Zero-Moment Point on a Bipedal Robot under Bio-Inspired Walking Control

Nicolas Van der Noot and Allan Barrea

Abstract— **Humanoid robots are currently still far from reaching the impressive human walking capabilities. Among the different methods used to design walking controllers, those based on the Zero-Moment Point (ZMP) criterion are among the most popular, even if they induce intrinsic limitations in terms of energy consumption and robustness. In parallel, bio-inspired controllers are emerging. They overcome the ZMP-based limitations, but still miss robust stabilization rules to be validated on real robots. This contribution studies how to efficiently compute the ZMP in real-time on a robot walking with bio-inspired control rules, in order to detect when the robot stability is compromised.**

I. INTRODUCTION

Simplest methods to achieve robot walking neglect the robot dynamics, leading to so-called *static walking*. In this case, stability is insured as long as the center of mass stays within the support polygon. The price to pay for this simplified approach is a drastic limitation of the maximum walking speed [1]. Indeed, humans rely on their body dynamics to control their gait, thus allowing a higher walking speed and a lower energy consumption.

In contrast, more advanced methods focus on dynamic walking, and the most popular one is the so-called Zero-Moment Point (ZMP) criterion, an indicator of dynamic stability [2], [3]. Many experimental validations were already conducted to perform ZMP-based dynamic walking with humanoid robots such as ASIMO [4] or HRP-2 [5]. However, there are some drawbacks associated with ZMP-based controllers: energy-inefficient walking, limited walking speed, poor resistance against external perturbations and no appropriate reaction when the equilibrium is lost [6]. On top of that, these methods are computationally greedy, rely on perfect knowledge of the robot parameters and of the environment, and exhibit non human-like walking features, like constant knee bending [7].

In particular, ZMP-based methods rely on full local controllability (i.e. each point of the gait cycle is stable), which is not necessary to ensure a stable walking gait, then leading to a higher energy consumption [6]. The emerging concept called 'Limit Cycle Walking' considers the gait as a limit cycle whose global stability is prevalent to the local stability [1], leading to more energy-efficient walkers. Bio-inspired controllers are emerging as a promising way to implement such limit cycle walking, even if they have been mainly studied in simulation so far.

In this paper, we implement such a bio-inspired controller. More precisely, we adapt the reflex-based approach developed by Geyer and Herr [8] to make a humanoid robot, namely the COMAN, walk dynamically. The purpose is then to extend it to compute the robot ZMP position in real-time, while walking in a human-like fashion. Indeed, the ZMP is a good indicator of dynamic stability. Then, even if not used as an input of the gait controller, it can detect when the robot dynamic stability is compromised, leading to the activation of additional recovery modules. In this contribution, we focus on simulation only, but keeping in mind the idea of transferring these results to the real robot. This brings up new challenges in terms of computation time (the real robot controller runs at 1 kHz) and on available inputs (using only data coming from the real robot sensors).

The two main contributions of this paper are (i) the presentation of a computationally-efficient ZMP computation and (ii) the illustration of this computation for a robot walking with a human-like gait. This type of gait induces some rough heel strikes, which are absent from most ZMP-based walking robots, thus decreasing the accuracy of the ZMP computation. We will illustrate that computing the ZMP on a bio-inspired gait is valuable to quantify the gait stability, mainly in the lateral direction.

This paper is organized as follows. In section II, the COMAN robotic platform and its bio-inspired gait are briefly outlined. In section III, the ZMP concept is introduced and in section IV, the efficient computation method is described. In section V, simulation results obtained from two different bio-inspired gaits are presented. Section VI discusses the possibility of using such a method on real human-like walking robots. Finally, section VII concludes the paper.

II. COMAN PLATFORM AND GAIT CONTROLLER

The robotic platform used to compute the ZMP is the COmpliant HuMANoid platform (called COMAN) developed by the Italian Institute of Technology (IIT), within the AMARSI European project [9]. The 95 cm tall COMAN has 23 actuated degrees of freedom (DOFs), equipped with series elastic actuators and position, velocity and torque sensors [10]. An Inertial Measurement Unit (IMU) is located in the robot waist and 6-DOF force and torque sensors in each ankle can measure the ground reaction forces. Further information about COMAN can be found in [11], [12], and [13]. Figure 1 shows the basic planes used to describe the robot motion. The controller

This work was supported by the Université catholique de Louvain (internationalization grant, IMMC11/13), and by the Belgian F.R.S.-FNRS.

