with the top showing the previous, purely IMU-based estimator (baseline), and the bottom having the limb dynamics included. A decrease

in the estimation errors is clearly visible, which is most apparent in $\ddot{\mathbf{m}}_{c,x}$. With all of the limbs swinging back and forth, their significant contribution to the overall system dynamics is simply omitted by the old method (as also shown in Fig. 5). An interesting result arose with the $\ddot{\mathbf{m}}_{c,z}$ estimate. Although the filter is able to capture more detail in the z-direction, it exhibits a higher standard deviation and average absolute error. The reason for this was made clear upon further inspection of the data. Increasing the values of the measurement covariance matrix to deal with numerical differentiation, resulted in the KF suppressing spikes of $\ddot{\mathbf{m}}_{c,z}$ occurring at every foot strike (support exchange). This is also visible in Fig. 5.

B. Feedforward Compensation

We demonstrate the output produced by the developed compensation mechanisms in Fig. 6, where in both cases the robot is asked to walk *on the spot*. Without the feedforward components, the robot is unable to get into, or maintain a uniform walking rhythm. The timing, duration and sizes of the steps are all completely off, due to the dynamically changing load on the joints. In the sagittal direction the robot consistently keeps walking forward, as if bending under its own weight. This is an issue that the CPG-based gaits previously used on the NimbRo-OP2X did not face, as they are first and foremost meticulously tuned by hand to achieve open-loop walking. Once the estimation of the feedforward terms has been enabled, it takes only a few steps for the robot to completely correct its state. If left undisturbed, it can maintain it without issues. The latency t_l of the system was estimated to be around $55\,\text{ms}$. Although the tracking precision is already quite high, the estimated parameters can be further fine-tuned to achieve higher precision. The small errors can also be left for the feedback controller, which has been significantly relieved in its duties. Through several experiments we observed that the coupled nature of the sagittal and lateral planes necessitates correcting for errors in both simultaneously. This is also true for the coupling between step placement, its timing and angle of attack. All of these factors complement each other and intricately influence the state tracking and visual gait quality, which is not faithfully represented through ablation tests.

TABLE I

STATE ESTIMATION ERROR IN SIMULATION

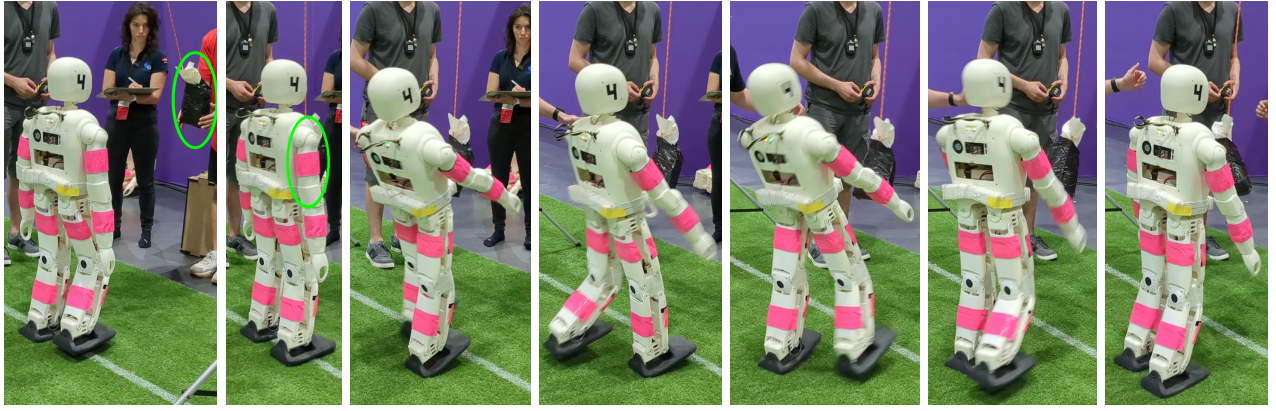|	$e(\ddot{\mathbf{m}}_{c,x})$	$e(\ddot{\mathbf{m}}_{c,y})$	$e(\ddot{\mathbf{m}}_{c,z})$	$e(\dot{\mathbf{m}}_{c,x})$	$e(\dot{\mathbf{m}}_{c,y})$	$e(\dot{\mathbf{m}}_{c,z})$
Purely IMU-based estimation:						
$\mathbb{M}(e)$	-0.5500	-0.4070	0.0767	-0.1146	0.0288	0.0424
$\text{SD}(e)$	2.6092	1.3079	**4.2658**	0.1039	**0.0356**	**0.0829**
$\frac{\Sigma|e|}{T}$	1.8023	0.6752	**1.8965**	0.1208	0.0367	0.0710
Including limb dynamics:						
$\mathbb{M}(e)$	**0.0097**	-0.4010	0.0540	-0.0913	0.0148	0.0286
$\text{SD}(e)$	**2.0921**	1.0489	5.5886	**0.0902**	0.0358	0.0892
$\frac{\Sigma|e|}{T}$	**1.1526**	0.6542	2.0439	**0.1058**	**0.0269**	**0.0671**

Fig. 7. Still frames from the robot performing push recovery at RoboCup. The $5\,\mathrm{kg}$ pendulum is circled in green. The first frame depicts the distance from which it was dropped, while the second one shows the moment of impact. After the impact, the robot undergoes an immediate and significant tilt. It manages to right itself using the developed reactive stepping approach.

