



HAL
open science

Compliant attitude control and stepping strategy for balance recovery with the humanoid COMAN

Nicolas Perrin, Nikolaos Tsagarakis, Darwin Caldwell

► **To cite this version:**

Nicolas Perrin, Nikolaos Tsagarakis, Darwin Caldwell. Compliant attitude control and stepping strategy for balance recovery with the humanoid COMAN. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2013), Nov 2013, Tokyo, Japan. pp.4145-4151, 10.1109/IROS.2013.6696950 . hal-03089490

HAL Id: hal-03089490

<https://hal.science/hal-03089490>

Submitted on 15 Feb 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Compliant Attitude Control and Stepping Strategy for Balance Recovery with the Humanoid COMAN

Nicolas Perrin and Nikolaos Tsagarakis and Darwin G. Caldwell

Abstract—In this paper we describe an approach for humanoid robot balance recovery that combines a novel attitude control algorithm adding compliance to the robot’s behavior and increasing the smoothness of its motion, and an omnidirectional stepping strategy that can trigger one or two steps based on a measured disturbance vector. The proposed method is validated through experiments with the inherently compliant humanoid COMAN.

I. INTRODUCTION AND RELATED WORK

Since avoiding to fall is almost always the task of highest priority for humanoid robots, their balancing is a vital and fundamental control problem. Humanoid robots have two legs, a high center of mass, and small feet, three properties that make them unstable when standing, and very sensitive to external perturbations such as pushes. Humans are assumed to use three main strategies to perform balance control and push recovery while standing or walking ([14], [40], [7]). Two of them are the so-called ankle and hip strategies. In the ankle strategy, only the ankle joints are used to control the robot like an inverted pendulum. The hip strategy consists of a bending at the hips and has two main components that can be used to control balance: the upper-body angular momentum and the horizontal shift of the center of mass. The third strategy is the stepping strategy, and it occurs when using only the two first ones would fail. Early theoretical studies considered the problem of stability of simple planar bipeds ([39], [5], [6]), and there has been a lot of research on trying to use similar or slightly more complex models to recreate efficient ankle and hip strategies on humanoid robots. In [32] for example, hip and ankle strategies are used on a planar double inverted pendulum model (a bit similar to Acrobot [31]) with a regulator based on the tracking of the center of pressure (CoP), i.e. a particular point calculated from the integral of the pressure distribution under the feet. Many approaches, including ours, use the tracking of the CoP in order to estimate whether balancing actions are required.

[9] superposes a static and a dynamic balancer on a compliant human-sized robot. [15] and [16] achieve rotational stability by focusing on regulating the centroidal angular momentum. Several approaches have focused on controlling and distributing properly contact forces so as to obtain compliant, balanced behaviors ([33], [29], [8], [22], [19], [18]). For strong pushes, stepping is unavoidable, and deciding when and where to step remains a scientific challenge, although

interesting results have already been demonstrated. In the 1980s, Marc Raibert [27] showed that for running or jumping robots, robust behaviors could be obtained with a few simple and decoupled control laws. Surprisingly, good stepping strategies seem more difficult to realize with walking robots, particularly in non-steady state conditions. An important principle that has been formalized and studied is the capture point and capture region ([25], [26], [17]), the latter one being the region on the ground where a humanoid robot can step in order to come to a complete stop. Although it is complicated to compute it with a detailed and faithful model of the robot, it becomes easier with simple models of walking such as the linear inverted pendulum model (LIPM) which is very common in the field of walking trajectory generation [11]. In [25], the LIPM is extended by adding a flywheel body that enables the humanoid to control its centroidal angular momentum, and the capture region is computed. In [23] the LIPM is again used to decide steps that perform balance control, but with a better handling of the double support stance phase. The inverted pendulum model has also been extended to a spring loaded model (SLIP) ([4], [28]). This model demonstrates open-loop stability, and walking controllers have been designed to reproduce its limit cycles, while also adding control actions to increase the robustness of the walking and its ability to reject external disturbances [3]. Studies of the capture point dynamics have also lead to walking stabilizers [2]. Other approaches for walking stabilization include quick modifications of the CoP reference based on perceived disturbances [21], or posture and force control to simplify the robot walking dynamics [12]. [34] and [1] express stepping strategies within a model predictive control scheme. In [38], a modified version of preview control based on the LIPM is used as a unified way to realize walking pattern generation and stepping for push recovery. In [20], reactive stepping is considered as a mean to suppress part of an external disturbance that a feedback controller tries to cancel out.

In the present paper, we focus on the stepping strategy and practically address the problem of considering the stepping limitations of the robot when deciding to perform one step or more. Additionally, we use the stabilization control law of [19] and add atop of it a novel attitude control scheme that slows down the dynamics of the robot, leading to smoother trajectories even in the presence of perturbations.

