

Coupling Humanoid Walking Pattern Generation and Visual Constraint Feedback for Pose-Regulation and Visual Path-Following

Noé G. Aldana-Murillo^a, Luis Sandoval^b, Jean-Bernard Hayet^a, Claudia Esteves^c and Hector M. Becerra^a

^aComputer Science Department, Centro de Investigación en Matemáticas (CIMAT), C.P. 36023, Guanajuato, Gto., México.

^bInstituto Tecnológico de Estudios Superiores de Monterrey (ITESM), Guadalajara, Jal. México.

^cMathematics Department, Universidad de Guanajuato, Gto., México.

ARTICLE INFO

Keywords:

Humanoid Robots Locomotion, Visual Servoing, Visual Path Following, Visual Geometric Constraint.

ABSTRACT

In this article, we show how visual constraints such as homographies and fundamental matrices can be integrated tightly into the locomotion controller of a humanoid robot to drive it from one configuration to another (pose-regulation), only by means of images. The visual errors generated by these constraints are stacked as terms of the objective function of a Quadratic Program so as to specify the final pose of the robot with a reference image. By using homographies or fundamental matrices instead of specific points, we avoid the features occlusion problem. This image-based strategy is also extended to solve the problem of following a visual path by a humanoid robot, which allows the robot to execute much longer paths and plans than when using just one reference image. The effectiveness of our approach is validated with a humanoid dynamic simulator.

1. Introduction


For years, the locomotion of humanoid robots has driven a lot of attention from the robotics community. In the continuity of the seminal work by Kajita et al. [17], Wieber [34] and Herdt et al. [15] have used a cart-table model as a simplified model for this complex mechanical system. They plan the trajectory of the Center of Mass (CoM) of the robot, while simultaneously enforcing its stability through the notion of Zero Moment Point (ZMP) [33] as a first step, and they derive its whole body motion by inverse kinematics, as a second step. The CoM trajectory is modeled as a piecewise cubic trajectory (piecewise constant jerks), and Model Predictive Control (MPC) is used to perform optimization of these jerk values in a horizon window, so as to anticipate the motions to be done in the future. Then, the first computed optimal control in the horizon window is applied to generate a reference CoM position to the inverse kinematics, and the algorithm is run again in the following cycles. In Herdt et al. [15], it is shown that the problem can be tackled in terms of reference velocities to be tracked, without specifying the footsteps beforehand, and that it can be efficiently solved as a Quadratic Program (QP), with quadratic terms smoothing the trajectory, other enforcing the stability of the robot (penalizing ZMP positions far from the center of the foot on the ground) and other enforcing the tracking of the reference velocities. All of these are expressed in terms of the values of the jerk of the CoM and the footsteps positions. In addition, hard constraints on the position of the ZMP and on feasible positions of the footsteps are encoded as linear constraints on the problem variables, which allows for an efficient resolution.

In many situations, e.g., when precise metric mapping

of the environment is difficult, it is useful to specify the objectives of the robot in terms of the values of its sensors. This has been a principle at the core of the visual control community for a couple of decades [5]. It avoids having to cope with the traditional problems associated to map building and SLAM strategies: Drifting errors, Computational load associated to map updates and loop detection, among others. However, SLAM strategies have also been used for humanoid locomotion, for instance [29, 30, 31]. After successful works that have demonstrated the benefits of this approach on wheeled robots, efforts have been done to integrate visual control within the locomotion of humanoid robots, i.e., to specify the target pose (position and orientation) of the humanoid through a reference image, which is known as pose-regulation. In works like Dune et al. [9], Delfin et al. [6], the locomotion with visual objectives is handled in a decoupled way: first, a target velocity is computed from visual errors and then, it is used as a reference velocity in the velocity-based walking pattern generator (WPG) of Herdt et al. [15]. A limitation of such an approach is that it needs a special and careful handling of the robot sway motion. In Garcia et al. [13], the authors use visual references instead of reference velocities in the objective function. This scheme implements position-based and image-based visual servoing, with determined points, by linearizing the image projection functions to keep the optimization as a QP.

Visual servoing is local by nature. In order to extend it to larger scale navigation, some authors, e.g. Ido et al. [16], have proposed to use a sequence of reference images (visual path). This requires a visual path-following controller and may generate discontinuous velocities when switching from one reference image to another. This problem has been tackled in Delfin et al. [7] by using transition functions to achieve smooth switching. Other methods exploit the geometry of

*Corresponding author

 noe.aldana@cimat.mx (N.G. Aldana-Murillo)

ORCID(s): 0000-0003-3015-3123 (N.G. Aldana-Murillo)

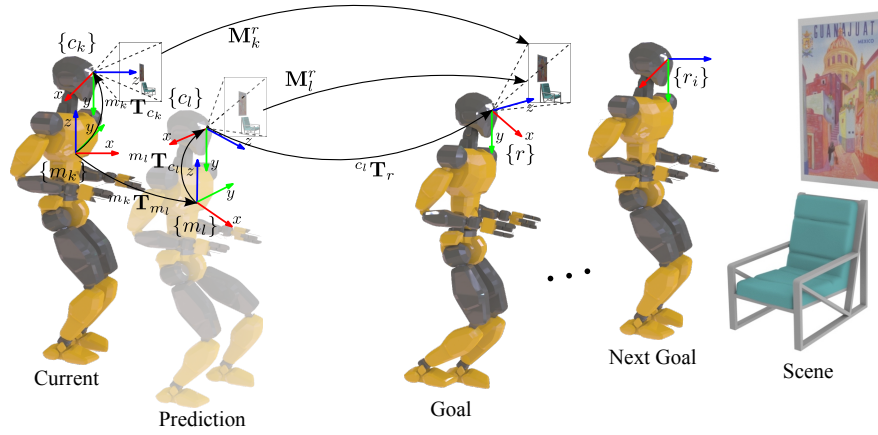


Figure 1: Setup of the proposed approach: A humanoid robot with an onboard camera has to walk while following a sequence of goal images, being driven by feedback of visual constraints errors. The key transformations and coordinate frames are shown here, where the leading superscript denotes the reference coordinate frame and the subscript denotes the frame being described. For example, ${}^{c_k}\mathbf{T}_{c_r}$ is the transformation that maps points expressed in $\{c_r\}$ to points in $\{c_k\}$.

the environment, for instance by using vanishing points that can be computed with corridors [11, 25].

In this work, we present five main contributions towards a tight coupling between walking pattern generation and visual references using the setup shown in Fig. 1:

1. We generalize the approach of Garcia et al. [13] to homography-based and essential matrix-based control. Sensibility to occlusion is one of the drawbacks of Garcia et al. [13], that limits its applicability. Our approach can handle lost point correspondences and partial occlusions of the reference image without altering the walking pattern. Homographies and essential matrices have been successfully exploited for robot visual control [3, 20, 6]. However, to our knowledge, this is the first time that they feed an MPC scheme within a humanoid locomotion controller.
2. We study and compare different strategies to handle the rotational component of the humanoid motion. In many previous works [15, 13], the orientation has been handled as a separate optimization process so as not to lose the QP structure. Here, we take advantage of being able to recover the relative orientation to the goal through the visual constraints (e.g., homographies) to drive the trunk and feet orientations within the walking pattern.
3. We extend our control strategy to navigate by following a visual path instead of following just one image. This allows larger scale visual control tasks: Given a succession of consecutive visual goals (images) to reach, we take advantage of the MPC approach to integrate visual errors of subsequent reference images directly within the WPG, with the benefit of generating smooth walking patterns when switching from one reference image to another.
4. We demonstrate that the visual constraints-driven lo-

comotion can be efficiently and robustly implemented in a dynamic simulator.

5. We demonstrate through experiments that map-less navigation within a visual memory is more efficient than classical approaches relying on SLAM.

The remainder of this paper is structured as follows: Section 2 describes other works related to our approach. In Section 3, we present the general ideas guiding our proposal of coupled visual constraint-based visual servoing and walking pattern generation. Section 4 describes two optimization schemes for implementing our proposal, one leading to a Quadratic Program, the other one involving Sequential Quadratic Programming. Section 5 gives details on the two cases of studied visual constraints, namely homographies and essential matrices. Section 6 discusses simulation results and Section 7 gives our conclusions.

2. Related work

The dynamic model formulation for humanoid locomotion proposed in Kajita et al. [17] assumes that: 1) longitudinal and lateral translations are decoupled and 2) the height of the Center of Mass position (CoM) is fixed. This simplified model is used in Herdt et al. [15] to formulate the WPG as a QP problem, optimizing the sequence of jerks and the footstep positions to track a given reference velocity while ensuring stability by keeping the ZMP within the sustentation polygon of the robot.

In addition to the assumptions made in these works, we consider a humanoid robot equipped with a calibrated, monocular camera placed on its head. For MPC, we assume that the camera motion is described by a decoupled translation on the plane and rotation around a vertical axis. We also assume that the head of the robot is fixed and therefore the relative camera-CoM pose is also fixed.

As mentioned above, the WPG of Herdt et al. [15] may

be used to drive the robot based on visual feedback, in a decoupled manner. In Dune et al. [9], a loose coupling is proposed where the longitudinal and lateral reference velocities for the CoM are determined based on visual servoing strategies, as a function of the visual error defined with respect to a reference goal image. The main drawback is that it needs to handle explicitly the image motion due to the robot balancing motion. In Garcia et al. [13], a tight coupling is proposed where, instead of computing the reference velocity separately and feeding the QP with it, the velocity term in the optimization problem of Herdt et al. [15] is replaced by a visual error term based on the Visual Predictive Control (VPC) approach proposed in Allibert et al. [1]. Garcia et al. [13] adds the third term of the following QP to encode visual errors:

$$\begin{aligned} \min_{U_k} & \frac{\alpha}{2} \|\ddot{X}_k\|^2 + \frac{\alpha}{2} \|\ddot{Y}_k\|^2 + \\ & \frac{\beta}{2} \sum_{\lambda=1}^V [S_\lambda^d - S_{k+1,\lambda}^m]^T \mathbf{W} [S_\lambda^d - S_{k+1,\lambda}^m] + \\ & \frac{\gamma}{2} \|Z_{k+1}^x - Z_{k+1}^{x-ref}\|^2 + \frac{\gamma}{2} \|Z_{k+1}^y - Z_{k+1}^{y-ref}\|^2, \\ \text{s.t. } & \mathbf{C}U_k \leq \bar{c}, \end{aligned} \quad (1)$$

where $U_k = (\ddot{X}_k, X_k^f, \ddot{Y}_k, Y_k^f)^T \in \mathbb{R}^{2N+2m}$ is a vector stacking the unknown jerks $(\ddot{X}_k, \ddot{Y}_k)^T \in \mathbb{R}^{2N}$ along the two translation directions and the m future footstep positions $(X_k^f, Y_k^f)^T \in \mathbb{R}^{2m}$, both in a horizon window of duration N starting at k . The vector $S_\lambda^d \in \mathbb{R}^N$ contains the stacked reference values for the visual features (indexed by λ) and $S_{k+1,\lambda}^m \in \mathbb{R}^N$ are the stacked values predicted by the VPC scheme. The matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is a weight matrix, $(Z_{k+1}^x, Z_{k+1}^y)^T \in \mathbb{R}^{2N}$ is the vector of stacked predicted ZMP positions in the horizon window and $(Z_{k+1}^{x-ref}, Z_{k+1}^{y-ref})^T \in \mathbb{R}^{2N}$ are the corresponding reference positions, which are functions of the footstep positions (X_k^f, Y_k^f) . Finally, α, β, γ are user-defined scalar weighting coefficients for the different terms of the objective function and $\mathbf{C} \in \mathbb{R}^{(4N+4m) \times (2N+2m)}$, $\bar{c} \in \mathbb{R}^{(4N+4m) \times 1}$ encode linear constraints on U_k (on the ZMP and on the footstep positions). These terms are all detailed in Garcia et al. [13]. In this QP problem, the future values $S_{k+1,\lambda}^m$ of the visual features are predicted by linearization of the perspective projection in the horizon window. It is worth noting that this scheme is not generic, since it relies on the feedback from point features and using any other visual information is not straightforward. Another drawback of the aforementioned approach is that the reference visual features need to remain visible during the navigation. Hard visibility constraints may be introduced in the QP algorithm, but this reduces the range of possible movements and does not ensure robustness with respect to image processing errors. To allow long-range navigation using visual path-following, as what has been done with wheeled robots [26], robustness is needed to feature appearance/disappearance or occlusions. We will show that our approach allows to account for occlusions.

3. Humanoid locomotion based on visual constraints: General principles

In the following, we describe how to modify the WPG scheme of Eq. 1 to include geometric constraints between pairs of images as a way to drive the robot to its objective.

3.1. Visual constraints in the WPG formulation

Our strategy still uses MPC, with the same variables as above, but instead of using the observed/predicted positions of specific point features, we use the predicted values of selected elements of a multiple-view constraint, namely a homography or an essential matrix. In both cases, this visual constraint can be represented by a 3×3 homogeneous matrix \mathbf{M}_k^r relating the two images indexed by k (“current” image \mathcal{I}^k taken from the robot) and r (the “reference” image \mathcal{I}^r the robot has to reach). In the following, we consider a subset of V scalar elements $\{m_{k,i}^r, i \in [1, V]\}$ of the matrix \mathbf{M}_k^r , where these elements are indexed by i . For example, we may consider the element $(1, 1)$ of a homography matrix, index it by $i = 1$, and denote it as $m_{k,1}^r$.

Let us also define the vectors $\bar{\mathbf{m}}_{k+1,i}^r$ and $\hat{\mathbf{m}}_{k+1,i}^r$, which stack the desired and predicted values, respectively, of the selected element i of a geometric constraint, in a prediction window of duration N , starting at $k + 1$:

$$\hat{\mathbf{m}}_{k+1,i}^r = \left(\hat{m}_{k+1,i}^r, \hat{m}_{k+2,i}^r, \dots, \hat{m}_{k+N,i}^r \right) \quad (2)$$

and similarly for $\bar{\mathbf{m}}_{k+1,i}^r$. We propose to modify Eq. 1 by replacing the visual errors term by the following one:

$$\frac{1}{2} \sum_{i=1}^V \beta_i [\bar{\mathbf{m}}_{k+1,i}^r - \hat{\mathbf{m}}_{k+1,i}^r]^T \mathbf{W} [\bar{\mathbf{m}}_{k+1,i}^r - \hat{\mathbf{m}}_{k+1,i}^r], \quad (3)$$

where $i \in [1, V]$ refers to one of the selected elements of the geometric constraint, and where β is individualized for each element i as β_i . For a future time-step $l > k$, we define

$$\bar{\mathbf{M}}_l^r = \mathbf{M}^* - \Upsilon_k, \quad (4)$$

$$\hat{\mathbf{M}}_l^r = \mathbf{m}(U_k), \quad (5)$$

as the desired (resp. predicted) values of the visual measurement (geometric constraint) from image l to reference image r . These predicted matrix values are written as a function \mathbf{m} of the control vector U_k . The exact nature of \mathbf{m} depends on which geometrical constraint is used and will be detailed in the cases of homographies and essential matrices in Sections 4 and 5. The term $\Upsilon_k = \mathbf{M}_k^r - \bar{\mathbf{M}}_k^r \in \mathbb{R}^{3 \times 3}$ is the prediction error at the current time step k . It is evaluated at k , based on the current observations \mathbf{M}_k^r (computed from image point correspondences) and kept constant in the horizon window, as suggested in Allibert et al. [1]. The diagonal matrix $\mathbf{W} = \text{diag}(w_{k+1}, \dots, w_{k+N})$ weights time indices in the prediction window. As matrices \mathbf{M}_k^r are projective, a special care has to be put on their scaling (see Section 5).

As a major difference with previous works in walking pattern generation, the use of visual constraints allows us not

to depend on a world reference frame in which to localize the robot, as the visual constraints implicitly operate in a relative frame with respect to the reference image. This means that the optimization of our controls (jerks and footsteps) can be done in a local reference frame, namely in the current CoM frame m_k . A global localization of the robot at every iteration with respect to a world reference frame requires more information than only the monocular images (as concluded in Garcia et al. [13]), and it is avoided here.

3.2. Predictive model for the visual constraints

As mentioned above, we do not use all the elements of the geometric constraint between the target and the current images but only a subset of V elements. To predict their values $\hat{m}_{l,i}^r$ in the horizon window, we express them in terms of the jerks to optimize. Based on Fig. 1 and referring to reference frames with indices “ c ” (resp. “ m ”) for the camera (resp. the CoM), the homogeneous rigid transformation ${}^{c_l}\mathbf{T}_{c_r}$ between the camera frame in a future iteration l and the camera reference frame is written as

$${}^{c_l}\mathbf{T}_{c_r} = {}^{c_r}\mathbf{T}_{m_l} ({}^{m_k}\mathbf{T}_{m_l})^{-1} ({}^{c_k}\mathbf{T}_{m_k})^{-1} {}^{c_k}\mathbf{T}_{c_r}, \quad (6)$$

with ${}^{c_k}\mathbf{T}_{m_k} = {}^{c_l}\mathbf{T}_{m_l}$ the CoM/camera transformation (roughly, a vertical translation), ${}^{m_k}\mathbf{T}_{m_l}$ the transformation encoding the planar CoM motion between $\{m_k\}$ and $\{m_l\}$, and ${}^{c_k}\mathbf{T}_{c_r}$ the transformation from the reference camera frame, c_r , to the camera frame at k , c_k . The pitch rotation angle of this transformation is referred to as ϕ_k . The transformation ${}^{c_k}\mathbf{T}_{c_r}$ (and its rotational component ϕ_k , in particular) is deduced, up to a scale, from the decomposition of the observed geometric constraint \mathbf{M}_k^r at k (see Section 5 for more details). Under the planar motion assumption used within the MPC, the vector ${}^{c_l}\mathbf{t}_{c_r}$ is expressed as:

$$\begin{bmatrix} ({}^{c_k}x_{c_r} + {}^{m_k}y_{m_l}) \cos({}^{m_k}\theta_{m_l}) + ({}^{c_k}z_{c_r} - {}^{m_k}x_{m_l}) \sin({}^{m_k}\theta_{m_l}) \\ 0 \\ -({}^{c_k}x_{c_r} + {}^{m_k}y_{m_l}) \sin({}^{m_k}\theta_{m_l}) + ({}^{c_k}z_{c_r} - {}^{m_k}x_{m_l}) \cos({}^{m_k}\theta_{m_l}) \end{bmatrix},$$

and similarly for the rotation part,

$${}^{c_l}\mathbf{R}_{c_r} = \begin{bmatrix} \cos(\phi_k + {}^{m_k}\theta_{m_l}) & 0 & \sin(\phi_k + {}^{m_k}\theta_{m_l}) \\ 0 & 1 & 0 \\ -\sin(\phi_k + {}^{m_k}\theta_{m_l}) & 0 & \cos(\phi_k + {}^{m_k}\theta_{m_l}) \end{bmatrix},$$

where the notation ${}^a s_b$ represents the s -component (in translation or orientation) of the transformation between frames a and b . Using the above equations, we deduce the elements of \mathbf{M}_l^r in function of the CoM coordinates at l , expressed in the CoM frame at k , which are, in turn, expressed in terms of the jerks. As we will describe it later, in all the cases we consider (homographies and essential matrices), the predicted terms take the form

$$\hat{m}_{l,i}^r = \mu_{l,i}({}^{m_k}x_{m_l}, {}^{m_k}y_{m_l}, {}^{m_k}\theta_{m_l}), \quad (7)$$

where the functions $\mu_{l,i}$ are, in general, non-linear. Similarly, the stacked values of the visual constraint elements can be expressed as a function of the stacked positions/orientations

$$\hat{\mathbf{m}}_{l,i}^r = \mu_{l,i}(X_{k+1}, Y_{k+1}, \Theta_{k+1}), \quad (8)$$

where X_{k+1} (resp. Y_{k+1}) stacks the N predicted values ${}^{m_k}x_{m_l}$ (resp. ${}^{m_k}y_{m_l}$) of the x (resp. y) positions of the CoM with respect to the frame m_k and Θ_{k+1} stacks the N predicted values of the relative orientation ${}^{m_k}\theta_{m_l}$ of the trunk/CoM with respect to its orientation in k . These displacements are expressed in terms of the jerks to be optimized, which allows to make Eq. 3 an explicit function of these jerks and of the initial CoM state (position, velocity, acceleration) at k , \hat{x}_k, \hat{y}_k . For axis x , we have:

$$X_{k+1} = (x_{k+1}, \dots, x_{k+N})^T = \mathbf{P}_{ps} \hat{x}_k + \mathbf{P}_{pu} \ddot{X}_k, \quad (9)$$

where the sequence of controls (jerks) on x is $\ddot{X}_k = (\ddot{x}_k, \dots, \ddot{x}_{k+N-1})^T$. The matrices $\mathbf{P}_{ps} \in \mathbb{R}^{N \times 3}$ and $\mathbf{P}_{pu} \in \mathbb{R}^{N \times N}$ are deduced easily from repeated integrations [15]. The same relations apply along the y -axis.

