

Linear model predictive control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels

Jory Lafaye, David Gouaillier, Pierre-Brice Wieber

► **To cite this version:**

Jory Lafaye, David Gouaillier, Pierre-Brice Wieber. Linear model predictive control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels. IEEE-RAS International Conference on Humanoid Robots (Humanoids), Nov 2014, Madrid, Spain. pp.336-341, 10.1109/HUMANOID.2014.7041381 . hal-02487653

HAL Id: hal-02487653

<https://hal.inria.fr/hal-02487653>

Submitted on 21 Feb 2020

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.

Linear Model Predictive Control of the locomotion of Pepper, a humanoid robot with omnidirectional wheels

Jory Lafaye, *Aldebaran Robotics, jlafaye@aldebaran-robotics.com*
David Gouaillier, *Aldebaran Robotics, dgouaillier@aldebaran-robotics.com*
Pierre-Brice Wieber, *Inria Grenoble, pierre-brice.wieber@inria.fr*

Abstract—The goal of this paper is to present a new real-time controller based on linear model predictive control for an omnidirectional wheeled humanoid robot. It is able to control both the mobile base of the robot and its body, while taking into account dynamical constraints. It makes it possible to have high velocity and acceleration motions by predicting the dynamic behavior of the robot in the future. Experimental results are proposed on the robot Pepper made by Aldebaran Robotics, showing good performance in terms of robustness and tracking control, efficiently managing kinematic and dynamical constraints.

Index Terms—Humanoid Robotics, Mobile Robotics, Omnidirectional Wheeled Robot, Linear Model Predictive Control, Balance Control, Quadratic Programming, Aldebaran Robotics Pepper Robot

I. INTRODUCTION

Pepper (Fig. 1) is a humanoid robot standing on a holonomic mobile base, equipped with three omnidirectional wheels. It is 1.37 m high and weights 30 kg, with a 2 Degrees of Freedom (DoF) head, two 6 DoF arms, and a unique 3 DoF leg. It is developed by Aldebaran Robotics for the Japanese Telecommunication Company Soft Bank, to welcome customers in their shops. The mobile base is 42 cm long, with a triangular shape, and is able to move at a nominal velocity of $0.5 \text{ m}\cdot\text{s}^{-1}$ and acceleration of $1.0 \text{ m}\cdot\text{s}^{-2}$. A maximum velocity of $1.4 \text{ m}\cdot\text{s}^{-1}$ and acceleration of $1.7 \text{ m}\cdot\text{s}^{-2}$ can also be realized for short periods of time.

The wheels in the mobile base do not have dampers, and the normal forces between the wheels and the ground are strongly dependant on the position and acceleration of the body. For this reason, the mobile base suffers from slippage, a common problem for mobile robots. Moreover, due to the relatively high body with respect to the size of the support base, the robot can easily tilt and fall, a common problem for humanoid robots. As a result, this robot combines problems from mobile robots and humanoid robots.

Extensive work has been done on dynamically balancing robots on two wheels, similarly to inverted pendulums [1], [2], [3], but balancing a robot with three wheels or more is a significantly different problem. Concerning omnidirectional wheeled robots, extensive work has been done on trajectory tracking [4], [5], [6], also introducing Model Predictive Control (MPC) to deal with velocity and acceleration limits [7], but never considering dynamic balance, *i.e.*, the risk of tilting and falling.



Fig. 1. Pepper is a humanoid robot standing on a holonomic mobile base, equipped with three omnidirectional wheels. It is developed by Aldebaran Robotics for the Japanese Telecommunication Company Soft Bank, to welcome customers in their shops.

Avoiding tilting and falling has already been approached for wheeled robots with manipulator arms, focusing on the Center of Pressure (CoP), often called the Zero Moment Point (ZMP) [8]. In this case, the motion of the manipulator arm has been adapted to help maintaining balance, but not the motion of the wheeled base [9], [10], [11], [12]. This significantly limits the capacity of the robot to realize efficient dynamic motions, and react to perturbations.

Dynamic balance is the main problem of biped walking robots, and Model Predictive Control has proved to be a very efficient approach to avoid tilting and falling in this case [13], [14]. It is at the heart of the walking algorithms of our Nao robot [15]. In a way, the only difference between Pepper and a biped walking robot is that its “foot” (the mobile base) moves smoothly on the ground, depending on wheel motion, instead of having to regularly leave the ground to move from step to step.

