HAL
open science

Cancelling the sway motion of dynamic walking in visual servoing

Claire Dune, Andrei Herdt, Olivier Stasse, Pierre-Brice Wieber, Kazuhito Yokoi, Eiichi Yoshida

To cite this version:

Claire Dune, Andrei Herdt, Olivier Stasse, Pierre-Brice Wieber, Kazuhito Yokoi, et al.. Cancelling the sway motion of dynamic walking in visual servoing. IROS 2010 - IEEE-RSJ International Conference on Intelligent Robots & Systems, Oct 2010, Taipei, Taiwan. pp.3175-3180, 10.1109/IROS.2010.5649126 . inria-00567664

HAL Id: inria-00567664

https://inria.hal.science/inria-00567664v1

Submitted on 21 Feb 2011

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.

Cancelling the sway motion of dynamic walking in visual servoing

C. Dune, A. Herdt, O. Stasse, P.-B. Wieber, K. Yokoi, E. Yoshida

***Abstract*— This paper introduces a visual servoing scheme for humanoid walking. Though most of the existing approaches follow a perception-decision-action scheme, we close the loop so that the control is robust to model error. Our approach is based on a new reactive pattern generator which modifies, at the control level, the footsteps, the center of mass and the center of pressure trajectories for the center of mass to track a reference velocity. And, in this paper, the reference velocity is directly given by a visual servoing control law. Since, the HRP-2 walk induces a sway motion that disturbs the regulation of the visual control law, we introduce a control law allowing convergence in the image space and taking into account this sway motion.**

I. INTRODUCTION

Humanoid robots are designed for human environments, defined as unstructured and dynamic environments [1] where objects move outside robots'control. In order to complete a specific task, humanoid robots must perceive and react to environmental changes. This paper focuses on positioning tasks where the robot has typically to reach a piece of furniture or go through a door.

Most of the humanoid robots are equipped with cameras that provide rich information without adding so much weight and size. The use of embedded camera is attractive because it avoids equipping the environment with additional sensors, and thus the system is more autonomous. Yet, extracting data from these cameras is a real challenge, especially while walking. In this paper, we introduce a eye-in-hand visual servoing scheme to control the HRP-2 walking with taking into account the peculiar motion of the on-board camera induced by the stepping.

Previous works on humanoid walking control assume that the robot path is defined before computing the actual joint control to realize it. They generally follow a perception-decision-action scheme: first, a sensor acquires data on the world and/or the robot state, then, suitable footsteps over a time horizon are decided, and the trajectories of the center of mass (CoM) and the center of pressure (CoP) are computed while respecting the stability constraints. Finally, the control of the legs is computed by inverse kinematics. This perception-decision-action loop has proven to be fast enough to realize impressive demonstrations for stair-climbing and obstacle avoidance [2], [3], [4], [5]. Yet, these methods have two drawbacks:

1) they rely on the object pose estimation that is sensitive to noise (calibration, image processing, optimisation)
2) once the footsteps are decided inside the window of the model predictive control we cannot make any modification.

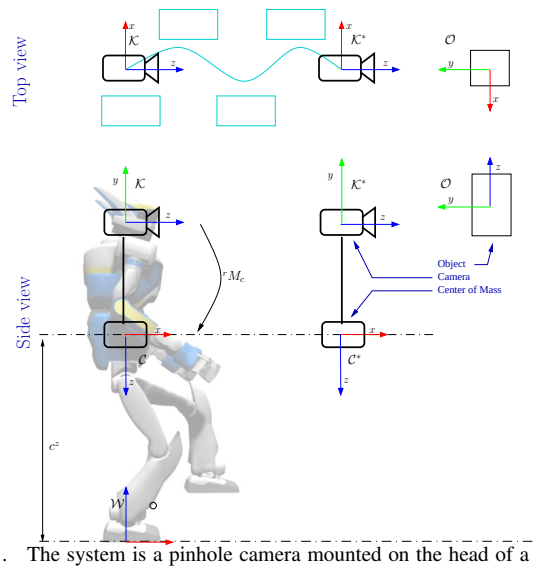

Fig. 1. The system is a pinhole camera mounted on the head of a walking robot

To cope with the pose estimation issue 1), we propose a visual servoing control scheme which is robust to model errors. Visual servoing proved to be successful for grasping tasks with standing [6], [7] or walking humanoids [8], [9]. In [8], visual servoing is used to control a humanoid avatar along landmarks. The upper body is approximated by the kinematic chain that links an on-board camera to the CoM. The lower body is controlled by adding two translational degrees of freedom to the CoM. The translational velocity of the CoM is sent to a kinematic locomotion module which control the legs motion. In [9] a whole body visual servoing scheme based on a hierarchical stack of task is introduced. However, the footsteps are predefined. The leg motion is thus set to be the task of higher priority. Therefore visual-servoing in this context is projected in the null-space of the pre-defined walking path. On the contrary in this work, the controller driving the walk is directly guided by vision.

