

Partial Yaw Moment Compensation Through Whole-Body Motion

Rafael Cisneros¹, Eiichi Yoshida² and Kazuhito Yokoi²

Abstract—This paper describes a generalized approach for compensating just the required yaw moment of a humanoid robot about the Zero Moment Point (ZMP) while performing an arbitrary motion, in order to prevent unwanted / unexpected yaw rotations. This is done by modifying the motion of any set of joints with low priority tasks that can be arbitrarily selected before-hand. Finally, some simulation results are provided, which intend to show the validity of this approach.

I. INTRODUCTION

Humanoid robots' most important characteristic is their bipedal locomotion. However, in order for it to be efficiently performed, not only the generation of dynamically stable movements is important but also the avoidance of unexpected motion due to slippage, as the performance of these ones strongly rely on the friction between the feet and the floor. For example, walking on a curved path or performing a kicking motion can cause unexpected yaw rotation, due to the inherent generation of yaw moment.

In order to compensate this yaw moment several attempts have been made, which mainly use the upper body motion; that is, the joints that are not directly related with the locomotion task, which is a high priority one in most of the cases. In this way, some researchers have proposed to use the motion of the arms only, simultaneously rotated through opposite directions [1] [2] [3]. However, if the humanoid robot is expected to use them for manipulation, then this strategy results impractical [4]. Some others have proposed to rotate the trunk of the robot to achieve this compensation, by changing the orientation of the waist (which may compromise the locomotion pattern) or by rotating the trunk independently if it is provided with the corresponding yaw-axis joint [5] [6] [4]. However, this strategy may influence the inertial sensory system, and if the humanoid robot's head is supposed to maintain its gaze then the neck must be provided with an additional joint to compensate the trunk's rotation.

That is, depending on the tasks aside from the locomotion that the robot has to accomplish, to use one set of joints or another is preferable according to the priority of their motions. All the joints (or some of them) with a low priority motion may be chosen to compensate the corresponding yaw moment.

¹R. Cisneros is with the Graduate School of Systems and Information Engineering, University of Tsukuba, 305-0006 Tsukuba, Japan and with the CNRS-AIST JRL (Joint Robotics Laboratory), UMI 3218/CRT, 305-8568 Tsukuba, Japan, rafael.cisneros at aist.go.jp

²E. Yoshida and K. Yokoi are with the CNRS-AIST JRL (Joint Robotics Laboratory), UMI 3218/CRT, 305-8568 Tsukuba, Japan and the Graduate School of Systems and Information Engineering, University of Tsukuba, 305-0006 Tsukuba, Japan, e.yoshida at aist.go.jp and kazuhito.yokoi at aist.go.jp

We have previously proposed a compensation approach attained via the control of the angular momentum with respect to the moving ZMP [7], by using the concept of Resolved Momentum Control [8] to relate this angular momentum with the motion of the joints. This approach was able to fully compensate (at least theoretically) the yaw moment by means of controlling the rate of change of the angular momentum with the corresponding acceleration of the selected joints. However, this generally led to an unbounded joint motion, unfeasible due to the presence of joint limits.

In this paper we propose a novel control method for whole-body yaw moment compensation that does not cause undesired residual motion. Instead of taking the net yaw moment to zero we establish a threshold value (kept below the frictional yaw moment given between the floor and the soles of the robot to prevent unwanted / unexpected yaw rotations), so that only the yaw moment that exceeds this value is compensated by means of the acceleration of the selected joints. Then, once the yaw moment has been taken to the desired limit range the joints can slow down at a lower rate, fulfilling the net yaw moment requirement and, at the same time, keeping the joint values bounded and within the limits without residual drift, and this is our main contribution in contrast with our previous work.

II. PROBLEM STATEMENT

Let us consider a humanoid robot which has to perform a high priority task. This task may be the locomotion itself and/or any other motion, not restricted to the lower part of the robot. For example, the robot may be required to grasp an object by using one hand (or two). In addition, it is necessary for the robot to keep its balance. This can be achieved by controlling the position of the Zero Moment Point (ZMP) to remain inside of the support polygon of the robot, maybe by moving its waist horizontally as described in [9]. In this way, several joints of the robot will be given a required high priority motion, while the remaining ones may be given constant values, or any low priority motion.

Then, once all these motions are defined (or proposed) and specifically their velocity profiles, the total linear and angular momenta of the robot with respect to some reference point (as it is the origin of the world frame 0), P_0 and L_0 respectively, will also be inherently given.

Let us denote by τ_p the net moment of the vertical ground reaction force f_p about the ZMP, whose position is denoted by $r_{p/0}$ as it is taken from the origin of the world frame, 0 (see Fig. 1). The moment of f_p about the origin of the world coordinate frame, τ_0 , can be calculated as

$$\tau_0 = r_{p/0} \times f_p + \tau_p. \quad (1)$$

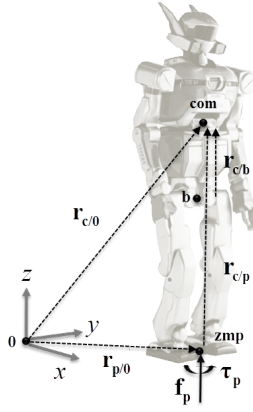


Fig. 1. Definition of variables [7]