Feedback from Inertia Measurement Units (IMUs) has been used extensively as a mean to control the balance of humanoid robots. In [10], Jenkins et al. introduce a method for inertial motion control inspired by how inverted pendulums

N. Perrin, N. Tsagarakis, and D. G. Caldwell are with the department of Advanced Robotics, Istituto Italiano di Tecnologia, Via Morego, 30, 16163 Genova. {nicolas.perrin, nikolaos.tsagarakis, darwin.caldwell} at iit.it

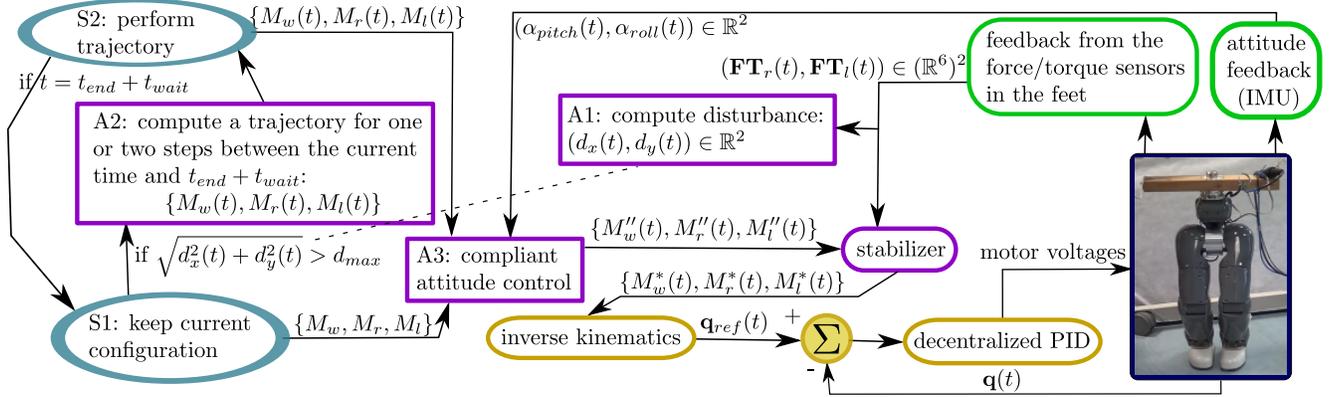


Fig. 1. The global architecture of our control scheme. Reference trajectories are sent as triples of homogeneous matrices representing the rigid body motion of the waist, right and left foot in the world frame. Both controllers (the compliant attitude controller and the stabilizer) use feedback from the robot to update the values of these homogeneous matrices. They are then transformed into joint values through inverse kinematics, and a decentralized PID controller (one PID per joint) regulates the motor inputs.

such as Segway PTs or Golem Krang [35] dynamically balance. They generalize this notion to humanoids, mimicking the action of a standard proportional-derivative (PD) motion controller that computes motor torques about each degree of freedom of an inverted pendulum, based on the differences between measured and desired angles and angular velocities. In [13], IMUs are used to measure the angular velocity of the robot, which makes it possible to control the dynamic balance of the humanoid robot with direct angular momentum feedback, following an idea introduced in [30]. In [36] a similar approach is used to improve the transient response of the trunk attitude in a control algorithm for gravity compensation on humanoid robot.

In our approach, we use the IMU feedback to give a more compliant behavior to the robot, which already has an intrinsic compliance due to its series elastic actuators [37], and some additional active compliance due to the stabilizer [19]. Our compliant attitude control algorithm uses as feedback only the measured attitude, not the angular velocities. It relies on one parameter (μ) that can be tuned to achieve a wide range of levels of compliance. Experiments on the lower body of the humanoid robot COMAN validated the combination of this compliant attitude control scheme with the stabilization control law of [19] and our stepping strategy. In the experimental results obtained, COMAN demonstrated a very robust behavior, making appropriate sequences of steps to keep its balance after strong pushes in any direction.

II. GLOBAL ARCHITECTURE OF THE CONTROL SCHEME

The robot used in this work is COMAN without its upper body, and we assume it is standing on a flat horizontal floor.

The global architecture of our control scheme is described on Fig. 1. It is based on a simple finite state machine that switches between two states. In the first state, S1, the robot tries to keep the current standing configuration, while possibly undergoing external perturbations. Based on the feedback from the force/torque sensors in its feet, the robot estimates how the CoP deviates from its desired reference

location. The magnitude and speed of this deviation are used by the algorithm A1 to compute a “disturbance vector” (d_x, d_y) . If the norm of this vector exceeds a threshold d_{max} , a stepping action is required, and (d_x, d_y) and the current foot placements are sent as inputs to the algorithm A2 which decides whether to make one or two steps, and computes the corresponding trajectory. At this point, the finite state machine switches to state S2 for a duration decided in advance. The moment when the finite state machine will switch back to state S1 is $t_{end} + t_{wait}$, t_{end} being the moment when the trajectory of one or two steps ends, and t_{wait} a small additional delay (150ms in our case). Basically, the reason why t_{wait} is added is because, although we have a compliant attitude controller and a stabilizer that make the robot motion smooth, there are still self-induced disturbances that occur just after the end of steps, mostly due to the impact of the swing foot when it lands. Without t_{wait} , just after switching back to S1 the disturbance vector might have a norm exceeding d_{max} , leading to an immediate switch back to S2 and to new steps being performed, even without any external disturbance. Therefore, a properly tuned t_{wait} helps avoiding unnecessary infinite sequences of steps. In this work, t_{wait} was experimentally tuned to compromise between stepping sensitivity and self-induced stepping. The smoother the robot motion is, the smaller t_{wait} can be chosen. We achieve smoothness thanks to two independent controllers. The first one implements an original compliant attitude control algorithm (A3) which takes as input the attitude estimation given by the IMU attached to the waist of the robot, and the second one is the stabilizer described in [19], which uses feedback from the force/torque sensors of the robot feet to perform an emulation of admittance control.

In the next three sections we describe and evaluate the algorithms A1, A2 and A3, respectively. In section VI, we present our experimental results and discuss future work.

III. DISTURBANCE ESTIMATION

The purpose of this section is to define the “disturbance vector” based upon which we decide to trigger steps.