Regarding the second issue 2), few works deal with footsteps, CoM and CoP trajectories modification inside the preview window. The work presented in [10] shades some light on this problem. It shows that modifying the next landing position of the flying foot might impose a new CoP trajectory going out of the support polygon. This can jeopardize the equilibrium of the robot. To solve the problem, the stepping period may be modified to reduce this instability [10], at the cost of slowing down the robot. A recent method propose to modify the foot steps according to a perturbation applied to the CoP [11]. In the current paper the desired velocity computed by a visual servoing based controller is directly used to change footsteps, while

ensuring walking stability constraints and with time intervals of constant length. Another difference lays on the fact that, on the one hand, the CoP is constrained at the center of the footprints, and on the other hand the CoP can move freely inside the support polygon.

Our approach is based on a new pattern generator (PG) that has been proposed by Herdt et al [12], [13]. It computes a reactive stable walking motion for the CoM to track an instant reference velocity without predefined footsteps. It paves the way to reactive walking motion based on current environmental perception. The main contribution of this paper is the introduction of a visual-servoing scheme *modifying* on-line the footsteps *and* the CoM-CoP trajectory based on [13].

Section II briefly presents the reactive pattern generator. Section III proposes to compute the reference velocity with a classical visual servoing control scheme as proposed in [8]. The HRP-2 walking motion induces an additional sway motion that disturbs the visual servoing regulation but has not to be regulated. In Section III-B.2 we propose a visual control that takes into account this disturbance in the visual error computation to obtain an exponential decrease of the visual error.

II. PREDICTIVE CONTROL SCHEME FOR REACTIVE MOTION

This section presents the on-line walking motion generator proposed in [12], [13]. The robot is modelled as a linear inverse pendulum which fits fairly well with the HRP-2 distribution of mass. The control is based on a Linear Model Predictive Control scheme that computes the footsteps and the optimal jerk of the point mass model to minimise the difference between a reference CoM velocity and the previewed one. In the next Section, the reference velocity is a visual servoing control law.

A. Systems Dynamics

The humanoid robot is modelled as a oriented mass point centred on the robot CoM. This paragraph describes the dynamics of a stable walking motion.

1) Motion of the Center of Mass: Let us consider a frame $\mathcal{C}$ attached to the position of the CoM of the robot and to the orientation of its trunk. The position and orientation of this frame will be noted $c = \begin{bmatrix} c^x & c^y & c^z & c^\varphi & c^\psi & c^\theta \end{bmatrix}$, with Cardan angles c^φ, c^ψ and c^θ.

The acceleration $\ddot{c}$ of this frame has to be continuous for being realized properly by usual actuators. We will consider here that it is in fact piecewise linear on time intervals of constant length τ, with a piecewise constant jerk $\dddot{c}$ (third derivative of the position) on these intervals. The trajectory of this frame over longer time intervals of length $n\tau$ can be obtained simply by integrating over time the piecewise constant jerk together with the initial speed $\dot{c}$ and acceleration $\ddot{c}$. For any coordinate $\alpha \in \{x, y, z, \varphi, \psi, \theta\}$, this leads to simple linear relationships (details on matrices $U_.$, $S_.$, $Z_.$ can be found in [12])

$$C^\alpha_{i+1} = S_p c^{\hat{\alpha}_i} + U_p \dddot{C}^\alpha_i, \tag{1}$$

$$\dot{C}^\alpha_{i+1} = S_v \hat{c}^\alpha_i + U_v \dddot{C}^\alpha_i, \tag{2}$$

$$\ddot{C}^\alpha_{i+1} = S_a \hat{c}^\alpha_i + U_a \dddot{C}^\alpha_i, \tag{3}$$

where the initial state is $\hat{c}^\alpha_i = \begin{bmatrix} c^\alpha(t_i) & \dot{c}^\alpha(t_i) & \ddot{c}^\alpha(t_i) \end{bmatrix}^T$. and the state on the prediction horizon is

$$C^\alpha_{i+1} = \begin{bmatrix} c^\alpha(t_{i+1}) \\ \vdots \\ c^\alpha(t_{i+n}) \end{bmatrix}, \dots \ \dddot{C}^\alpha_{i+1} = \begin{bmatrix} \dddot{c}^\alpha(t_{i+1}) \\ \vdots \\ \dddot{c}^\alpha(t_{i+n}) \end{bmatrix}$$