In addition, we know that the following relationships hold:

$$\dot{P}_0 = \tilde{m}g + f_p, \quad (2)$$

$$\dot{L}_0 = r_{c/0} \times \tilde{m}g + \tau_0, \quad (3)$$

where $g = [0 \ 0 \ -g]^T$ is the gravity vector and g is the acceleration due to gravity, whereas \tilde{m} stands for the total mass of the robot and $r_{c/0}$, for the position of the center of mass (CoM) of the robot with respect to the origin of the world coordinate frame. By substituting (1) and (2) into (3) and solving with respect to τ_p , we get [10]

$$\tau_p = \dot{L}_0 - r_{c/0} \times \tilde{m}g - r_{p/0} \times (\dot{P}_0 - \tilde{m}g). \quad (4)$$

By definition of the ZMP the x and y components of τ_p are zero, while the vertical one (z) is calculated as [7]

$$\tau_{p,z} = \dot{L}_{0,z} - r_{p/0,x} \dot{P}_{0,y} + r_{p/0,y} \dot{P}_{0,x}; \quad (5)$$

that is, the vertical component of τ_p depends on the derivative (the rate of change) of the linear and angular momenta. This moment may be compensated with the one created by the horizontal frictional forces between the robot's feet and the ground if the static frictional moment is not exceeded. Otherwise, slippage will occur and the robot will suffer an undesired yaw rotation.

It is our objective to avoid this yaw rotation by means of yaw moment compensation, modifying \dot{P}_0 and \dot{L}_0 with an appropriate adjustment of the motion of the joints related to low priority tasks, referred from now on as “free” joints (as their value is not completely determined), θ_f . On the other hand, the joints related with high priority tasks will be referred as “high priority” joints, θ_h .

However, as the linear and angular momenta are directly related to the joint velocities, modifying their derivative implies the acceleration of the “free” joints. This fact has an inherent implication: unless the overall motion of the robot is completely symmetrical (in terms of joint displacement and time), there will be a residual joint velocity causing the “free” joints to continuously increment/decrement their value, overpassing the joint limits of the robot. Therefore, the robot should be provided with joints capable of continuous rotation for this approach to be feasible.

Then, instead of fully compensating the yaw moment as proposed in [7], we newly introduce a partial compensation scheme, in order to maintain it below a threshold value given by the static frictional moment, in such a way that we can still prevent the undesired yaw rotation. In this way, we can eliminate the drift in the motion caused by the joint accelerations by slowing them down at a lower rate, as long as the small moment caused by this process meets the requirement imposed by the threshold.

III. MOMENTUM EQUATIONS

Let us represent the humanoid robot as a mechanism whose root is a free-flying base link (its waist), with reference point b , whose linear and angular velocities are given by v_b and ω_b . In addition, we define a joint velocity vector $\dot{\theta} \in \mathbb{R}^n$, where n is the total number of joints. The linear and angular momenta about the CoM are given by

$$P_c = \tilde{m}v_b + \tilde{m}\hat{r}_{c/b}^T \omega_b + M_c \dot{\theta}, \quad (6)$$

$$L_c = \tilde{I}_c \omega_b + H_c \dot{\theta}, \quad (7)$$

where $\hat{\cdot} : \mathbb{R}^3 \rightarrow \mathbb{R}^{3 \times 3}$ maps a vector into a skew-symmetric matrix, $\hat{r}_{c/b}$ is a vector defined from b to the CoM of the robot and \tilde{I}_c stands for the total inertia matrix of the robot with respect to the CoM. The matrices $M_c, H_c \in \mathbb{R}^{3 \times n}$ can be regarded as inertia matrices which indicate how the joint speeds affect the linear and angular momenta, respectively, as defined by Kajita et al [8].

Let us calculate the angular momentum for the robot with respect to the ZMP by using the expressions given in (6) and (7), in addition with $r_{c/p}$, a vector defined from the ZMP to the CoM of the robot [7]:

$$\begin{aligned} L_p &= r_{c/p} \times P_c + L_c \\ &= \tilde{m}\hat{r}_{c/p} v_b + \left(\tilde{m}\hat{r}_{c/p}\hat{r}_{c/b} + \tilde{I}_c \right) \omega_b \\ &\quad + \left(\hat{r}_{c/p} M_c + H_c \right) \dot{\theta}. \end{aligned} \quad (8)$$

By defining

$$H_p = \hat{r}_{c/p} M_c + H_c, \quad (9)$$

we can express (8) as

$$L_p = (\tilde{m}\hat{r}_{c/p}) v_b + \left(\tilde{m}\hat{r}_{c/p}\hat{r}_{c/b} + \tilde{I}_c \right) \omega_b + H_p \dot{\theta}. \quad (10)$$

A. Inertia Matrices

As seen in (6) and (10), the role of the inertia matrices M_c and H_p is of great importance as they dictate how the linear and angular momenta change according to the motion of the joints. The calculation of these ones is explained in detail in [8], but can be summarized as follows:

Let us assume that the humanoid robot with n joints has a tree structure whose root corresponds to its base link and that all of its joints are rotational. The joint j directly drives the link j . The position of joint j with respect to the origin of the world coordinate frame can be described by $r_{(j)/0}$, while its axis orientation by the unit vector a_j and its joint variable by θ_j .

The link j has a mass distribution characterized by the mass m_j , the position of its center of mass described by $\mathbf{r}_{c(j)/0}$ and the tensor of inertia with respect to this point (and described in the world frame 0) $\mathbf{I}_{c,j}$. Additionally, the joint j also drives all the link structure connected to the link j . Let us denote as \tilde{m}_j , $\tilde{\mathbf{r}}_{c(j)/0}$ and $\tilde{\mathbf{I}}_{c,j}$ the mass, the position of the center of mass and the inertia tensor of all this link structure driven by the joint j . See Fig. 2.