C. Balance

The final component that we test is the balancing capability, which utilizes the presented approach in its entirety. The push-recovery capabilities have been presented during the Technical Challenges of RoboCup 2023, which was held in Bordeaux, France. During the challenge the robot has to complete a trial consisting of three consecutive pushes to its CoM, induced by a free-falling pendulum. One from the front, one from the back, and the third from either of the two. Using our approach, NimbRo-OP2X was able to successfully complete several trials with a $3\,\mathrm{kg}$ and $5\,\mathrm{kg}$ pendulum. The strongest pushes that the robot sustained were with a mass of $5\,\mathrm{kg}$ which was let go at a horizontal distance of $90\,\mathrm{cm}$ to the robot. An example of this trial is shown in Fig. 7. The impact quickly starts to noticeably tilt the robot, which is met with an immediate response from the developed controllers. Firstly, the CoM controller quickly responds by shifting the weight of the robot towards the front, along with the arms. Simultaneously, the swing leg gets repositioned in accordance to the tilt and recomputed step position. The robot then continues walking until it fully disperses the push. Overall, this was the best performance in the 2023 push-recovery challenge in the AdultSize class, more than doubling the score of the second team. This result also beats our team's previous performance achieved with the Capture Steps [16] framework, which in the 2019 and 2022 competitions was able to sustain perturbations of the $5\,\mathrm{kg}$ pendulum from a distance of $60\,\mathrm{cm}$ [32].

We also compare our approach to the mentioned Capture Steps framework quantitatively in simulation, as it was known to produce the best push-recovery results up to date with the NimbRo-OP2X robots. Both approaches were tuned to achieve the highest capable performance for the evaluation. The experiments consist of a series of pushes from the back and front, where our interface to MuJoCo applies the force to the center of the torso link for a duration of $200\,\mathrm{ms}$. After 10 pushes from both directions, the force gradually increases. Results of these experiments are summarized in Table II. Both approaches can tolerate pushes of up to $60\,\mathrm{N}$. Above that, Capture Steps can only regain balance when pushed towards the front, without being able to recover a single push of $75\,\mathrm{N}$. This is most likely due to the noise suppression, adaptation and model blending factors, all of which just contribute to a slower response. The framework most likely would perform better for similar impulses, but having smaller forces spread across a longer time period. This is not a limitation in our approach, as the dynamics are accurately estimated and accounted for. This allows to tolerate even some pushes with a force of $85\,\mathrm{N}$. Opposite to Capture Steps, the pushes towards the back are tolerated better, due to the kinematics. As the robot cannot overextend its knee, taking a step back mostly moves the shank, while going forward requires to move the thigh (which is carrying the knee and the shank). At forces of $90\,\mathrm{N}$ neither approach can recover, as the robot tilts and falls to the ground instantly.

VI. CONCLUSIONS

We presented an approach to bipedal locomotion and balancing, specifically targeting platforms constrained by their hardware. The limitations include lack of sensing, imperfect actuation or even kinematic capabilities. We have quantitatively and qualitatively shown, that hardware with such limitations can also achieve relatively high performance. For this, we developed a novel state estimator based on a five-mass model with limb dynamics. The precise estimates were then used to establish a feedforward compensation scheme, operating on the centroidal state as opposed to the usual joint-space schemes. Finally, step feedback was extended to account for tilting due to kinematic limitations.

TABLE II
NUMBER OF WITHSTOOD SIMULATED PUSHES OUT OF 10

Force [N]	from	50	60	70	75	80	85	90
Presented framework	back	10	10	7	4	3	0	0
	front	10	10	9	9	7	6	0
Capture steps [16]	back	10	10	4	0	0	0	0
	front	10	2	0	0	0	0	0

In future work, we would like to investigate a meaningful way to combine both sagittal and lateral movement. The lateral movement in this work was mostly handled by pure CoM-ZMP and gain-based timing control. The reality is that the timing is much more nuanced and we hope to properly integrate it with CAM regulation for a robust gait. We also want to explore possible improvements to the state estimator, which would incorporate joint current (torque) readings for more refined acceleration estimates.

REFERENCES

[1] M. Al Borno, M. De Lasa, and A. Hertzmann, "Trajectory optimization for full-body movements with complex contacts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 8, pp. 1405–1414, 2012.

[2] 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.

[3] S. Kajita, M. Morisawa, K. Miura, S. Nakaoka, K. Harada, K. Kaneko, F. Kanehiro, and K. Yokoi, "Biped walking stabilization based on linear inverted pendulum tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 4489–4496.