2) Motion of the Center of Pressure: The position z of the Center of Pressure (CoP) on the ground can be approximated by considering only the inertial effects due to the translation of the CoM, neglecting the other effects due to the rotations of the different parts of the robot. This proves to be a very effective approximation, which leads to the simple relationships

$$z^x_i = c^x_i - (c^z_i - z^z_i)\ddot{c}^x_i/g, \text{ and } z^y_i = c^y_i - (c^z_i - z^z_i)\ddot{c}^y_i/g,$$

where the difference $c^z_i - z^z_i$ corresponds to the height of the CoM above the ground, and g is the norm of the gravity force. We will consider here only the simple case where the height of the CoM above the ground is constant. In that case, we can obtain a relationship similar to (1)-(3):

$$Z^x_{i+1} = S_z \hat{c}^x_i + U_z \dddot{C}^x_i \text{ and } Z^y_{i+1} = S_z \hat{c}^y_i + U_z \dddot{C}^y_i,$$
$$\text{with } S_z = S_p - (c^z_i - z^z_i) S_a/g,$$
$$U_z = U_p - (c^z_i - z^z_i) U_a/g.$$

3) Foot step generation: Basically, humanoid nominal walking cycle can be divided into two stages: a double support phase, where the two feet are on the ground and single support phase, where only one foot is firmly on the ground on the other one is flying from its previous position to the next one. In this paper the stepping period is set to be $800ms$ with a double support phase of $100ms$ and single support phase of $700ms$.

The new pattern generator selects on-line the feasible footsteps F_{i+1} on the preview window with regards to the robot mechanical properties [14]. The position of the footsteps is then used twice : first to ensure the stability constraints on the CoP trajectory and secondly it is included in the cost function to attract the CoP trajectory towards the center of the polygon of support.

B. Constraints definition

To be stable, the dynamics control of the walking motion must comply with stability constraints that are listed in this paragraph.

1) Constraints on the CoP: Since the feet of the robot can only push on the ground, the CoP can lie only within the support polygon, that is the convex hull of the contact points between the feet and the ground [15]. Any trajectory not satisfying this constraint cannot be realized properly. This needs to be taken into account when computing a walking

motion with the MPC scheme (4). The foot on the ground is assumed to have a polygonal shape, so that this constraint can be expressed as a set of constraints on the position of the CoP which are linear with respect to the position of the foot on the ground but nonlinear with respect to its orientation.

2) Constraints on the foot placement: We need to assure that the footsteps decided by the above algorithm are feasible with respect to maximum leg length, joint limits, self-collision avoidance, maximum joint speed and similar geometric and kinematic limitations. In order to keep the Linear MPC structure of the algorithm, simple approximations of all these limitations are expressed in the form of linear constraints defined in [14].

C. Following a reference velocity

This Section set the optimisation problem to solve to ensure that the CoM velocity tracks a reference velocity. In order to keep the constraints linear, the optimisation is split in two steps : first, translations are treated, then rotations along the vertical axis are considered.

1) Translational velocity: It has been proposed in [12] to generate walking motions by directly following a reference speed $\dot{C}^*$. Only horizontal translations were considered. Secondary objectives were also introduced to help obtaining a more satisfying behaviour: centring the position of the feet with respect to the position of the CoP, and minimizing the jerk $\dddot{c}(t)$ to slightly smooth the resulting trajectory.

$$\begin{aligned}\min\ & \frac{\alpha}{2}\left\|\dot{C}^x_{i+1}-\dot{C}^{x,*}_{i+1}\right\|^2+\frac{\alpha}{2}\left\|\dot{C}^y_{i+1}-\dot{C}^{y,*}_{i+1}\right\|^2\\ &+\frac{\beta}{2}\left\|\bar{C}^x_{i+1}-\dot{C}^{x,*}_{i+1}\right\|^2+\frac{\beta}{2}\left\|\bar{C}^y_{i+1}-\dot{C}^{y,*}_{i+1}\right\|^2\\ &+\frac{\gamma}{2}\left\|F^x_{i+1}-Z^x_{i+1}\right\|^2+\frac{\gamma}{2}\left\|F^y_{i+1}-Z^y_{i+1}\right\|^2\\ &+\frac{\varepsilon}{2}\left\|\dddot{C}^x_i\right\|^2+\frac{\varepsilon}{2}\left\|\dddot{C}^y_i\right\|^2 \end{aligned} \tag{4}$$

with $\bar{C}$ the mean speed of the CoM over two steps. Introducing the vector $u_i = \begin{bmatrix}\dddot{C}^x_i & F^x_{i+1} & \dddot{C}^y_i & F^y_{i+1}\end{bmatrix}$ of motion parameters which automatically computed, this optimization problem can be expressed as a canonical Quadratic Program with the aforementioned constraints [12].