Then, the *additional* linear and angular momenta with respect to the origin of the world coordinate frame for the robot which are yielded by $\dot{\theta}_j$ ($\mathbf{P}_{j,0}$ and $\mathbf{L}_{j,0}$ respectively) can be calculated as

$$\mathbf{P}_{j,0} = \tilde{m}_j \mathbf{a}_j \dot{\theta}_j \times (\tilde{\mathbf{r}}_{c(j)/0} - \mathbf{r}_{c(j)/0}) = \mathbf{M}_{j,0} \dot{\theta}_j, \quad (11)$$

$$\mathbf{L}_{j,0} = (\tilde{\mathbf{r}}_{c(j)/0} \times \mathbf{M}_{j,0} + \tilde{\mathbf{I}}_{c,j} \mathbf{a}_j) \dot{\theta}_j = \mathbf{H}_{j,0} \dot{\theta}_j. \quad (12)$$

Such that,

$$\mathbf{M}_0 = \begin{bmatrix} \mathbf{M}_{1,0} & \mathbf{M}_{2,0} & \cdots & \mathbf{M}_{n,0} \end{bmatrix}, \quad (13)$$

$$\mathbf{H}_0 = \begin{bmatrix} \mathbf{H}_{1,0} & \mathbf{H}_{2,0} & \cdots & \mathbf{H}_{n,0} \end{bmatrix}, \quad (14)$$

$$\mathbf{H}_c = \mathbf{H}_0 - \hat{\mathbf{r}}_{c/0} \mathbf{M}_0. \quad (15)$$

Finally, let us consider (9) and substitute (15) into it in order to get a final expression for \mathbf{H}_p [7]:

$$\begin{aligned} \mathbf{H}_p &= \hat{\mathbf{r}}_{c/p} \mathbf{M}_c + (\mathbf{H}_0 - \hat{\mathbf{r}}_{c/0} \mathbf{M}_0) \\ &= \mathbf{H}_0 - \hat{\mathbf{r}}_{p/0} \mathbf{M}_0, \end{aligned} \quad (16)$$

given that $\mathbf{M}_0 = \mathbf{M}_c$ as $\mathbf{P}_{j,0} = \mathbf{P}_{j,c}$.

IV. PARTIAL YAW MOMENT COMPENSATION

Once an arbitrary motion is defined for a humanoid robot, there will be a resulting moment (produced by the uncompensated motion) whose vertical component, the *resulting* yaw moment $\tau_{p,z}^r$, may exceed the static frictional moment between the robot's feet and the ground, $\tau_{p,z}^{fr}$. Let us define a threshold value $\tau_{p,z}^{th}$ such that $\tau_{p,z}^r < \tau_{p,z}^{fr}$, and generate a *desired* yaw moment (the expected result of the partial compensation), $\tau_{p,z}^*$, by means of saturating (or clipping) the signal $\tau_{p,z}^r$; that is,

$$\tau_{p,z}^* = \begin{cases} \tau_{p,z}^{th} & \text{if } \tau_{p,z}^r > \tau_{p,z}^{th} \\ \tau_{p,z}^r & \text{if } -\tau_{p,z}^{th} \leq \tau_{p,z}^r \leq \tau_{p,z}^{th} \\ -\tau_{p,z}^{th} & \text{if } \tau_{p,z}^r < -\tau_{p,z}^{th} \end{cases}, \quad (17)$$

while $\tau_{p,x}^* = \tau_{p,x}^r$ and $\tau_{p,y}^* = \tau_{p,y}^r$.

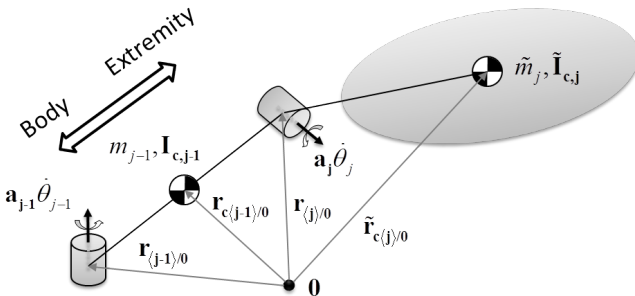


Fig. 2. Link configuration of the robot

In addition, let us consider the relationship between the angular momentum with respect to the ZMP and the one with respect to the origin of the world coordinate:

$$\mathbf{L}_0 = \mathbf{r}_{p/0} \times \mathbf{P}_0 + \mathbf{L}_p. \quad (18)$$

By differentiating (18) with respect to time we get

$$\dot{\mathbf{L}}_0 = \mathbf{r}_{p/0} \times \dot{\mathbf{P}}_0 + \dot{\mathbf{r}}_{p/0} \times \mathbf{P}_0 + \dot{\mathbf{L}}_p, \quad (19)$$

which can be substituted into (4) to get

$$\boldsymbol{\tau}_p = \dot{\mathbf{L}}_p + \dot{\mathbf{r}}_{p/0} \times \mathbf{P}_0 + \mathbf{r}_{p/c} \times \tilde{m} \mathbf{g}. \quad (20)$$