The driving idea of this paper is to build on this small

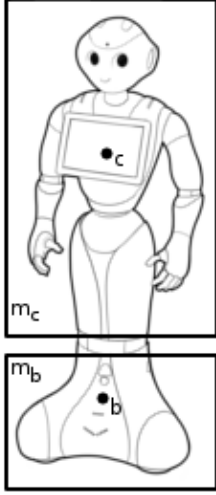


Fig. 2. We propose to model Pepper with two independent point masses, b and c for the mobile base and upper body, and consider the corresponding Center of Pressure p .

difference, and adapt control methods from biped walking robots, to humanoid robots with omnidirectional wheels, considering more precisely the MPC scheme with adaptive footsteps originally proposed in [14]. The advantage of such a scheme is that it can very easily manage a variety of constraints such as kinematic limits, maximum velocity and acceleration of the mobile base, dynamic balance of the CoP, and anticipate the future motions of the robot, gracefully and efficiently avoiding motions that could lead to tilting and falling.

The main difference with the MPC scheme originally proposed for biped walking will be related to the smooth motion of the foot/mobile base on the ground. Another important difference will be the need to consider two independent point masses, because of the specific mass distribution of Pepper. The proposed dynamical model will be presented in Section II, and the specific MPC design in Section III, with experimental validation discussed in Section IV.

II. DYNAMICAL MODEL OF PEPPER'S LOCOMOTION

A. Linear model with two independent point masses

Approximately half of the mass of Pepper rests in its mobile base, while the other half lies in its articulated upper body. The dynamical effects due to the rotations of the different parts of the robot, and those due to the motion of the arms, can usually be neglected. As a result, we propose to consider a two point-mass model (Fig. 2), with b the position of the Center of Mass (CoM) of the mobile base, and c the position of the CoM of the upper body.

The Newton and Euler equations for this two point mass model can be written:

$$m_b(\ddot{b} + \mathbf{g}) + m_c(\ddot{c} + \mathbf{g}) = F, \quad (1)$$

$$b \times m_b(\ddot{b} + \mathbf{g}) + c \times m_c(\ddot{c} + \mathbf{g}) = p \times F, \quad (2)$$

with F the contact force with the ground and p the corresponding Center of Pressure (CoP), m_b and m_c the masses

of the mobile base and upper-body.

For all the vectors in the equation, we associate the superscript x, y, z for respectively the two horizontal and the vertical axes. The gravity vector \mathbf{g} is simply $\mathbf{g}^x = \mathbf{g}^y = 0$, $\mathbf{g}^z = g$. If the robot moves on a horizontal ground, we can also consider that the height of the CoP is constant $p^z = 0$. In this case, the height of the mobile base is also constant: $b^z = l$. Neglecting the effect of the vertical motion of the upper body, we can also consider its height to be constant: $c^z = h$.

With these assumptions, the Newton and Euler equations can be combined and simplified to give an equation which is linear with respect to the motion of the two point masses:

$$b^{xy}m_bg - lm_b\ddot{b}^{xy} + c^{xy}m_cg - hm_c\ddot{c}^{xy} = p^{xy}(m_b + m_c)g, \quad (3)$$

which can be reorganized to give directly the position of the CoP as a linear function of the position and acceleration of the two point masses:

$$p^{xy} = \frac{b^{xy}m_bg + c^{xy}m_cg - \ddot{b}^{xy}m_b l - \ddot{c}^{xy}m_c h}{(m_b + m_c)g}, \quad (4)$$

Note that if these two point masses were at the same height ($l = h$), then this equation would be equivalent to that of a single point mass model:

$$p^{xy} = r^{xy} - \frac{h}{g}\ddot{r}^{xy}, \quad (5)$$

where $r^{xy} = (b^{xy}m_b + c^{xy}m_c)/(m_b + m_c)$ would be the overall CoM of the whole system. But since they are at different heights, this equation behaves significantly differently, reflecting the effect of the motion of the upper-body with respect to the mobile base on the balance of the robot.

B. Sampled dynamics and prediction

In order to generate smooth motions of the robot, with smooth contact force profiles, we consider that the two point masses follow a third order dynamics, with a sampling period T :

$$\hat{c}_k^{xy} = \begin{bmatrix} c_k^{xy} \\ \dot{c}_k^{xy} \\ \ddot{c}_k^{xy} \end{bmatrix} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \hat{c}_{k-1}^{xy} + \begin{bmatrix} \frac{T^3}{6} \\ \frac{T^2}{2} \\ T \end{bmatrix} \ddot{c}_{k-1}^{xy}, \quad (6)$$

the same for \hat{b}_k^{xy} . We will consider the motion of these two point masses over a horizon of N future samples. Introducing $C^{xy} = [c_1^{xy} \dots c_N^{xy}]^t$, the same for \dot{C}^{xy} , \ddot{C}^{xy} , B^{xy} , \dot{B}^{xy} , \ddot{B}^{xy} and P^{xy} , we can iterate the third order dynamics (6) to obtain