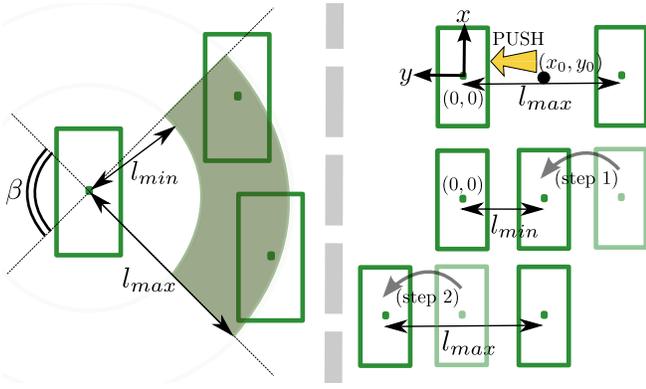


Fig. 2. On the left: the stepping capabilities of the robot. The filled area represents the locations where the center of the swing foot can be put after a right step. The stepping capabilities are symmetric. We have $\beta = 42^\circ$, $l_{min} = 0.14m$, and $l_{max} = 0.245m$. On the right: a lateral push to the left obliges the robot to perform a step. In this particular situation, since it cannot move its left foot further on the left, it instead performs a quick sequence of two steps, starting with a right step.

The force/torque sensors of the robot feet provide an estimation of the position of the CoP. When the robot is not performing a step, we want the CoP to have the same horizontal position as the center of mass (right between the foot centers). Therefore, we can easily compute the CoP deviation. In the stabilizer, this deviation is converted into a desired horizontal modification of the center of mass (CoM) reference: $(\Delta x_{com}, \Delta y_{com})$ (cf. [19]). This horizontal modification regulates the perceived cartesian compliance at the waist, and it can be used to estimate the direction of the CoM motion during an imminent fall, if it were to happen. We base our decisions for triggering steps on the evolution of the vector $(\Delta x_{com}, \Delta y_{com})$. Basically, we want to stay in a situation where the CoM reference is not significantly modified by the stabilizer. Thus, our goal is for the norm of $(\Delta x_{com}, \Delta y_{com})$ to remain lower than a certain value Δ_{max} . We could simply use $(\Delta x_{com}, \Delta y_{com})$ as our disturbance vector, but if we also observe its rate of change, we can predict in advance its future values, and thus guess if the vector will soon either leave or return to the disk of radius Δ_{max} . We therefore express our disturbance vector with the following heuristic:

$$\begin{pmatrix} d_x(t) \\ d_y(t) \end{pmatrix} = \begin{pmatrix} \Delta x_{com}(t) \\ \Delta y_{com}(t) \end{pmatrix} + K \frac{d}{dt} \begin{pmatrix} \Delta x_{com}(t) \\ \Delta y_{com}(t) \end{pmatrix},$$

with K tuned manually during experiments where the robot undergoes perturbations. The chosen value was $K = 0.13s$. Similarly, d_{max} (the disturbance vector norm threshold) was tuned experimentally, and its chosen value was $0.01m$.

IV. STEPPING STRATEGY

During a step, the position and orientation of the next footprint is limited to a region that depends on intrinsic physical constraints of the robot, and on limitations of the algorithm that generates the walking motion. It is convenient to define a subset of this region with simple geometry so as to always have a simple set of next possible steps. To generate

stepping motions, we use the algorithm introduced in [24]. After a series of tests where we tried various sequences of steps on the robot, we defined the stepping region as shown on Fig. 2 (on the left). This region is such that during double support phases the distance between the feet is always between l_{min} ($= 0.14m$) and l_{max} ($= 0.245m$). Let us now consider the situation described on the right of Fig. 2: the robot is in a standing configuration with distance l_{max} between its feet, and it is being pushed laterally to the left. After such a push, the normal reflex is usually to step with the left foot and move it a bit further away from the right one. But if the feet are already far apart, as in the situation described on Fig. 2, it is impossible. In that case, a human would quickly put his right foot next to the left one and then perform a lateral step with the left foot, as shown on Fig. 2. It is for this reason that our algorithm A2 can decide between making one or two steps.

We use the same duration for all the single steps, but when a sequence of two steps is decided, we make the individual steps faster so as to reduce the difference between the duration of a sequence of two steps ($1.0s$) and that of a single step ($0.7s$). We add the following restrictions on the steps:

- after a single step, the distance between the foot centers must be l_{max} ;
- during a sequence of two steps, after the first step the distance between the foot centers must be l_{min} , and after the second one the distance must be l_{max} .

Such restrictions are not compulsory, but they give uniform characteristics to the steps performed: the single steps always go from a stance with foot centers at distance l_{max} to another stance with foot centers at distance l_{max} , and thus are rather “large” steps, while the steps during a sequence of two steps are always between a stance with foot centers at distance l_{min} and a stance with foot centers at distance l_{max} , and thus are smaller steps, which makes it reasonable to perform them faster.

Let us now explain precisely how we choose between one or two steps.

We assume that there is a point P on the floor above which the robot can aim at positioning its CoM so as to absorb and suppress the effect of the current external perturbation. Although inspired from the concept of capture point [17], P is not defined as such. We leave theoretical justifications for its existence and position as future work, and give a simple heuristic for the position of P : if (x_0, y_0) is the initially desired horizontal position of the CoM (before the push), i.e. the center of the support polygon, and (d_x, d_y) the disturbance vector, then we define P as $(x_0 + \lambda d_x, y_0 + \lambda d_y)$, $\lambda > 0$ being an arbitrary value that can be experimentally tuned. If the stepping strategy is well adjusted, very large values of λ will always lead to a decent behavior. In our case, we set $\lambda = 300$.