Let us write the last equation for both, the resulting and the desired yaw moment (without and with partial compensation), τ_p^r and τ_p^* , such that

$$\tau_p^r = \dot{\mathbf{L}}_p^r + \dot{\mathbf{r}}_{p/0} \times \mathbf{P}_0^r + \mathbf{r}_{p/c} \times \tilde{m} \mathbf{g}, \quad (21)$$

$$\tau_p^* = \dot{\mathbf{L}}_p^* + \dot{\mathbf{r}}_{p/0} \times \mathbf{P}_0^* + \mathbf{r}_{p/c} \times \tilde{m} \mathbf{g}. \quad (22)$$

This means that clipping the yaw moment requires the modification of both, the derivative of the resulting angular momentum and the resulting linear one (not its derivative), $\dot{\mathbf{L}}_p^r$ and \mathbf{P}_0^r , such that (22) holds. The modified signals are marked as $\dot{\mathbf{L}}_p^*$ and \mathbf{P}_0^* . Notice that the dependance on the derivative of the linear momentum was suppressed by considering the ZMP as a point of reference. Instead, the expression now depends on the velocity of the desired ZMP.

In order to calculate the amount of modification let us subtract (21) from (22) to have

$$\begin{aligned} \tau_p^* - \tau_p^r &= \dot{\mathbf{L}}_p^* - \dot{\mathbf{L}}_p^r + \dot{\mathbf{r}}_{p/0} \times (\mathbf{P}_0^* - \mathbf{P}_0^r) \\ \Delta \tau_p &= \Delta \dot{\mathbf{L}}_p + \dot{\mathbf{r}}_{p/0} \times \Delta \mathbf{P}_0. \end{aligned} \quad (23)$$

Intuitively, the compensation should be done mainly through the manipulation of the derivative of the angular momentum, by modifying the motion of the joints. However, this also modifies the linear momentum, but slightly as the center of mass of the robot is supposed to follow a path as a result of the predefined motion. Remember that $\mathbf{P}_0^r = \tilde{m} \dot{\mathbf{r}}_{c/0}$ also. This means that, in practical terms, $\mathbf{P}_0^* \approx \mathbf{P}_0^r$, such that

$$\Delta \dot{\mathbf{L}}_p \approx \Delta \tau_p. \quad (24)$$

Now, let us consider (10) and break down the joint speed vector into two parts: one composed by the n_h "high priority" joints, $\dot{\boldsymbol{\theta}}_h \in \mathbb{R}^{n_h}$, and another one composed by the n_f "free" joints, $\dot{\boldsymbol{\theta}}_f \in \mathbb{R}^{n_f}$ ($n_f = n - n_h$). Then,

$$\mathbf{H}_p \dot{\boldsymbol{\theta}} = \mathbf{H}_{h,p} \dot{\boldsymbol{\theta}}_h + \mathbf{H}_{f,p} \dot{\boldsymbol{\theta}}_f, \quad (25)$$

where $\mathbf{H}_{h,p} \in \mathbb{R}^{3 \times n_h}$ and $\mathbf{H}_{f,p} \in \mathbb{R}^{3 \times n_f}$. Furthermore, we can consider that $\dot{\boldsymbol{\theta}}_f = \dot{\boldsymbol{\theta}}_f^{ref} + \Delta \dot{\boldsymbol{\theta}}_f$; that is, a predefined motion according to a low priority task plus a correction term that will partially compensate the yaw moment. Therefore (10) can be written as

$$\begin{aligned} \mathbf{H}_{f,p} \Delta \dot{\boldsymbol{\theta}}_f &= \mathbf{L}_p^* - (\tilde{m} \hat{\mathbf{r}}_{c/p}) v_b - \left(\tilde{m} \hat{\mathbf{r}}_{c/p} \hat{\mathbf{r}}_{c/b} + \tilde{\mathbf{I}}_c \right) \boldsymbol{\omega}_b \\ &\quad - \mathbf{H}_{h,p} \dot{\boldsymbol{\theta}}_h - \mathbf{H}_{f,p} \dot{\boldsymbol{\theta}}_f^{ref}, \end{aligned} \quad (26)$$

where we have used \mathbf{L}_p^* , the desired moment.

Identifying that the rest of the terms at the right calculate the momentum resulting from the predefined motion L_p^r (by using $\dot{\theta}_d$ and $\dot{\theta}_f^{ref}$), then we can write (26) as

$$H_{f,p} \Delta \dot{\theta}_f = L_p^* - L_p^r = \Delta L_p, \quad (27)$$

by using the term calculated from the numerical integration of (24). Then, it is possible to find a solution for $\Delta \dot{\theta}_f$ in (27) by using the *least squares* method; that is,

$$\Delta \dot{\theta}_f = H_{f,p}^\dagger \Delta L_p, \quad (28)$$

where $H_{f,p}^\dagger$ is the *Moore-Penrose pseudo-inverse* of $H_{f,p}$.

In fact, taking into account that the only non-zero component of ΔL_p is the vertical one, $\Delta L_{p,z}$, we can consider to use only the third row of $H_{f,p}$, denoted by $H_{f,p,z} \in \mathbb{R}^{1 \times n_f}$, so that its pseudo-inverse be a column vector that distributes the “compensation factor” $\Delta L_{p,z}$ to the respective joints according to their influence in the overall momentum. Then, (28) may be written as

$$\Delta \dot{\theta}_f = H_{f,p,z}^\dagger \Delta L_{p,z}. \quad (29)$$

This last equation can be numerically integrated to get $\Delta \theta_f$, the term that added to θ_f^{ref} gives us the modified trajectories for the “free” joints. See Fig. 3.