3.3. Constraints in the optimization problem

The optimization hard constraints are similar to Herdt et al. [15]. One of them forces the ZMP inside the sustentation polygon at each timestep ($4N$ constraints), which translates into constraints on the ZMP position (hence, on the controls) and on the footsteps positions. The only difference here is that all the quantities are expressed relatively to the frame $\{m_k\}$. The same applies for the constraints on the support foot position in the horizon window ($4m$ constraints). The sustentation polygon orientation is driven by the flying foot orientation values in the prediction window, which in turn induce orientation values for the support foot. We denote these flying foot orientation values as $\Psi_{k+1} \in \mathbb{R}^N$. The $4N + 4m$ constraints above are stacked as

$$\mathbf{C}(\Psi_{k+1})U_k \leq \bar{c}.$$

Moreover, when optimizing orientations, we set constraints on the predicted values of the flying foot orientations Ψ_{k+1} and trunk orientations Θ_{k+1} . This is reviewed in 4.2.2.

4. Solving the optimization problem

Hereafter, we describe and compare two approaches to solve the optimization problem described above.

4.1. Linear approach

In this first approach, we assume: (1) that the predicted trunk and feet angles Θ_{k+1} and Ψ_{k+1} are determined before solving for the jerks in x and y (see Section 4.2) and (2) that the scaling of the visual constraint matrix is done in such a way as to induce linearity in the jerks (see Section 5). Then, Eq. 5 takes a much simpler, linear form

$$\hat{m}_{l,i}^r = a_{l,i} {}^{m_k}x_{m_l} + b_{l,i} {}^{m_k}y_{m_l} + c_{l,i},$$

where $a_{l,i}, b_{l,i}, c_{l,i}$ are constant at k . We also get the vectors $\hat{\mathbf{m}}_{k+1,i}^r$ that stack the predicted visual measurements in terms of the stacked jerks applied from $\{m_k\}$, \ddot{X}_k, \ddot{Y}_k , as

$$\hat{\mathbf{m}}_{k+1,i}^r = \mathbf{A}_{k,i} X_{k+1} + \mathbf{B}_{k,i} Y_{k+1} + \mathbf{C}_{k,i}, \quad (10)$$

with $\mathbf{A}_{k,i}, \mathbf{B}_{k,i} \in \mathbb{R}^{N \times N}$ diagonal matrices that depend only on the known value of the pitch-to-go ϕ_k , for the case of the homography matrix, and on ${}^m k \theta_{m_l}$ (assumed as pre-computed here, see Section 4.2) or ϕ_k , for the case of the essential matrix. $\mathbf{C}_{k,i} \in \mathbb{R}^{N \times 1}$ is a constant vector that depends on the configuration of the target camera at k (${}^c k x_{c_r}, {}^c k z_{c_r}, \phi_k$) and on the angle ${}^m k \theta_{m_l}$, both pre-determined in this case.

Like the position (see Eq. 9), the velocity and acceleration of the CoM, and the position of the ZMP can be predicted in the horizon window as linear functions of the initial state \hat{x}_k, \hat{y}_k and of the sequence of jerks to be applied:

$$\dot{X}_{k+1} = (\dot{x}_{k+1}, \dots, \dot{x}_{k+N})^T = \mathbf{P}_{vs} \hat{x}_k + \mathbf{P}_{vu} \ddot{X}_k, \quad (11)$$

$$\ddot{X}_{k+1} = (\ddot{x}_{k+1}, \dots, \ddot{x}_{k+N})^T = \mathbf{P}_{as} \hat{x}_k + \mathbf{P}_{au} \ddot{X}_k, \quad (12)$$

$$\mathbf{Z}_{k+1}^x = (z_{k+1}^x, \dots, z_{k+N}^x)^T = \mathbf{P}_{zs} \hat{x}_k + \mathbf{P}_{zu} \ddot{X}_k. \quad (13)$$

The matrices $\mathbf{P}_{vs}, \mathbf{P}_{as}, \mathbf{P}_{zs} \in \mathbb{R}^{N \times 3}$ and $\mathbf{P}_{vu}, \mathbf{P}_{au}, \mathbf{P}_{zu} \in \mathbb{R}^{N \times N}$ translate the integration processes [15] and apply similarly along y . The reference ZMP positions are expressed linearly in terms of the current and future positions of the support foot (X_k^c, Y_k^c) and (X_k^f, Y_k^f):

$$\mathbf{Z}_{k+1}^{x_ref} = \mathbf{V}_{k+1}^c X_k^c + \mathbf{V}_{k+1}^f X_k^f, \quad (14)$$

with $\mathbf{V}_{k+1}^c \in \mathbb{R}^N, \mathbf{V}_{k+1}^f \in \mathbb{R}^{N \times m}$ constant binary matrices selecting when each position of a footstep is taken into account in the horizon window.

Finally, Eqs. (3) and (10) are integrated into

$$\begin{aligned} \min_{U_k} & \frac{\alpha}{2} \|\ddot{X}_k\|^2 + \frac{\alpha}{2} \|\ddot{Y}_k\|^2 + \frac{\eta}{2} \|\ddot{X}_{k+1}\|^2 + \frac{\eta}{2} \|\ddot{Y}_{k+1}\|^2 + \\ & \frac{1}{2} \sum_{i=1}^V \beta_i [\hat{\mathbf{m}}_{k+1,i}^r - \hat{\mathbf{m}}_{k+1,i}^r]^T \mathbf{W} [\hat{\mathbf{m}}_{k+1,i}^r - \hat{\mathbf{m}}_{k+1,i}^r] + \\ & \frac{\gamma}{2} \|\mathbf{Z}_{k+1}^x - \mathbf{Z}_{k+1}^{x_ref}\|^2 + \frac{\gamma}{2} \|\mathbf{Z}_{k+1}^y - \mathbf{Z}_{k+1}^{y_ref}\|^2, \end{aligned} \quad (15)$$

$$\text{s.t. } \mathbf{C} U_k \leq \bar{c},$$

where $(\alpha, \eta, \beta_1, \dots, \beta_V, \gamma)$ are weighting coefficients that impose a desired behavior for each term, i.e., increasing a weight has the effect of imposing a faster convergence of the corresponding term. Compared to Eq. 1, we included an additional term related to the acceleration in order to reduce this dynamic effect, for instance at the initial steps. Because of the linear expressions of $\ddot{X}_k, \ddot{X}_{k+1}, \dot{X}_{k+1}, X_{k+1}, Z_{k+1}, Z_{k+1}^{x_ref}, \hat{\mathbf{m}}_{k+1,i}^r$ in function of the jerks and of the footsteps positions, the terms to minimize are all quadratic in the control variables, and the constraints

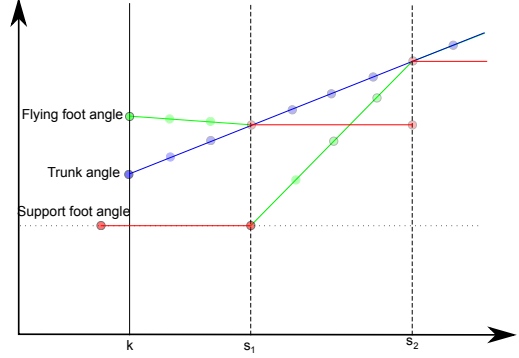


Figure 2: Interpolating the flying foot Ψ_{k+1} (green) and trunk angles Θ_{k+1} (blue).

described in Section 3.3 are linear with fixed \mathbf{C} and \bar{c} . Hence, Eq. 15 is a Quadratic Problem. Its canonical form is

$$\begin{aligned} \min_{U_k} & \frac{1}{2} U_k^T \mathbf{Q}_k U_k + q_k^T U_k, \\ \text{s.t. } & \mathbf{C} U_k \leq \bar{c}, \end{aligned} \quad (16)$$

with \mathbf{Q}_k, q_k deduced from the previous developments. The number of variables is $2N + 2m$ where m is the number of support foot positions in the prediction window, and the number of constraints is $4N + 4m$. Although the mathematical formulation differs, the final structure of the optimization problem is the same as in Herdt et al. [15], so there are no significant differences in the execution time for the solver, for example.

4.2. Handling trunk and feet rotations

In what has been presented up to now, the rotations of the support foot and of the trunk are assumed to be constant, and determined before the translational displacement optimization of Eq. 15. However, to drive the robot from one configuration to another, these rotations should be controlled in a thorough way. Note that the visual constraint at k gives us the angle ϕ_k required to reach the final configuration.

Hereafter, we describe two approaches to determine the orientation values in the prediction window: by simple interpolation and by optimization. Let Θ_{k+1} (resp. Ψ_{k+1}) be the vectors stacking the trunk (resp. flying foot) orientations.

4.2.1. Interpolation.

In this approach, we interpolate the trunk orientations from 0 to ϕ^* (target orientation) within the horizon window:

$$\Theta_{k+1} = \text{interp}(0, \phi^*, N), \quad (17)$$

where $\text{interp}(a, b, n)$ is a simple linear interpolator of n values in the interval $[a, b]$. The target orientation value ϕ^* we use here is obtained from the following expression:

$$\phi^* = \min(\phi_k, \phi_{max}), \quad (18)$$

where ϕ_k is the pitch-to-go angle obtained from the visual constraint at k and ϕ_{max} is a maximal angular displacement

allowed during N steps. This parameter smooths the trajectories of the orientations during a walking execution.

The flying foot orientation trajectories Ψ_{k+1} are piecewise continuous between two support changes, so they are interpolated linearly as illustrated in Fig. 2, i.e. as a linear function starting at the current support foot orientation and reaching the trunk orientation from Θ_{k+1} at the next support change (indices s_i in Fig. 2).

4.2.2. Orientation optimization.

Here, we use the same methodology proposed in Herdt et al. [15] and optimize both, the orientation of the flying foot and the orientation of the trunk. This is solved in the prediction window as:

$$\min_{\ddot{\Theta}_{k+1}, \ddot{\Psi}_{k+1}} \frac{\alpha_R}{2} \|\ddot{\Theta}_{k+1}\|^2 + \frac{\alpha_R}{2} \|\ddot{\Psi}_{k+1}\|^2 + \frac{\beta_R}{2} \|\Theta_{k+1} - \Theta^*\|^2 + \frac{\gamma_R}{2} \|\Psi_{k+1} - \Theta^*\|^2, \quad (19a)$$

$$\begin{aligned} \text{s.t. } & |\Theta_{k+1} - \Psi_{k+1}| \leq \Delta, \\ & \Psi_{k+1} = g(\hat{\psi}_k, \varphi_k, \ddot{\Psi}_k), \\ & \Theta_{k+1} = \mathbf{P}_{ps} \hat{\theta}_k + \mathbf{P}_{pu} \ddot{\Theta}_k, \end{aligned} \quad (19b)$$

where $\ddot{\Theta}_{k+1}$ and $\ddot{\Psi}_{k+1}$ stack the jerks of the corresponding orientations, Θ^* is the reference value computed through Eq. 17, that we want the orientations to be close to, and Δ a vector of maximal allowed absolute differences between the trunk and foot angles. The vectors $\hat{\theta}_k$ and $\hat{\psi}_k$ are the initial states that give the values, first, and second derivatives of the orientations at k . Finally, g is a function (illustrated in green in Fig. 2) of the flying foot initial state $\hat{\psi}_k$, of the current support foot angle φ_k and of the optimized jerks. It can be shown that it has the following form:

$$g(\hat{\psi}_k, \varphi_k, \ddot{\Psi}_k) = \Psi^v(\hat{\psi}_k, \varphi_k) + \mathbf{P}_{puk} \ddot{\Psi}_k,$$

with $\Psi^v(\hat{\psi}_k, \varphi_k) \in \mathbb{R}^{N \times 1}$ and $\mathbf{P}_{puk} \in \mathbb{R}^{N \times N}$. We only consider $m = 2$ footstep positions, for simplicity and because this is the value used in the experiments. We have

$$\Psi^v(\hat{\psi}_k, \varphi_k) = \begin{bmatrix} \mathbf{P}_{ps}^{[0:s_1-1,0:2]} \hat{\psi}_k \\ \varphi_k \mathbf{P}_{ps}^{[0:s_2-s_1-1,0:0]} \\ (\mathbf{P}_{ps}^{[s_1-1:s_1-1,0:2]} \hat{\psi}_k) \mathbf{P}_{ps}^{[0:N-s_2-1,0:0]} \end{bmatrix}.$$

The matrix \mathbf{P}_{puk} is not detailed here for lack of space, but it can be built as a matrix of 3×3 blocks where the blocks are taken from \mathbf{P}_{pu} and \mathbf{P}_{ps} . The notation $\mathbf{A}_{pu}^{[a:b,c:d]}$ selects a specific block indicated by the rows (resp. columns) range $[a, b]$ (resp. $[c, d]$) of the matrix \mathbf{A} . Here, s_1 and s_2 are the instants of the first and second change of the support foot within the prediction window.

The orientation of the support foot is determined from the orientation of the flying foot. As illustrated in Fig. 2, this orientation is assigned when the flying foot touches the floor at each support switch. The next flying foot maintains the orientation of the previous support foot.

Formulated this way, the problem has, again, the structure of a QP problem. Note that all the values are

defined relatively to $\{m_k\}$, the reference frame of the trunk at k . Further details on this step are given in Section 6.

4.3. Non-linear approach

In this second approach, we extend the model predictive control through piecewise-constant jerk values to both positions and orientations and incorporate the orientation jerks $\ddot{\Theta}_{k+1}$ and $\ddot{\Psi}_{k+1}$ in the same objective function as shown above. We use a non-linear solver to handle the non-linear nature of the problem. The number of variables is now $4N + 2m$. We use a non-linear least square formulation to solve the problem, which has the form:

$$\min_{U_k} \frac{1}{2} \|f(U_k)\|_2^2, \quad (20a)$$

$$\text{s.t. } \underline{c} \leq c(U_k) \leq \bar{c}. \quad (20b)$$

To solve this problem, we use Sequential Quadratic Programming (SQP), that quadratizes the objective function and linearizes the constraints around a reference value $U_k^{(0)}$. The method iterates $U_k = U_k^{(0)} + \Delta U_k$ where ΔU_k is the solution of the QP obtained by linearizing f in Eq. 20a and c in Eq. 20b:

$$\min_{\Delta U_k} \frac{1}{2} \|f(U_k^{(0)}) + (\nabla_{U_k} f(U_k^{(0)})) \Delta U_k\|_2^2, \quad (21a)$$

$$\text{s.t. } \underline{c} \leq c(U_k^{(0)}) + (\nabla_{U_k} c(U_k^{(0)})) \Delta U_k \leq \bar{c}. \quad (21b)$$

Reformulating Eq. 21a as a QP in canonical form

$$\min_{\Delta U_k} \frac{1}{2} \Delta U_k^T \tilde{\mathbf{Q}}_k \Delta U_k + \tilde{p}_k^T \Delta U_k, \quad (22a)$$

$$\text{s.t. } \underline{\tilde{c}}_k \leq \tilde{\mathbf{C}}_k \Delta U_k \leq \bar{\tilde{c}}_k, \quad (22b)$$

with

$$\begin{aligned} \tilde{\mathbf{Q}}_k &= \nabla_{U_k} f(U_k^{(0)}) (\nabla_{U_k} f(U_k^{(0)}))^T, \\ \tilde{p}_k &= \nabla_{U_k} f(U_k^{(0)}) f(U_k^{(0)}), \\ \tilde{\mathbf{C}}_k &= (\nabla_{U_k} c(U_k^{(0)}))^T, \\ \underline{\tilde{c}}_k &= \underline{c} - c(U_k^{(0)}), \\ \bar{\tilde{c}}_k &= \bar{c} - c(U_k^{(0)}). \end{aligned}$$

Our final optimization problem is the following one:

$$\begin{aligned} \min_{U_k} & \frac{\alpha}{2} \|\ddot{X}_k\|^2 + \frac{\alpha}{2} \|\ddot{Y}_k\|^2 + \frac{\eta}{2} \|\ddot{X}_{k+1}\|^2 + \frac{\eta}{2} \|\ddot{Y}_{k+1}\|^2 + \\ & \frac{1}{2} \sum_{i=1}^V \beta_i [\hat{m}_{k+1,i}^r - \hat{m}_{k+1,i}^r(\Theta_{k+1})]^T \mathbf{W} [\hat{m}_{k+1,i}^r - \hat{m}_{k+1,i}^r(\Theta_{k+1})] + \\ & \frac{\gamma}{2} \|Z_{k+1}^x - Z_{k+1}^{x,ref}\|^2 + \frac{\gamma}{2} \|Z_{k+1}^y - Z_{k+1}^{y,ref}\|^2 + \\ & \frac{\alpha_R}{2} \|\ddot{\Theta}_k\|^2 + \frac{\alpha_R}{2} \|\ddot{\Psi}_k\|^2 + \\ & \frac{\beta_R}{2} \|\Theta_{k+1} - \Theta^*\|^2 + \frac{\gamma_R}{2} \|\Psi_{k+1} - \Theta^*\|^2, \\ \text{s.t. } & \underline{c} \leq c(U_k) \leq \bar{c}, \end{aligned} \quad (23)$$

where $U_k = ((\ddot{X}_k)^T, (X_k^f)^T, (\ddot{Y}_k)^T, (Y_k^f)^T, (\ddot{\Psi}_k)^T, (\ddot{\Theta}_k)^T)^T$, and where $(\alpha, \eta, \beta_i, \gamma, \alpha_R, \beta_R, \gamma_R)$ are weighting factors. The vectors $\ddot{\Psi}_k \in \mathbb{R}^{N \times 1}$ and $\ddot{\Theta}_k \in \mathbb{R}^{N \times 1}$ are the values of the

piecewise-constant jerks of the feet and trunk orientations, respectively. The vector $\Theta^* \in \mathbb{R}^{N \times 1}$ is a vector of reference orientations. The remaining variables are $\ddot{X}_k, \ddot{Y}_k \in \mathbb{R}^{N \times 1}$, and $X_k^f, Y_k^f \in \mathbb{R}^{m \times 1}$, where N is the prediction horizon and m the number of predicted footsteps.