2) Following a reference rotational velocity: If the robot trunk has to rotate, then the orientations of the feet have to be adapted properly. Introducing θ as a variable in II-B.2 though would result in non-linear constraints. In order to keep the linear form Herdt et al choose to predetermine the orientation of the feet before solving the translational Quadratic Program.

To increase the robustness of trunk rotational motion, the feet orientations have to be aligned with the trunk orientation. Furthermore, feet and trunk acceleration and velocity have to be limited to avoid infeasible trajectories. This leads to the formulation of a decoupled Quadratic Program:

$$\min_{u_i^\theta}\quad \tfrac{\delta}{2}||C^\theta_{i+1}-F^\theta_{i+1}||^2+\tfrac{\epsilon}{2}||\dot{C}^\theta_{i+1}-\dot{C}^{\theta,*}_{i+1}||^2 \tag{5}$$

$$s.t.\qquad \dot{F}^{\theta,s}_{i+1}=0 \tag{6}$$

$$||F^{\theta,r}_{i+1}-F^{\theta,l}_{i+1}||<\theta^{rl}_{max} \tag{7}$$

$$||F^\theta_{i+1}-C^\theta_{i+1}||<\theta^{FT}_{max} \tag{8}$$

$$||\dot{F}^\theta_{i+1}-\dot{C}^\theta_{i+1}||<\dot{\theta}^{FT}_{max} \tag{9}$$

$$||\ddot{F}^\theta_{i+1}-\ddot{C}^\theta_{i+1}||<\ddot{\theta}^{FT}_{max}, \tag{10}$$

The two terms of the above objective assure that the trunk follows the desired rotational velocity and at the same time the feet are aligned with the trunk as much as possible. The constraints assure the feasibility of the desired motions.

III. VISUAL SERVOING FOR WALKING

In this section, a visual servoing scheme is introduced to compute the reference velocity that is given as a reference to the reactive PG. The system we consider is an on-board camera rigidly linked to the robot CoM (Fig. 1). Let $\mathcal{C}$ and $\mathcal{K}$ be the frame attached to the CoM and the camera and $\dot{c}$ and $\dot{k}$ their velocities.

A. Visual Servoing

Visual servoing of a system equipped with a camera regulates to zeros the error vector $e = s - s^*$ between some current visual features s and some desired visual features s^* [16]. The key feature in this control scheme is the *interaction matrix* L which links the time variation of the visual features $\dot{s}$ to the relative camera/object kinematics screw $\dot{k}$. It is defined by

$$\dot{s} = L\dot{k} \tag{11}$$

Then, the classical control law that regulates e with an exponential decrease $\dot{e} = -\lambda e$ is [16]:

$$\dot{k} = -\lambda \widehat{L}^+ e \tag{12}$$

where $\widehat{L}^+$ denotes the Moore-Penrose pseudo inverse of an approximation or a model of L, and λ is a positive scalar value.

B. Control of the pattern generator

In order to compute a proper control law for the walk, we have to understand the over-all behaviour of the pattern generator. The PG ensures that the CoM tracks a reference velocity yet on average and in the limit of the dimension of the robot (length of legs, actuator torque limit, etc.). In this section, we will describe these two aspects of the PG (see Fig. 2).

1) Limiting the velocity: In order to ensure the tracking of the reference velocity, the three velocity components have to be limited to feasible ones, i.e. velocities that respect the walking constraints which mainly depends on the robot geometry and actuators capabilities. It can be shown that the maximum speed for the HRP-2 robot is : $\dot{c}_{limit} = \begin{pmatrix}0.35 & 0.2 & 0.15\end{pmatrix}$ for the considered PG [13]. Yet

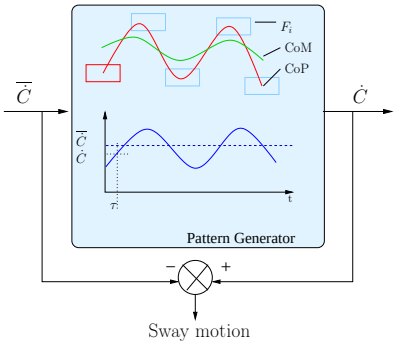

Fig. 2. The pattern generator ensures that the input velocity is tracked on average on the preview horizon. The output of the Model Predictive Control is the first control computed on the preview horizon. The difference between the reference velocity and the real velocity is mostly a sway motion due to the stepping.

this velocity cannot be reached instantly from a stationary position.