$$C^{xy} = U_c \ddot{C}^{xy} + S_c \hat{c}_0^{xy}, \quad (7)$$

$$\dot{C}^{xy} = U_{\dot{c}} \ddot{C}^{xy} + S_{\dot{c}} \hat{c}_0^{xy}, \quad (8)$$

$$\ddot{C}^{xy} = U_{\ddot{c}} \ddot{C}^{xy} + S_{\ddot{c}} \hat{c}_0^{xy}, \quad (9)$$

with

$$U_c = \begin{bmatrix} \frac{T^3}{6} & 0 & 0 \\ \vdots & \ddots & 0 \\ \frac{(1-3N+3N^2)T^3}{6} & \dots & \frac{T^3}{6} \end{bmatrix},$$

$$S_c = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ \vdots & \vdots & \vdots \\ 1 & NT & \frac{N^2 T^2}{2} \end{bmatrix}$$

$$U_{\dot{c}} = \begin{bmatrix} \frac{T^2}{2} & 0 & 0 \\ \vdots & \ddots & 0 \\ \frac{(2N-1)T^2}{2} & \dots & \frac{T^2}{2} \end{bmatrix}, S_{\dot{c}} = \begin{bmatrix} 0 & 1 & T \\ \vdots & \vdots & \vdots \\ 0 & 1 & NT \end{bmatrix},$$

$$U_{\ddot{c}} = \begin{bmatrix} T & 0 & 0 \\ \vdots & \ddots & 0 \\ T & \dots & T \end{bmatrix}, S_{\ddot{c}} = \begin{bmatrix} 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 1 \end{bmatrix}.$$

The same can be done for B^{xy} , \dot{B}^{xy} and \ddot{B}^{xy} as functions of \ddot{B}^{xy} and \hat{b}_0^{xy} . Note that the matrices U_c , $U_{\dot{c}}$ and $U_{\ddot{c}}$ are lower triangular and therefore easily invertible. Using the linear relation (4), we also obtain:

$$P^{xy} = U_{pb} \ddot{B}^{xy} + U_{pc} \ddot{C}^{xy} + S_{pb} \hat{b}_0^{xy} + S_{pc} \hat{c}_0^{xy} \quad (10)$$

with

$$U_{pb} = \frac{m_b(gU_b - lU_{\dot{b}})}{(m_b + m_c)g}, \quad U_{pc} = \frac{m_c(gU_c - hU_{\dot{c}})}{(m_b + m_c)g}, \quad (11)$$

$$S_{pb} = \frac{m_b(gS_b - lS_{\dot{b}})}{(m_b + m_c)g}, \quad S_{pc} = \frac{m_c(gS_c - hS_{\dot{c}})}{(m_b + m_c)g}. \quad (12)$$

III. CONTROL DESIGN

Our controller has to decide the third order derivatives of both the mobile base and the upper body, in both x and y directions. The working variable is therefore $X = [\ddot{C}^x \ \ddot{C}^y \ \ddot{B}^x \ \ddot{B}^y]^t$. The following control objectives and feasibility constraints will be expressed respectively as quadratic and linear functions of this variable X .

A. Control objectives

1) *Trajectory tracking*: In order for the mobile base of the robot to follow a given reference trajectory ($B_{\text{ref}}^{xy}, \dot{B}_{\text{ref}}^{xy}$), the first objectives to minimize are

$$O_1 = \frac{1}{2} \|\dot{B}^{xy} - \dot{B}_{\text{ref}}^{xy}\|^2 = \frac{1}{2} X^t Q_1 X + p_1^t X + \epsilon_1 \quad (13)$$

with

$$Q_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & U_b^t U_b & 0 \\ 0 & 0 & 0 & U_b^t U_b \end{bmatrix}, \quad p_1 = \begin{bmatrix} 0 \\ 0 \\ U_b^t (S_b \hat{b}_0^x - \dot{B}_{\text{ref}}^x) \\ U_b^t (S_b \hat{b}_0^y - \dot{B}_{\text{ref}}^y) \end{bmatrix}, \quad (14)$$

and

$$O_2 = \frac{1}{2} \|B^{xy} - B_{\text{ref}}^{xy}\|^2 = \frac{1}{2} X^t Q_2 X + p_2^t X + \epsilon_2 \quad (15)$$

with

$$Q_2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & U_b^t U_b & 0 \\ 0 & 0 & 0 & U_b^t U_b \end{bmatrix}, \quad p_2 = \begin{bmatrix} 0 \\ 0 \\ U_b^t (S_b \hat{b}_0^x - B_{\text{ref}}^x) \\ U_b^t (S_b \hat{b}_0^y - B_{\text{ref}}^y) \end{bmatrix}. \quad (16)$$

