HAL
open science

Closed-loop MPC with Dense Visual SLAM-Stability through Reactive Stepping

Arnaud Tanguy, Daniele de Simone, Andrew I. Comport, Giuseppe Oriolo, Abderrahmane Kheddar

To cite this version:

Arnaud Tanguy, Daniele de Simone, Andrew I. Comport, Giuseppe Oriolo, Abderrahmane Kheddar. Closed-loop MPC with Dense Visual SLAM-Stability through Reactive Stepping. 2018. hal-01883725

HAL Id: hal-01883725

https://hal.science/hal-01883725

Preprint submitted on 28 Sep 2018

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

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

Closed-loop MPC with Dense Visual SLAM - Stability through Reactive Stepping

Arnaud Tanguy[1,2*], Daniele De Simone[4*], Andrew I. Comport[1], Giuseppe Oriolo[4] and Abderrahmane Kheddar[3,2]

***Abstract*— Model Predictive Control (MPC) is a widely used technique for humanoid gait generation due to its capability to handle several constraints that characterize humanoid locomotion. The use of simplified models to describe the humanoid dynamics (the Linear Inverted Pendulum) allows to perform computations in real time, giving the robot the fundamental capacity to replan its motion to follow external inputs (e.g. reference velocity, footstep plans). However, usually the MPC does not take into account the current state of the robot when computing the reference motion, losing the ability to react to external disturbances. In this paper a closed-loop MPC scheme is proposed to estimate the robot's real state through Simultaneous Localization and Mapping (SLAM) and proprioceptive sensors (force/torque). With the proposed control scheme it is shown that the robot is able to react to external disturbances (push), by stepping to recover from the loss of balance. Moreover the localization allows the robot to navigate to target positions in the environment without being affected by the drift generated by imperfect open-loop control execution. We validate the proposed scheme through two different experiments with a HRP-4 humanoid robot.**

I. Introduction

In recent years, humanoid robots have been given increasingly more attention due to their ability to perform complex tasks, thanks to their highly redundant kinematic structure. However, those systems are also challenging to control because they have very complex dynamics. Therefore researchers tend to use simplified models to approximate the humanoid robot dynamics. When dealing with humanoid locomotion, the most commonly used model to approximate the dynamics of walking is the Linear Inverted Pendulum (LIP) [1], and the use of Model Predictive Control (MPC) has become predominant [2, 3]. In traditional MPC-based gait generation techniques, a reference motion is generated without taking into account the current robot state, however, planning a motion starting from the current robot state is often necessary to react to unexpected situations and to obtain more robust motions. In this paper the problem of closed-loop MPC is tackled [4, 5], and the robot state is estimated using SLAM and force/torque sensor measurements.

The use of SLAM for humanoid locomotion planning has been widely used, since the introduction of dense RGB-D approches, such as [6, 7] because they both provide a dense 3D map of the explored environment and 6D localization at sensor framerate with centimeteric precision. With this information, it becomes possible to plan robot actions *w.r.t.* the environment. In [8], a pair of Multisense stereo cameras are used to generate RGB-D inputs for use with Kintinuous SLAM [6]. A footstep plan for walking over a rough brick field is generated from its map, while the robot state is estimated with a kinematic-inertial estimator [9]. Footsteps are selected within the map, but no attempt is made to change the next robot's footstep should its state differ from a feasible one. In [10], kinematic-inertial measurements are used to improve the robustness of ElasticFusion [11], and are applied to a cartesian walking controller to repeatedly walk towards goal positions in the environment.