In this case, we will show that all predicted visual measurements can be expressed with the following structure:

$$\hat{\mathbf{m}}_{k+1,i}^r = \mathbf{A}_{k,i} X_{k+1} + \mathbf{B}_{k,i} Y_{k+1} + \mathbf{C}_{k,i} + \mu_i(\Theta_{k+1}, \Phi_k) + v_i(\Phi_k), \quad (24)$$

where $\mathbf{A}_{k,i}, \mathbf{B}_{k,i} \in \mathbb{R}^{N \times N}$ are diagonal matrices with each diagonal term depending on the known pitch-to-go angle ϕ_k and on ${}^m \theta_{m_i}$ (variable in this approach). $\mathbf{C}_{k,i} \in \mathbb{R}^{N \times 1}$ is a constant vector that depends on the configuration of the target camera at k , i.e., $({}^k x_{c_r}, {}^k z_{c_r}, \phi_k)$. The nonlinear function $\mu_i(\Theta_{k+1}, \Phi_k) \in \mathbb{R}^{N \times 1}$ depends on the predicted values of the trunk orientation Θ_{k+1} and on $\Phi_k = \phi_k(1, \dots, 1)^T$. Finally, $v_i(\Phi_k) \in \mathbb{R}^{N \times 1}$ is a nonlinear in Φ_k .

We recall that the foot/trunk predicted values are integrated from their corresponding jerks through

$$\Theta_{k+1} = \mathbf{P}_{ps} \hat{\Theta}_k + \mathbf{P}_{pu} \ddot{\Theta}_k, \quad \Psi_{k+1} = \Psi^v + \mathbf{P}_{puk} \ddot{\Psi}_k.$$

We use SQP to solve this Non-linear Model Predictive Control (NMPC) in each iteration k . The computational time can be increased due to the iterations it takes to solve each SQP. If the SQP is initialized carefully ($U_k^{(0)}$), then a simple QP can be solved. More details will be given in Section 6.1.

5. Visual constraints and their estimation

The two visual constraints used in this work relate pairs of views sharing some part of their field of view.

The homography matrix \mathbf{H}_k^r gives an explicit mapping of the points from the current image \mathcal{I}_k to the reference image \mathcal{I}_r , provided that the points are projections of a planar scene. It can be computed from the pair of current and reference images with 4 point correspondences at minimum [14].

We will reason in terms of the normalized homography \mathbf{H}_k^r relating these two images as shown in Fig. 3. By “normalized”, we mean that, if \mathbf{K} are the intrinsic parameters of the camera (supposed to be known), and if the image-to-image homography between \mathcal{I}^r and \mathcal{I}^k is \mathcal{H}_k^r , then $\mathbf{H}_k^r = \mathbf{K}^{-1} \mathcal{H}_k^r \mathbf{K}$. The normalized homography at frame k can be decomposed as:

$$\mathbf{H}_k^r = ({}^r \mathbf{R}_{c_k}) \left(I - {}^k \mathbf{t}_{c_r} \frac{\mathbf{n}^T}{d} \right), \quad (25)$$

where ${}^r \mathbf{R}_{c_k}$ is the rotation matrix mapping points from the camera frame $\{c_k\}$ to points in the reference frame $\{c_r\}$, and ${}^k \mathbf{t}_{c_r}$ is the position of the reference camera frame $\{c_r\}$ in the current camera frame $\{c_k\}$. The underlying plane has an equation $\mathbf{n}^T \mathbf{p} + d = 0$ in the camera frame $\{c_k\}$. This decomposition is not unique as it is not possible a priori to disambiguate the scale factor in ${}^k \mathbf{t}_{c_r}$ and d .

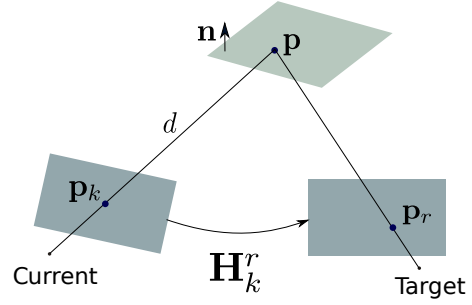


Figure 3: The homography matrix: two-view geometry in the case of a planar scene.

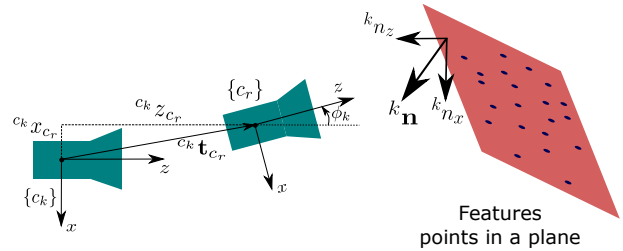


Figure 4: Notations used for the relative pose between the current frame c_k and the reference frame c_r for locomotion based on homography.

The essential matrix \mathbf{E}_k^r is a more general visual constraint that does not require the scene to be planar. It encodes the epipolar geometry and relates corresponding image points p^r, p^k between views through $p^{rT} \mathbf{K}^{-T} \mathbf{E}_k^r \mathbf{K}^{-1} p^k = 0$. It can be computed from the current and reference images with 7 point correspondences at minimum [14]. However, its computation becomes ill-posed when the current image gets close to the reference image.

The essential matrix at frame k can be decomposed as:

$$\mathbf{E}_k^r \propto {}^r \mathbf{R}_{c_k} ({}^k \mathbf{t}_{c_r})_{\times}. \quad (26)$$

Again, this decomposition is not unique, as the scale of ${}^k \mathbf{t}_{c_r}$ cannot be recovered.

5.1. Locomotion based on the homography

The homography matrix was introduced for visual servoing in 6 degrees of freedom in Benhimane and Malis [3] by using the complete matrix for the errors computation. It has been shown in López-Nicolás et al. [20] and Delfin et al. [6] that if the robotic system is subject to motion constraints, like nonholonomy or planar motion, only a few elements of the homography are enough to compute the visual errors. This idea is exploited here.

We develop Eq. 25 by considering the planar motion assumption and by normalizing the matrix by element (2, 2), given that this element is constant and different from zero

with planar motion. We get the following homography at l :

$$\mathbf{H}_l^r = \begin{bmatrix} \mathbf{H}_{l,11}^r & \mathbf{H}_{l,12}^r & \mathbf{H}_{l,13}^r \\ 0 & 1 & 0 \\ \mathbf{H}_{l,31}^r & \mathbf{H}_{l,32}^r & \mathbf{H}_{l,33}^r \end{bmatrix}. \quad (27)$$

Let us describe the normal vector to the plane expressed in the frame $\{m_l\}$, at time l , as ${}^l\mathbf{n} = ({}^l n_x, {}^l n_y, {}^l n_z)^T$. Similarly, the distance is denoted as ${}^l d$. Then the theoretical values of the elements of the homography \mathbf{H}_l^r , using the notation depicted in Fig. 4, are the following:

$$\begin{aligned} \mathbf{H}_{l,11}^r &= -\frac{{}^l n_x}{{}^l d} ({}^{m_k} x_{m_l} \sin \phi_k + {}^{m_k} y_{m_l} \cos \phi_k) + \\ &\quad \frac{{}^l n_x}{{}^l d} (-{}^{c_k} x_{c_r} \cos \phi_k + {}^{c_k} z_{c_r} \sin \phi_k) + \cos(\phi_k + {}^{m_k} \theta_{m_l}), \\ \mathbf{H}_{l,12}^r &= -\frac{{}^l n_y}{{}^l d} ({}^{m_k} x_{m_l} \sin \phi_k + {}^{m_k} y_{m_l} \cos \phi_k) + \\ &\quad \frac{{}^l n_y}{{}^l d} (-{}^{c_k} x_{c_r} \cos \phi_k + {}^{c_k} z_{c_r} \sin \phi_k), \\ \mathbf{H}_{l,13}^r &= -\frac{{}^l n_z}{{}^l d} ({}^{m_k} x_{m_l} \sin \phi_k + {}^{m_k} y_{m_l} \cos \phi_k) + \\ &\quad \frac{{}^l n_z}{{}^l d} (-{}^{c_k} x_{c_r} \cos \phi_k + {}^{c_k} z_{c_r} \sin \phi_k) - \sin(\phi_k + {}^{m_k} \theta_{m_l}), \\ \mathbf{H}_{l,21}^r &= 0, \quad \mathbf{H}_{l,22}^r = 1, \quad \mathbf{H}_{l,23}^r = 0 \\ \mathbf{H}_{l,31}^r &= \frac{{}^l n_x}{{}^l d} ({}^{m_k} x_{m_l} \cos \phi_k - {}^{m_k} y_{m_l} \sin \phi_k) - \\ &\quad \frac{{}^l n_x}{{}^l d} ({}^{c_k} x_{c_r} \sin \phi_k + {}^{c_k} z_{c_r} \cos \phi_k) + \sin(\phi_k + {}^{m_k} \theta_{m_l}), \\ \mathbf{H}_{l,32}^r &= \frac{{}^l n_y}{{}^l d} ({}^{m_k} x_{m_l} \cos \phi_k - {}^{m_k} y_{m_l} \sin \phi_k) + \\ &\quad \frac{{}^l n_y}{{}^l d} (-{}^{c_k} x_{c_r} \sin \phi_k - {}^{c_k} z_{c_r} \cos \phi_k), \\ \mathbf{H}_{l,33}^r &= \frac{{}^l n_z}{{}^l d} ({}^{m_k} x_{m_l} \cos \phi_k - {}^{m_k} y_{m_l} \sin \phi_k) - \\ &\quad \frac{{}^l n_z}{{}^l d} ({}^{c_k} x_{c_r} \sin \phi_k + {}^{c_k} z_{c_r} \cos \phi_k) + \cos(\phi_k + {}^{m_k} \theta_{m_l}). \end{aligned} \quad (28)$$

Given the rotation ${}^{m_k} \mathbf{R}_{m_l}$ undergone by the CoM between $\{m_k\}$ and $\{m_l\}$, we deduce the normal ${}^l \mathbf{n}$ from ${}^l \mathbf{n} = ({}^{m_k} \mathbf{R}_{m_l})^T k \mathbf{n}$, which we emphasize is a function of ${}^{m_k} \theta_{m_l}$. We can write it as:

$${}^l \mathbf{n} = \begin{bmatrix} k n_x \cos({}^{m_k} \theta_{m_l}) + k n_z \sin({}^{m_k} \theta_{m_l}) \\ k n_y \\ -k n_x \sin({}^{m_k} \theta_{m_l}) + k n_z \cos({}^{m_k} \theta_{m_l}) \end{bmatrix}.$$

We use the previous expression in Eq. 28 to express the homography elements as functions of the optimization variables.

The normalized homography of Eq. 25 should be equated to $\mathbf{H}^* = \mathbf{I}_{3 \times 3}$ so as to reach the configuration corresponding to \mathcal{I}^r . We use some of the matrix elements as visual measurements, namely the element 11 (denoted by $\iota = 1$ in the following), the element 12 (denoted by $\iota = 2$), the element 13 (denoted by $\iota = 3$), the element 31 (denoted by $\iota = 4$), the element 32 (denoted by $\iota = 5$), and the element 33 (denoted by $\iota = 6$). Based on Eq. 25, the selected

homography elements are written as specified by Eq. 7. For the sake of clarity, we will only give details for $\iota = 1$ and emphasize the dependency on ${}^{m_k} \theta_{m_l}$ (relative orientation of the trunk/CoM with respect to its orientation in k). The other cases are similar. For $l > k$,

$$\begin{aligned} h_{l,1}^r &= \frac{{}^l n_x ({}^{m_k} \theta_{m_l})}{{}^l d} (-{}^{m_k} x_{m_l} \sin \phi_k - {}^{m_k} y_{m_l} \cos \phi_k) + \\ &\quad \frac{{}^l n_x ({}^{m_k} \theta_{m_l})}{{}^l d} (-{}^{c_k} x_{c_r} \cos \phi_k + {}^{c_k} z_{c_r} \sin \phi_k) + \\ &\quad \cos({}^{m_k} \theta_{m_l} + \phi_k), \end{aligned}$$

which can be stacked to give the form for $\hat{\mathbf{h}}_{k,1}^r$ presented in Eq. 10 for the linear approach or in Eq. 24 for the non-linear approach, as follows:

$$\begin{aligned} \hat{\mathbf{h}}_{k,1}^r &= \begin{bmatrix} -\frac{{}^l n_x}{{}^l d} \sin \phi_k & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{{}^l n_x}{{}^l d} \sin \phi_k \end{bmatrix} \begin{bmatrix} {}^{m_k} x_{m_{k+1}} \\ \vdots \\ {}^{m_k} x_{m_{k+N}} \end{bmatrix} + \\ &\quad \begin{bmatrix} -\frac{{}^l n_x}{{}^l d} \cos \phi_k & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{{}^l n_x}{{}^l d} \cos \phi_k \end{bmatrix} \begin{bmatrix} {}^{m_k} y_{m_{k+1}} \\ \vdots \\ {}^{m_k} y_{m_{k+N}} \end{bmatrix} + \\ &\quad \begin{bmatrix} \frac{{}^l n_x}{{}^l d} (-{}^{c_k} x_{c_r} \cos \phi_k + {}^{c_k} z_{c_r} \sin \phi_k) \\ \vdots \\ \frac{{}^l n_x}{{}^l d} (-{}^{c_k} x_{c_r} \cos \phi_k + {}^{c_k} z_{c_r} \sin \phi_k) \end{bmatrix} + \\ &\quad \begin{bmatrix} \cos({}^{m_k} \theta_{m_{k+1}} + \phi_k) \\ \vdots \\ \cos({}^{m_k} \theta_{m_{k+N}} + \phi_k) \end{bmatrix}. \end{aligned} \quad (29)$$

where the elements ${}^l n_x, {}^l n_y, {}^l n_z$ have been described above. We omitted their dependency on ${}^{m_k} \theta_{m_{k+l}}$ for clarity.

The translation elements ${}^{m_k} x_{m_l}, {}^{m_k} y_{m_l}$ of ${}^{m_k} \mathbf{T}_{m_l}$ are expressed in terms of the jerk vectors to be applied in the prediction window, since this transformation encodes the displacement induced by these controls from k to l . In the linear case (Section 4.1), all ${}^{m_k} \theta_{m_l}$ are pre-determined before this step so that the predictive equations are linear in function of the jerks, with the last two terms of Eq. 29 being $\mathbf{C}_{k,1}$,

$$\hat{\mathbf{h}}_{k,1}^r = \mathbf{A}_{k,1} X_{k+1} + \mathbf{B}_{k,1} Y_{k+1} + \mathbf{C}_{k,1}. \quad (30)$$

In the non-linear case (Section 4.3), ${}^{m_k} \mathbf{R}_{m_l}$ depends on the orientation jerks $\ddot{\Theta}_k$, which makes Eq. 29 a non-linear predictive model,

$$\hat{\mathbf{h}}_{k,1}^r = \mathbf{A}_{k,1} X_{k+1} + \mathbf{B}_{k,1} Y_{k+1} + \mu_1(\Theta_{k+1}, \Phi_k) + \nu_1(\Phi_k). \quad (31)$$

One problem arises with the distance to the plane ${}^l d$: It depends on the displacement from k to l , in a way that would make the formulation of the predicted coefficients $\mathbf{h}_{l,l}^r$ non-linear in the jerks. However, we suppose that the plane is far enough with respect to the translation done during an iteration, hence we assume ${}^l d \approx {}^k d$.

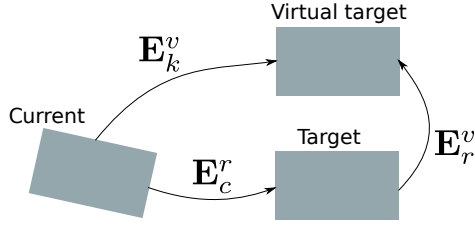


Figure 5: The essential matrix: two-view geometry in the case of non planar scenes, with a virtual target used to avoid degeneracies.

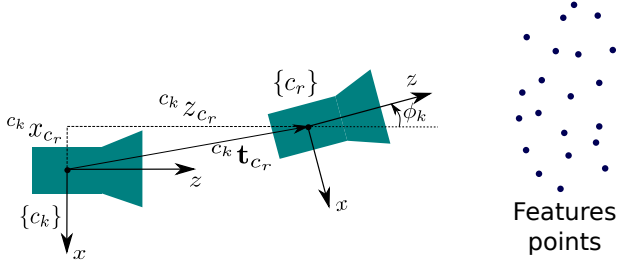


Figure 6: Notations used for the displacements between the current frame c_k and the reference frame c_r for locomotion based on the essential matrix.

5.2. Locomotion based on the essential matrix

The epipolar geometry has been exploited for visual servoing, for instance in Rives [27], Becerra et al. [2]. In particular, it has been shown that some elements of the essential matrix are enough to solve a visual servoing problem for nonholonomic robots undergoing planar motion [21, 19]. However, the estimation of these visual measurements directly from images is ill-conditioned when the relative distance between the current and target positions is too short, which is known as the *short-baseline problem*. To overcome this problem, it has been proposed the use of a virtual target image by López-Nicolás et al. [21]. This virtual image is built by using the epipolar constraint and an essential matrix \mathbf{E}_k^v that considers a vertical displacement of the camera position in the target configuration. The following projection is used to generate the virtual target images:

$$\mathbf{p}_v = (\mathbf{E}_k^v)^T \mathbf{p}_k \times (\mathbf{E}_r^v)^T \mathbf{p}_r, \quad (32)$$

where \mathbf{p}_k and \mathbf{p}_r are corresponding points in the current image \mathcal{I}^k and target image \mathcal{I}^r , respectively. Then, the visual measurements are estimated without degeneration issues from the virtual target points and the corresponding points in the current image \mathcal{I}^k . The principle is illustrated in Fig. 5 and the reader can refer to López-Nicolás et al. [21] for the details about the estimation of the essential matrix without degeneracies for planar motion.