N. Van der Noot is with the Center for Research in Energy and Mechatronics; Institute of Mechanics, Materials and Civil Engineering; Université catholique de Louvain; B-1348 Louvain-la-Neuve, Belgium. nicolas.vandernoot@uclouvain.be

A. Barrea is with the Institute of Neuroscience; Université catholique de Louvain; B-1200 Brussels, Belgium and the Institute of Information and Communication Technologies, Electronics and Applied Mathematics; Université catholique de Louvain; B-1348 Louvain-la-Neuve, Belgium. allan.barrea@uclouvain.be

design was done in the Robotran simulation environment [13], [14], accurately modelling the physics of the COMAN and its environment.

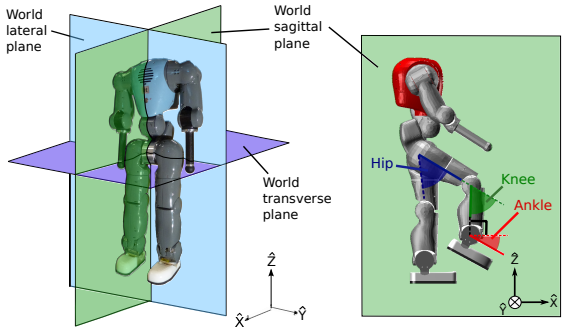


Fig. 1. Real COMAN, along with the related world planes and the inertial frame (left) and simulated COMAN in the world sagittal plane, along with the inertial frame and the angles describing the leg sagittal joints (right).

Two kinds of walking gaits are considered. First, the *2D gait* artificially constrains the robot waist to stay in the world sagittal plane. Second, the *3D gait* relaxes this constraint. In both cases, the design of the leg sagittal joints controller (6 DOFs) is based on the bio-inspired reflex rules described in [8]. These are the most important joints for walking since they propel the body forward. The remaining DOFs are controlled under the rules described in [15] and [16]. Finally, the whole controller is tuned via a Particle Swarm Optimization (PSO) algorithm [17], [18].

III. ZERO-MOMENT POINT

A. Zero-Moment Point overview

The bipedal gait is composed of several phases. The *single support* phase happens when only one foot contacts the ground, while the *double support* phase happens when both feet are in contact with the ground. Their corresponding support polygons can be seen in Figure 2.

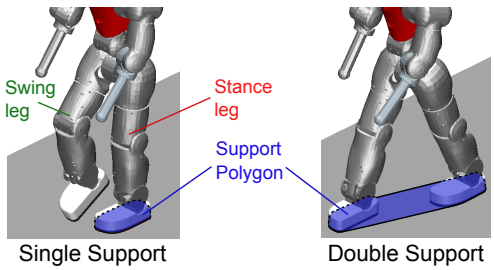


Fig. 2. Single support (left) and double support (right) phases.

For static bodies, the stability criterion is quite straightforward: the vertical projection of the center of mass (COM) on the ground must stay within the support polygon [1]. Nevertheless, this is not true for moving bodies where linear and angular accelerations must be taken into account. This is the purpose of the Zero-Moment Point (ZMP), which can be viewed as the generalization of the COM concept in dynamic conditions. More formally, the ZMP is the point on the ground where the tipping moment (i.e. the component of the moment that is tangential to the supporting surface) acting on the biped, due to gravity and inertia forces, equals zero [19]. Consequently, the body does not fall as long as its ZMP does not reach the boundaries of the support polygon. Otherwise, the robot would start falling by rotating around the edge of the support polygon where the ZMP is located. Thus, the distance between the ZMP and the nearest border of the support polygon is a good indicator of dynamic stability. The purpose of ZMP-based controllers is thus to maximize this distance, as described in Figure 3.

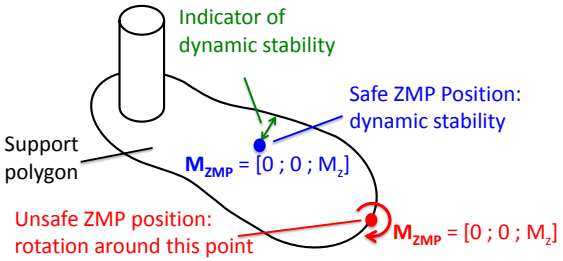


Fig. 3. The ZMP position is the point where the net moment M_{ZMP} of inertial gravity forces has only a vertical component, i.e. along the z-axis. When the ZMP lies within the support polygon (blue point), the distance to the nearest support polygon edge is an image of the dynamic stability. Otherwise, when the ZMP lies on the support polygon boundary (red point), the robot is dynamically unstable and starts to rotate around the point where the ZMP is located.