We can distinguish three phases in the walking motion: i) an initial state ($40ms$) where the robot is standing in double support, i.e. the two feet are on the ground and the robot stands still, ii) a nominal walking phase with a constant period of $\tau_{step} = 80ms$ and iii) a final phase where the robot stands in double support. To switch from the initial double support state to the nominal walking phase, there is a transitory phase during which the CoP is brought from the center of the two feet to the center of the left foot. Then the robot starts walking and the velocity of the CoM increases gradually during the first steps to reach a steady state where the reference velocity can be tracked up to $\dot{c}$. We then set a transient maximum velocity for the two first steps. The maximum velocity is then:

$$\begin{cases} \dot{c}_{max} = \dfrac{t}{2\tau_{step}}\dot{c}_{limit} \text{ if } t < 2\tau_{step} \\ \dot{c}_{max} = \dot{c}_{limit} \end{cases} \tag{13}$$

2) Cancelling the Sway Motion: In most of the existing PG, the stepping motion induces a lateral sway motion that prevents the CoM velocity from following instantaneously the expected one. The sway motion is mandatory for a proper walk and the control law should not compensate for it but cancels its effects on the visual error computation.

Let us define $\dot{b}$ the additional sway motion of period $T = \tau_{step}/\tau$, such that $\sum_{l=i}^{i+T} \dot{b}_l = 0$ (see Fig 1). The behaviour of the PG can be approximated by $\dot{c} = \overline{\dot{c}} + \dot{b}_c$ where $\overline{\dot{c}}$ is the velocity if there were no sway. It induces a motion of the camera of $\dot{k} = \overline{\dot{k}} + {}^kV_c\dot{b}_c$, where ${}^cV_k$ is the twist matrix associated to the cam-com transform ${}^cM_k$. If we assume ${}^cM_k$ to be constant over the time [1] the camera velocity can be written $\dot{k} = \overline{\dot{k}} + \dot{b}_k$ The features will then oscillate in the image and the feature variation can be written:

$$\dot{e} = \dot{s} = L\overline{\dot{k}} + L\dot{b}_k \tag{14}$$

[1]In this paper, we assume that the position of the camera is fixed with regards to the CoM. Yet, the upper body and the pan/tilt camera could be used as redundant degrees of freedom to regulate the control law. This will be investigated in a future work.

Let us define a virtual camera (Fig. 3) $\overline{\mathcal{K}}$ that corresponds to the position of the on-board camera if there was no sway motion. The velocity of this virtual camera is $\overline{\dot{k}}$, it is actually exactly the velocity that is sent as input in the reactive PG. Its value is given in(18). In order to compute a control law that does not include the sway motion, we will servo this virtual camera $s(\overline{k})$ to $s(\overline{k}^*)$.

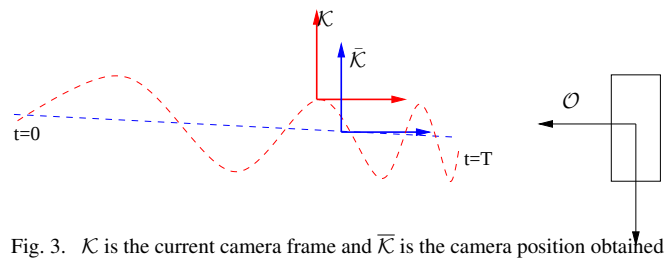

Fig. 3. $\mathcal{K}$ is the current camera frame and $\overline{\mathcal{K}}$ is the camera position obtained if the visual servoing velocity is applied without the walking constraints.

We have now to express $\overline{s} = s(\overline{k})$ with regards to the current measurement $s = s(k)$. With (11) we can write:

$$s(t) - s(0) = \int_0^t L\dot{k}dt = \int_0^t L(\overline{\dot{k}} + \dot{b}_k)dt \tag{15}$$

$$\text{and } \overline{s}(t) - \overline{s}(0) = \int_0^t L\overline{\dot{k}}dt \tag{16}$$

Then assuming that $s(0) = \overline{s}(0)$ and using (15) and (16) we obtain $s(t) = \overline{s}(t) + \int_0^t L\dot{b}_k dt$, from which we can deduce the corrected visual error

$$\overline{e}(t) = \overline{s}(t) - s^* = e(t) - \int_0^t L\dot{b}_k dt \tag{17}$$

Notice that when $\overline{e} \longrightarrow 0$ then $e \longrightarrow \int_0^t L\dot{b}_k$. In this study, we do not expect e to converge to zero but to oscillate around zero with a period T due to the sway motion. The convergence of the control law is then reached when $\int_{t-T}^t e dt = 0$, which is obtained if $\int_{t-T}^t \int_0^t L\dot{b}_k dt = 0$. Let us define $E = \int_{t-T}^t \int_0^t L\dot{b}_k dt$ and note that in general $E \neq 0$. It can be estimated over one period of time T. We can then use a sliding windows to define the current error $\overline{e} = e - (\int_0^t L\dot{b}_k - E)$ and deduce the control law

$$\overline{\dot{k}} = -\lambda L^+(e - (\int_0^t L\dot{b}_k - E)) \tag{18}$$

IV. SIMULATION RESULTS

3) Simulation set up and assumptions: Fig. 1 presents the experimental set up : the robot starts from an arbitrary position and has to reach a position where the current image features match the desired one.