It is worth considering that as (24) is just an approximation, there will be some compensation error, specifically when the ZMP moves with higher velocity; that is, when $\dot{r}_{p/0}$ is significant. This causes some overshoot over our established threshold. However, if we run this process several times by taking as the input $\tau_{p,z}^r$ the moment generated as a result of the compensation in the previous iteration, we can reduce this overshoot until it becomes sufficiently small. That is, for every iteration the overshoot will become the input for the calculation of the compensation term $\Delta L_{p,z}$, proportional to $\Delta \dot{\theta}_f$, so that if the magnitude of the overshoot decreases, the compensation for the joints also does.

V. PRELIMINARY RESULTS:

NUMERICAL COMPUTATION FOR A KICKING MOTION

Let us evaluate the performance of our approach until this point by introducing the case study of a humanoid robot that is required to approach a ball by following a curved trajectory and then perform a kicking motion on it.

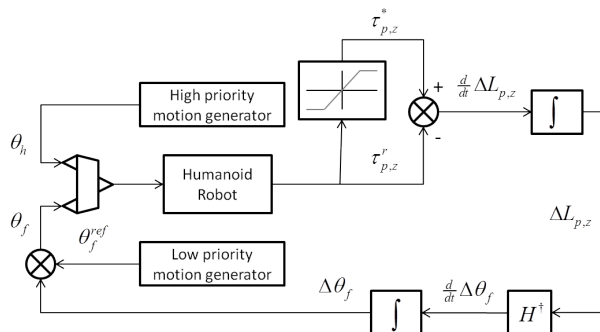


Fig. 3. Preliminary partial yaw moment compensation algorithm

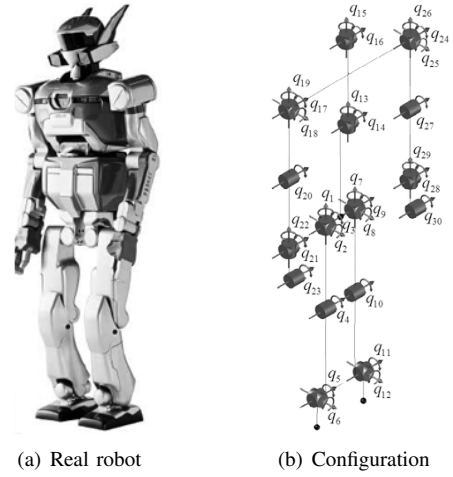


Fig. 4. HRP-2 Humanoid Robot

For this case study we will consider the HRP-2 humanoid robot [11]. This humanoid is 1.54 m height and weights 58 kg (Fig. 4(a)). It possesses 30 degrees of freedom (dof): 6 at each leg, 6 at each arm, 1 at each hand, 2 at the waist and 2 at the neck, arranged according to the kinematic structure shown in Fig. 4(b).

The proposed high priority tasks for this robot is described as follows: starting from the position $(-1, 1)$ m the robot has to approach a ball positioned at $(0, 0)$ m by following the footsteps specified in Fig. 5, which also includes the kicking phase. The desired trajectory for the ZMP is also shown in Fig. 5, as well as the one for the projection of the waist position that stabilizes the robot. Both, the kicking motion and the stabilization process, are calculated as explained in [9]. Also, as a part of the high priority tasks it was decided for the robot to use the yaw-axis joint of its neck to look towards to the position of the ball.

This means that the joints corresponding to the trunk (chest) and the arms can be given any low priority motion, as they are not related with the high priority tasks. For example, their value can be initially proposed to remain constant.

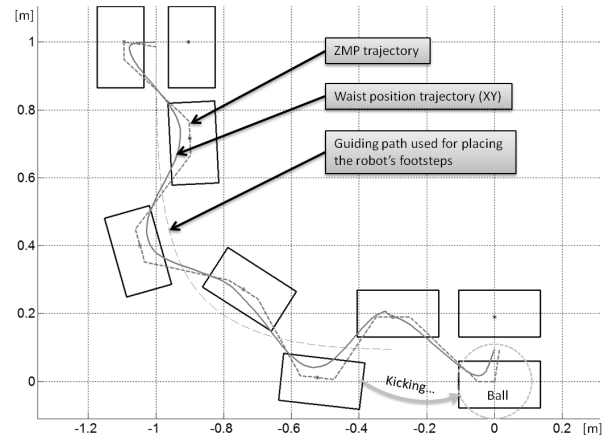


Fig. 5. Planned motion for the robot showing its footsteps

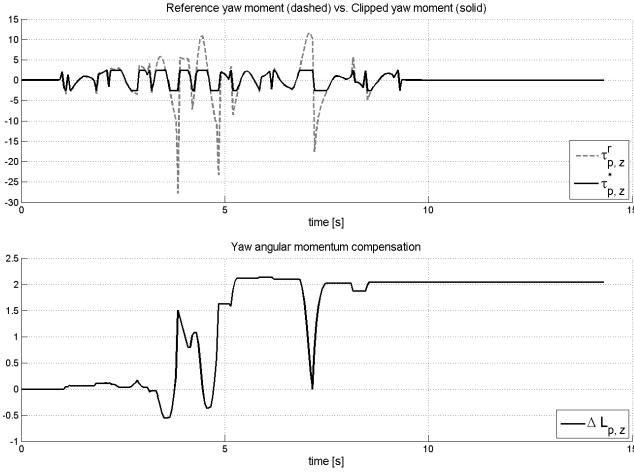


Fig. 6. $\tau_{p,z}^r$, $\tau_{p,z}^*$, $\Delta L_{p,z}$ at 1st iteration for the preliminar compensation