Once a stepping action is triggered, our goal is to find steps that minimize the final horizontal distance between the CoM and P (i.e. the distance between the “CoM target” and P). We add a penalty $\delta > 0$ to favour single steps: we perform

two steps if and only if the ‘‘improvement’’ (compared with a single step) is at least δ .

Let us consider again the example on the right of Fig. 2. $(0, 0)$ is the position of the left foot center. The lateral push causes the point P to be somewhere on the y -axis, at a position that can be written $(0, -\frac{l_{max}}{2} + \lambda d)$. If we choose to perform only one step, then, assuming that $\lambda d \geq \frac{l_{max}}{2}$, the best choice is to step in place (we recall that after a single step the distance between the foot centers must be l_{max}), which corresponds to the CoM target $(0, -\frac{l_{max}}{2})$.

If on the other hand we decide to perform two steps, the first step would move the right foot laterally and put it at distance l_{min} from the left one (step 1 on Fig. 2), and assuming also $\lambda d \geq l_{max} - l_{min}$, the second step would be the step 2 of Fig. 2, and it would lead to the following new position of the center of the support polygon, i.e. to the following CoM target: $(0, \frac{l_{max}}{2} - l_{min})$.

We can compute the distances between P and the CoM target in both cases:

- with one step: λd ;
- with two steps: $\lambda d - (l_{max} - l_{min})$

Considering the penalty δ , the algorithm A2 would decide to perform two steps if and only if $\delta + \lambda d - (l_{max} - l_{min}) < \lambda d$, that is: $\delta < l_{max} - l_{min}$. This particular property can be seen on the left column of three maps on Fig. 3: the map at the bottom is the only one for which $\delta \geq l_{max} - l_{min}$, and the only one where sequences of two steps are never chosen in response to purely lateral perturbations (i.e. along the y -axis).

In total, there are 4 distinct stepping actions that can be chosen: one left step (c1), one right step (c2), two steps with first a left step (c3), and two steps with first a right step (c4). For each of these actions, we try to find a step or sequence of two steps that brings the CoM target as close to P as possible. For one step, an analytical solution is easy to find, but it becomes a bit more complicated with two steps. Several optimization methods are possible, depending on how we parametrize the steps. In our implementation, we use polar coordinates and perform gradient descents just as if we were computing the motion of a manipulator arm using directions obtained from the pseudo-inverse of the Jacobian matrix. The advantage of this approach is that we could easily extend it and use state-of-the-art inverse kinematics algorithms to obtain solutions with sequences of 3 or more steps.

For a given stance of the robot (i.e. its relative foot placements) and a given value of δ , the choice between c1, c2, c3 and c4 according to the position of P divides the plane in 4 regions. Fig. 3 shows such divisions of the plane with the stepping capabilities of COMAN (cf. Fig. 2) for different values of δ and different stances. We can see a sudden ‘‘topological’’ change when δ becomes larger than $l_{max} - l_{min}$, i.e. $0.105m$ in our case. For our experiments, we set $\delta = 0.10m$, a value just slightly smaller than $l_{max} - l_{min}$.

V. COMPLIANT ATTITUDE CONTROL

As explained in the caption of Fig. 1, desired configurations are sent to the robot through triples of homogeneous matrices: M_w for the waist, M_r for the right foot, and M_l

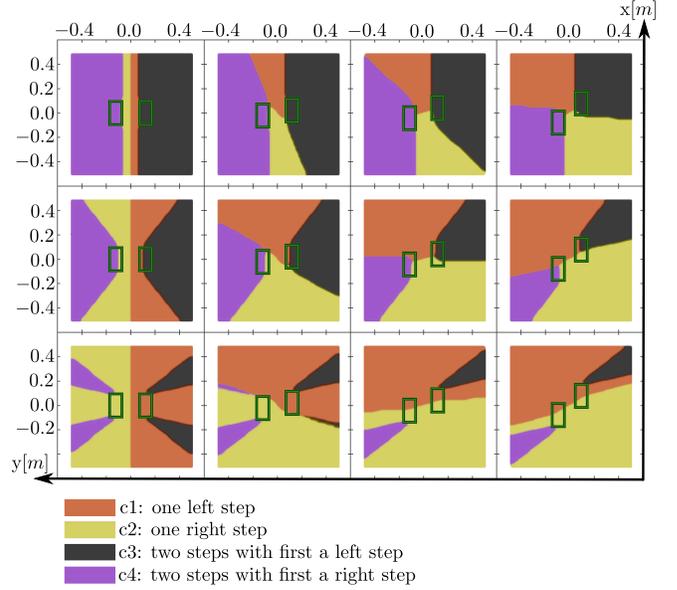


Fig. 3. Maps of stepping strategies chosen according to the position of P , for various values of δ and various stances. The first row at the top corresponds to $\delta = 0.05m$, the second one to $\delta = 0.10m$, and the last one at the bottom to $\delta = 0.108m$.

for the left foot. Each of these homogeneous matrices defines the position and orientation of the corresponding body in a fixed reference frame. Since each leg of the robot has 6 joints, these matrices entirely define the robot configuration. This configuration being relative, and not absolute, we could always choose $M_w = I_{4,4}$, which is the convention we use in the triples initially sent to the robot. However it is more convenient to allow the control algorithms to modify M_w .