With the notations of Fig. 6, the theoretical values of the elements of the essential matrix \mathbf{E}_l^v normalized by the element 13 are:

$$\begin{aligned} \mathbf{E}_{l,11}^v &= -\tan(\phi_k + {}^{mk}\theta_{m_l}), \\ \mathbf{E}_{l,12}^v &= \frac{({}^{ck}x_{c_r} + {}^{mk}y_{m_l}) \sin {}^{mk}\theta_{m_l} + ({}^{mk}x_{m_l} - {}^{ck}z_{c_r}) \cos {}^{mk}\theta_{m_l}}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_l})}, \\ \mathbf{E}_{l,13}^v &= 1, \\ \mathbf{E}_{l,21}^v &= \frac{({}^{ck}x_{c_r} + {}^{mk}y_{m_l}) \sin \phi_k + ({}^{ck}z_{c_r} - {}^{mk}x_{m_l}) \cos \phi_k}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_l})}, \\ \mathbf{E}_{l,22}^v &= 0, \\ \mathbf{E}_{l,23}^v &= \frac{-({}^{ck}x_{c_r} + {}^{mk}y_{m_l}) \cos \phi_k + ({}^{ck}z_{c_r} - {}^{mk}x_{m_l}) \sin \phi_k}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_l})}, \\ \mathbf{E}_{l,31}^v &= -1, \\ \mathbf{E}_{l,32}^v &= \frac{({}^{ck}x_{c_r} + {}^{mk}y_{m_l}) \cos {}^{mk}\theta_{m_l} + ({}^{ck}z_{c_r} - {}^{mk}x_{m_l}) \sin {}^{mk}\theta_{m_l}}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_l})}, \\ \mathbf{E}_{l,33}^v &= -\tan(\phi_k + {}^{mk}\theta_{m_l}), \end{aligned} \quad (33)$$

where ${}^{ck}y_{c_v}$ is the predefined height of the virtual camera. The target essential matrix to reach the robot configuration at the reference image \mathcal{I}^r is the following:

$$\mathbf{E}^* = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}. \quad (34)$$

Considering the planar motion assumption and the dependency on the optimization variables, we have four useful elements of the essential matrix, namely the element 12 (denoted by $\iota = 1$ in the following), the element 21 (denoted by $\iota = 2$), the element 23 (denoted by $\iota = 3$) and the element 32 (denoted by $\iota = 4$). For the sake of clarity, we only present the case of $\iota = 1$:

$$e_{l,1}^r = \frac{{}^{mk}x_{m_l} \cos {}^{mk}\theta_{m_l}}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_l})} + \frac{{}^{mk}y_{m_l} \sin {}^{mk}\theta_{m_l}}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_l})} + \frac{{}^{ck}x_{c_r} \sin {}^{mk}\theta_{m_l} - {}^{ck}z_{c_r} \cos {}^{mk}\theta_{m_l}}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_l})}.$$

Again, these predicted elements can be stacked to give the form mentioned in Eq. 24, as $\tilde{\mathbf{e}}_{k,1}^r$:

$$\begin{bmatrix} \frac{\cos({}^{mk}\theta_{m_{k+1}})}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_{k+1}})} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{\cos({}^{mk}\theta_{m_{k+N}})}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_{k+N}})} \end{bmatrix} \begin{bmatrix} {}^{mk}x_{m_{k+1}} \\ \vdots \\ {}^{mk}x_{m_{k+N}} \end{bmatrix} + \begin{bmatrix} \frac{\sin({}^{mk}\theta_{m_{k+1}})}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_{k+1}})} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \frac{\sin({}^{mk}\theta_{m_{k+N}})}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_{k+N}})} \end{bmatrix} \begin{bmatrix} {}^{mk}y_{m_{k+1}} \\ \vdots \\ {}^{mk}y_{m_{k+N}} \end{bmatrix} + \begin{bmatrix} \frac{{}^{ck}x_{c_r} \sin({}^{mk}\theta_{m_{k+1}}) - {}^{ck}z_{c_r} \cos({}^{mk}\theta_{m_{k+1}})}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_{k+1}})} \\ \vdots \\ \frac{{}^{ck}x_{c_r} \sin({}^{mk}\theta_{m_{k+N}}) - {}^{ck}z_{c_r} \cos({}^{mk}\theta_{m_{k+N}})}{{}^{ck}y_{c_v} \cos(\phi_k + {}^{mk}\theta_{m_{k+N}})} \end{bmatrix}.$$

Recall that ${}^{m_k}\theta_{m_l}$ is precomputed in the linear approach of Section 4.1. Hence, we can express the stacked predicted elements as:

$$\hat{\mathbf{e}}_{k,1}^r = \mathbf{A}_{k,1}X_{k+1} + \mathbf{B}_{k,1}Y_{k+1} + \mathbf{C}_{k,1}, \quad (35)$$

and similarly for all the elements. In the nonlinear formulation of Section 4.3, the example for $\iota = 1$ can be written as:

$$\hat{\mathbf{e}}_{k,1}^r = \mathbf{A}_{k,1}X_{k+1} + \mathbf{B}_{k,1}Y_{k+1} + \mu_1(\Theta_{k+1}, \Phi_k). \quad (36)$$

In the case of the essential matrix, all the elements have the same form as the example above.

6. Simulation results

The results in this section are divided in three parts. First, single visual servoing tasks are presented, where only one reference image is considered. In the second part, results of an extension of our approach to follow a visual path using a sequence of reference images are given. Finally, in a third part, results obtained on a dynamic simulator are analyzed.

6.1. Implementation details

All the simulations presented hereafter have been generated in C++ and run on an *Intel 2.50 GHz i7 Core* processor. We have used the OPENCV [4] library for most of the required computer vision functions, in particular the estimation and decomposition of the homography matrix and the essential matrix. We have also used the QPOASES [12] library as a solver for the QPs to solve. This is an open source solver in C++ that implements an online active sets strategy.

In all cases, we have supposed that the calibration matrix \mathbf{K} of the camera is known.

6.2. Simulations of single visual servoing tasks

In the simulations described in this section, we have considered the physical parameters of a HRP-2 humanoid robot [18]. The initial pose of the CoM in the motion plane is always taken as $\mathbf{q}_0 = (0\text{m}, 0\text{m}, 0\text{ deg})$, while the final pose is denoted by \mathbf{q}_r . In the reported results, the task termination condition is given by a maximal number of iterations (which is specified in each case). The following objective functions weights are kept constant as $\alpha = 1e^{-4}$, $\gamma = 10$, $\eta = 0.025$, $\alpha_R = 0.01$, $\beta_R = 100$ and $\gamma_R = 100$.

6.2.1. Homography-based visual servoing.

The linear approach for homography-based visual predictive control, proposed in Sections 4.1 and 5.1, is evaluated here for different configurations of visual servoing tasks with a single reference image \mathcal{I}^r . In all the experiments, the reference and the current images \mathcal{I}^r and \mathcal{I}^k are simulated by projecting a set of 3D points that lie in a plane. Gaussian noise was added to the projected points with a standard deviation of 2 pixels. The homography is computed based on the projected points with the DLT algorithm [14]. We use the homography decomposition algorithm proposed in

Triggs [32] to compute the predicted elements $h'_{l,1}$, as explained in Section 5.1. The distance from the target configuration to the plane where the points lie was 4m.

The first test, shown in Fig. 7, corresponds to an experiment to reach the desired pose $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{ deg})$. In Fig. 7(a), the footsteps, altogether with the CoM and ZMP evolution, are shown in the x, y plane. With the red rectangle, we depict the desired pose \mathbf{q}_r , while the cyan footprints give the last configuration of the robot. The desired pose is reached with a good final accuracy in both position and orientation after around 200 iterations. As it can be seen, a smooth trajectory for the CoM is obtained without having to model the sway motion induced in the camera by the locomotion. In Fig. 7(b), the image points paths are shown in the image plane. We have included the evolution of the controlled elements of the homography matrix (Fig. 7(c)), with both the one-step-ahead predicted values from the MPC (in dashed lines) and the observed (measured) values estimated from the point matches. Note that the elements h_{12} and h_{32} are almost zero through the experiment. This is because we use a vertical plane and the components ${}^l n_y$ (see Eq. 28) are almost null. The CoM velocities in the local frame m_k are shown in the Fig. 7(d). In the first iterations, the angular velocity becomes saturated while trying to reach the reference orientation, because of the constraint on maximal angular displacement in the optimization problem of Eq. 19a. Fig. 7(e) shows the optimized objective function values for the translation component of Eq. 15. The value of the weights of the visual features are $\beta_1 = 3$, $\beta_2 = 1$, $\beta_3 = 3$, $\beta_4 = 2$, $\beta_5 = 1$, $\beta_6 = 1$. We will modify these values in the subsequent experiments.

Using feedback of a visual constraint instead of specific points as in Garcia et al. [13] provides robustness of our approach to visual occlusions. This is shown in Fig. 8, where we present the results of an experiment similar to the previous one in which half of the image points are occluded at the middle of the experiment, for the rest of the iterations. Occluded points are depicted in black in Fig. 8(b). As it can be seen, not only the goal is reached with good accuracy but also the occlusion effect is not perceptible on the velocity profiles, which shows the robustness of the approach for these cases and motivates the use of multiple view constraints. In Fig. 8(e), we illustrate the orientation control process, implemented as explained in Section 4.2.

As expected, the weighting coefficients β_i related to the visual errors are important to determine the transient response and the accuracy to reach the desired robot position. We report the effect of varying the weights β_3 and β_6 related to the sagittal visual error in Fig. 9, to reach the pose $\mathbf{q}_r = (5\text{m}, 1.82\text{m}, 20\text{ deg})$, while the other weights are kept constant. In most cases, the robot reaches the desired pose, but the path toward the target is significantly different. Taken into account this results, the values of the visual weights chosen for the next experiments were $\beta_1 = 3$, $\beta_2 = 1$, $\beta_3 = 3$, $\beta_4 = 2$, $\beta_5 = 1$, $\beta_6 = 0.5$ since they provide the best accuracy to reach the target (Fig. 9(d)).

Fig. 10 compares humanoid walks to reach $\mathbf{q}_r =$

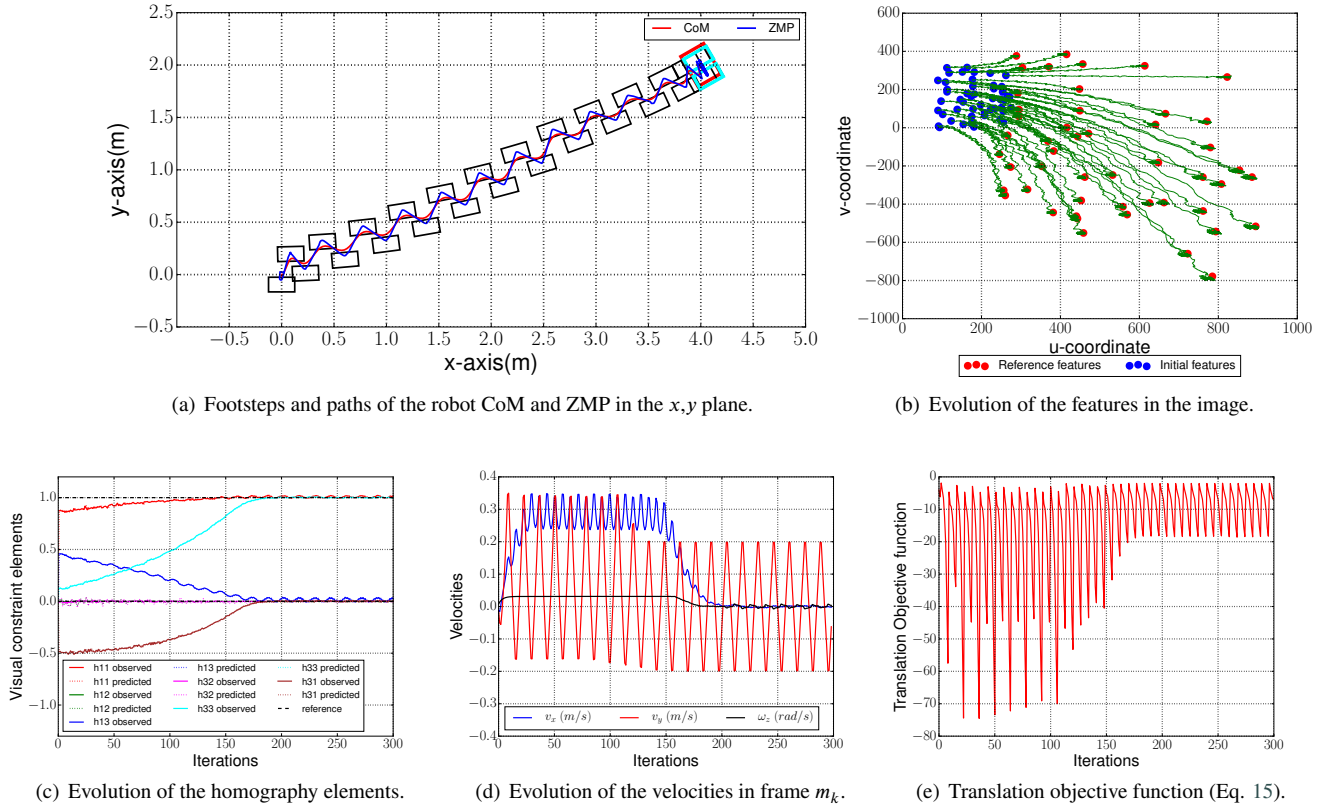


Figure 7: Simulation (300 iterations) of a single visual servoing task with separate position and orientation control for the homography-based locomotion linear setup (Sections 4.1 and 5.1). The desired pose is $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{deg})$.

(2m, 0.8m, 20 deg) for three different ways of evaluating the distance d to the plane used in the predicted homography elements (see Eq. 25). In the first case, d is lower to the real value, while in the second case, it is higher. In both cases, d is kept constant for the duration of the experiment. In the third case, d is first fixed to the real value and then updated along the trajectory, at the beginning of each iteration k , according to the robot motion (i.e., we integrate the value of d used in $k - 1$ and the motion undergone by the robot from $k - 1$ to k , up to time step l). In principle, this should give a better approximation of d . However, as seen in Fig. 10, there is no significant performance differences between the three cases, so it is reasonable (and more computationally efficient) to set d as a constant value in the horizon window.

In Fig. 11, we have tested the robustness of the approach against an external transient perturbation introduced as an additional acceleration to the CoM. The goal is $\mathbf{q}_r = (2.5\text{m}, -1.0\text{m}, -30\text{deg})$. Since we proposed to include a term penalizing high accelerations in Eq. 15, the effect of including it or not is also tested. The perturbation direction and the instant of its application are depicted in the left part of the figure. In both cases (with or without the acceleration term) the humanoid reaches effectively the target pose regardless the perturbation. However, the ZMP has a smoother behavior when the acceleration is penalized (top row of Fig. 11). We have observed that this acceleration term is useful at the beginning of visual servoing tasks,

when large visual errors may cause the ZMP to be very close to the footprint boundaries. Another evaluation of the acceleration term in Eq. 15 is shown in Fig. 612. We repeat the experiment shown in Fig. 7, where the final desired pose is $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{deg})$, with and without the acceleration term. In both cases, the humanoid robot reaches the desired pose, however, with the acceleration penalization, the velocities are lower and the trajectories of the CoM and ZMP are smoother.

Figure 13 shows a walking experiment where all the visual features were occluded (depicted through black lines) from iteration 150 and during a relatively long period of other 150 iterations. This means that the robot is blind during iterations $150 \leq k \leq 300$ and a homography cannot be computed. Instead, the controller uses the homography predicted at $k - 1$ for time k as our new visual features. The desired configuration is $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{deg})$. As it can be seen, the robot reaches the desired pose after an overshooting in position and comes back when the visual features become available again at iteration 300. Between iterations 150 and 300, the robot uses only its predictions, which results in accumulated drift, visible in Fig. 13(c). At iteration 300, the drift between the predicted and ground truth values for the visual features shrinks again. This drift also occurs in the angle estimations since the reference angle values in Fig. 13(e) are estimated based on the decomposition of the predicted homographies, introducing a large drift.

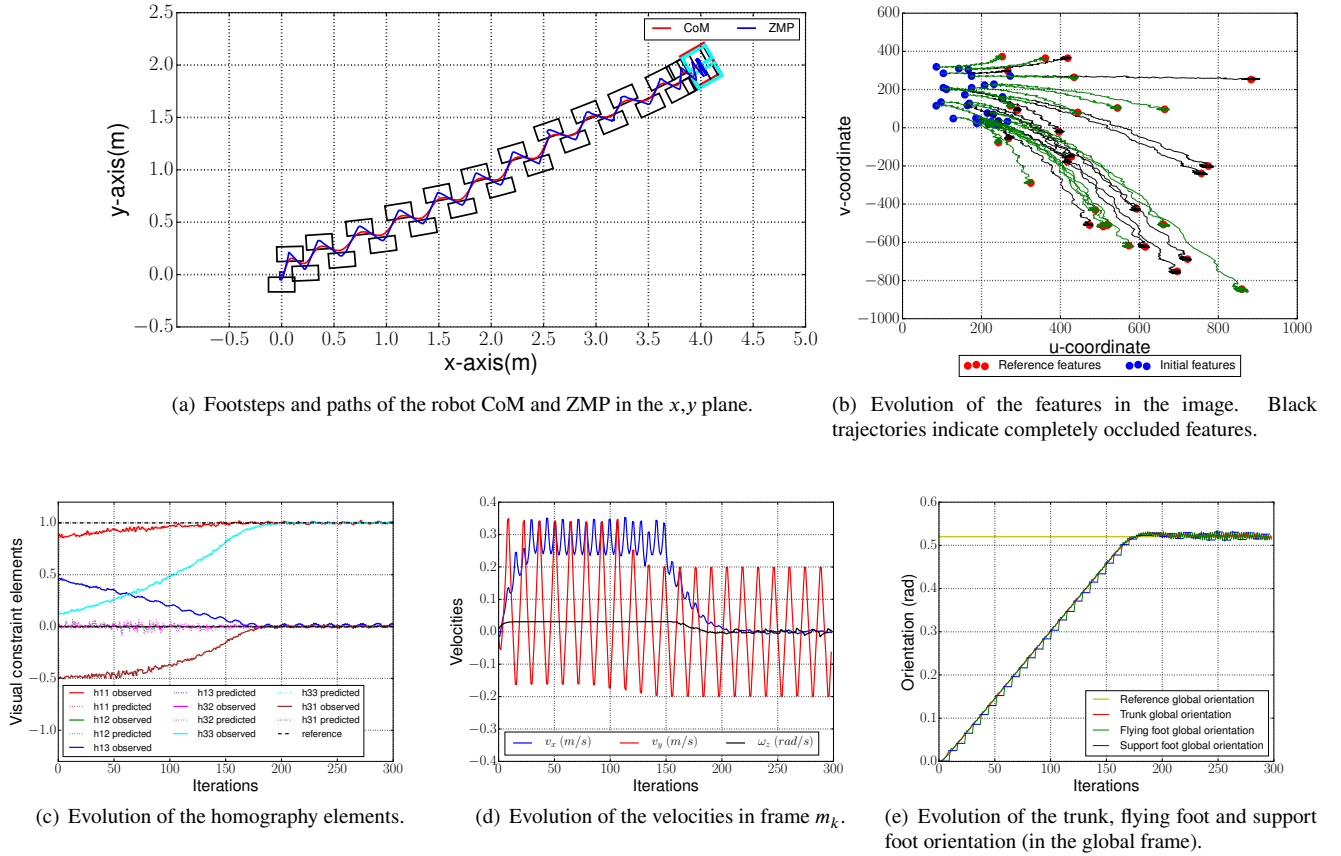


Figure 8: Simulation (300 iterations) of a single visual servoing task with separate position and orientation control for the homography-based locomotion linear setup (Sections 4.1 and 5.1), in presence of partial occlusion (depicted as black lines in 8(b)). The desired pose is $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{ deg})$.