[4] D. Dimitrov, A. Sherikov, and P.-B. Wieber, "A sparse model predictive control formulation for walking motion generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 2292–2299.

[5] R. Tedrake, S. Kuindersma, R. Deits, and K. Miura, "A closed-form solution for real-time zmp gait generation and feedback stabilization," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2015, pp. 936–940.

[6] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2006, pp. 200–207.

[7] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4420–4427.

[8] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

[9] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.

[10] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body Motion Planning with Centroidal Dynamics and Full Kinematics," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2014.

[11] Y.-M. Chen, G. Nelson, R. Griffin, M. Posa, and J. Pratt, "Integrable whole-body orientation coordinates for legged robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[12] R. Schuller, G. Mesesan, J. Englsberger, J. Lee, and C. Ott, "Online centroidal angular momentum reference generation and motion optimization for humanoid push recovery," *IEEE Robotics and Automation Letters (RAL)*, vol. 6, no. 3, pp. 5689–5696, 2021.

[13] G. Ficht and S. Behnke, "Direct Centroidal Control for Balanced Humanoid Locomotion," in *Climbing and Walking Robots Conference (CLAWAR)*. Springer, 2023, pp. 242–255.

[14] G. Ficht, H. Farazi, A. Brandenburger, D. Rodriguez, D. Pavlichenko, P. Allgeuer, M. Hosseini, and S. Behnke, "NimbRo-OP2X: Adult-sized open-source 3D printed humanoid robot," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.

[15] P. Allgeuer and S. Behnke, "Omnidirectional bipedal walking with direct fused angle feedback mechanisms," in *16th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, Cancún, Mexico, 2016.

[16] M. Missura, M. Bennewitz, and S. Behnke, "Capture steps: Robust walking for humanoid robots," *International Journal of Humanoid Robotics (IJHR)*, vol. 16, no. 06, p. 1950032, 2019.

[17] M. Schwarz and S. Behnke, "Compliant robot behavior using servo actuator models identified by iterative learning control," in *RoboCup International Symposium*, Eindhoven, Netherlands, 2013.

[18] C. Gehring, S. Coros, M. Hutter, C. D. Bellicoso, H. Heijnen, R. Diethelm, M. Bloesch, P. Fankhauser, J. Hwangbo, M. Hoepflinger *et al.*, "Practice makes perfect: An optimization-based approach to controlling agile motions for a quadruped robot," *IEEE Robotics & Automation Magazine*, vol. 23, no. 1, pp. 34–43, 2016.

[19] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[20] P. M. Wensing, A. Wang, S. Seok, D. Otten, J. Lang, and S. Kim, "Proprioceptive actuator design in the MIT Cheetah: Impact mitigation and high-bandwidth physical interaction for dynamic legged robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 509–522, 2017.

[21] M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, "The MIT humanoid robot: Design, motion planning, and control for acrobatic behaviors," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2021, pp. 1–8.

[22] 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, no. 21-22, pp. 1380–1389, 2020.

[23] P. Allgeuer and S. Behnke, "Fused Angles: A representation of body orientation for balance," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 366–373.

[24] G. Ficht and S. Behnke, "Fast whole-body motion control of humanoid robots with inertia constraints," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 6597–6603.

[25] P. Allgeuer and S. Behnke, "Robust sensor fusion for biped robot attitude estimation," in *Proceedings of 14th IEEE-RAS Int. Conference on Humanoid Robotics (Humanoids)*, Madrid, Spain, 2014.

[26] S. A. Ludwig and K. D. Burnham, "Comparison of Euler estimate using extended Kalman Filter, Madgwick and Mahony on quadcopter flight data," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 1236–1241.

[27] R. Cisneros, A. Benallegue, Y. Chitour, M. Morisawa, F. Kanehiro *et al.*, "Lyapunov-stable orientation estimator for humanoid robots," *IEEE Robotics and Automation Letters (RAL)*, vol. 5, no. 4, pp. 6371–6378, 2020.

[28] H. Herr and M. Popovic, "Angular momentum in human walking," *Journal of Experimental Biology*, vol. 211, no. 4, pp. 467–481, 2008.

[29] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3D linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001, pp. 239–246.

[30] S. Kajita, "Linear inverted pendulum-based gait," *Humanoid Robotics: A Reference*, pp. 905–922, 2019.

[31] Y. Choi, D. Kim, Y. Oh, and B.-J. You, "Posture/walking control for humanoid robot based on kinematic resolution of com jacobian with embedded motion," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1285–1293, 2007.

[32] D. Pavlichenko, G. Ficht, A. Amini, M. Hosseini, R. Memmesheimer, A. Villar-Corrales, S. M. Schulz, M. Missura, M. Bennewitz, and S. Behnke, "RoboCup 2022 AdultSize winner NimbRo: Upgraded perception, capture steps gait and phase-based in-walk kicks," in *Robot World Cup*. Springer, 2022, pp. 240–252.