Both our control algorithm A3 and the stabilizer [19] modify the triple $\{M_w, M_r, M_l\}$ based on some feedback. For the stabilizer, this feedback comes from the force/torque sensors of the feet. For the compliant attitude control algorithm A3, this feedback is the attitude angles that are sent by the IMU located in the robot waist. These angles are an estimation of the orientation of the robot waist in the space, i.e. the actual value of the rotational part of M_w .

In this work, we only take into consideration the pitch (α_{pitch}) and roll (α_{roll}) angles, not the yaw. These angles can be represented by the homogeneous matrix $M_{rot} = M_{pitch} \cdot M_{roll}$. If the reference sent to the robot is $\{I_{4,4}, M_r, M_l\}$, then based on the estimated pitch and roll angles, we can assume that the real configuration in the space (without considering the yaw or the actual position of the waist) is $\{M_{rot}, M_{rot} \cdot M_r, M_{rot} \cdot M_l\}$.

If we correct the reference in order to keep the orientation of the feet as initially desired, we obtain the following triple:

$$\left\{ M_{rot}, \left[\begin{array}{c|c} R_r & R \cdot \mathbf{p}_r \\ \hline 0 & 1 \end{array} \right], \left[\begin{array}{c|c} R_l & R \cdot \mathbf{p}_l \\ \hline 0 & 1 \end{array} \right] \right\}$$

with

$$M_{rot} = \left[\begin{array}{c|c} R & 0 \\ \hline 0 & 1 \end{array} \right], M_r = \left[\begin{array}{c|c} R_r & \mathbf{p}_r \\ \hline 0 & 1 \end{array} \right], M_l = \left[\begin{array}{c|c} R_l & \mathbf{p}_l \\ \hline 0 & 1 \end{array} \right]$$

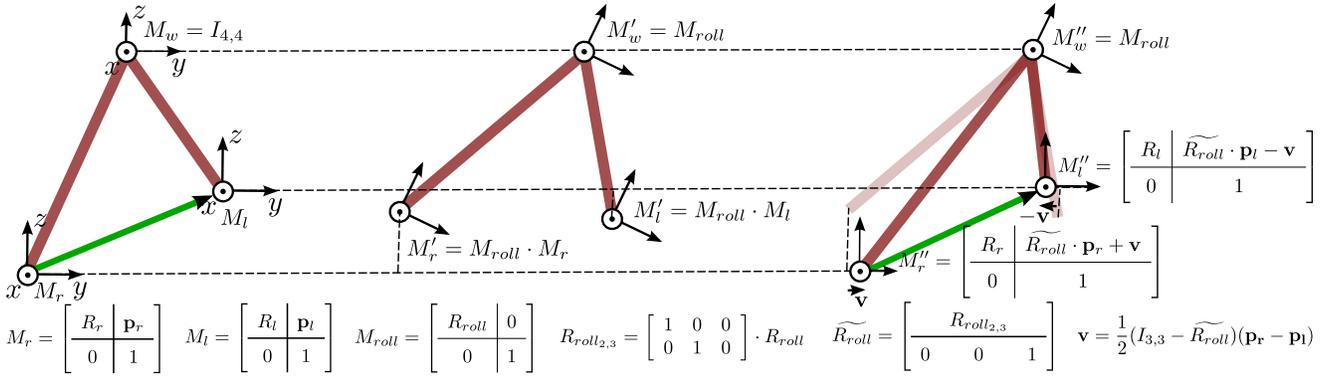


Fig. 4. The steps of the computation of the compliant attitude control algorithm to modify the triple $\{M_w, M_r, M_l\}$, when a roll is measured by the IMU attached at the robot waist.

Our goal is to “accept” the disturbance on the waist orientation, but without moving the feet, and without moving the waist downward. To keep the waist and feet at the same height, we simply define $R_{2,3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \cdot R$, and $\tilde{R} = \begin{bmatrix} R_{2,3} \\ 0 & 0 & 1 \end{bmatrix}$. The following triple leads to both feet being at the same height as in the initial reference triple:

$$\left\{ M_{rot}, \left[\begin{array}{c|c} R_r & \tilde{R} \cdot \mathbf{p}_r \\ \hline 0 & 1 \end{array} \right], \left[\begin{array}{c|c} R_l & \tilde{R} \cdot \mathbf{p}_l \\ \hline 0 & 1 \end{array} \right] \right\}$$

The last problem is that we want the relative position of the feet to remain unchanged, in order to avoid potential slipping motions during double support phases. However, with the previous triple, we usually have:

$$\tilde{R} \cdot \mathbf{p}_l - \tilde{R} \cdot \mathbf{p}_r \neq \mathbf{p}_l - \mathbf{p}_r$$

So, we pose $\mathbf{v} = \frac{1}{2}(I_{3,3} - \tilde{R})(\mathbf{p}_r - \mathbf{p}_l)$, and finally obtain the following triple:

$$\{M''_w, M''_r, M''_l\} = \left\{ M_{rot}, \left[\begin{array}{c|c} R_r & \tilde{R} \cdot \mathbf{p}_r + \mathbf{v} \\ \hline 0 & 1 \end{array} \right], \left[\begin{array}{c|c} R_l & \tilde{R} \cdot \mathbf{p}_l - \mathbf{v} \\ \hline 0 & 1 \end{array} \right] \right\}$$

Fig. 4 illustrates the whole computation when only a roll is measured.