A final evaluation of the homography-based locomotion under camera modeling errors is presented in Fig. 14. In particular, we considered imprecise camera intrinsic parameters. Recall that the predicted values of the homography elements are computed based on the decomposition of the estimated homographies from point matches, which uses the matrix of intrinsic parameters \mathbf{K} . The real parameters of the camera are $\alpha_u = 391.59$, $\alpha_v = 391.59$, $u_0 = 338.82$ and $v_0 = 274.59$. We perturbed these parameters by adding Gaussian noise with standard deviation proportional to the parameters values (30% for the focal length and 25% for the principal point), along a series of 200 simulations. The desired configuration is $\mathbf{q}_r = (4.5\text{m}, -1.5\text{m}, -20\text{ deg})$. On the left of Fig. 14, we depict the distribution of the final poses for the case of using the real camera parameters (but with variations due to image noise and selection of reference points). On the right of the figure, considering uncertain camera parameters, one can see that the robot reaches the desired pose rather precisely in spite of the introduced inaccuracies, with a bounded variance on the final position.

6.2.2. Essential matrix-based visual servoing.

In this case, the simulated image features are generated by projecting a set of 3D points, randomly distributed in

the 3D space in front of the robot, onto the robot camera. As explained in Section 5.2, we follow the virtual target image approach described in López-Nicolás et al. [21] to avoid the degeneracy problem of the fundamental matrix estimation close to the goal configuration. The virtual target (consisting in a set of image points) is only generated at the first iteration and is kept as a constant set of reference image points during the walk. At each iteration, we estimate the fundamental matrix between the virtual image points and the current image points by using the 8-point algorithm. Then, we recover the essential matrix and use the algorithm proposed in Ma et al. [22] to decompose it and compute its predicted elements with Eqs. 33. The value of the height of the virtual image (${}^c y_{c_v}$) was set to 6 meters.

In Fig. 15, the scheme based on the essential matrix is evaluated and point features occlusion is included. The desired pose is $\mathbf{q}_r = (4\text{m}, 1.5\text{m}, 30\text{ deg})$. We depict the same quantities as shown in previous experiments, and, in addition, we depict the virtual reference features in Fig. 15(b) (in green). As it can be seen, the robot is correctly driven to its target and the essential matrix elements converge to their target values. This behavior is obtained even though all the visual features are occluded between iterations 50 and 150. During this total occlusion,

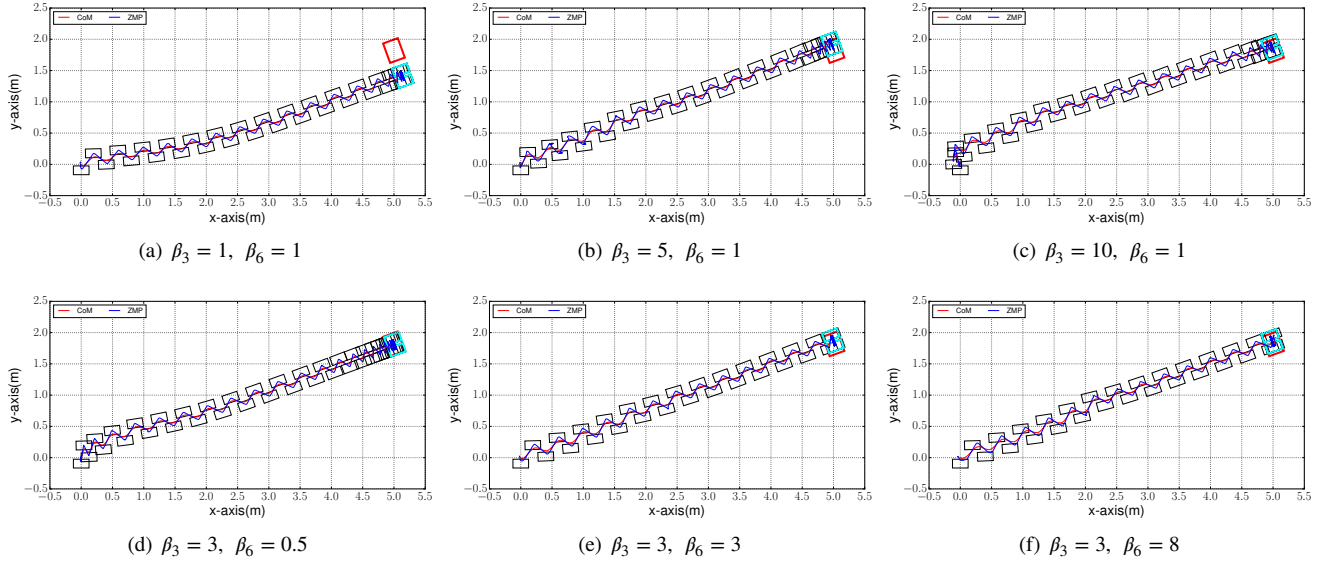


Figure 9: Simulation (400 iterations) of a single visual servoing task under the homography-based locomotion linear setup (Sections 4.1 and 5.1) for different weights β_3 and β_6 in the visual term of Eq. 15. $\beta_2 = \beta_5 = 1, \beta_1 = 3$ and $\beta_4 = 2$. Footsteps and evolution of the CoM and ZMP in the x, y plane are shown. The desired pose is $\mathbf{q}_r = (5\text{m}, 1.82\text{m}, 20\text{deg})$.

the controller uses the essential matrix elements predicted at $k - 1$ for time k as visual features, since an essential matrix cannot be computed from the images. Because of the accumulated drift, as can be seen in Fig. 15(c), the robot walks faster to compensate the drift at iteration 150 and finally reaches the desired pose. The values of the weights on visual errors were $\beta_1 = 4.0, \beta_2 = 3.0, \beta_3 = 8.0$ and $\beta_4 = 1.0$ (we recall that in this case only 4 elements are used).

In Fig. 16, we compare three humanoid walks to reach the pose $\mathbf{q}_r = (4.5\text{m}, -1.5\text{m}, -20\text{deg})$, for different values of the height of the virtual image, ${}^c y_{c_v}$. In the first case, ${}^c y_{c_v}$ is set to 6m, in the second case, to 7m, and in the third case, to 9m. As it can be seen, for the first and third cases, the final position has a small lateral error. This is explained by the fact that varying the value ${}^c y_{c_v}$ is equivalent to modifying the weight coefficients of the visual term in Eq. 3.

We have also tested the robustness of the approach based on the essential matrix against an external transient perturbation and it is illustrate in Fig. 17 with and without the acceleration term in Eq. 15. The objective is to reach $\mathbf{q}_r = (3.0\text{m}, 1.0\text{m}, 20\text{deg})$. Similarly as in the homography-based approach, in spite of the perturbation, in both cases, with or without the acceleration term (top and bottom figures), the humanoid effectively reaches the target pose. It is clear in the plots to the right in Fig. 17 that the magnitude of the accelerations is smaller and the evolution of them is smoother when the acceleration term is used.

Finally, we have evaluated essential matrix-based locomotion under modeling errors. We introduce uncertain values in the camera intrinsic parameters \mathbf{K} , which are used to decompose the essential matrix and compute the predicted values for the MPC scheme. In Fig. 18, we see that in spite of moderate errors on \mathbf{K} (as in the homography case, 30%

of standard deviation for focal length, 25% for the principal point), the desired pose can still be reached successfully; the variance on the final position ($\approx 20\text{cm}$) is a bit larger than in the homography case, but still acceptable.

6.2.3. Evaluation of the non-linear approach.

The nonlinear formulation presented in Section 4.3 uses an analytical linearization of the constraints and a quadratization of the objective function to obtain a standard QP with linear constraints. We evaluated two different ways to define the linearization/quadratization point in the SQP at time k : either we simply set $U_k^{(0)}$ to zero

$$U_k^{(0)} = 0, \quad (37)$$

or we adapt the controls U_{k-1} computed at $k - 1$,

$$U_k^{(0)} = \tau(U_{k-1}), \quad (38)$$

where the function τ “shifts” the previous solution by one time unit to the left (this is effectively a shifting for the jerks in x and y but it is a bit trickier for the other parts of the state). We compare the linear approach versus the non-linear approach in a series of experiments using the homography-based visual servoing. The objective to reach is the position $(6.0\text{m}, 3.46\text{m})$ and the desired angles is varied from 0 to 50 degrees in 5-degree increments. The results are presented in Table 1 (average final position errors), Table 3 (average computational times) and Table 2 (average final orientation errors). The results presented are average values over 100 experiments. As it can be seen, the final position errors are significantly reduced by using the non-linear approach and even more when using the initialization strategy of Eq. 38. The error in orientation has similar values for all three cases,

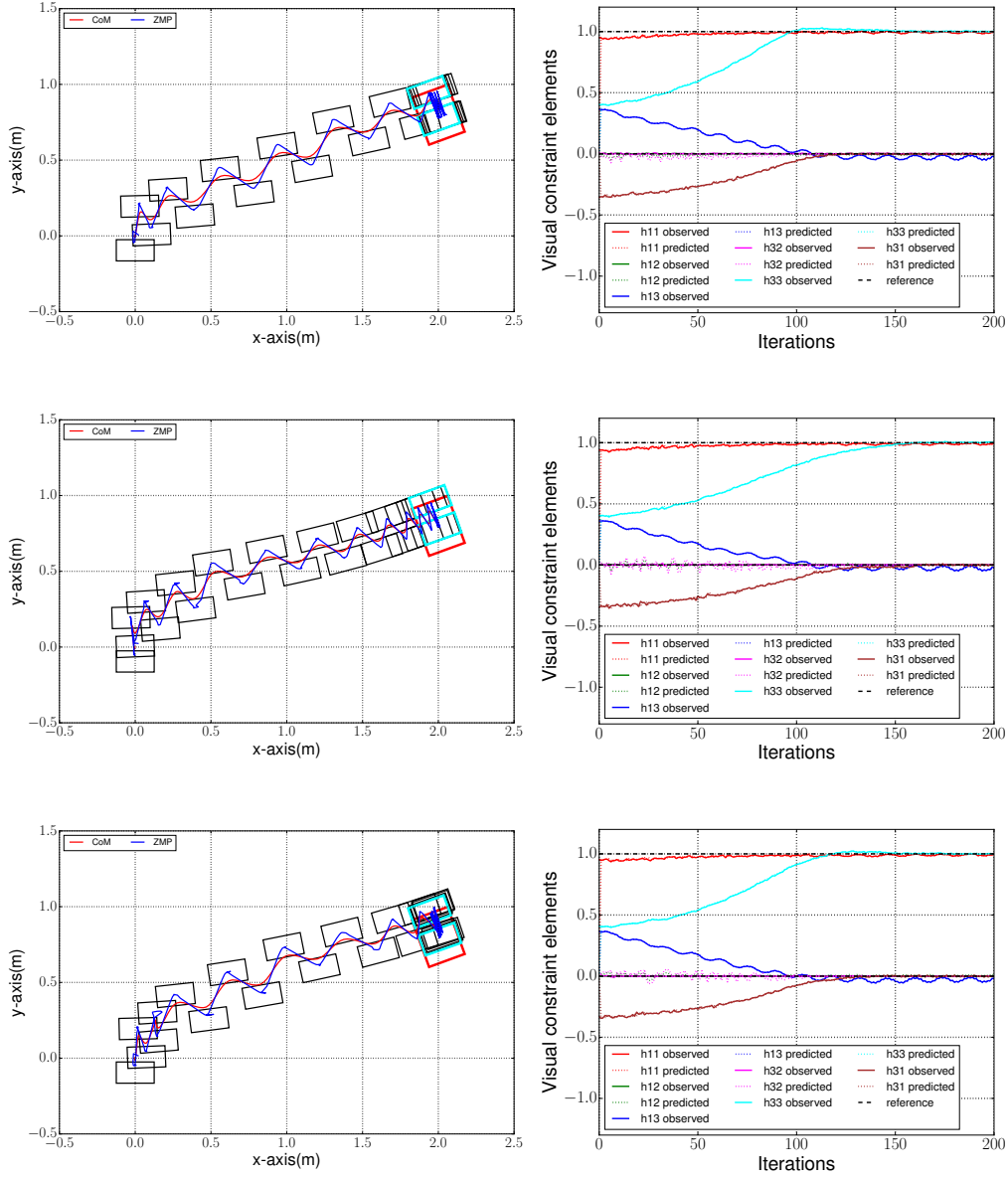


Figure 10: Simulation (200 iterations) of a single visual servoing task under the homography-based locomotion linear setup (Sections 4.1 and 5.1) with two different settings for estimating the distance d to the plane generating the homography. The desired pose is $\mathbf{q}_r = (2\text{m}, 0.8\text{m}, 20\text{deg})$ and the real distance to the plane is 4 meters. Top row: d is fixed to 3 meters along the walk. Middle row: d is fixed to 5 meters. Bottom row: d is set to 4 meters and is updated along the horizon window using the robot self-motion information.

while computational times are higher with the non-linear approaches (but keep in mind that in the linear approach, another QP has to be solved, the one of Eq. 19a). Finally, in Fig. 19, we depict the resulting distributions of the final positions for these three cases, when the desired orientation is set to 45 deg. In all these cases, the final variance on the position is comparable to the one observed in Fig. 14.

6.3. Applications to visual path-following.

In this section, we extend the two-images method described above to use it with a set of images as consecutive targets, which is what we call a visual path. This

strategy allows the robot to reach its final goal by going to intermediate reference images first [26]. We suppose that such a visual path is given, and we aim at controlling the robot to follow this path. We present two strategies to do it.

6.3.1. Weighted visual errors averages.

In this first approach, we extend the VPC term in Eq. 15 to handle multiple reference images $\mathcal{I}^{r_k(i)}$, where the function $r_k(i)$ specifies, at time k , which is the i -th reference

Humanoid Locomotion from Visual Constraints

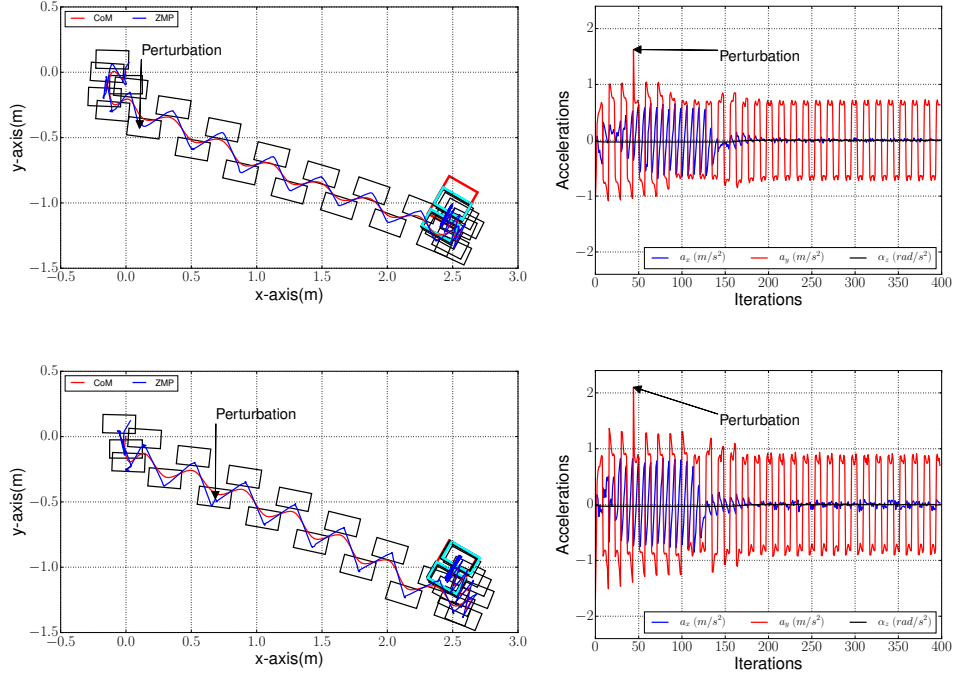


Figure 11: Simulation (400 iterations) of a single visual servoing task with a transient perturbation, using the homography-based locomotion linear setup (Sections 4.1 and 5.1). The desired pose is $\mathbf{q}_r = (2.5\text{m}, -1.0\text{m}, -30\text{ deg})$. Top row: the acceleration term in Eq. 15 is included. Bottom row: this term is not included. Left column: Footsteps and paths of the robot CoM and ZMP in the x,y plane. The instant and orientation of the perturbation are indicated by the black arrow. Right column: accelerations.

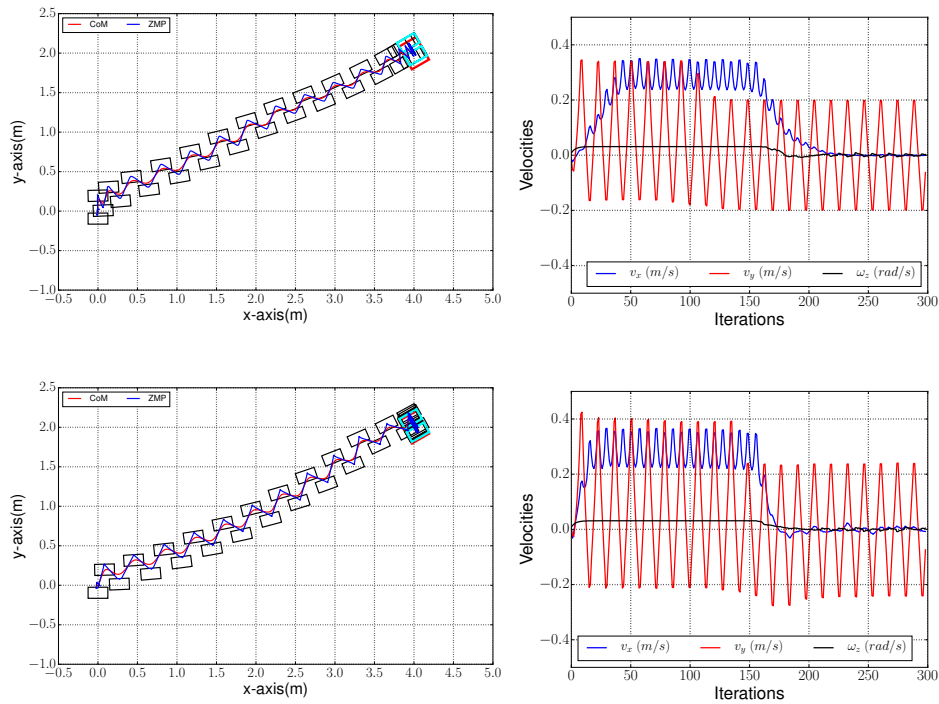
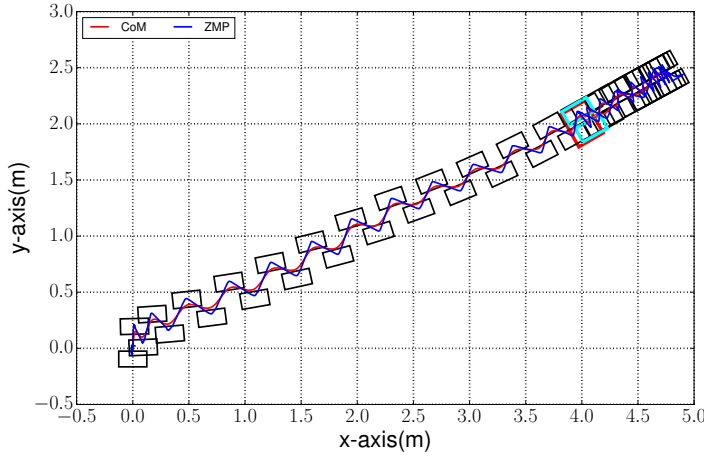
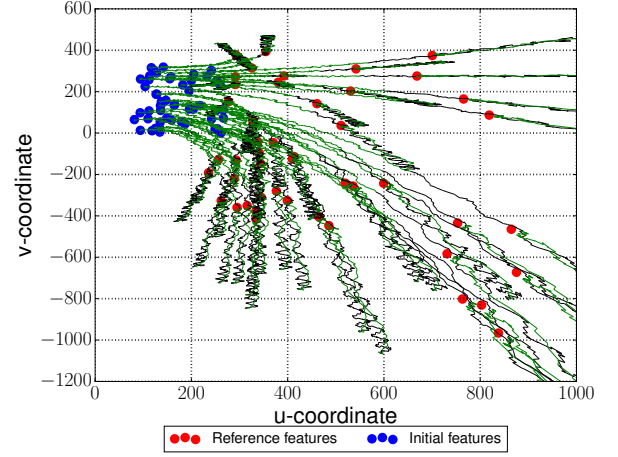
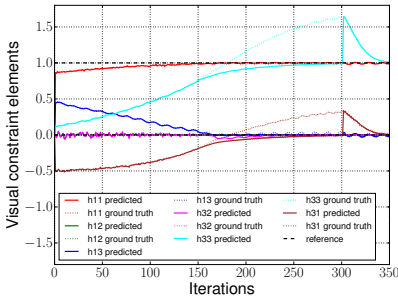


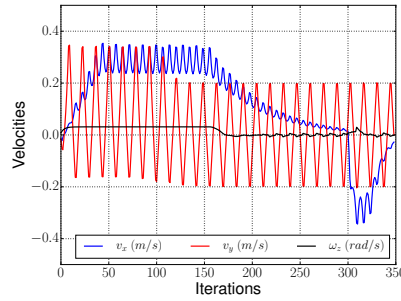
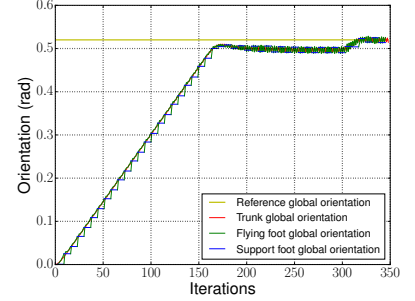
Figure 12: Simulation (300 iterations) of a single visual servoing task comparing the effect of including the acceleration term, with the homography-based locomotion linear setup (Sections 4.1 and 5.1). The desired pose is $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{ deg})$. Top row: the acceleration term in Eq. 15 is not included. Bottom row: the acceleration term is included. Left column: footsteps and paths of the robot CoM and ZMP in the x,y plane. Right column: evolution of the velocities in frame m_k .