2) *Robustness to perturbations*: Without any precise model of potential perturbations applied on the robot, robustness can only be evaluated heuristically. Two important aspects of the motion of the robot that enable fast and efficient compensations of perturbations are centering of the CoP under the mobile base, and centering of the upper body CoM above the mobile base. This leads to minimizing the weighted sum

$$O_3 = \frac{\zeta_f}{2} \|P - B\|^2 + \frac{1 - \zeta_f}{2} \|C - B\|^2 \quad (17)$$

$$= \frac{1}{2} X^t Q_3 X + p_3^t X + \epsilon_3 \quad (18)$$

with

$$Q_3 = \zeta_f Q_{3p} + (1 - \zeta_f) Q_{3c}, \quad p_3 = \zeta_f p_{3p} + (1 - \zeta_f) p_{3c}, \quad (19)$$

$$Q_{3p} = \begin{bmatrix} U_{pc}^t U_{pc} & 0 & U_{pc}^t U_{pbb} & 0 \\ 0 & U_{pc}^t U_{pc} & 0 & U_{pc}^t U_{pbb} \\ U_{pbb}^t U_{pc} & 0 & U_{pbb}^t U_{pbb} & 0 \\ 0 & U_{pbb}^t U_{pc} & 0 & U_{pbb}^t U_{pbb} \end{bmatrix}, \quad (20)$$

$$p_{3p} = \begin{bmatrix} U_{pc}^t (S_{pbb} \hat{b}_0^x + S_{pc} \hat{c}_0^x) \\ U_{pc}^t (S_{pbb} \hat{b}_0^y + S_{pc} \hat{c}_0^y) \\ U_{pbb}^t (S_{pbb} \hat{b}_0^x + S_{pc} \hat{c}_0^x) \\ U_{pbb}^t (S_{pbb} \hat{b}_0^y + S_{pc} \hat{c}_0^y) \end{bmatrix}, \quad (21)$$

$$U_{pbb} = U_{pb} - U_b, \quad S_{pbb} = S_{pb} - S_b, \quad (22)$$

$$Q_{3c} = \begin{bmatrix} U_c^t U_c & 0 & -U_c^t U_b & 0 \\ 0 & U_c^t U_c & 0 & -U_c^t U_b \\ -U_b^t U_c & 0 & U_b^t U_b & 0 \\ 0 & -U_b^t U_c & 0 & U_b^t U_b \end{bmatrix}, \quad (23)$$

$$p_{3c} = \begin{bmatrix} U_c^t (S_c \hat{c}_0^x - S_b \hat{b}_0^x) \\ U_c^t (S_c \hat{c}_0^y - S_b \hat{b}_0^y) \\ U_b^t (S_c \hat{c}_0^x - S_b \hat{b}_0^x) \\ U_b^t (S_c \hat{c}_0^y - S_b \hat{b}_0^y) \end{bmatrix}. \quad (24)$$

3) *Motion smoothness*: In order to generate smooth motions, we propose to penalize the third derivative of the motion of both point masses, minimizing

$$O_4 = \frac{1}{2} \|X\|^2 = \frac{1}{2} X^t Q_4 X + p_4^t X + \epsilon_4 \quad (25)$$

with

$$Q_4 = I, \quad p_4 = 0. \quad (26)$$

B. Feasibility constraints

1) *Dynamic balance*: Dynamic balance of the motion of the robot imposes that the CoP p^{xy} stays within the triangular support polygon D (see Fig. 3). In order to have this feasibility constraint invariant with respect to the orientation of the robot, we would consider instead a circle $D' \subset D$ inscribed in the support triangle, but to keep this constraint linear, we propose in the end to consider an octagon $D'' \subset D'$ inscribed in this circle. We end up therefore with a dynamic balance constraint

$$P^{xy} - B^{xy} \in D'', \quad (27)$$

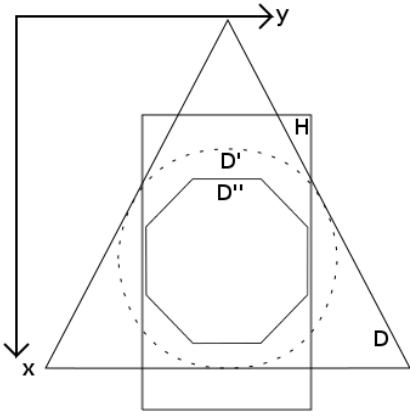


Fig. 3. Representation of the constraints. D is the real CoP constraint. D' is the circle inscribed in D . D'' is the octagonal constraint used in the controller to approximate the circle D' . H is the boundary limits of the upper body CoM position relative to the base.