With this control algorithm, any perturbation on the waist pitch or roll would result in a new reference where the desired pitch and roll correspond exactly to the one measured. This control would definitely diverge. However, it is possible to “follow” only part of the measured pitch and roll by doing the same computations, but with the matrices M_{pitch} and M_{roll} corresponding respectively to the angles $\mu\alpha_{pitch}$ and $\mu\alpha_{roll}$, with $\mu \in [0, 1)$. The IMU uses a Kalman filter before sending angular values, but we also add an averaging filter. After tuning this filter, we verified experimentally that it was possible to obtain a stable control with values of μ ranging from 0 to about 0.7. With $\mu = 0$, the control algorithm does nothing. But with $\mu > 0$, it adds compliance to the robot behavior, and for adequate values it tends to increase the stability and the smoothness of its motion.

This effect has been experimentally validated with the following “fixed frequency swaying waist experiment”: while

the robot stands with both feet on the ground and its foot centers at distance l_{max} from each other, we send three cycles of a reference trajectory that consists in a lateral sinusoidal motion of the waist along the y-axis (without moving the feet). The frequency of this trajectory is slightly above 1Hz, and we try to perform it with various amplitudes and various values of μ .

We did experiments with the following values of μ : 0.00 (the feedback control is only done by the stabilizer), 0.15, 0.30, 0.45 and 0.60. For each of these values, we progressively increased the amplitude until it lead to a motion making the robot fall. Table I shows the results.

TABLE I
FIXED FREQUENCY SWAYING WAIST EXPERIMENT

Value of μ	Max. withstandable amplitude
0.00	0.062m
0.15	0.070m
0.30	0.085m
0.45	0.099m
0.60	0.120m

With $\mu = 0.60$, the maximum withstandable amplitude is almost twice as large as the maximum withstandable amplitude obtained with $\mu = 0$, which clearly shows an increased robustness brought by our attitude control algorithm.

Fig. 5 and 6 show experimental results for an amplitude of 0.062m. Fig. 5 shows that increasing μ leads to a significant reduction of the vertical component of the reaction forces measured by the force/torque sensors in the robot feet, which indicates that the motion becomes much smoother. What’s more, on Fig. 6, we can see that the measured variations of the waist roll tend to have a smaller amplitude when μ increases. Thus, increasing μ improves the tracking of the horizontal attitude.

VI. EXPERIMENTAL RESULTS AND FUTURE WORK

Combining our compliant attitude control algorithm, the stabilizer [19] and our stepping strategy according to the organization described on Fig. 1, we obtained a robust behavior of balance recovery with the humanoid robot CO-MAN. When pushed in any direction, the robot measures

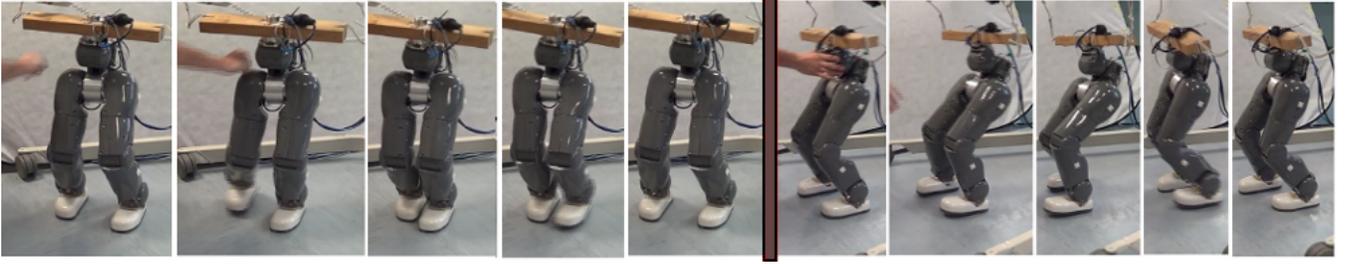


Fig. 7. On the left: after a lateral push, the robot performs a sequence of two steps. On the right: after a frontal push, the robot decides to perform a single step backwards. However, just after this step the balance is not fully recovered, and the robot measures a disturbance indicating that it might be about to fall. As a result, it decides to perform a step backwards one more time, at the end of which the balance is fully recovered.

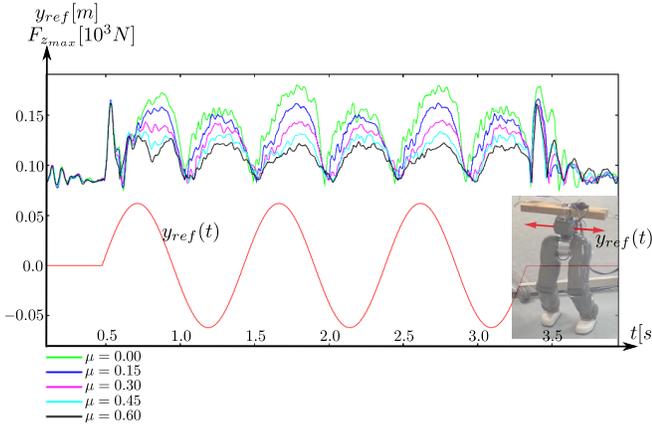


Fig. 5. The reference trajectory $y_{ref}(t)$ is sent to the control algorithm. For different values of μ , we record the vertical component of the reaction forces measured by the force/torque sensors in the robot feet. Denoting by $f_{right}(t)$ and $f_{left}(t)$ these forces, we define $F_{z_{max}}(t) = \max(f_{right}(t), f_{left}(t))$. Large values of μ lead to a smoother motion with smaller reaction forces.

a disturbance and triggers appropriate steps. It can decide to perform two quick steps when one step only would not help much. Since the robot has just been pushed, it doesn't necessarily track well the trajectory it tries to perform, but the compliance added by our attitude control algorithm ensures the smoothness of the motion (we use $\mu = 0.25$ in standard experiments). At the end of a step or sequence of two steps, the robot might still be unbalanced, for example if the initial push was strong. In that case, the robot can quickly decide to perform a new step or sequence of two steps (cf. Fig. 7).