In these simulations we have arbitrarily chosen to use a five dot planar target that is vertically set, the purpose of this paper beeing to show that the sway motion can be compensate for any visual information, The interaction matrix L corresponding to the projection of the target is a stack of the five interaction matrices corresponding to each point. The interaction matrix for a point is defined

in [16][2]. In order to compute the interaction matrix L, we need an estimated depth of the point or to set it arbitrarily. In the context of these examples, we assume the depth to be estimated at each iteration, *e*g by estimating the target pose.

We assume that the camera has an infinite image plane, which means that the features are visible if the object is in front of the image plane. For real applications, this assumption does not hold. An additional degree of freedom, for example, a camera tilt motion, could be used to compensate for the feature motion or image constraints could be added. We will investigate these solutions in future work.

4) First experiment: 2D motion: We arbitrarily fix the relative camera-CoM position to be $\begin{bmatrix}0 & 0 & -1 & 0 & \pi/2 & \pi/2\end{bmatrix}$. The reference frame is the initial position of the CoM. The target is fixed at $1.5m$ on the x axis. The desired features are computed for a CoM at $\begin{bmatrix}1 & 0.3 & 0 & 0 & 0 & 0\end{bmatrix}$.

Fig. 4 depicts the results obtained when applying a saturated visual servoing as an input of the reactive pattern generator. After 100 iteration (10s), the CoM oscillates around the desired position. We can see the repercussion of the sway motion on the error vector that is oscillating around zero. This motion results in oscillation of the features in the image plane that the visual servoing tries to compensate. Which explains the remaining lateral reference velocity (Fig 4.b red dotted lines). Then the visual servoing never converges and oscillates. This is this effect that we want to cancel : when the error oscillates around zero, the convergence is obtained and the control law should be zero.

This is the result we obtain Fig 5. The sway motion is cancelled from the error computation. The corrected error vector appears in dotted lines in Fig 5.a while the real error is in plain line. The corrected error follows an exponential decrease. We can use the sum square value of this error as a convergence criterion to stop the servoing.

The cancellation of the sway produces a smaller lateral velocity (see Fig 4.b and Fig 5.b) which means a smaller oscillation ofg the features in the image plane (see Fig 4.a and Fig 5.a). Comparing the CoP trajectories in Fig 4.c and Fig 5.c, we can see that the correction prevents the CoP from going to far in the support polygon which ensures a better stability.

5) Second experiment: 2D motion with disturbance: Now that we have obtained an exponential decrease of the error, let us expose the system to more complex situations. Let us consider the case where the robot is pushed at time $t = 4s$. We applied an additionnel acceleration of $[110001]$ during Te. The disturbance clearly appears on the Fig. 6, yet the system is stable. This behaviour to instant disturbance allows to react quickly and to correct the position as soon as the features can be tracked in the image. It is the main advantage of this method with regards to planned trajectory.

6) Third Experiment: 3D motion: Then, let us add a rotational motions. Here, the desired CoM position is set to be $\begin{bmatrix}1 & 0.6 & 0.8 & 0 & \frac{\pi}{18} & 0\end{bmatrix}$ Fig. 7 illustrates the results. The visual task does not converge perfectly. A bias is introduced by the correction because the sagital velocity can not be tracked due to the walking constraints.

Indeed, the proposed Model Predictive Control scheme ensures that the average CoM velocity $\dot{c}$ tracks the reference velocity $\dot{c}^*$, but the dynamics and constraints of the walking motion induce some instant disturbances: the desired velocity may reach the system speed limits and a sway motion is induced by the stepping. To cope with this issue, the constraints need to be explicitly taken into account to correct the measure of the current features s. It could be done by introducing the visual criteria directly inside the cost function of the proposed Model Predictive Control Scheme.

V. CONCLUSION

The new reactive pattern generator proposed by Herdt et al. paves the way to vision based reactive control of walking motion. In this paper, we propose a method to ignore the sway motion while controlling the walking motion with visual information. It leads to an exponential decrease of the visual error. Besides, it remove the unwanted lateral motion which tried to compensate for the sway motion. In order to improve the robot behavior, the reference velocity is limited to a feasible one and a transitory start-up phase is considered.