Let us perform the corresponding numerical computation on Matlab by implementing the controller shown in Fig. 3 and taking into account the humanoid dynamics. Also, let us establish a threshold value of $\tau_p^* = 2.5$ Nm. The yaw moment resulting from the prescribed motion ($\tau_{p,z}^r$) as well as the clipped one ($\tau_{p,z}^*$) during the first iteration are shown in the upper graph of Fig. 6. The yaw component of the angular momentum required for compensation ($\Delta L_{p,z}$) is shown in the lower graph. As can be seen, every time that a compensation is required, $\tau_{p,z}^r \neq \tau_{p,z}^*$, $\Delta L_{p,z}$ is modified and remains constant when that is not the case. However, when the latter happens the constant values are non-zero, and because $L_{p,z}$ is directly related with the velocity of the joints, these ones increment their value continuously. As an example, the final motion for the chest joints (computed after 8 iterations) is shown in Fig. 7.

This wouldn't matter so much if it were not for the joint limits. For example, for the yaw-rotation joint on the chest the joint limit is ± 0.7854 rad, which is clearly exceeded as it reaches more than 6 rad at the end of the simulation (and continues growing).

VI. DRIFT COMPENSATION

It would be desirable to modify $\Delta L_{p,z}$ so that it preserve the high frequency changes required for the compensation and, at the same time, asymptotically approach to zero (as it represents the joint velocities); that is, to have a final value of 0 for $\Delta L_{p,z}$ after the compensation.

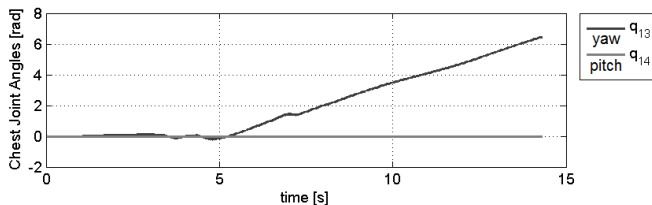


Fig. 7. Computed chest joint trajectories for the preliminar compensation

One way to do that is to subtract from this signal a drift compensation factor proportional to the accumulated “error” of the compensation (the area under its curve): an integral compensation term. Let us propose a control signal $L_{p,z}^{ctrl}$ calculated by

$$L_{p,z}^{ctrl}(t) = k_p \Delta L_{p,z}(t) - k_i \int_{t_0}^{t-\Delta t} L_{p,z}^{ctrl}(\tau) d\tau, \quad (30)$$

where k_p a proportional gain, k_i is the integral one, t_0 is the initial time, t is the current time and Δt is a delay required by its implementation. Such that (29) shall be rewritten as

$$\Delta \dot{\theta}_f = H_{f,p,z}^{\dagger} L_{p,z}^{ctrl}. \quad (31)$$

By doing this and properly tune the gain k_i we should achieve a final zero velocity compensation, but we are not sure that the net displacement of the compensation angle for joints will be zero; that is, if for the “free” joints an initial constant value is proposed, we would like that at the end of the motion the joint angles would return to their initial values. This can be done by modifying (30) as

$$L_{p,z}^{ctrl}(t) = k_p \Delta L_{p,z}(t) - k_i \int_{t_0}^{t-\Delta t} L_{p,z}^{ctrl}(\tau) d\tau - k_{ii} \int_{t_0}^{t-\Delta t} \int_{t_0}^{\tau} L_{p,z}^{ctrl}(\tau) d^2\tau, \quad (32)$$

where k_{ii} is the double integral gain. The reasoning behind this choice is explained as follows: if the angular moment is directly related to the joint velocities, its integral is directly related the joint displacements caused by the compensation, which we want to make zero. This can be done by subtracting again its accumulated “error”, represented by the second integral.

The final version of the algorithm including the drift compensation can be depicted as shown in Fig. 8.

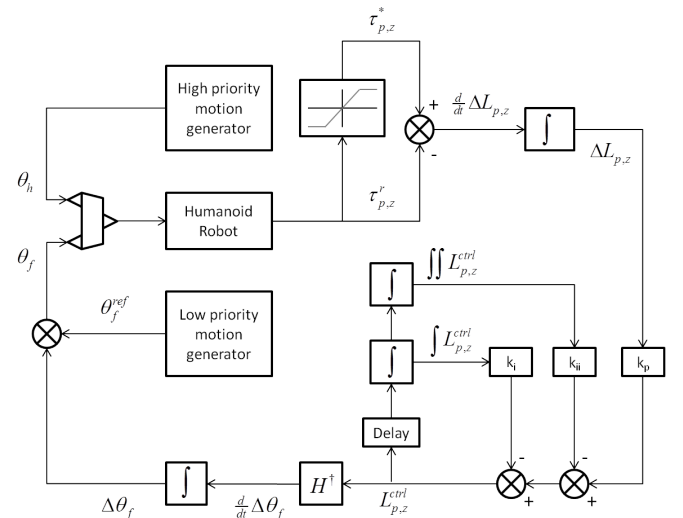


Fig. 8. Partial yaw moment compensation algorithm without drift

VII. NUMERICAL COMPUTATION WITH DRIFT COMPENSATION

Let us revisit the case study considered in Section V by including the drift compensation just discussed. First, let us show the effect of considering only the integral gain k_i by comparing it with $\Delta L_{p,z}$. These values are shown in Fig. 9 with a dashed line and a solid gray one, respectively, where the values $k_p = 1$ and $k_i = 3$ were used.