The ZMP criterion is a sufficient condition for dynamic stability but not a necessary one, as it requires full local controllability [6]. For instance, a ZMP at the front edge of the stance foot causes the body to start rotating around this edge, but then, it is possible to recover stable walking when the swing leg strikes the ground (see Figure 4). The strict application of the ZMP criterion constrains the stance foot to remain flat on the floor at all time, decreasing the performances in terms of efficiency, disturbance handling, and natural appearance as compared to human walking [1].

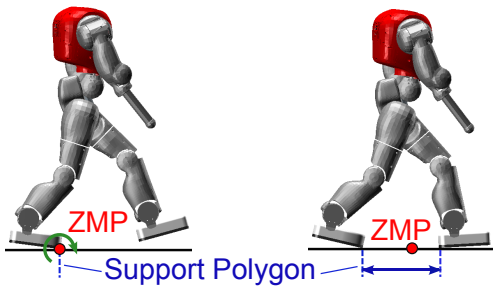


Fig. 4. On the left-hand side, the ZMP is located at the edge of the support polygon, causing the foot to start rotating around it. On the right-hand side, after the strike of the swing leg, the ZMP is no longer located at the edge of the support polygon (ZMP stability criterion recovered).

Among the numerous ZMP-based biped robots built around the world, few are provided with anthropomorphic soles [19], which prevents them from walking with a human-like gait, especially during the double support phases [20]. Indeed, most of these biped robots walk with bended knees. This is because controlling the ZMP becomes tedious when the knee is fully stretched, i.e. when the leg is in a singular configuration [7].

Importantly, even for limit cycle walking, the ZMP should never be located on the left or right side of the support polygon, indicating a lateral fall.

B. Center of pressure

The ZMP is equivalent to another concept widely used in bio-mechanics: the center of pressure (COP). When the field of pressure forces exerted by the feet on the ground is replaced by a single resultant force, the COP is the point where the resultant moment is zero, as defined by [19]. Based on this definition, they proved that the ZMP and the COP are strictly equivalent in a balanced gait. Interestingly, they are computed in a different way: the ZMP is computed from the body kinematics while the COP is generally referred to as the point computed from measured forces. So, the COP can be used to validate the ZMP computation, since both should coincide. Moreover, in simulations, the COP can be easily computed as a weighted sum of the interaction forces created by the contact model between the robot feet and the ground.

IV. ZERO-MOMENT POINT COMPUTATION

A. Main assumptions

The ZMP computation uses forward kinematics, relying on the following assumptions [3]:

a) *The robot is made of n rigid links*;
b) *All time-invariant quantities (inertia...) are known:* in simulation, these values are obtained from the CAD files of the real robot;
c) *All kinematic information is perfectly measurable:* this is **challenging after heel strike** because of the high accelerations it induces;
d) *The floor is rigid and motionless*;
e) *The feet do not slide over the floor:* there is a **small sliding** after some rough heel strikes.

These assumptions are valid, except $c)$ and $e)$ after heel strike.

B. Symbolic equations

There are very tight constraints on the time allocated for the ZMP computation, as it must be fast enough to let the controller fulfil its real-time constraints. A first naive approach to compute the ZMP would be to numerically compute the kinematics of all bodies (absolute position, velocity and acceleration) as well as the linear and angular momenta. However, this method requires solving several numerical loops, which is not efficient enough to be used in practice. In contrast, the proposed approach allows to get the exact ZMP position (i.e. without model simplification) while being fast enough in the computation.

The approach of this paper consists in getting the *symbolic* ZMP equations (as if they were computed by hand), automatically generated by a custom Matlab script. This approach is called the symbolic approach[1] and is at the heart of the Robotran simulator (see [14] for more insights). All time-independent bodies constants (masses, inertia matrices,...) are gathered. Symbolic variables are then defined and the ZMP is symbolically computed. The corresponding symbolic equations (computed in the Matlab script) can further be printed in a C file which can be integrated to the robot controller. Section IV-D details the method used to generate this script.