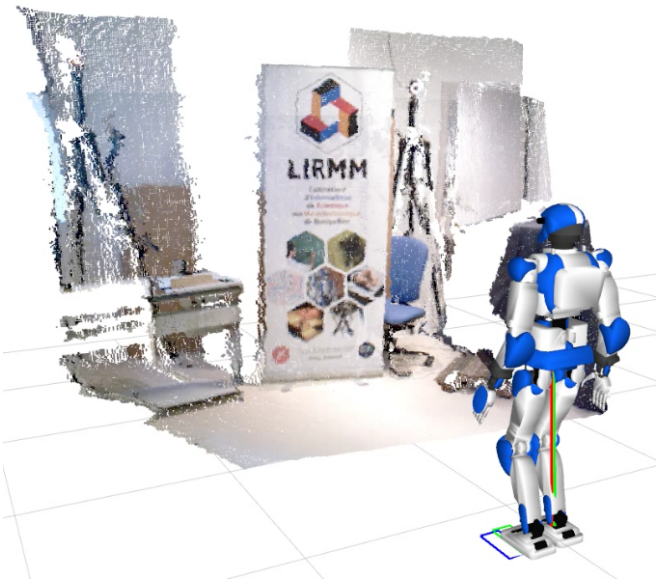

Fig. 1. HRP-4 walking with the proposed closed-loop MPC with dense visual SLAM feedback. The estimated LIP pendulum state obtained from SLAM and force-sensor measurements is shown as a red line going from the ZMP estimate to the robot's CoM. The green line represents the desired pendulum state. Blue and green squares are respectively the desired optimal footstep, and a non-optimal footstep that minimizes changes of the footstep trajectory.

II. MPC-based gait generation

When dealing with a complex system like a humanoid robot, it is a common practice to rely on a simplified model, the Linear Inverted Pendulum (LIP) [1] to describe the humanoid behavior during locomotion. The reason why the LIP is widely used when dealing with humanoid locomotion is because its dynamics approximate well the motion of the Center of Mass (CoM) of a biped robot, and the differential equations governing the two directions of motion x and y are linear, identical and decoupled. So, from now on, only the sagittal component of motion will be referred to.

This work is supported by grant from the COMANOID H2020 project

[1] CNRS-University of Nice Sophia Antipolis, I3S, France
[2] CNRS-University of Montpellier, LIRMM, Interactive Digital Humans, France
[3] CNRS-AIST Joint Robotics Laboratory, UMI3218/RL, Japan
[4] Sapienza University of Rome, DIAG, Italy
* Joint first authorship

Consider, without loss of generality, the evolution of the sagittal component of motion x

$$\begin{pmatrix} \dot{x}_\mathrm{c} \\ \ddot{x}_\mathrm{c} \\ \dot{x}_\mathrm{z} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ \eta^2 & 0 & -\eta^2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_\mathrm{c} \\ \dot{x}_\mathrm{c} \\ x_\mathrm{z} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \dot{x}_\mathrm{z} \quad (1)$$

where x_c is the position of the CoM, x_z the position of the Zero-Moment Point (ZMP), $\eta = \sqrt{g/h}$ and h the constant height of the CoM. In the motion model (1) we assume controls $\dot{x}_\mathrm{z}$ which are piece-wise constant over time intervals of duration δ.

The gait generation is based on the intrinsically stable MPC [12], where the decision variables are the ZMP velocities $(\dot{x}_\mathrm{z}^i, \dot{y}_\mathrm{z}^i)$, $i = 1, \ldots, N$ and the footstep positions and orientations $(x_\mathrm{f}^j, y_\mathrm{f}^j, \theta_\mathrm{f}^j)$, $j = 1, \ldots, M$, over a prediction horizon $T_h = N\delta$.

The choice of a MPC-based gait generation allows us to give the robot a high-level task, through an appropriate cost function to be minimized, and also to impose constraints to enforce stability, to maintain balance and to guarantee the kinematic feasibility of the robot motion.

A. Cost Function

In the formulation proposed here, the aim is to have the robot track high-level reference sagittal and coronal velocities $(v_\mathrm{x}, v_\mathrm{y})$, and an angular velocity ω, by an appropriate selection of ZMP velocities $(\dot{x}_\mathrm{z}^i, \dot{y}_\mathrm{z}^i)$ and footstep positions and orientations $(x_\mathrm{f}^j, y_\mathrm{f}^j, \theta_\mathrm{f}^j)$. However to maintain linearity in the MPC formulation, the footstep orientations must be chosen before the computation of their positions [2].

Therefore, a first optimization problem is solved, minimizing a function that takes into account the reference angular velocity ω to determine the footstep orientations θ_f^j

$$\sum_{j=1}^{M} \left(\frac{\theta_\mathrm{f}^j - \theta_\mathrm{f}^{j-1}}{T_s} - \omega \right)^2, \quad (2)$$

where T_s is the constant duration of a step, subject to the linear constraint $|\theta_\mathrm{f}^j - \theta_\mathrm{f}^{j-1}| \leq \theta_{max}$, that limits to θ_{max} the maximum difference in orientation between two consecutive footsteps.