As can be seen, the high frequency variations are preserved while the signal asymptotically approaches to zero, as it was desired. This control leads to the chest joint trajectories shown in Fig. 10, which are effectively bounded (and within the joint limits); that is, they finish with zero joint velocity. However, the yaw-axis joint finished its motion with a different value with respect to the proposed constant one (the initial one).

Next, let us show the effect of considering both gains, k_i and k_{ii} . The corresponding signal is also shown in Fig. 9 with a solid black line and compared to the previous ones, where the values $k_p = 1$, $k_i = 3$ and $k_{ii} = 4$ were used.

As can be seen, both drift compensation schemes take the control signal asymptotically to zero and are very similar, being the main difference that the net area between the black solid curve and the time axis is zero. This final version of the control leads to the joint trajectories shown in Fig. 11, which are not only bounded but all of them finish their motion with the same initial value; that is, with a zero net displacement for the joints with respect to the proposed constant value.

Finally, we show that the yaw moment achieved by means of this version of the algorithm (after 8 iterations) could be effectively maintained below 2.5 Nm, as shown in Fig. 14 where it is compared to the one calculated when no compensation is performed and to the one calculated when there is no drift compensation.

VIII. SIMULATION RESULTS

In order to assess the validity of our approach the full algorithm described in this paper was implemented as a closed-loop compensator for the virtual model of the HRP-2 on the dynamic integrated simulation platform OpenHRP3 [12], by taking as inputs the readings of the force/moment sensors placed at its feet, calculating the current yaw moment, filtering it with the aid of a first order digital filter and generating the appropriate joint motion compensation.

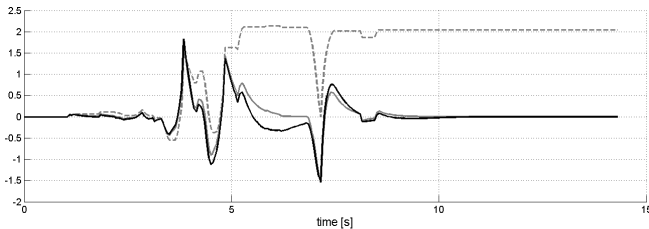


Fig. 9. Comparison between the original ΔL_p (dashed), the control signal $L_{p,z}$ considering only k_i (solid gray) and the one considering both k_i and k_{ii} (solid black), during the first iteration for the final algorithm

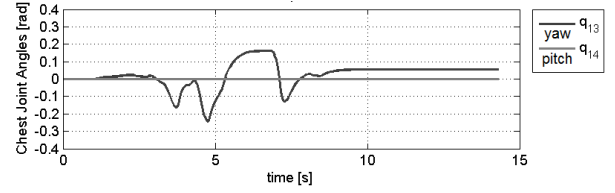


Fig. 10. Computed chest joint trajectories by using the compensation that considers only the gain k_i

The snapshots shown in Fig. 12 depict the behavior of the robot while performing the task described in Section V on a very slippery floor (with coefficient of static friction $\mu_s = 0.075$) without applying the algorithm presented on this paper. As can be seen, the robot suffers unplanned yaw rotations which result on its fall.

On the other hand, the snapshots shown in Fig. 13 depict the robot's behavior under the same low friction circumstances when applying our algorithm by using the following parameters: $\tau_{p,z}^{th} = 1$ Nm, $k_p = 4$, $k_i = 4$ and $k_{ii} = 3$. For this case, the robot is not only able to execute the task without falling but it also exhibits a low yaw rotation.

The yaw moment measured by the force/moment sensors during the simulation (and filtered because of the noise), when no compensation is performed and when it is, are shown in Fig. 15. For this case a coefficient of static friction $\mu_s = 0.4$ was considered to prevent the robot from falling in any case and get a clear comparison. This time $\tau_{p,z}^{th} = 2$ Nm, a threshold that is almost maintained as desired, except for some overshoot.

IX. CONCLUSIONS

In this paper we presented a method to partially compensate the yaw moment for a humanoid robot as it is not possible to fully compensate it for an *arbitrary* motion without experiencing an unwanted drift of the joints.

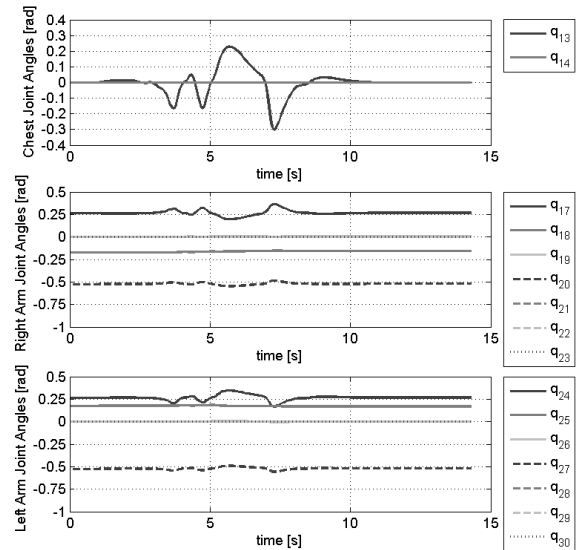


Fig. 11. Computed "Free" joint trajectories by using the compensation that considers both gains k_i and k_{ii}