what can be written as

$$v_1^- \leq V_1 X \leq v_1^+ \quad (28)$$

with

$$V_1 = \begin{bmatrix} U_{pc} & 0 & U_{pbb} & 0 \\ 0 & U_{pc} & 0 & U_{pbb} \\ U_{pc} & U_{pc} & U_{pbb} & U_{pbb} \\ U_{pc} & -U_{pc} & U_{pbb} & -U_{pbb} \end{bmatrix}, \quad (29)$$

$$v_1^- = \begin{bmatrix} -3\tilde{r} - S_{pc}\hat{c}_0^x - S_{pbb}\hat{b}_0^x \\ -3\tilde{r} - S_{pc}\hat{c}_0^y - S_{pbb}\hat{b}_0^y \\ -4\tilde{r} - S_{pc}(\hat{c}_0^x + \hat{c}_0^y) - S_{pbb}(\hat{b}_0^x + \hat{b}_0^y) \\ -4\tilde{r} - S_{pc}(\hat{c}_0^x - \hat{c}_0^y) - S_{pbb}(\hat{b}_0^x - \hat{b}_0^y) \end{bmatrix}, \quad (30)$$

$$v_1^+ = \begin{bmatrix} 3\tilde{r} - S_{pc}\hat{c}_0^x - S_{pbb}\hat{b}_0^x \\ 3\tilde{r} - S_{pc}\hat{c}_0^y - S_{pbb}\hat{b}_0^y \\ 4\tilde{r} - S_{pc}(\hat{c}_0^x + \hat{c}_0^y) - S_{pbb}(\hat{b}_0^x + \hat{b}_0^y) \\ 4\tilde{r} - S_{pc}(\hat{c}_0^x - \hat{c}_0^y) - S_{pbb}(\hat{b}_0^x - \hat{b}_0^y) \end{bmatrix}, \quad (31)$$

and $\tilde{r} = \frac{r}{\sqrt{10}}$ with r the radius of the circle \mathbb{D}' .

2) *Maximum velocity and acceleration*: Maximum velocity and acceleration of the mobile base can be taken into account in the following way:

$$-\dot{b}_{\max}^{xy} \leq \dot{B}^{xy} \leq \dot{b}_{\max}^{xy}, \quad (32)$$

$$-\ddot{b}_{\max}^{xy} \leq \ddot{B}^{xy} \leq \ddot{b}_{\max}^{xy}, \quad (33)$$

what can be written as

$$v_2^- \leq V_2 X \leq v_2^+ \quad (34)$$

with

$$V_2 = \begin{bmatrix} 0 & 0 & U_{\dot{b}} & 0 \\ 0 & 0 & 0 & U_{\dot{b}} \\ 0 & 0 & U_{\ddot{b}} & 0 \\ 0 & 0 & 0 & U_{\ddot{b}} \end{bmatrix}, \quad (35)$$

$$v_2^- = \begin{bmatrix} -\dot{b}_{\max} - S_{\dot{b}}\hat{b}_0^x \\ -\dot{b}_{\max} - S_{\dot{b}}\hat{b}_0^y \\ -\ddot{b}_{\max} - S_{\ddot{b}}\hat{b}_0^x \\ -\ddot{b}_{\max} - S_{\ddot{b}}\hat{b}_0^y \end{bmatrix}, \quad v_2^+ = \begin{bmatrix} \dot{b}_{\max} - S_{\dot{b}}\hat{b}_0^x \\ \dot{b}_{\max} - S_{\dot{b}}\hat{b}_0^y \\ \ddot{b}_{\max} - S_{\ddot{b}}\hat{b}_0^x \\ \ddot{b}_{\max} - S_{\ddot{b}}\hat{b}_0^y \end{bmatrix}. \quad (36)$$

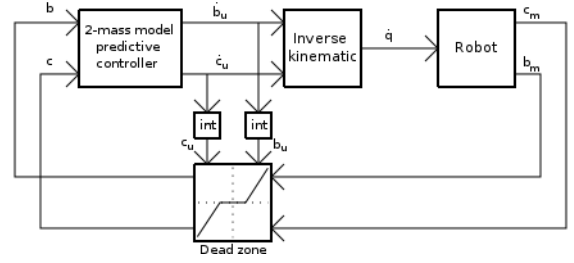


Fig. 4. Closed-loop control scheme

3) *Maximum reach of the upper body*: Due to joint limits, the CoM of the upper body of Pepper is limited to a rectangular zone H above the mobile base (see Fig. 3), what can be expressed as