One very interesting property is that we can easily introduce artificial disturbances and use our balance recovery scheme as a robust gait generator. More precisely, an artificial disturbance (\hat{d}_x, \hat{d}_y) of norm slightly larger than d_{max} can be added to the measured disturbance (d_x, d_y) , and it would result in the robot walking in the direction of the vector (\hat{d}_x, \hat{d}_y) , but with the ability to withstand external perturbations and perform steps in any direction if needed. Besides, if a human operator wants to stop the walking motion of the robot, it is very easy to do so by pulling or pushing the robot in the direction opposite to its motion, with the real disturbance in this case naturally compensating the artificial one.

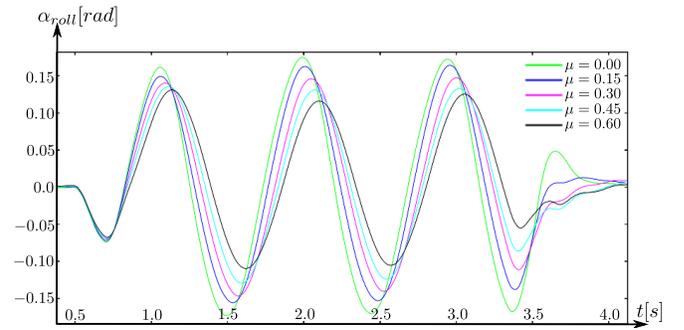


Fig. 6. In the same experiment as in Fig. 5 (the “fixed frequency swaying waist experiment”), we record the roll angle measured by the IMU attached to the waist of the robot. The amplitude of its variations decreases when μ increases, showing that a large value of μ results in a better tracking of the horizontal attitude.

In terms of future work, our first goal is to apply our algorithm to COMAN with its upper body, which will allow us to add a hip strategy. Then, we have identified three main axes of research that should help us to further improve the experimental results:

- *Torque control*: implementing torque control in the robot legs will allow us to use a more expressive sensory feedback in order to make appropriate stepping decisions, and to further increase the smoothness of the motion.
- *Online modification of the trajectories*: although planning initial trajectories for the steps seems reasonable, it would be very beneficial if the steps could be modified in real-time, based on the disturbances that are measured during the single support phases. Basically, the planning should be merged into the control.
- *Machine learning*: using lifelong machine learning should enable a continuous improvement of the performance and an automatic tuning of the different parameters of our algorithms.

ACKNOWLEDGMENT

This work is supported by the ICT-248311 AMARSI and the FP7-ICT-2013-10 WALK-MAN European Commission projects. We would like also to thank and acknowledge the contribution to this work of Zhibin Li whose help was essential for the implementation of stabilizer module in [19].