The propose approach is efficient when the CoM velocity tracks well the reference velocity. The sway motion only disturbs the system. Furthermore, the proposed approach is robust to instant disturbance of the target position.In order to deal with the walking constraints properly, the visual criterion should is included as an additional cost function in the Model Predictive Control scheme.

Visual Model Predictive Control Scheme has been studied to deal with constraints, eg to ensure the visibility of the target or avoid joint limits [17]. In order to improve the results presented in this paper, we propose to write a general non linear model predictive control scheme to select the optimal jerk of the CoM $\dddot{C}$ regarding some visual criteria. Then the function to minimise is now

$$\min_{\dddot{C},\overline{F}} \frac{1}{2}\sum_{i=1}^{N} \|s(k_i) - s_i^*\|^2 \tag{19}$$

It will be studied in a future work.

VI. ACKNOWLEDGMENTS

The first author is supported by a grant from the Japanese Society for the Promotion of Science (JSPS). The second, third and fourth authors are supported by a grant from the RBLINK Project, Contrat ANR-08-JCJC-0075-01.

[2]Let $\begin{bmatrix}X & Y & Z & 1\end{bmatrix}^T$ be the homogeneous position of a 3d point expressed in the camera frame and $\begin{bmatrix}x & y & 1\end{bmatrix}^T$ be its projection in an image plane, the interaction matrix of a point can then be written:

$$L = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & (1+y^2) & -xy & -x \end{bmatrix}$$

.

a. Error Norm b. Camera velocity c. Top view d. Image features

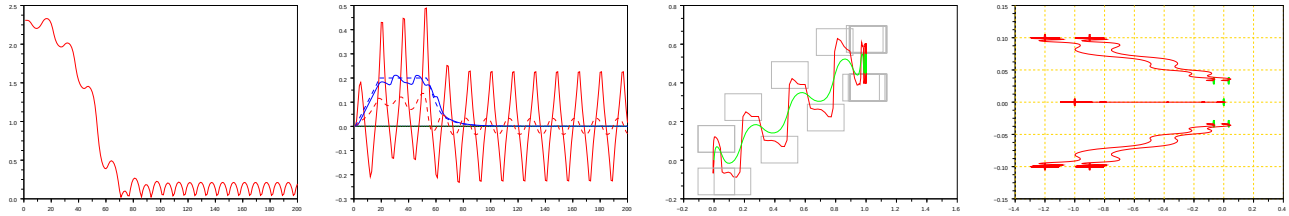

Fig. 4. Visual servoing with limiting the velocity to comply with the HRP-2 capabilities.

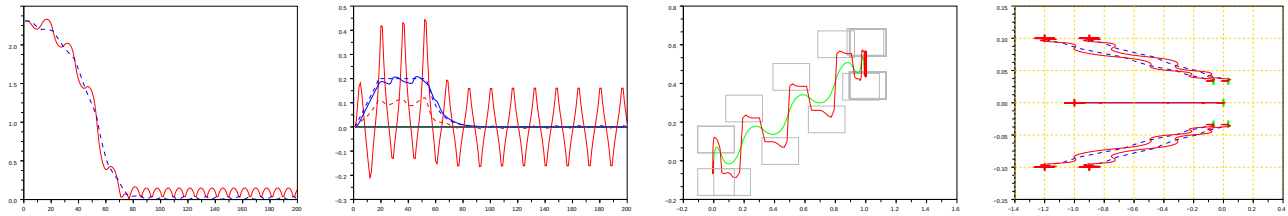

Fig. 5. Visual servoing with a correction of the sway motion to remove the useless lateral control.

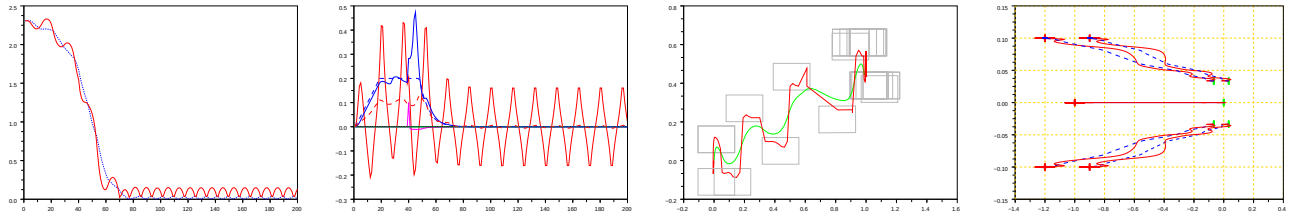

Fig. 6. Visual servoing with a correction of the sway motion when the system undergoes an instant disturbance.