[1]The symbolic approach, which appeared in the eighties, is a powerful tool to drastically simplify mathematical expressions and to confer the equations a high portability towards other scientific disciplines like control, optimization, dimensioning, etc. [21]

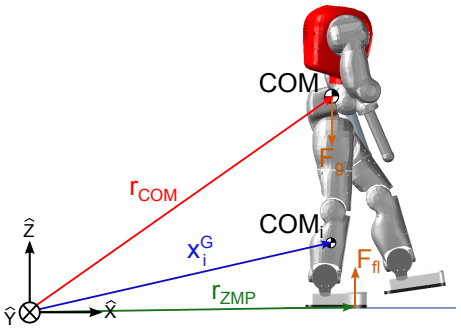


Fig. 5. The absolute position vectors r_{COM}, x_i^G and r_{ZMP} are computed with respect to the inertial frame. The only forces acting on the system are due to gravity (F_g) and contacts with the floor (F_{fl}).

To get the ZMP position, all forces acting on the COMAN are considered, namely the gravitational force $\boldsymbol{F_g}$ applied at the whole-body center of mass (COM) and the ground contact force $\boldsymbol{F_{fl}}$ [3]. The position vector of the COM is denoted $\boldsymbol{r_{COM}}$, $\boldsymbol{x_i^G}$ is the position vector of a single body i center of mass COM_i and r_{ZMP} is the position vector of the ZMP, as shown in Figure 5. Then, the net moment M_{ZMP} of inertial and gravitational forces is computed. Its x and y components (i.e. parallel to the ground as illustrated in Figure 5) must then be equal to zero. The ZMP position equations are presented in (1), where $\boldsymbol{r_{ZMP}} = [r_{ZMP,x}, r_{ZMP,y}, r_{ZMP,z}]^T$, m_{tot} is the total mass of COMAN, g_z is the gravitational acceleration, $\dot{\boldsymbol{N}}$ is the time derivative of the linear momentum and $\dot{\boldsymbol{H}}$ is the time derivative of the angular momentum.

$$\begin{cases} r_{ZMP,x} = \dfrac{m_{tot} \cdot r_{COM,x} \cdot g_z + r_{ZMP,z} \cdot \dot{N}_x^0 - \dot{H}_y^0}{\dot{N}_z^0 + m_{tot} \cdot g_z} \\ r_{ZMP,y} = \dfrac{m_{tot} \cdot r_{COM,y} \cdot g_z + r_{ZMP,z} \cdot \dot{N}_y^0 + \dot{H}_x^0}{\dot{N}_z^0 + m_{tot} \cdot g_z} \end{cases} \quad (1)$$

C. Inputs to the ZMP computation

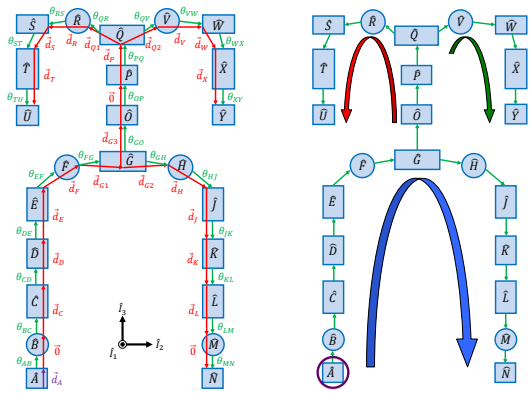
All quantities presented in (1) must be described relative to an inertial frame (visible in Figure 5). There are two different ways to get an inertial frame with the COMAN sensors: (i) using the Inertial Measurement Unit (IMU) located in the robot waist or (ii) assuming that the foot with the highest measured ground reaction force (called the *supporting foot*) is flat on the ground. Because the ZMP computation relies on sharp absolute acceleration variations, using the IMU might not be accurate enough, so the second possibility was chosen, i.e. there is always one foot assumed to be motionless and flat on the ground.

Then, the relative joint accelerations need to be computed while only the relative joint positions and velocities are accessible on the real robot. Because the relative joint velocities are

quite noisy, numerical differentiation must be rejected. To filter and differentiate these signals, a third-order polynomial fitting the velocity is computed, and then analytically differentiated. Like any filter, this approach induces a small delay (about 25 ms in this case).