REFERENCES

- [1] Z. Aftab, T. Robert, and P.-B. Wieber. Ankle, hip and stepping strategies for humanoid balance recovery with a single model predictive control scheme. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, 2012.
- [2] J. Engelsberger, C. Ott, M.A. Roa, A. Albu-Schaffer, and G. Hirzinger. Bipedal walking control based on capture point dynamics. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4420–4427, sept. 2011.
- [3] G. Garofalo, C. Ott, and A. Albu-Schaffer. Walking control of fully actuated robots based on the bipedal slip model. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1456–1463, may 2012.
- [4] H. Geyer, A. Seyfarth, and R. Blickhan. Compliant leg behaviour explains basic dynamics of walking and running. *Proceedings of the Royal Society B: Biological Sciences*, 273(1603):2861–2867, 2006.
- [5] Jr. Golliday, C. and H. Hemami. Postural stability of the two-degree-of-freedom biped by general linear feedback. *Automatic Control, IEEE Transactions on*, 21(1):74–79, feb 1976.
- [6] H. Hemami and B.-R. Chen. Stability analysis and input design of a two-link planar biped. *Int. J. Rob. Res.*, 3(2):93–100, 1984.
- [7] F.B. Horak and L.M. Nashner. Central programming of postural movements: Adaptation to altered support surface configuration. *J. Neurophysiology*, 55(6):1369–1381, 1986.
- [8] Sang-Ho Hyon. Compliant terrain adaptation for biped humanoids without measuring ground surface and contact forces. *Robotics, IEEE Transactions on*, 25(1):171–178, feb. 2009.
- [9] Sang-Ho Hyon, Rieko Osu, and Yohei Otaka. Integration of multi-level postural balancing on humanoid robots. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 1549–1556, may 2009.
- [10] O.C. Jenkins, P. Wrotek, and M. McGuire. Dynamic humanoid balance through inertial control. In *Humanoid Robots (Humanoids), 2007 7th IEEE-RAS International Conference on*, pages 501–506, 2007.
- [11] 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 *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 1, pages 239–246 vol.1, 2001.
- [12] 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 *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4489–4496, oct. 2010.
- [13] S. Kajita, K. Yokoi, M. Saigo, and K. Tanie. Balancing a humanoid robot using backdrive concerned torque control and direct angular momentum feedback. In *Robotics and Automation, 2001. ICRA 2001. IEEE International Conference on*.
- [14] A.D. Kuo. An optimal control model for analyzing human postural balance. *Biomedical Engineering, IEEE Transactions on*, 42(1):87–101, jan. 1995.
- [15] Sung-Hee Lee and A. Goswami. Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4667–4672, april 2007.
- [16] Sung-Hee Lee and A. Goswami. Ground reaction force control at each foot: A momentum-based humanoid balance controller for non-level and non-stationary ground. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3157–3162, oct. 2010.
- [17] Twan Koolen, Tomas De Boer, John Rebula, Ambarish Goswami, and Jerry Pratt. Capturability-based analysis and control of legged locomotion, part 1: Theory and application to three simple gait models. *Int. J. Rob. Res.*, 31(9):1094–1113, August 2012.
- [18] Z. Li, N. Tsagarakis, and D. Caldwell. A passivity based admittance control for stabilizing the compliant humanoid coman. In *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, 2012.
- [19] Zhibin Li, B. Vanderborght, N.G. Tsagarakis, L. Colasanto, and D.G. Caldwell. Stabilization for the compliant humanoid robot coman exploiting intrinsic and controlled compliance. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2000–2006, may 2012.
- [20] M. Morisawa, F. Kanehiro, K. Kaneko, N. Mansard, J. Sola, E. Yoshida, K. Yokoi, and J. Laumond. Combining suppression of the disturbance and reactive stepping for recovering balance. In *Intelligent Robots and Systems (IROS), 2010. IEEE/RSJ International Conference on*, pages 3150–3156, 2010.
- [21] K. Nishiwaki and S. Kagami. Strategies for adjusting the zmp reference trajectory for maintaining balance in humanoid walking. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4230–4236, may 2010.
- [22] C. Ott, M.A. Roa, and G. Hirzinger. Posture and balance control for biped robots based on contact force optimization. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 26–33, oct. 2011.
- [23] F. Parietti and H. Geyer. Reactive balance control in walking based on a bipedal linear inverted pendulum model. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5442–5447, may 2011.
- [24] N. Perrin, O. Stasse, F. Lamiroux, and E. Yoshida. A biped walking pattern generator based on “half-steps” for dimensionality reduction. In *IEEE Int. Conf. on Robotics and Automation (ICRA'11)*, pages 1270–1275, 2011.
- [25] J. Pratt, J. Carff, S. Drakunov, and A. Goswami. Capture point: A step toward humanoid push recovery. In *Humanoid Robots, 2006 6th IEEE-RAS International Conference on*, pages 200–207, dec. 2006.
- [26] Jerry E. Pratt and Russ Tedrake. Velocity-based stability margins for fast bipedal walking. In *In Fast Motions in Biomechanics and Robotics*, pages 299–324. Springer, 2006.
- [27] Marc H. Raibert. *Legged robots that balance*. Massachusetts Institute of Technology, Cambridge, MA, USA, 1986.
- [28] J. Rummel, Y. Blum, and A. Seyfarth. Robust and efficient walking with spring-like legs. *Bioinspiration & Biomimetics*, 5(4), 2010.
- [29] H. Sang-Ho, J.G. Hale, and G. Cheng. Full-body compliant human-humanoid interaction: Balancing in the presence of unknown external forces. *Robotics, IEEE Transactions on*, 23(5):884–898, oct. 2007.
- [30] A. Sano and J. Furusho. Realization of natural dynamic walking using the angular momentum information. In *Robotics and Automation, 1990. ICRA 1990. IEEE International Conference on*, pages 1476–1481, 1990.
- [31] M.W. Spong. The swing up control problem for the acrobot. *Control Systems, IEEE*, 15(1):49–55, feb 1995.
- [32] Benjamin Stephens. Integral control of humanoid balance. In *Proceedings of the IEEE/RSJ 2007 International Conference on Intelligent Robots and Systems*, 2007.
- [33] B.J. Stephens and C.G. Atkeson. Dynamic balance force control for compliant humanoid robots. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1248–1255, oct. 2010.
- [34] B.J. Stephens and C.G. Atkeson. Push recovery by stepping for humanoid robots with force controlled joints. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 52–59, dec. 2010.
- [35] M. Stilman, J. Olson, and W. Gloss. Golem Krang: Dynamically stable humanoid robot for mobile manipulation. In *Robotics and Automation, 2010. ICRA 2010. IEEE International Conference on*, pages 3304–3309, 2010.
- [36] Nakamura Y. Sugihara, T. Gravity compensation on humanoid robot control with robust joint servo and non-integrated rate-gyroscope. In *Humanoid Robots (Humanoids), 2006 6th IEEE-RAS International Conference on*, pages 194–199, 2006.
- [37] N.G. Tsagarakis, Zhibin Li, J. Saglia, and D.G. Caldwell. The design of the lower body of the compliant humanoid robot cub. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2035–2040, 2011.
- [38] J. Urata, K. Nishiwaki, Y. Nakanishi, K. Okada, S. Kagami, and M. Inaba. Online walking pattern generation for push recovery and minimum delay to commanded change of direction and speed. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3411–3416.
- [39] M. Vukobratovic, A. A. Frank, and D. Juricic. On the stability of biped locomotion. *Biomedical Engineering, IEEE Transactions on*, BME-17(1):25–36, jan. 1970.
- [40] D. Winter. Human balance and posture control during standing and walking. *Gait & Posture*, 3(4):193–214, December 1995.