 (a) Footsteps and paths of the robot CoM and ZMP in the x,y plane.


(b) The portion of the black trajectories corresponds to the period of total occlusion of the visual features.



(c) Evolution of the homography elements.


 (d) Evolution of the velocities in frame m_k .


(e) Evolution of the trunk, flying foot and support foot orientation (in the global frame).

Figure 13: Simulation (350 iterations) of a single visual servoing task with separate position and orientation control for the homography-based locomotion linear setup (Sections 4.1 and 5.1), including a total occlusion of the visual features between iterations 150 and 300. The desired pose is $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{deg})$.

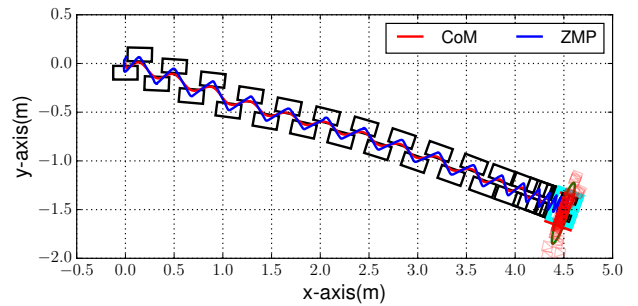
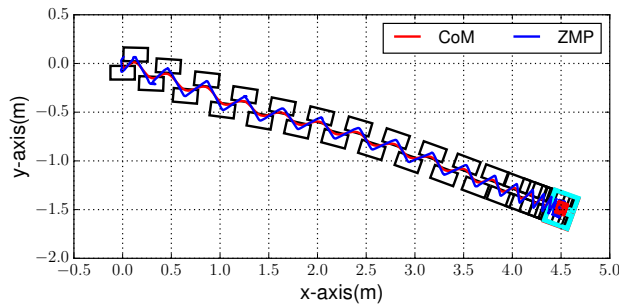


Figure 14: 200 simulations (400 iterations for each simulation) of a single visual servoing task comparing the effect of imprecise camera intrinsic parameters (30% error on the focal length, 25% error on the principal point), with the homography-based locomotion linear setup (Sections 4.1 and 5.1). The desired pose is $\mathbf{q}_r = (4.5\text{m}, -1.5\text{m}, -20\text{deg})$. Left: real camera intrinsic parameters. Right: imprecise camera intrinsic parameters.

image in the visual path to follow. It is written as:

$$\frac{1}{2} \sum_{i=1}^{N_I(k)} \rho(i) \left[\sum_{l=1}^M \beta_l [\bar{\mathbf{h}}_{k,l}^{r_k(i)} - \hat{\mathbf{h}}_{k,l}^{r_k(i)}]^T \mathbf{W} [\bar{\mathbf{h}}_{k,l}^{r_k(i)} - \hat{\mathbf{h}}_{k,l}^{r_k(i)}] \right], \quad (39)$$

where $N_I(k)$ is the number of images remaining in the visual path to be followed from k . This is a weighted sum of the visual errors for all the reference images $\mathcal{I}^{r_k(i)}$, where $\rho(i)$ is a function of the image index i . The weights $\rho(i)$ are positive and sum to one. We select them empirically to give priority to the closest reference images, with smaller

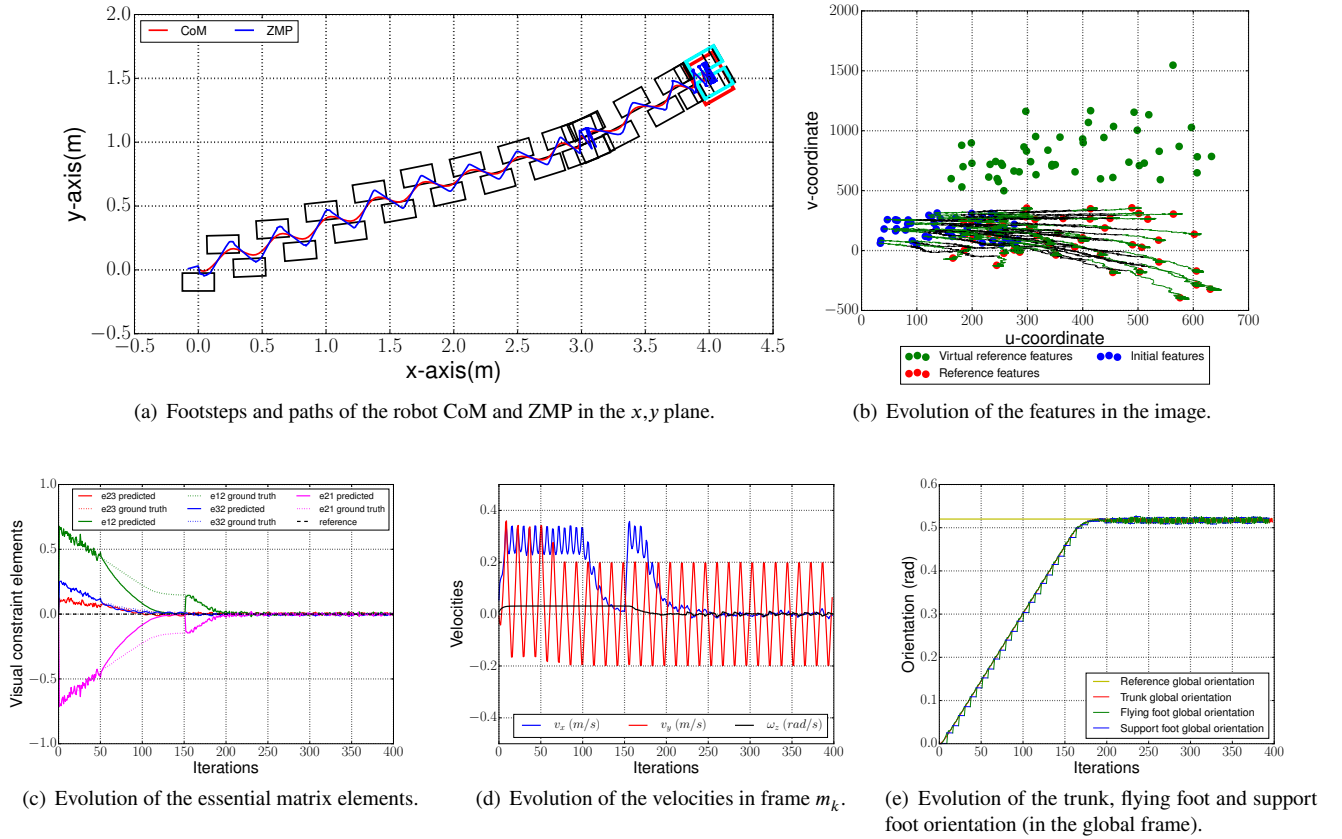


Figure 15: Simulation (400 iterations) of a single visual servoing task with separate position and orientation control for the essential matrix-based locomotion linear setup (Sections 4.1 and 5.2), including a total occlusion of the visual features between iterations 50 and 150. The desired posed is $\mathbf{q}_r = (4m, 1.5m, 30 \text{ deg})$.

Table 1
Final CoM error to the desired position in meters.

Desired orientation (deg)	Linear approach	Non-linear approach with Eq. 37	Non-linear approach with Eq. 38
0	0.61	0.21	0.15
5	0.47	0.20	0.12
10	0.43	0.20	0.13
15	0.43	0.20	0.12
20	0.41	0.20	0.12
25	0.40	0.19	0.12
30	0.13	0.25	0.17
35	0.29	0.25	0.17
40	0.56	0.25	0.17
45	0.57	0.25	0.17
50	0.60	0.25	0.18
Mean	0.45	0.22	0.15
Std. dev.	0.138	0.027	0.026

Table 2
Absolute value of errors in orientation in degrees.

Desired orientation (deg)	Linear approach	Non-linear approach with Eq. 37	Non-linear approach with Eq. 38
0	0.6	1.1	1.1
5	0.2	0.2	0.7
10	0.3	0.3	0.9
15	0.1	0.5	0.5
20	0.1	0.6	0.6
25	0.4	0.2	0.8
30	0.2	0.2	0.2
35	0.1	0.1	0.6
40	0.5	0.5	0.5
45	0.3	0.3	0.3
50	0.4	0.2	0.2
Mean	0.3	0.4	0.6
Std. dev.	0.17	0.29	0.29

weights to indexes farther from 1, and 0 to the reference images that do not have point correspondences with the current image \mathcal{I}^k . Since the number of reference images $N_I(k)$ decreases with k , the function $\rho(i)$ is adjusted to deal with the remaining reference images.

Note that the term of Eq. 39 preserves the quadratic problem structure, and only induces small, straightforward modifications to matrices \mathbf{Q}_k and \mathbf{q}_k of the QP.

The switch from a subset of reference images $\{r_k(1), \dots, r_k(N_I(k))\}$ to the next one $\{r_{k+1}(1), \dots, r_{k+1}(N_I(k+1))\}$ is

Humanoid Locomotion from Visual Constraints

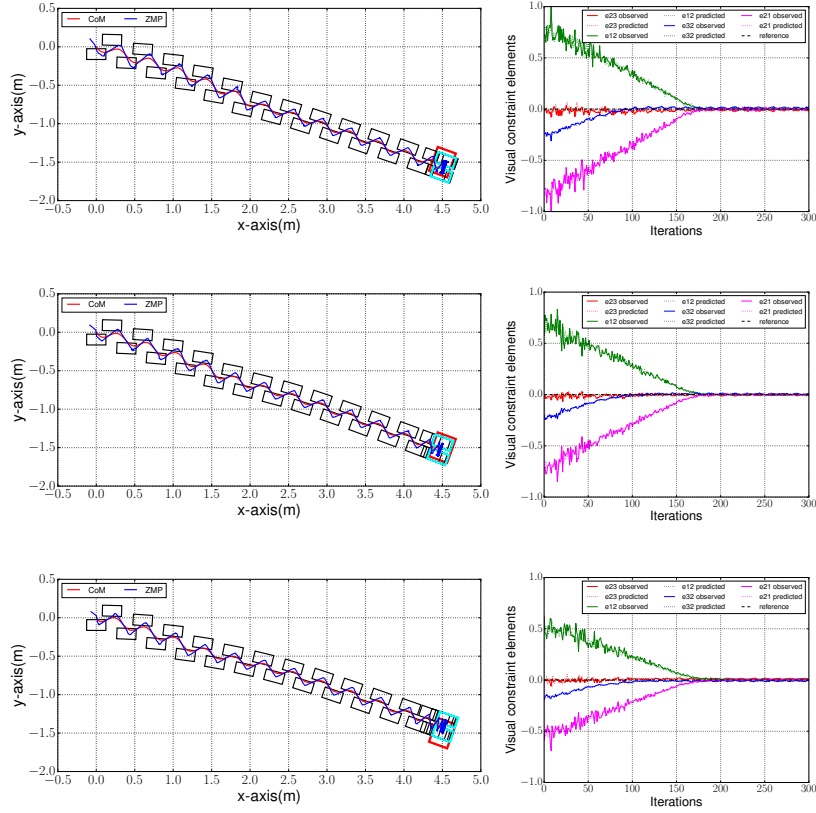


Figure 16: Simulation (300 iterations) of a single visual servoing task under the essential matrix-based locomotion linear setup (Sections 4.1 and 5.2) with three different values for the virtual camera height. The desired pose is $\mathbf{q}_r = (4.5\text{m}, -1.5\text{m}, -20\text{ deg})$. Top row: h is fixed to 6m along the walk. Middle row: h is fixed to 7m. Bottom row: h is fixed to 9m.

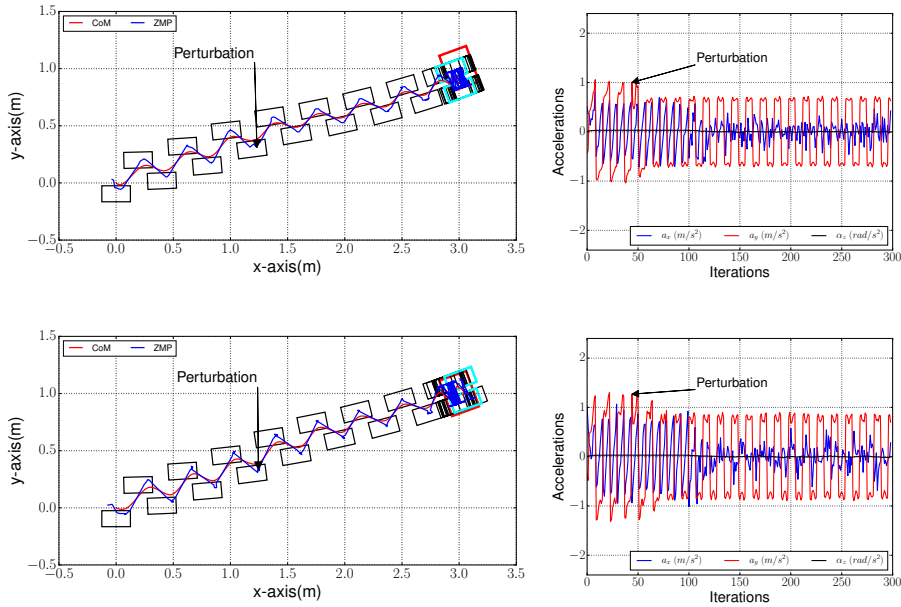


Figure 17: Simulation (300 iterations) of a single visual servoing task with a transient perturbation, using the essential matrix-based locomotion linear setup (Sections 4.1 and 5.2). The desired pose is $\mathbf{q}_r = (3.0\text{m}, 1.0\text{m}, 20\text{ deg})$. Top row: the acceleration term in Eq. 15 is included. Bottom row: this term is not included. Left column: Footsteps and paths of the CoM and ZMP in the x,y plane. The time instant and the orientation of the perturbation are indicated by the black arrow. Right column: accelerations.

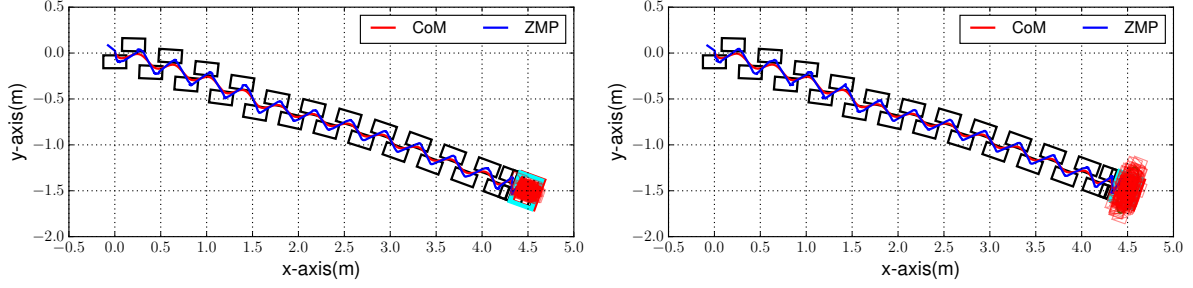


Figure 18: 200 simulations (300 iterations for each simulation) of a single visual servoing task comparing the effect of including imprecise camera intrinsic parameters (30% error on the focal length, 25% error on the principal point), with the essential matrix-based locomotion linear setup (Sections 4.1 and 5.2). The desired pose is $\mathbf{q}_r = (4.5\text{m}, 1.5\text{m}, -20\text{deg})$. Left: correct camera intrinsic parameters. Right: imprecise camera intrinsic parameters.

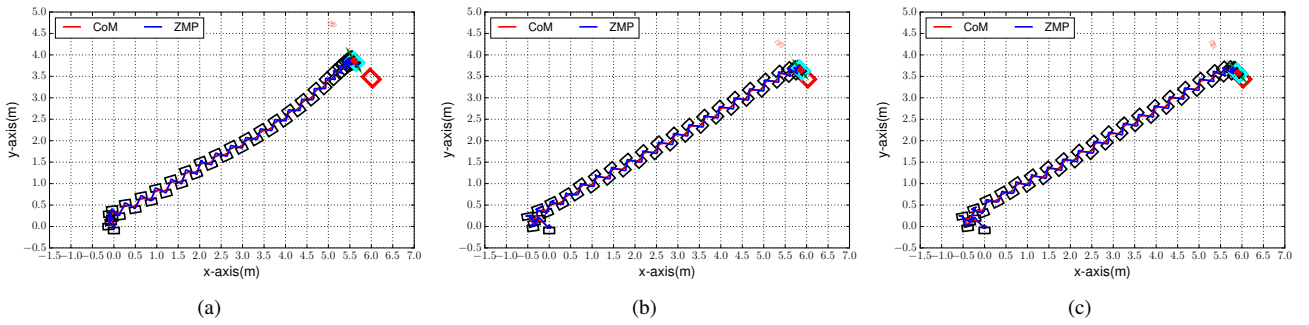


Figure 19: Simulations (600 iterations) of a single visual servoing task comparing the linear (Section 4.1) versus the non-linear (Section 4.3) approach, with the homography-based locomotion (Section 5.1). The desired pose is $\mathbf{q}_r = (6.0\text{m}, 3.46\text{m}, 45\text{deg})$. (a) Linear approach. (b) Non-linear approach with Eq. 37. (c) Non-linear approach with Eq. 38.