D. Recursive forward kinematics

To compute all the quantities in (1), the recursive forward kinematics method [14] is used. To do so, a proper view of all body relationships in the robot is needed. The schematic on the left panel of Figure 6 gives an overview of the 24 bodies of the COMAN with appropriate notations. The computation of the forward kinematics starts from the supporting foot, denoted by $\hat{A}$ and supposed to be motionless. Then, a recursive path is defined (visible on the right panel of Figure 6) to list all the bodies in a precise order: first the leg bodies (blue arrow), then the trunk and first arm bodies (red arrow) and finally the other arm bodies (green arrow).



(a) Joint and positions (b) Forward kinematics path

Fig. 6. Left: schematic representation of the 24-bodies COMAN, along with reference frames, joints and relative positions. Right: same representation, but with the forward kinematic path being indicated.

When the forward kinematics of a body $i-1$ is computed, the forward kinematics of the next body i is obtained by computing first its absolute angular velocity $\boldsymbol{\omega_i}$ and acceleration $\dot{\boldsymbol{\omega}}_{\boldsymbol{i}}$ based on its relative angular velocity $\boldsymbol{\Omega_i}$ and acceleration $\dot{\boldsymbol{\Omega}}_{\boldsymbol{i}}$ and on the body $i-1$ kinematics, according to the following recursive equations:

$$\begin{cases} \boldsymbol{\omega_i} = & \boldsymbol{\omega_{i-1}} + \boldsymbol{\Omega_i} \\ \dot{\boldsymbol{\omega}}_{\boldsymbol{i}} = & \dot{\boldsymbol{\omega}}_{\boldsymbol{i-1}} + \dot{\boldsymbol{\Omega}}_{\boldsymbol{i}} + \boldsymbol{\omega_i} \times \boldsymbol{\Omega_i} \end{cases} \tag{2}$$

Thereafter, the absolute position $\boldsymbol{x_i^G}$ and acceleration $\ddot{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{G}}$ vectors of any body i are computed (velocities are not necessary to compute the ZMP). $\boldsymbol{d_i}$ is the relative position vector between COM_{i-1} and COM_i while R_i (recursively computed) is the absolute rotation matrix between the relative frame of body i and the inertial frame.

$$\begin{cases} \boldsymbol{x_i^G} = \boldsymbol{x_{i-1}^G} + R_i \boldsymbol{d_i} \\ \ddot{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{G}} = \ddot{\boldsymbol{x}}_{\boldsymbol{i-1}}^{\boldsymbol{G}} + R_i(\dot{\boldsymbol{\omega}}_{\boldsymbol{i}} \times \boldsymbol{d_i} + \boldsymbol{\omega_i} \times (\boldsymbol{\omega_i} \times \boldsymbol{d_i})) \end{cases} \tag{3}$$

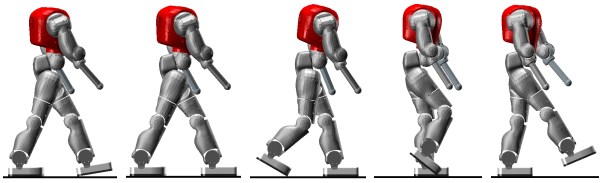
Once this recursive forward kinematics step is done, the rate of change of angular and linear momentum ($\dot{\boldsymbol{H}}^{\boldsymbol{0}}$ and $\dot{\boldsymbol{N}}^{\boldsymbol{0}} = \sum_{i=1}^{N} m_i \cdot \ddot{\boldsymbol{x}}_{\boldsymbol{i}}$) are computed, along with the whole-body COM position ($\boldsymbol{r_{COM}}$). m_i is the mass of body i while I_i^G (recursively computed) is its absolute inertial matrix.

$$\dot{\boldsymbol{H}}^{\boldsymbol{0}} = \sum_{i=1}^{N} (\boldsymbol{x_i^G} \times (m_i \, \ddot{\boldsymbol{x}}_{\boldsymbol{i}}^{\boldsymbol{G}}) + I_i^G \, \dot{\boldsymbol{\omega}}_{\boldsymbol{i}} + \boldsymbol{\omega_i} \times (I_i^G \, \boldsymbol{\omega_i})) \tag{4}$$

The last step consists in computing the ZMP with equations (1). During this recursive process, each symbolic equation computed with Matlab is printed in a C file. At the end, thousands of lines written in C are generated, which would have been impossible to derive by hand. In sum, this produces a computationally-efficient C file allowing to compute the ZMP in real-time.