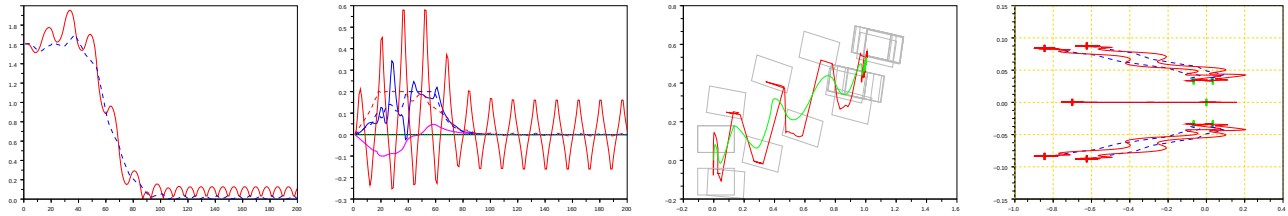

Fig. 7. Visual servoing while cancelling sway motion with 3 degrees of freedom.

REFERENCES

[1] C. Kemp, A. Edsinger, and E. Torres-Jara, "Challenges for robot manipulation in human environments," *IEEE Robotics and Automation Magazine*, pp. 20–29, march 2007.

[2] O. Lorch, A. Albert, J. Denk, M. Gerecke, R. Cupec, J. Seara, W. Gerth, and G. Schmidt, "Experiments in vision-guided biped walking," in *IEEE Int. Conf. on Robotics and Automation*, 2002, pp. 2484–2490.

[3] J. Chestnutt, P. Michel, J. Kuffner, and T. Kanade, "Locomotion among dynamic obstacles for the honda asimo," in *IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 2572–2573.

[4] P. Michel, J. Chestnutt, S. Kagami, K. Nishiwaki, J. Kuffner, and T. Kanade, "Gpu-accelerated real-time 3d tracking for humanoid locomotion and stair climbing," in *IEEE Int. Conf. on Robotics and Automation*, October 2007, pp. 463–469.

[5] J.-S. Gutmann, M. Fukuchi, and M. Fujita, "3d perception and environment map generation for humanoid robot navigation," *International Journal of Robotics Research*, vol. 27, no. 10, pp. 1117–1134, 2008.

[6] J. Coelho, J. Piater, and R. Grupen, "Developing haptic and visual perceptual categories for reaching and grasping with a humanoid robot," *Robotics and Autonomous Systems*, vol. 37, no. 2-3, pp. 195–217, November 2001.

[7] G. Taylor and L. Kleeman, "Flexible self-calibrated visual servoing for a humanoid robot," in *Australian Conference on Robotics and Automation (ACRA2001)*, 2001, pp. 79–84.

[8] N. Courty, E. Marchand, and B. Arnaldi, "Through-the-eyes control of a virtual humanod," in *IEEE Computer Animation 2001*, November 2001, pp. 74–83.

[9] N. Mansard, O. Stasse, F. Chaumette, and K. Yokoi, "Visually-guided grasping while walking on a humanoid robot," in *IEEE Int. Conf. on Robotics and Automation*, April 2007, pp. 3041–3047.

[10] M. Morisawa, K. Harada, S. Kajita, S. Nakaoka, K. Fujiwara, F. Kanehiro, K. Kaneko, and H. Hirukawa, "Experimentation of humanoid walking allowing immediate modification of foot place based on analytical solution," in *IEEE Int. Conf. on Robotics and Automation*, 2007, pp. 3989–3994.

[11] K. Nishiwaki and S. Kagami, "Online walking control system for humanoids with short cycle pattern generation," *Int. Journal of Robotics Research*, vol. 28, pp. 729–742, 2009.

[12] A. Herdt, D. Holger, P.-B. Wieber, D. D., K. Mombaur, and D. Moritz, "Online walking motion generation with automatic foot step placement," *Advanced Robotics*, vol. 24, no. 5-6, April 2010.

[13] A. Herdt, N. Perrin, and P.-B. Wieber, "Walking without thinking about it," 2010.

[14] N. Perrin, S. Stasse, F. Lamiraux, and E. Yoshida, "Approximation of feasibility tests for reactive walk on hrp-2," in *IEEE Int. Conf. on Robotics and Automation*, 2010.

[15] P.-B. Wieber, "On the stability of walking systems," in *International Workshop on Humanoid and Human Friendly Robotics (2002)*, 2002.

[16] F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, December 2006.

[17] G. Allibert, E. Courtial, and F. Chaumette, *Visual Servoing via Advanced Numerical Methods*. Springer, Avril 2010, ch. 20- Visual Servoing via Nonlinear Predictive Control, pp. 393–412.