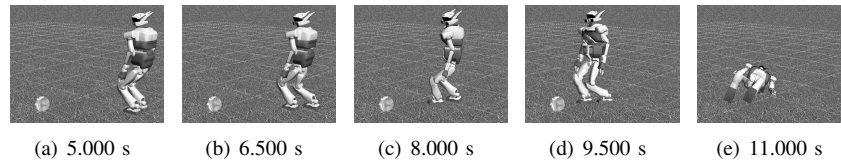


Fig. 12. Kicking motion without yaw moment compensation

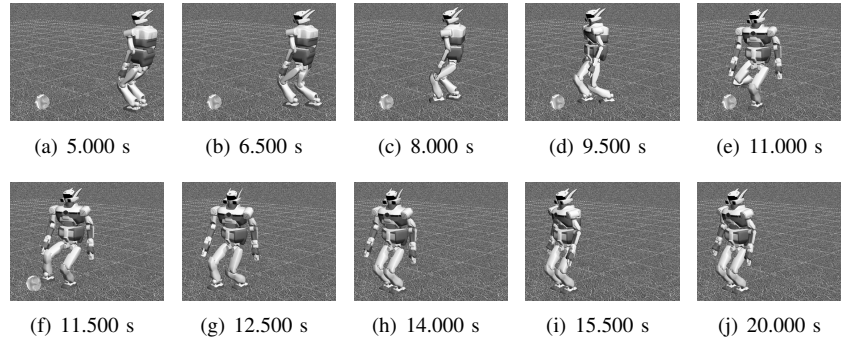


Fig. 13. Kicking motion with yaw moment compensation

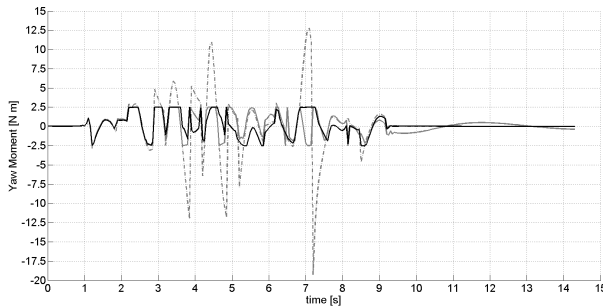


Fig. 14. Numerical computation of the yaw moment: without compensation (dashed) vs yaw moment compensation with drift (solid gray) vs yaw moment and drift compensation (solid black)

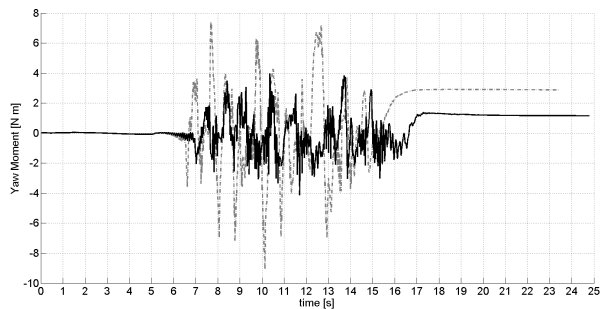


Fig. 15. Simulation results: yaw moment without compensation (dashed) vs with compensation and no drift (solid)

This is because any drift compensation generates a moment in the opposite direction of the yaw compensation motion. However, if its magnitude is lower than the frictional moment between the feet and the ground there is no practical consequence. Remember that some friction is unavoidable.

This approach let us select any arbitrary set of joints, depending on alternate tasks required by the robot. This is a clear advantage over the classical methods that work on a specific set of joints for the compensation.

We have demonstrated in our analysis and simulation the effectivity of the method, whose practical limitation relies on the accuracy of the sensors and the processing speed of the controller. As a future work we would like to improve the performance of the compensator, and test it on the real robot. Also, instead of pre-selecting the “free” joints, we would like to achieve the compensation by using the null space of the high priority motion as a generalization of our work, as well as discuss the stability of the controller in order to appropriately choose the value for the corresponding gains.

REFERENCES

- [1] J. Yang et al. Walking pattern generation for humanoid robot considering upper body motion. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2006.
- [2] D.P. Xing and J.B. Su. Arm / trunk motion generation for humanoid robot. *Science China: Information Services*, 2010.
- [3] S. Zhang et al. The mechanism of yaw torque compensation in the human and motion design for humanoid robots. *INTECH International Journal of Advanced Robotics Systems*, 2013.
- [4] B. Ugurlu et al. Yaw moment compensation for bipedal robots via intrinsic angular momentum constraint. *International Journal of Humanoid Robotics*, 2012.
- [5] J. Ueda et al. Momentum compensation for the fast dynamic walk of humanoids based on the pelvic rotation of contact sport athletes. *INTECH Humanoid Robots, Human-like Machines*, 2007.
- [6] S. Nakaoka et al. Task model of lower body motion for a biped humanoid robot to imitate human dances. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2005.
- [7] R. Cisneros, K. Yokoi, and E. Yoshida. Yaw moment compensation by using full body motion. In *IEEE/RAS Int. Conf. on Mechatronics and Automation*, 2014.
- [8] S. Kajita et al. Resolved momentum control: Humanoid motion planning based on the linear and angular momentum. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.
- [9] R. Cisneros, K. Yokoi, and E. Yoshida. Impulsive pedipulation of a spherical object for reaching a 3d goal position. In *IEEE/RAS Int. Conf. on Humanoid Robots*, 2013.
- [10] S. Kajita et al. *Humanoid Robot*. Ohmsha Ltd., 2005. In Japanese.
- [11] K. Kaneko et al. Humanoid robot HRP-2. In *IEEE Int. Conf. on Robotics & Automation*, 2004.
- [12] S. Nakaoka et al. Constraint-based dynamics simulator for humanoid robots with shock absorbing mechanisms. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2007.