V. RESULTS

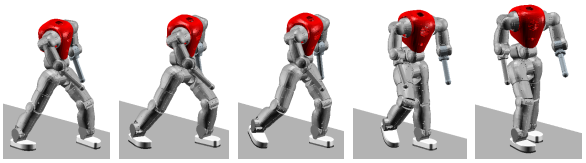
The ZMP computation was tested in simulation on two different kinds of gait. The first one is the *2D walking gait*. A video of this gait can be downloaded at [22] and a few snapshots of this video are shown in Figure 7. It can be observed that the robot exhibits straight knees at some phases of the gait (in contrast to many existing ZMP-based walkers) and strikes the ground with the heel first, similarly to humans.



(a) t=0.42 s (b) t=0.45 s (c) t=0.61 s (d) t=0.79 s (e) t=0.99 s

Fig. 7. Snapshots of the 2D walking gait of the COMAN in the Robotran simulator, corresponding to panels (a) and (b) of Figure 9.

The other gait is the *3D gait*. It can be watched at [23], some corresponding snapshots being presented in Figure 8. This second gait is clearly less robust, more jerky and less human-like than the 2D one.



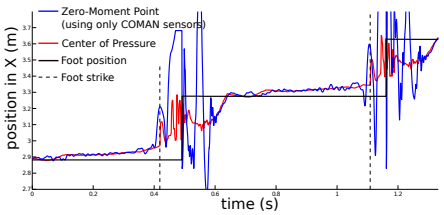
(a) t=0.38 s (b) t=0.47 s (c) t=0.52 s (d) t=0.66 s (e) t=0.83 s

Fig. 8. Snapshots of the 3D walking gait of the COMAN in the Robotran simulator, corresponding to panels (c) and (d) of Figure 9 and to Figure 10.

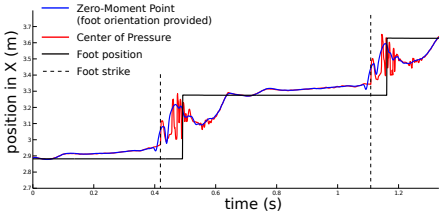
A. Computation time

One of the requirements of the ZMP computation was to be fast enough to be computable by the real COMAN controller without compromising its realtimeness. The computation time was evaluated around 0.014 ms, which is fast enough given the controller sampling rate, namely 1 ms [2].

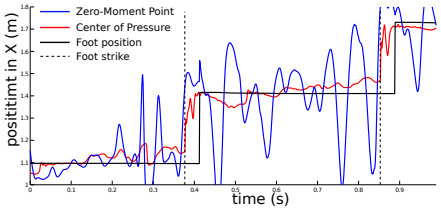
[2]This was tested on a Dell OptiPlex 7010 computer with quad-core Intel(R) Core(TM) i7-3770 CPU, 3.4GHz and 8 Go RAM.



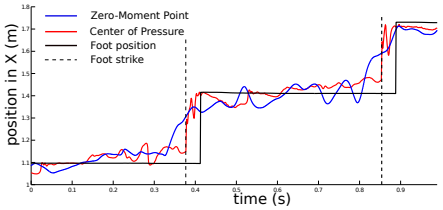
(a) ZMP computed for the 2D walking gait, using only sensors available on the real COMAN, without post-process filtering.



(b) ZMP computed for the 2D walking gait, with the foot orientation provided, without post-process filtering.



(c) ZMP computed for the 3D walking gait, with the foot orientation provided, without post-process filtering.



(d) ZMP computed for the 3D walking gait, with the foot orientation provided, post-processed with a 100 ms-wide running average.

Fig. 9. The ZMP position in X relative to an inertial frame is presented, along with the COP position, the supporting foot position and the foot strike instants. The ZMP position is shifted to compensate the 25 ms delay introduced by the filters.

B. 2D gait

Because the robot waist is constrained to stay within the world sagittal plane during the 2D walking gate, it is nonsense to compute the Y component of the ZMP in this case. Panel (a) in Figure 9 shows the X position (i.e. along the world sagittal plane) of the ZMP relative to an inertial frame. To compute this position, only sensors available on the real COMAN were used. To analyze the accuracy of this ZMP computation, the computed center of pressure (COP) absolute position is shown in the same figure. This COP position is considered to be errorless as it is computed from a weighted sum of the ground reaction forces. As the ZMP and COP are equivalent in balanced gaits (see section III-B), the matching between these two signals shows the accuracy of the ZMP computation proposed here.