$$-k^{xy} \leq C^{xy} - B^{xy} \leq k^{xy}, \quad (37)$$

what can be written as

$$v_3^- \leq V_3 X \leq v_3^+ \quad (38)$$

with

$$V_3 = \begin{bmatrix} U_c & 0 & -U_b & 0 \\ 0 & U_c & 0 & -U_b \end{bmatrix}, \quad (39)$$

$$v_3^- = \begin{bmatrix} -k^x - S_c\hat{c}_0^x + S_b\hat{b}_0^x \\ -k^y - S_c\hat{c}_0^y + S_b\hat{b}_0^y \end{bmatrix}, \quad v_3^+ = \begin{bmatrix} k^x - S_c\hat{c}_0^x + S_b\hat{b}_0^x \\ k^y - S_c\hat{c}_0^y + S_b\hat{b}_0^y \end{bmatrix}. \quad (40)$$

C. Resulting Quadratic Program

All the objectives and constraints presented in this Section can be gathered in a single Quadratic Program (QP), which can be solved efficiently [16]:

$$\min_X \alpha_1 O_1 + \alpha_2 O_2 + \alpha_3 O_3 + \alpha_4 O_4 \quad (41)$$

$$\text{s.t. } v_1^- \leq V_1 X \leq v_1^+ \quad (42)$$

$$v_2^- \leq V_2 X \leq v_2^+ \quad (43)$$

$$v_3^- \leq V_3 X \leq v_3^+ \quad (44)$$

Here, the different objectives O_i are weighted with parameters α_i . Different choices for these parameters will naturally lead to different behaviors of the robot. They must be chosen therefore according to the behavior that is preferred. The effect of the weights α_1 , α_2 and α_4 , given to trajectory tracking and motion smoothness, are straightforward to analyze. Particularly interesting is when a large weight α_3 is given to the robustness objective. With $\zeta_f = 0$, this results in straightening the upper-body above the mobile base, what can be appealing visually. With ζ_f closer to 1, the behavior of the robot will be closer to the one of an inverted pendulum: when asked to go forward, the mobile base will start backwards for a brief moment before actually moving forward.

IV. EXPERIMENTS

The MPC scheme proposed in the previous Section is applied now to Pepper (Fig. 4). The sampling period is $T = 100$ ms, and the preview horizon is 2 s, equivalent to $N = 20$ sampling periods. As a result, the vector X is of size 80. At each sampling period, the current states \hat{b}_0^{xy} and \hat{c}_0^{xy} of the two point masses are estimated, based on wheel and joint position sensors (the wheel material has been chosen carefully to minimize slippage, giving a good estimation of the mobile base motion in most situations). The QP (41)-(44) is solved, and the computed control is applied to the robot until the next sampling period, following a standard MPC approach. Since the wheels and joints of the robot are controlled in velocity, the third-order derivative obtained from the QP is converted into a velocity using the state equation (6).

In order to compare different control design options, four tests are performed with the same experimental scenario: tracking a reference trajectory which goes forward for a short amount of time and then makes a sharp 90 degree change in the direction of motion. This sharp change is technically infeasible by the robot, and we will see how the different design options balance the various objectives and feasibility constraints. The standard choice for the weights is to emphasize the trajectory tracking objective, then robustness, and consider smoothness only in the end: $\alpha_1, \alpha_2 > \alpha_3 > \alpha_4$

We can see in Fig. 5 that in this situation, the proposed MPC scheme clearly and efficiently anticipates the sharp turn: first of all, the mobile base starts turning slightly in advance in order to minimize the long term tracking error. But more interestingly, we can see that the upper body leans toward the inside of the turn, resulting in a CoP which is kept safely away from the boundary of the support polygon.

In case there is no feedback of the current state of the robot, the MPC scheme still manages to drive the robot correctly, but obviously, there will be some drift from the reference trajectory, as can be seen in Fig. 8, and no reaction to unexpected perturbations. We can observe also that the resulting motion appears to be slightly less smooth.

In case a model with a single point mass is used, we can see in Fig. 7 that the mobile base continues to track the reference trajectory adequately, but the upper body doesn't contribute anymore to anticipating the sharp turn. As a result, the CoP wanders closer to the boundary of the support polygon, reducing the robustness of the motion to perturbations. This is a serious problem in case of motions with high speed, where the model with a single point mass quickly fails maintaining the balance of the robot, while the proposed model with two point masses works very well in coordinating efficiently the motions of the mobile base and upper body.