Table 3

Time to solve the (S)QP in milliseconds.

Desired orientation (deg)	Linear approach	Non-linear approach with Eq. 37	Non-linear approach with Eq. 38
0	9.1	18.2	18.2
5	9.0	18.3	18.2
10	9.1	17.9	17.8
15	9.0	17.8	17.8
20	9.1	17.8	17.7
25	9.0	17.8	17.8
30	9.0	17.8	17.9
35	8.9	17.8	17.8
40	8.9	18.0	17.9
45	8.9	18.1	18.0
50	9.2	18.3	18.2
Mean	9.0	18.0	17.9
Std. dev.	2.81	6.89	6.86

triggered when the visual error goes below a threshold. This error is measured as the simple moving average (SMA) of the last S values of the visual features $\mathbf{h}_{k,t}^{r_k(1)}$, evaluated for the first reference image $d_t = \bar{\mathbf{h}}_{k,t}^{r_k(1)} - \mathbf{h}_{k,t}^{r_k(1),\text{SMA}}$. The switch

occurs when, for all the features t ,

$$|d_t| < \epsilon \quad (40)$$

for a specified $\epsilon > 0$. In our experiments, we evaluate these conditions only during the single support foot change.

6.3.2. Shared prediction windows.

The idea for this approach is to use the following errors:

$$\hat{\mathbf{h}}_{k+1,t} = \left(\hat{h}_{k+1,t}^{r_k(1)}, \hat{h}_{k+2,t}^{r_k(1)}, \dots, \hat{h}_{k+S,t}^{r_k(1)}, \hat{h}_{k+S+1,t}^{r_k(2)}, \dots, \hat{h}_{k+N,t}^{r_k(2)} \right)^T, \quad (41)$$

where the first part of the vector is relative to the next reference image, $r_k(1)$, and the second part is relative to the following reference image, $r_k(2)$ (when it exists). The time index S is determined by using the previous optimization solution \mathbf{u}_k^* , and evaluating the predicted errors for the next reference image $r_k(1)$ in the prediction window, $d_t = \bar{\mathbf{h}}_{k,t}^{r_k(1)} - \mathbf{h}_{k,t}^{r_k(1)}(\mathbf{u}_k^*)$ and defining:

$$S = \begin{cases} \min\{l \in [1, N] \text{ s.t. } |d_t(l)| < \epsilon \text{ for all } t\} & \text{if this set } \neq \emptyset \\ N & \text{otherwise.} \end{cases}$$

The visual term of the objective function keeps a similar form as the original one, since

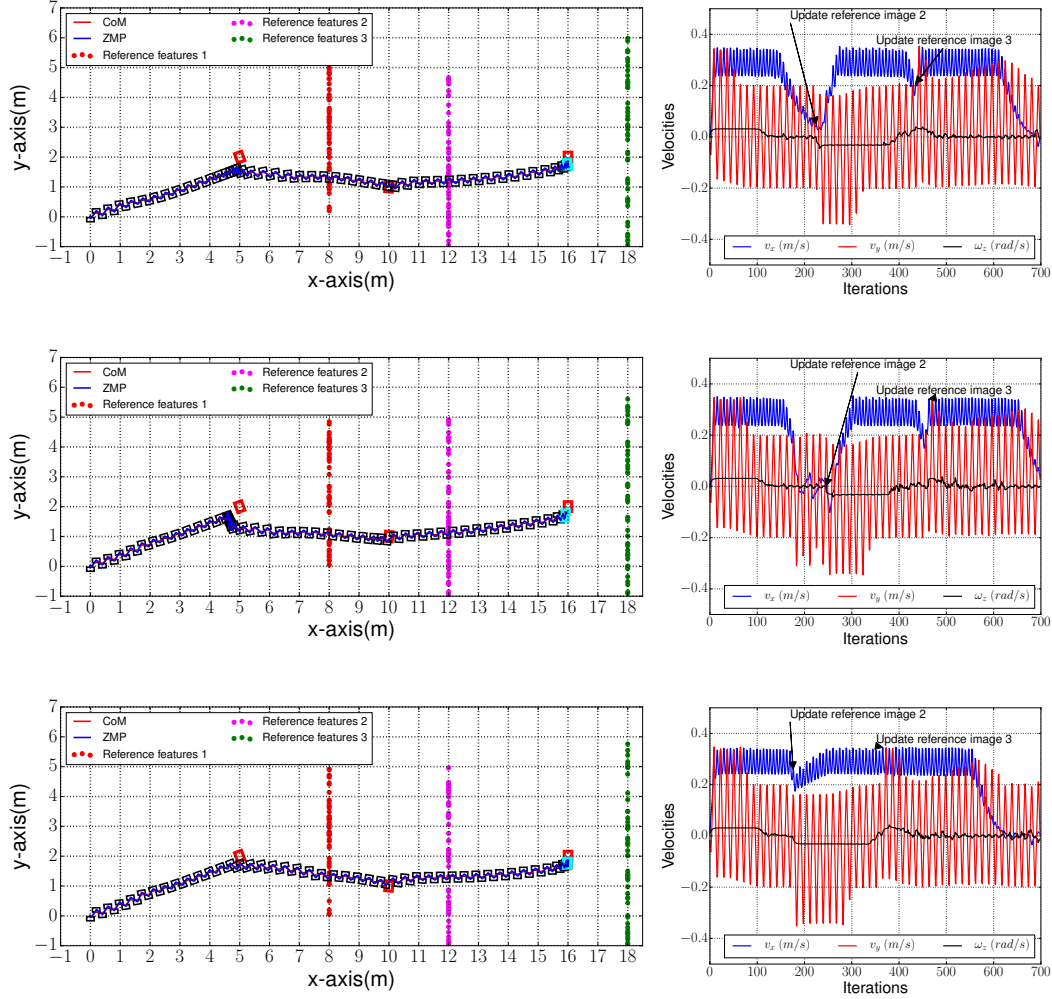


Figure 20: Simulations (700 iterations) of a visual path-following with the homography-based locomotion (Section 5.1) using 3 images. The desired pose is $\mathbf{q}_r = (16\text{m}, 2\text{m}, 0 \text{ deg})$.

$$\hat{\mathbf{h}}_{k,t} = \mathbf{D}_S \hat{\mathbf{h}}_{k,t}^{r_k(1)} + (\mathbf{I} - \mathbf{D}_S) \hat{\mathbf{h}}_{k,t}^{r_k(2)},$$

where \mathbf{D}_S is a diagonal matrix with ones in its first S elements, and zero afterwards. Hence, the visual term of the objective function is kept as a quadratic. The switching policy is kept similar to the one above.

6.3.3. Evaluation of the path-following approaches.

In Fig. 20, we evaluate the proposed approaches for following visual paths. The setup for this evaluation is similar to the previous ones, but with now a set of 3 consecutive reference images. Each reference image is associated to a different plane used for homography estimation (depicted with different colors in Fig. 20). The desired poses for the reference images are $\mathbf{q}_{r(1)} = (5\text{m}, 2\text{m}, 20 \text{ deg})$, $\mathbf{q}_{r(2)} = (10\text{m}, 1\text{m}, -5 \text{ deg})$ and $\mathbf{q}_{r(3)} = (16\text{m}, 2\text{m}, 0 \text{ deg})$. In the top row, we depict the results obtained with the scheme of Eq. 39, with the weights set as $\rho(1) = 0.75$, $\rho(2) = 0.2$ and $\rho(3) = 0.05$. At the switches between reference images, a sharp decrease in velocity is

visible. In the second row, we depict the results obtained with the scheme of Eq. 41. Again, sharp velocity decreases occur at transitions. As a third experiment (bottom row), we use the same scheme as in Eq. 41 and introduce a new term in the objective function to make \dot{X}_k as close as possible to a fixed, reference velocity, whenever new reference images do exist in the visual path. As it can be seen, the longitudinal velocity transitions are now much smoother in the vicinity of the reference image.

6.4. Experiments with a dynamic simulator

Finally, our approach was validated using the dynamic simulator *Pymanoid*¹, a humanoid robotics controller prototyping environment based on *OpenRAVE* [8]. The inverse kinematics included in *Pymanoid* and used here is based on a quadratic programming formulation [10]. In the simulations described in this section, we have used the Japan Virtual Robotics Challenge (*JRVC-1*) humanoid robot [8]. The initial pose of the CoM in the motion plane is always

¹<https://github.com/stephane-caron/pymanoid>

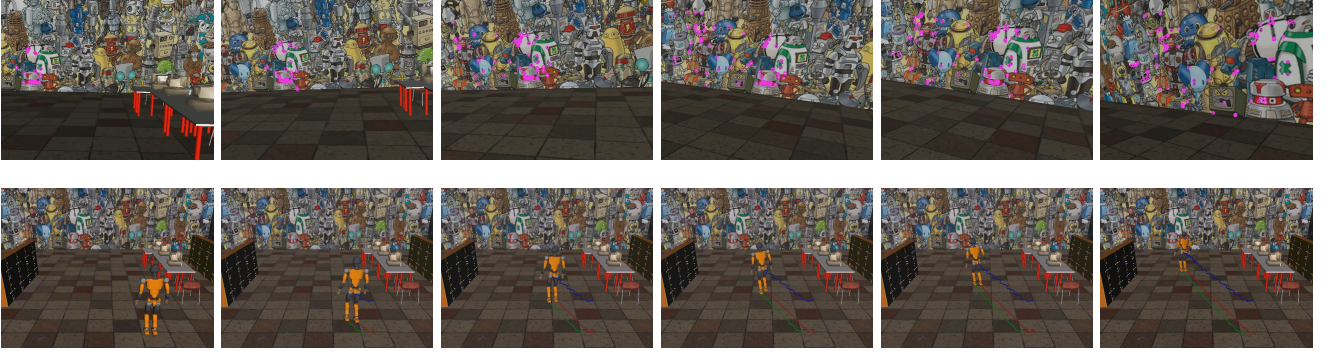


Figure 21: Samples of images taken from the robot camera (top row) and from an external camera (bottom row) during the experiment of pose regulation using a dynamic simulator for the footsteps shown in Fig. 22(a).

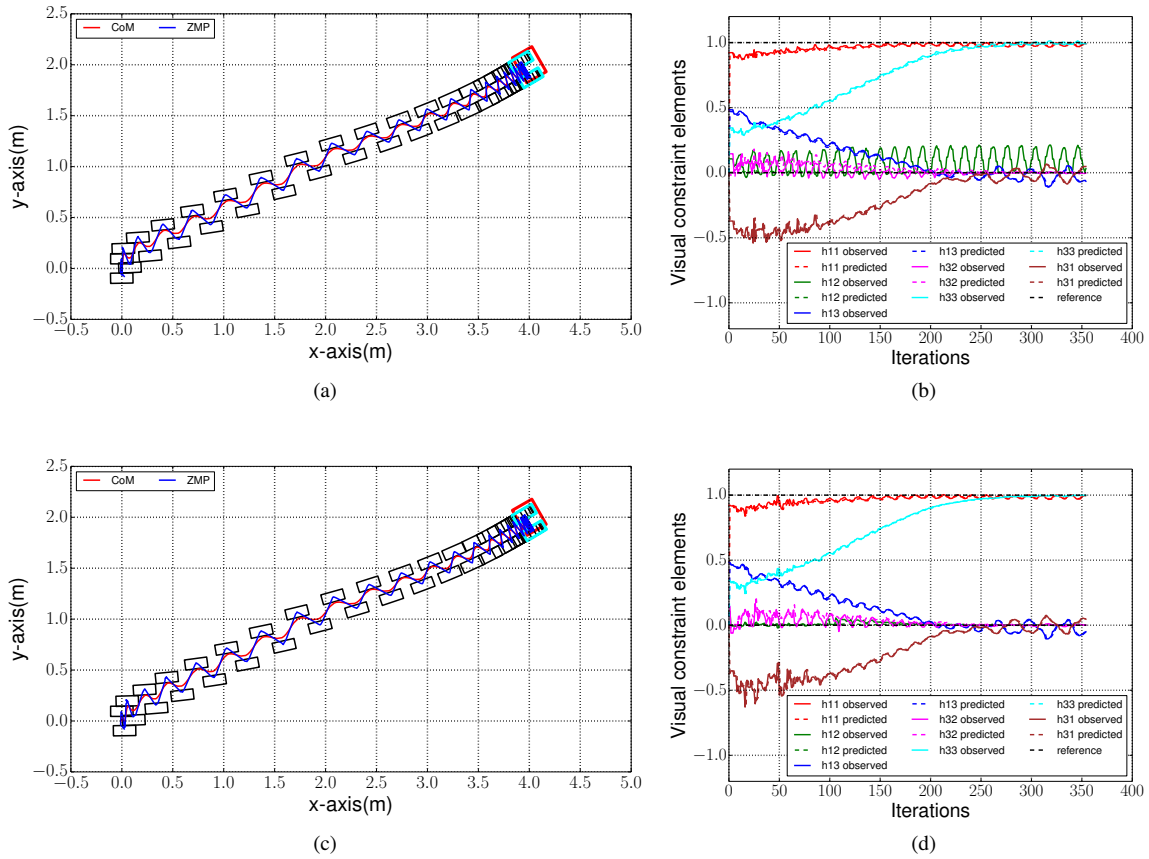


Figure 22: Simulation (350 iterations) of a homography-based locomotion experiment (Section 5.1) using a dynamic simulator. The desired pose is $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{ deg})$. Top row: using the raw homography matrix. Bottom row: using the rectified homography matrix. Left column: footsteps and paths of the robot CoM and ZMP in the x, y plane. Right column: evolution of the homography matrix elements.

taken as $\mathbf{q}_0 = (0\text{m}, 0\text{m}, 0\text{ deg})$, while the final pose \mathbf{q}_r varies. In the reported results, the termination condition is given by a maximal number of iterations of 350. The weights in the objective function are kept constant as $\alpha = 1e^{-4}$, $\gamma = 10$, $\eta = 0.025$, $\alpha_R = 0.06$, $\beta_R = 100$ and $\gamma_R = 100$. The weights on the visual features are $\beta_1 = 1.0$, $\beta_2 = 1.0$, $\beta_3 = 3.0$, $\beta_4 = 0.5$, $\beta_5 = 1.0$, $\beta_6 = 1.0$.

The linear approach for homography-based visual predictive control, described in Sections 4.1 and 5.1, is evaluated here for a visual servoing experiment with a single reference image. The images captured by the robot camera have a resolution of 640×480 pixels and are processed using the OpenCV library. We use ORB descriptors [28] as image features. RANSAC is applied to match robustly all the points

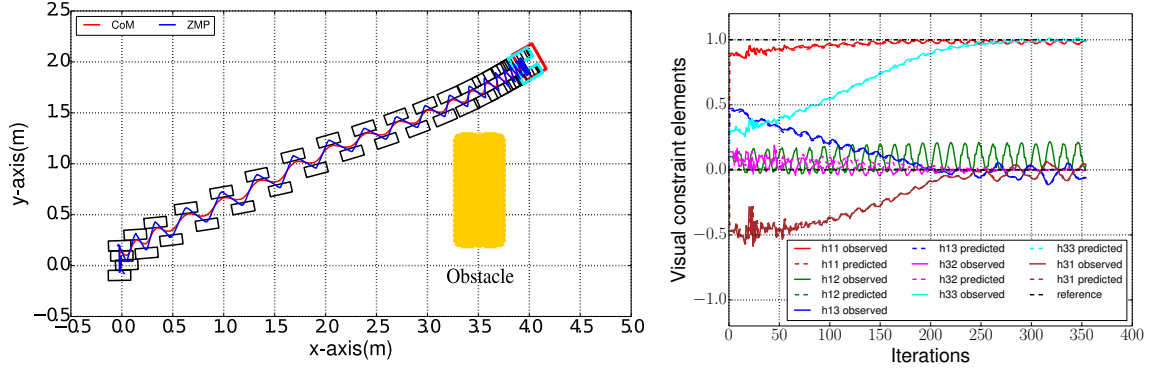


Figure 23: Simulation (350 iterations) of a homography-based locomotion experiment (Section 5.1) using a dynamic simulator and with partial occlusions. The desired pose is $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{deg})$. Left column: footsteps and paths of the robot CoM and ZMP in the x, y plane. Right column: evolution of the homography matrix elements.

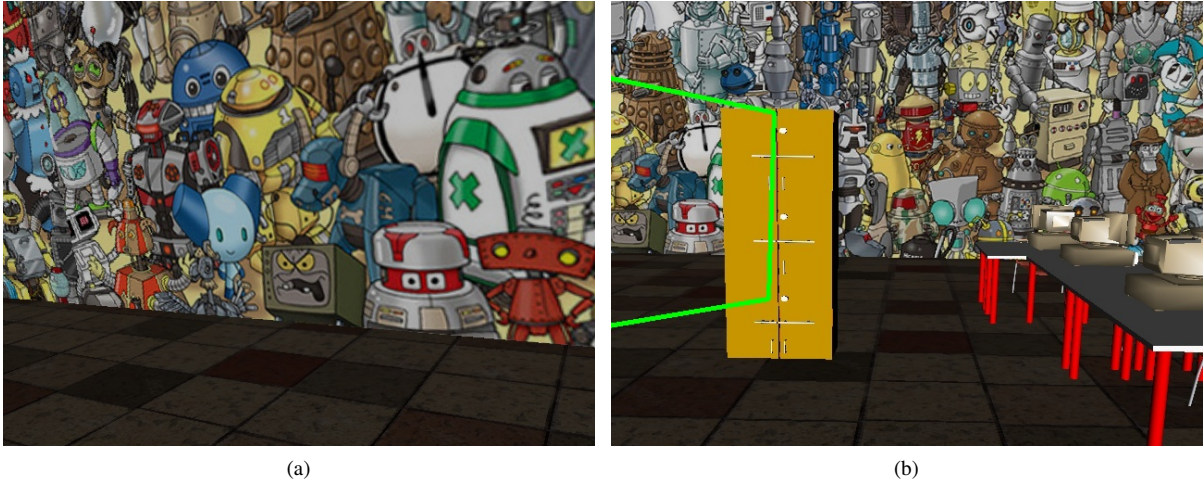


Figure 24: Reference and current images for the experiment with partial occlusion. Left column: reference image. Right column: current image in the third iteration.

between the current image and the reference one.