Figure 9(a) shows that the matching between both signals is quite good, except just after a heel strike. This is due to the assumption that the supporting foot is perfectly flat on the ground (see section IV-C), which is not always the case with the gait obtained from our bio-inspired controller, mainly after heel strike. Coherently, providing the COMAN with the supporting foot absolute orientation (along with its acceleration), an information unavailable on the real COMAN, gives the results shown in Figure 9(b). The blue curve in Figure 9(b) corresponds to the blue one in Figure 9(a), but with the foot orientation provided. The matching in Figure 9(b) appears to be much better as the ZMP and the COP coincide, except that the ZMP is smoother (indeed, filters are used on the ZMP inputs).

C. 3D gait

The generated 3D gait was much more jerky, due to the lack of stability of our bio-inspired controller. Consequently, largest accelerations were induced and the ZMP computation was much more prone to errors. Therefore, only results where the absolute foot orientation was used in the computation (i.e. relaxing the "flat foot" hypothesis) are presented, a situation which does not make sense on the real robot. The X position of the ZMP is shown in Figure 9(c). As expected, the ZMP still matches the COP position, although with much more noise. Processing the ZMP with a 100 ms-wide running average post-process filtering, gives the signal shown in Figure 9(d). This shows that the filtered version of the ZMP matches the COP. Their positions in the transverse plane are shown in Figure 10.

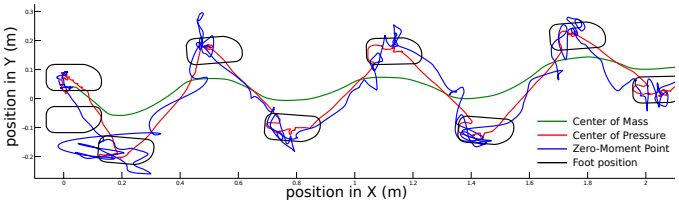


Fig. 10. Position of the COM, the COP, the ZMP and the feet in the transverse world plane for the 3D walking gait. To improve the graph clarity, the COP position was low-pass filtered with a 100 ms-wide running average. The ZMP is filtered like in Figure 9(d).

VI. DISCUSSION

For gaits being smooth enough like the 2D one, the method presented here exhibits very good results using only the sensors available on the real robot, except just after heel strikes, since the "flat foot" hypothesis is strongly violated at that moment. Nevertheless, this is not a real problem as the final purpose of this ZMP computation on this bio-inspired walker is to provide a measure of the gait stability in order to monitor possible fall (and trigger corresponding reactions). Because heel strike initiates the beginning of the double support phase, i.e. the one with the largest support polygon, the ZMP-based stability monitoring can be disregarded during this phase.

For jerky gaits, like the 3D one, results are deteriorated. So, Figure 10 shows that even after post-process filtering the ZMP position sometimes leaves the support polygon surface,

although the COP does not. This happens especially for the first steps, i.e. the jerkiest ones. Consequently, the robot would detect a fall, although this is not the case. Different solutions exist to overcome this problem. The first one would be to investigate more efficient filters for the ZMP inputs. Another one would be to use the COP position (computed thanks to the 6-DOF force and torque sensors included in the robot ankles) instead of the ZMP. Nevertheless, the problems related to the sharp acceleration variations would also alter the COP position. Then, a last method would be to use a simplified version of the robot model to compute the ZMP, but with no guarantee that this would improve the results. Finally, further research might be carried out to test this ZMP computation on the real robot.

VII. CONCLUSION

In this contribution, we presented a computationally-efficient method to compute the Zero-Moment Point on a robot walking with a human-like gait (i.e. with straight knees and heel strikes) obtained from a bio-inspired controller. Because our final objective is to implement this method on a real robot, two important requirements were (i) to minimize the computation time and (ii) to only use inputs corresponding to real robot sensors. In the present paper, we presented simulation studies.

As for the first requirement, the symbolic approach led to a very short computational time given realistic controller time constraints. Following this approach, we were able to automatically generate a custom C-code file, which would have been impossible to produce manually.

As for the second requirement, we established that the exact ZMP computation requires knowing the absolute orientation of the robot bodies, a measurement which is difficult to obtain with actual sensors. However, we proposed a second method, assuming that one foot is flat on the ground, which provided good matching with the real ZMP, except just after heel strikes. This is however not the most critical gait phase, since it corresponds to the largest support polygon. Future work will focus first on the improvements of the 3D gait controller, with the objective to make it less jerky in order to illustrate the usefulness of the ZMP computation as stability indicator. Then, we will develop an experiment to test this approach with a real robot.