We can see in Fig. 6 the result of increasing the weight α_3 of the robustness objective by a huge factor ($5 \cdot 10^4$): the CoP is kept even closer to the center of the support polygon, but at the expense of an increased tracking error of the reference trajectory. The compromise between the different

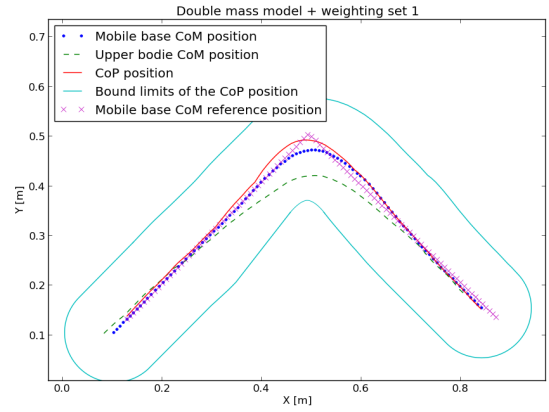


Fig. 5. Tracking control experiment: Closed-loop 2-mass model and weighting set 1 ($\alpha_1 = \alpha_2 = 10, \alpha_3 = 1, \alpha_4 = 10^{-5}$)

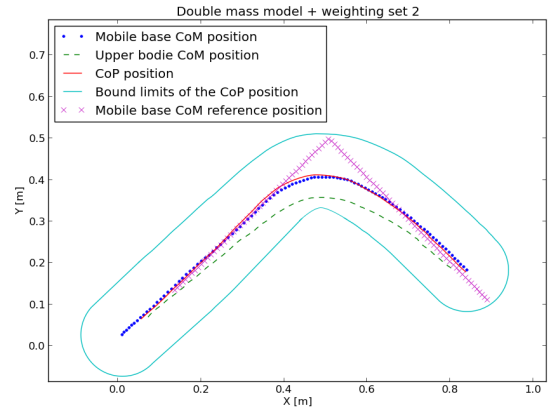


Fig. 6. Tracking control experiment: Closed-loop 2-mass model and weighting set 2 ($\alpha_1 = \alpha_2 = 10, \alpha_3 = 5 \cdot 10^4, \alpha_4 = 10^{-5}$)

control objectives appears very clearly here, and reflects the choice of the different weights α_i in a very straightforward way.

Finally, the Fig. 9 shows the tracking of a more complex trajectory. We can see that for each curve of the reference trajectory, the upper body is always leaning inside of the curve. This is due to the predictive effect of the controller, which lean the upper body in advance to improve the tracking control while preserving the CoP inside the support polygon. Also, we can note that during each turn, the CoP is about to reach its constraint. So, if we increase the tracking control weights (α_1 and α_2), this will lead to lean the upper body even more. The limit here is when the upper body reaches its constraint relative to the base position. Once it is reached, no more improvements in the tracking control can be performed.

V. CONCLUSION

In this paper, we have presented a new linear Model Predictive Control scheme for a humanoid robot with omnidirectional wheels, directly inspired by the MPC schemes being used for biped walking robots.

The advantage of this control scheme is that it can very easily manage a variety of constraints such as kinematic limits, maximum velocity and acceleration of the mobile

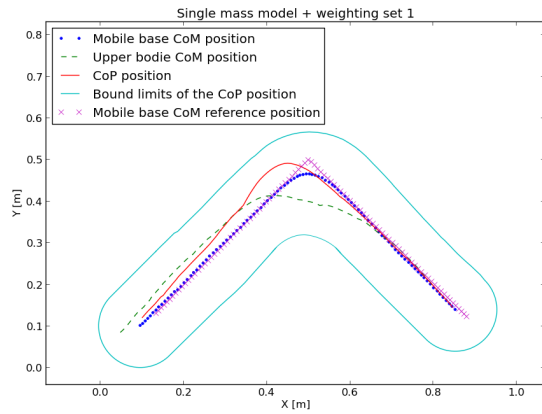


Fig. 7. Tracking control experiment: Closed-loop 1-mass model and weighting set 1 ($\alpha_1 = \alpha_2 = 10$, $\alpha_3 = 1$, $\alpha_4 = 10^{-5}$)

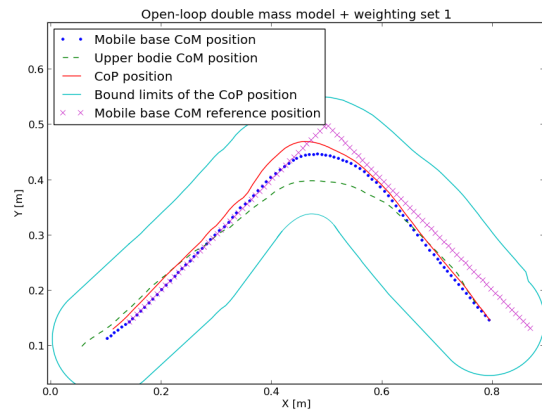


Fig. 8. Tracking control experiment: Open-loop 2-mass model and weighting set 1 ($\alpha_1 = \alpha_2 = 10$, $\alpha_3 = 1$, $\alpha_4 = 10^{-5}$)

base, dynamic balance of the CoP, and gracefully avoid motions that could lead to tilting and falling, while considering a variety of control objectives such as trajectory tracking, robustness to perturbations, or smoothness of the motion, all in a very straightforward way.