Fig. 21 shows the setup of the experiments in the dynamic simulator, through samples of images taken by the robot camera and by an external camera, respectively. We have included a *video as supplementary material* to show the experiments reported in this section. The gait pattern corresponding to the experiment of Fig. 21 is shown in Fig. 22(a), where the robot has to reach the desired pose $\mathbf{q}_r = (4\text{m}, 2\text{m}, 30\text{deg})$. We can see that the robot reaches the desired position quite accurately, both in position and orientation. As seen in Fig. 22(b), the element h_{12} is different from zero due to sway motion (see the video attachment to observe this effect). To mitigate this effect, we compensate the rotations in roll and yaw in the estimated homography, such that the rectified homography does not include those motions, as assumed in the model of Eq. 27. The homography is rectified as in Montijano et al. [23]:

$$\mathbf{H}_l^r = \mathbf{R}_{\epsilon_i} \mathbf{R}_{\delta_i} \mathbf{K}^{-1} \mathbf{H}_k^r \mathbf{K}, \quad (42)$$

where the roll (δ_i) and yaw (ϵ_i) angles are obtained from the decomposition of the raw homography as estimated from point matches. In Fig. 22(d), we can observe that the element h_{12} of the rectified homography is almost zero through the experiment. Note in Fig. 22 that, even without rectifying the homography, the visual control allows the robot to reach the desired pose, showing robustness of the scheme against no modeling effects. Thus, in the next experiment, no rectification of the homography is done.

In Fig. 23, we present the results of an experiment similar to the previous one, but with an obstacle partially occluding the features from the reference image. The obstacle is placed at the position (3.5m, 0.8m) in the world. Fig. 24 shows the reference image and the current image at the third iteration. The green shape depicts the viewpoint shared by both cameras. It can be seen that the obstacle occludes a part of this shared area. The robot reaches the desired position in spite of the partial occlusion, showing the robustness of the approach and the advantage of using locomotion based on

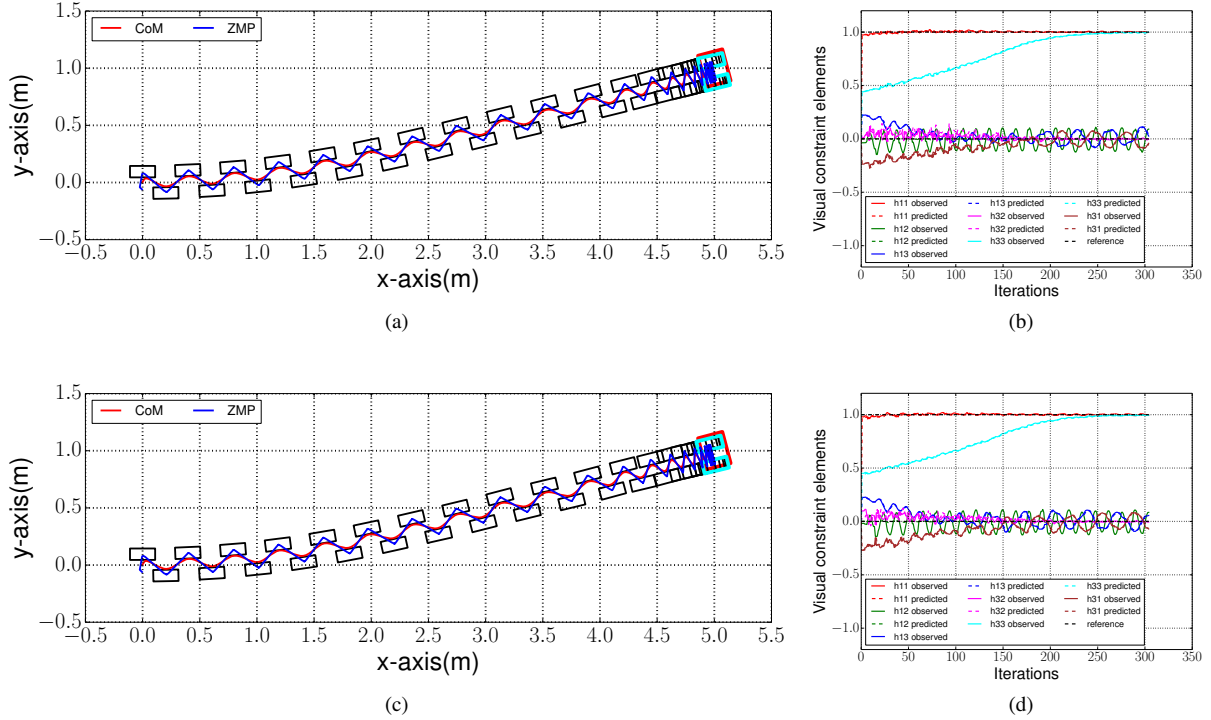


Figure 25: Simulation (300 iterations) of a homography-based locomotion experiment (Section 5.1) using a dynamic simulator. The desired pose is $\mathbf{q}_r = (5\text{m}, 1\text{m}, 15\text{deg})$. Top row: using the background image of the Fig. 22(a). Bottom row: using a less textured image as a background. Left column: footsteps and paths of the robot CoM and ZMP in the x, y plane. Right column: evolution of the homography matrix elements.

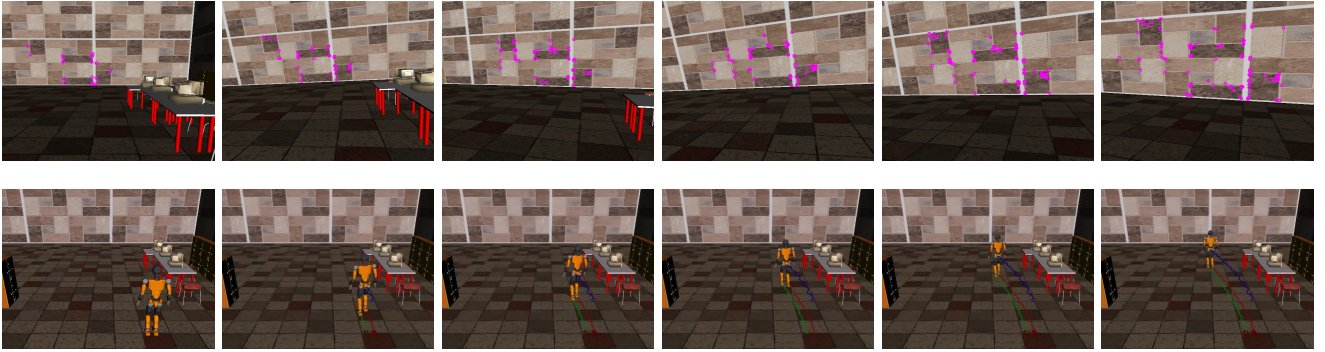


Figure 26: Image samples taken from the robot camera (top row) and from an external camera (bottom row) during the pose regulation experiment with a new background image also illustrated in Fig. 25(a).

geometric constraints.

Finally, in Fig. 25, we present two experiments led to evaluate the influence of the image texture to the visual control. The bottom row shows the results of a locomotion experiment with a background texture having much less interest points than the one used in the upper row, as seen in Fig. 26, where we depict onboard camera views and external views of the scene. The average of matched points is 91 for Fig. 25(c) and the mean of matched points for Fig. 25(a) is 135. As it can be seen, in spite of having much less interest points, the goal position is still achieved with high precision.

6.5. Comparison to SLAM-based navigation

In this last Section, through a couple of comparative experiments, we demonstrate the benefits of using an image-driven navigation method instead of using navigation on a metric map. For this purpose, in a first experiment illustrated in Fig. 27, we compare two standard, trivial locomotion experiments, where a given position \mathbf{q}_r has to be reached by the robot. This experiment is led in two ways: First (left side of Fig. 27), with the image-based algorithm described in Section 5.1, which only uses the current and final images to determine the control to execute; Second (right side of Fig. 27), with a state of the art visual SLAM system [24] and

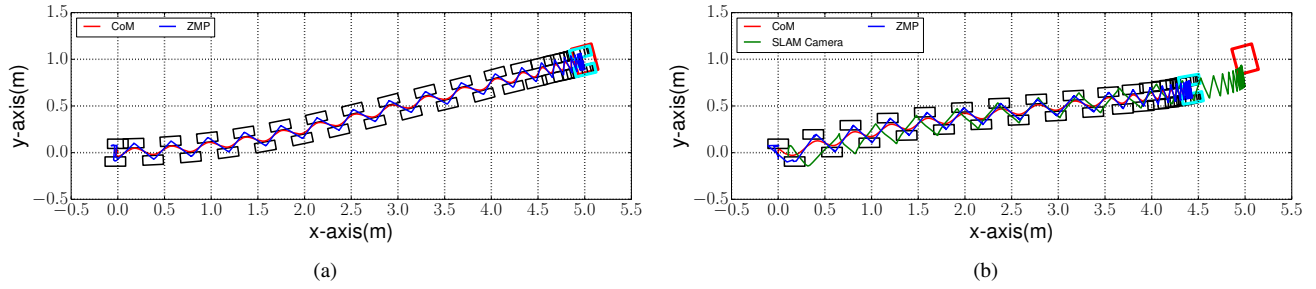


Figure 27: Simulation (300 iterations) of a humanoid locomotion experiment using a dynamic simulator. The desired pose is $\mathbf{q}_r = (5\text{m}, 1\text{m}, 15\text{deg})$. Left column: using an image-based visual servoing control strategy with the raw homography matrix (Section 5.1). Right column: using ORB-SLAM and a simple planning algorithm (Section 6.5). The footsteps and the path of the robot CoM and ZMP in the x, y plane (camera position in the ORB-SLAM experiment) are respectively depicted in blue, red, and blue. The SLAM camera pose estimate appears in green, which reaches close to the target pose not so the robot's CoM.

Table 4

Computational times in milliseconds.

Homography+QPs	Tracking+QPs	Mapping updates
72	65	423

a standard humanoid locomotion algorithm [15], where the reference velocity is updated based on the relative position to the goal. One can see that the precision reached in the first case is higher; in the second case, estimation errors lead to a poor positioning to the objective. Of course, using maps have also advantages that are not illustrated in this experiment: They allow high-level reasoning when the planning aspect is critical (e.g., with obstacles).

Another benefit of using a visual servoing scheme is lower computational times. In Table 4 (left), we give average computational times for one process iteration of our method, which includes the homography computation and the solver call for the QP locomotion problem. On the last two columns, the times are given for the SLAM-based approach: Note that it includes two different sub-parts, one (at high frequency) for iterations of the tracking/localization, similar to the visual control approach, and one (at low frequency) for map updates. The latter is quite slow and is clearly a disadvantage vs. visual control-based approaches.

Finally, we have evaluated homography matrix-based locomotion and visual SLAM under modeling errors by introducing uncertain values in the camera intrinsic parameters \mathbf{K} . The errors were chosen randomly with 30% of standard deviation for focal length errors and 25% for the principal point errors. In Fig. 28, the histograms of the final position errors in 50 experiments, for each approach, are presented. As it can be seen, the final position errors, measured as a distance to the target position, are significantly reduced by using the homography approach.

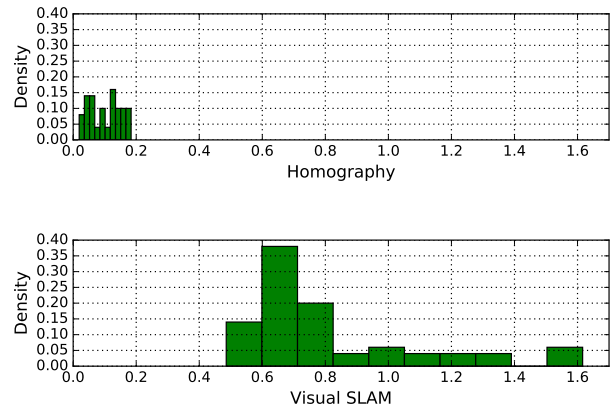


Figure 28: Distribution of the final CoM error to the desired position, in meters. Top row: Under our homography approach (Section 5.1). The mean is 0.1 and standard deviation is 0.048. Bottom row: Visual SLAM approach (Section 6.5). The mean is 0.809 and standard deviation is 0.279.

7. CONCLUSIONS

In this article, we have proposed a novel visual control approach for humanoid robots that uses visual errors extracted from two-view visual constraints such as homographies and essential matrices in a Model Predictive Control scheme for visual walking pattern generation. We have shown that this approach can simultaneously solve the visual servoing tasks and the walking pattern generation by only using visual information. The controls optimization is done in a local reference frame, namely in the current CoM reference frame, which differs from most existing approaches. The optimization problem to solve for the controls has been formulated in two ways: linear and nonlinear; the first one with the advantage of lower computational cost and the second higher accuracy of the regulation task. As with any visual-based method (e.g., visual SLAM), the success of this approach depends on the presence of textured scenes where feature points can be extracted to estimate the visual constraints. However, it has shown to be robust to losses of point features (due to missed correspondences or occlusions) and to

inaccuracies on camera parameters. The proposed approach has been extended to the problem where the humanoid has to follow a sequence of target images (visual path). As a future work, we plan to evaluate the use of other geometric constraints such as the trifocal tensor, as well as the use of more degrees of freedom, in particular the head angles of the robot.

Acknowledgments

This work was supported by CONACYT.

References

- [1] Allibert, G., Courtial, E., Chaumette, F., 2010. Visual servoing via nonlinear predictive control, in: Chesi, G., Hashimoto, K. (Eds.), *Visual Servoing via Advanced Numerical Methods*. Springer London. volume 401 of *Lecture Notes in Control and Information Sciences*, pp. 375–393.
- [2] Becerra, H.M., López-Nicolás, G., Sagüés, C., 2011. A sliding-mode-control law for mobile robots based on epipolar visual servoing from three views. *IEEE Transactions on Robotics* 27, 175–183.
- [3] Benhimane, S., Malis, E., 2007. Homography-based 2d visual tracking and servoing. *The International Journal of Robotics Research* 26, 661–676.
- [4] Bradski, G., 2000. *The OpenCV Library*. Dr. Dobb's Journal of Software Tools .
- [5] Chaumette, F., Hutchinson, S., 2006. Visual servo control, part i: Basic approaches. *IEEE Robotics and Automation Magazine* 13, 82–90.
- [6] Delfin, J., Becerra, H.M., Arechavaleta, G., 2016. Visual servo walking control for humanoids with finite-time convergence and smooth robot velocities. *International Journal of Control* 89, 1342–1358.
- [7] Delfin, J., Becerra, H.M., Arechavaleta, G., 2018. Humanoid navigation using a visual memory with obstacle avoidance. *Robotics and Autonomous Systems* 109, 109 – 124.
- [8] Diankov, R., 2010. *Automated Construction of Robotic Manipulation Programs*. Ph.D. thesis. Carnegie Mellon University, Robotics Institute.
- [9] Dune, C., Herdt, A., Stasse, O., Wieber, P.B., Yoshida, E., Yokoi, K., 2010. Cancelling the sway motion of dynamic walking in visual servoing, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3175–3180.
- [10] Escande, A., Mansard, N., Wieber, P.B., 2014. Hierarchical quadratic programming: Fast online humanoid-robot motion generation. *The International Journal of Robotics Research* 33, 1006–1028.
- [11] Faragasso, A., Oriolo, G., Paolillo, A., Vendittelli, M., 2013. Vision-based corridor navigation for humanoid robots, in: *IEEE/RAS International Conference on Robotics and Automation*, pp. 3175–3180.
- [12] Ferreau, H., Kirches, C., Potschka, A., Bock, H., Diehl, M., 2014. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation* 6, 327–363.
- [13] García, M., Stasse, O., Hayet, J.B., Dune, C., Esteves, C., Laumond, J.P., 2015. Vision-guided motion primitives for humanoid reactive walking: Decoupled versus coupled approaches. *The International Journal of Robotics Research* 34, 402–419.
- [14] Hartley, R., Zisserman, A., 2006. *Multiple View Geometry in computer vision*. Second ed., Cambridge University Press.
- [15] Herdt, A., Holger, D., Wieber, P.B., Dimitrov, D., Mombaur, K., Moritz, D., 2010. Online walking motion generation with automatic foot step placement. *Advanced Robotics* 24, 719–737.
- [16] Ido, J., Shimizu, Y., Matsumoto, Y., Ogasawara, T., 2009. Indoor navigation for a humanoid robot using a view sequence. *The International Journal of Robotics Research* 28, 315–325.
- [17] Kajita, S., Kanehiro, F., Kaneko, K., Fujiwara, K., Harada, K., Yokoi, K., 2003. Biped walking pattern generation by using preview control of zero-moment point, in: *IEEE/RAS International Conference on Robotics and Automation*, pp. 1620–1626.
- [18] Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., Akachi, K., Isozumi, ., 2004. Humanoid robot HRP-2, in: *IEEE/RAS International Conference on Robotics and Automation*, pp. 1083–1090.
- [19] López-Nicolás, G., Sagüés, C., 2011. Vision-based exponential stabilization of mobile robots. *Autonomous Robots* 30, 293–306.
- [20] López-Nicolás, G., Sagüés, C., Guerrero, J.J., 2007. Homography-based visual control of nonholonomic vehicles, in: *IEEE/RAS International Conference on Robotics and Automation*, pp. 1703–1708.
- [21] López-Nicolás, G., Sagüés, C., Guerrero, J.J., 2009. Parking with the essential matrix without short baseline degeneracies, in: *IEEE/RAS International Conference on Robotics and Automation*, pp. 1098–1103.
- [22] Ma, Y., Soatto, S., Kosecka, J., Sastry, S.S., 2003. *An invitation to 3-D vision: From images to geometrical approaches*. Springer-Verlag.
- [23] Montijano, E., Cristofalo, E., Zhou, D., Schwager, M., Sagüés, C., 2016. Vision-based distributed formation control without an external positioning system. *IEEE Transactions on Robotics* 32, 339–351.
- [24] Mur-Artal, Raúl, M.J.M.M., Tardós, J.D., 2015. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics* 31, 1147–1163.
- [25] Paolillo, A., Faragasso, A., Oriolo, G., Vendittelli, M., 2017. Vision-based maze navigation for humanoid robots. *Autonomous Robots* 41, 293–309.
- [26] Remazeilles, A., Chaumette, F., 2007. Image-based robot navigation from an image memory. *Robotics and Autonomous Systems* 55, 345–356.
- [27] Rives, P., 2000. Visual servoing based on epipolar geometry, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 602–607.
- [28] Rublee, E., Rabaud, V., Konolige, K., Bradski, G., 2011. ORB: An efficient alternative to SIFT or SURF, in: *IEEE International Conference on Computer Vision*, pp. 2564–2571.
- [29] Scona, R., Nobili, S., Petillot, Y.R., Fallon, M., 2017. Direct visual slam fusing proprioception for a humanoid robot, in: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1419–1426.
- [30] Stasse, O., 2018. *Self-localization and Map building*. Springer. chapter Self-localization and Map building.
- [31] Stasse, O., Verrelst, B., Davison, A., Mansard, N., Saïdi, F., Vanderborcht, B., Esteves, C., Yokoi, K., 2008. Integrating walking and vision to increase humanoid autonomy. *International Journal of Humanoid Robotics, special issue on Cognitive Humanoid Robots* 5, 287–310.
- [32] Triggs, B., 1998. Autocalibration from planar scenes, in: Burkhardt, H., Neumann, B. (Eds.), *Computer Vision - ECCV'98*. Springer Berlin Heidelberg. volume 1406 of *LNCS*, pp. 89–105.
- [33] Vukobratovic, M., Borovac, B., 2004. Zero-moment point: thirty five years of its life. *International Journal of Humanoid Robotics* 1, 157–173.
- [34] Wieber, P.B., 2006. Trajectory Free Linear Model Predictive Control for Stable Walking in the Presence of Strong Perturbations, in: *IEEE/RAS International Conference on Humanoid Robots*, Genova, Italy. pp. 137–142.