VIII. ACKNOWLEDGMENT

We would like to thank our advisor, Prof. Renaud Ronsse and also the staff from the BioRobotics Laboratory at EPFL for the support they provided during this work. Prof. Paul Fisette had also been of really good advice about the mechanical theory used in this work.

REFERENCES

[1] D. G. Hobbelen and M. Wisse, *Humanoid Robots: Human-like Machines*. Vienna, Austria: I-Tech Education and Publishing, June 2007, ch. Limit Cycle Walking, pp. 277–294.

[2] M. Vukobratovic and B. Borovac, "Zero-moment point - thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 1, pp. 157 – 173, 2004.

[3] M. Dekker, "Zero-moment point method for stable biped walking," Eindhoven, University of Technology, Tech. Rep., July 2009, internship report.

[4] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep Planning for the Honda ASIMO Humanoid," *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 629–634, 2005. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1570188

[5] K. Kaneko, F. Kanehiro, S. Kajita, K. Yokoyama, K. Akachi, T. Kawasaki, S. Ota, and T. Isozumi, "Design of prototype humanoid robotics platform for hrp," in *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, vol. 3, 2002, pp. 2431–2436.

[6] H. Dallali, "Modelling and dynamic stabilization of a compliant humanoid robot, CoMan," Ph.D. dissertation, University of Manchester, 2012.

[7] R. Kurazume, S. Tanaka, M. Yamashita, T. Hasegawa, and K. Yoneda, "Straight legged walking of a biped robot," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, 2005, pp. 337–343.

[8] H. Geyer and H. Herr, "A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 3, pp. 263 – 273, 2010.

[9] AMARSI, "Amarsi homepage," http://http://www.amarsi-project.eu/, 2013.

[10] G. Pratt and M. Williamson, "Series elastic actuators," in *Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1, 1995, pp. 399–406 vol.1.

[11] N. Tsagarakis, S. Morfey, H. Dallali, and G. Medrano-Cerda, "COmpliant HuMANoid Platform (COMAN)," http://www.iit.it/en/robots/coman.html, 2013.

[12] N. Tsagarakis, Z. Li, J. Saglia, and D. Caldwell, "The design of the lower body of the compliant humanoid robot "cCub"," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 2035 – 2040.

[13] H. Dallali, M. Mosadeghzad, G. a. Medrano-Cerda, N. Docquier, P. Kormushev, N. Tsagarakis, Z. Li, and D. Caldwell, "Development of a dynamic simulator for a compliant humanoid robot based on a symbolic multibody approach," *2013 IEEE International Conference on Mechatronics (ICM)*, pp. 598–603, Feb. 2013.

[14] J.-C. Samin and P. Fisette, *Symbolic modeling of multibody systems*. Dordrecht ; Boston : Kluwer Academic Publishers, 2003.

[15] J. Wang, S. Hamner, S. Delp, and V. Koltun, "Optimizing locomotion controllers using biologically-based actuators and objectives," *ACM Transactions on Graphics (TOG)*, vol. 31, p. 25, 2012.

[16] K. Yin, K. Loken, and M. van de Panne, "Simbicon: simple biped locomotion control," *ACM Trans. Graph.*, vol. 26, no. 3, July 2007. [Online]. Available: http://doi.acm.org/10.1145/1276377.1276509

[17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, November/December 1995, pp. 1942 – 1948.

[18] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 1, pp. 58 – 73, February 2002.

[19] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. center of pressure-zero moment point," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 34, no. 5, pp. 630–637, 2004.

[20] ——, "Zero-moment point Measurements from a human walker wearing robot feet as shoes." *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 34, no. 5, pp. 638–648, 2004. [Online]. Available: http://dblp.uni-trier.de/db/journals/tsmc/tsmca34.html#SardainB04a

[21] CEREM, *Modeling Multibody Systems with ROBOTRAN*, http://www.robotran.be/, September 2009.

[22] "Dynamic 2D walking of the COMAN in Robotran (Youtube video)," http://www.youtube.com/watch?v=7xXbVi3h62Q, December 2013.

[23] "Dynamic 3D walking of the COMAN in Robotran (Youtube video)," http://www.youtube.com/watch?v=xR2Jh2iK-sE, December 2013.