It allows Pepper to move dynamically with high speed and acceleration, without risking to tilt and fall in most normal situations. Interesting improvements in the future are to take into account the dynamical effects due to the motion of the arms, and focus on situations where the robot has been pushed strongly enough that it begins tilting, usually resting only on two wheels, and try to recover balance from such extreme situations.

REFERENCES

- [1] K. Pathak, J. Franch, and S. K. Agrawal, "Velocity and position control of a wheeled inverted pendulum by partial feedback linearization." *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 505–513, 2005.
- [2] B. Mahler and J. Haase, "Mathematical model and control strategy of a two-wheeled self-balancing robot," in *39th Annual Conference of the IEEE Industrial Electronics Society, IECON 2013*, 2013, pp. 4198–4203.
- [3] J. Wu, "A robust control method of two-wheeled self-balancing robot," in *2011 6th International Forum on Strategic Technology (IFOST)*, 2011, pp. 1031–1035.
- [4] K. Watanabe, "Control of an omnidirectional mobile robot." in *KES (1)*. IEEE, 1998, pp. 51–60.

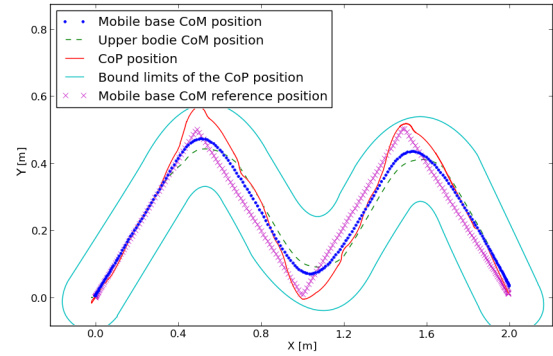


Fig. 9. Tracking control experiment on a more complex trajectory, using a closed-loop 2-mass model and the weighting set 1

- [5] Y. Liu, X. Wu, J. Zhu, and J. Lew, "Omni-directional mobile robot controller design by trajectory linearization," in *Proceedings of the American Control Conference*, vol. 4, 2003, pp. 3423–3428.
- [6] T. Kalmr-nagy and P. Ganguly, "Near-optimal dynamic trajectory generation and control of an omnidirectional vehicle," *Robotics and Autonomous Systems*, vol. 46, pp. 47–64, 2004.
- [7] K. Kanjanawanishkul and A. Zell, "Path following for an omnidirectional mobile robot based on model predictive control," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA 2009)*, Kobe, Japan, May 12-17 2009, pp. 3341–3346.
- [8] P. Sardain and G. Bessonnet, "Forces acting on a biped robot. Center of Pressure Zero Moment Point," *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, vol. 34, no. 5, pp. 630–637, september 2004.
- [9] S. A. A. Moosavian and K. Alipour, "On the dynamic tip-over stability of wheeled mobile manipulators," *Int. J. Robot. Autom.*, vol. 22, no. 4, pp. 322–328, 2007.
- [10] Q. Huang, S. Sugano, and K. Tanie, "Stability compensation of a mobile manipulator by manipulator motion: Feasibility and planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1997, pp. 1285–1292.
- [11] Y. Li, D. Tan, Z. Wu, and H. Zhong, "Dynamic stability analyses based on zmp of a wheel-based humanoid robot," in *International Conference on Robotics and Biomimetics*, Kunming, China, 2006.
- [12] K. Mingeuk, C. Dongil, and O. Jun-Ho, "Stabilization of a rapid four-wheeled mobile platform using the zmp stabilization method," in *International Conference on Advanced Intelligent Mechatronics*, Montréal, Canada, 2010.
- [13] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, and H. Hirukawa, "Biped walking pattern generation by using preview control of Zero Moment Point," in *Proceedings of the IEEE International Conference on Robotics & Automation*, september 2003, pp. 1620–1626.
- [14] A. Herdt, H. Diedam, P.-B. Wieber, D. Dimitrov, K. Mombaur, and M. Diehl, "Online walking motion generation with automatic foot step placement," *Advanced Robotics*, vol. 24, no. 5-6, pp. 719–737, 2010.
- [15] D. Gouaillier, C. Collette, and C. Kilner, "Omni-directional closed-loop walk for NAO," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2010.
- [16] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York: Springer, 2006.