Once the foot orientations are decided, the footstep locations and the ZMP velocities can be computed via the minimization of a second cost function

$$\begin{aligned} \sum_{i=1}^{N} \big(& (\dot{x}_\mathrm{z}^{k+i})^2 + (\dot{y}_\mathrm{z}^{k+i})^2 + \\ & k_x(\dot{x}_\mathrm{c}^{k+i} - v_x \cos(i\omega\delta) + v_y \sin(i\omega\delta))^2 + \\ & k_y(\dot{y}_\mathrm{c}^{k+i} - v_x \sin(i\omega\delta) - v_y \cos(i\omega\delta))^2 \big), \end{aligned} \quad (3)$$

where the first two terms penalize the control effort, while the last two terms are meant to minimize the deviation from the reference velocities (v_x, v_y).

B. Constraints

In the following the three constraints enforced in the proposed MPC scheme will be briefly presented. The first, introduced in [12], is the stability constraint, which takes the form

$$\frac{1}{\eta} \frac{1 - e^{\delta\eta}}{1 - e^{N\delta\eta}} \sum_{i=1}^{N} e^{i\delta\eta} \dot{x}_\mathrm{z}^{k+i} = x_\mathrm{c}^k + \frac{\dot{x}_\mathrm{c}^k}{\eta} - x_\mathrm{z}^k. \quad (4)$$

This constraint will guarantee the boundedness of the computed CoM trajectory.

In order to guarantee balance during locomotion, we must guarantee that the ZMP is at any time instant inside the robot current support polygon. To do so we define a rectangle with sides $d_x^\mathrm{z}, d_y^\mathrm{z}$, and therefore the balance constraint takes the form

$$R_j^T \begin{pmatrix} \delta \sum_{l=k}^{k+i-1} \dot{x}_\mathrm{z}^l - x_\mathrm{f}^j \\ \delta \sum_{l=k}^{k+i-1} \dot{y}_\mathrm{z}^l - y_\mathrm{f}^j \end{pmatrix} \leq \frac{1}{2} \begin{pmatrix} d_x^\mathrm{z} \\ d_y^\mathrm{z} \end{pmatrix} - R_j^T \begin{pmatrix} x_\mathrm{z}^k \\ y_\mathrm{z}^k \end{pmatrix}, \quad (5)$$

where R_j is the rotation matrix associated to the angle θ_f^j.

The last constraint is to guarantee that the choice of the next footstep is in a location that avoids self collisions and is inside the robot kinematic limits:

$$R_{j-1}^T \begin{pmatrix} x_\mathrm{f}^j - x_\mathrm{f}^{j-1} \\ y_\mathrm{f}^j - y_\mathrm{f}^{j-1} \end{pmatrix} \leq \pm \begin{pmatrix} 0 \\ \ell \end{pmatrix} + \frac{1}{2} \begin{pmatrix} d_x^\mathrm{f} \\ d_y^\mathrm{f} \end{pmatrix}, \quad (6)$$

where d_x^f and d_y^f are the sides of a rectangle defining the feasibility zone, and l is a reference distance between two consecutive footsteps.

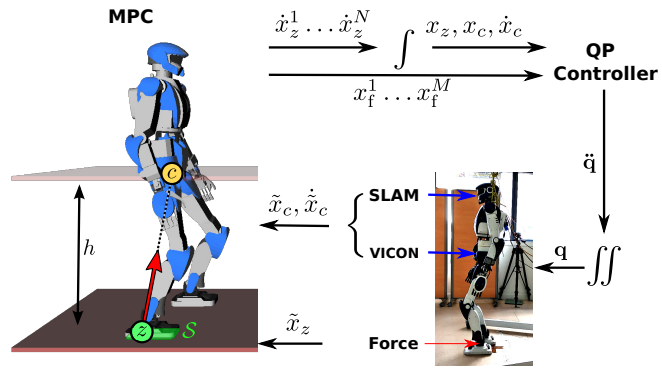

Fig. 2. Overview of the proposed closed-loop MPC. A LIP reduced model is used to represent the dynamics of a humanoid robot walking on flat floor. The pendulum state is estimated from HRP-4 sensors: its CoM position and velocity is obtained with dense visual slam from RGB-D measurements, and its ZMP from force-sensor measurements. This state is used to compute the MPC predictions, *i.e.* the future ZMP velocities and associated footsteps. Through integration, these references converted to reference position are then tracked by a quadratic programming controller that sends a desired whole-body configuration to the position-controlled robot.

III. Closed-Loop MPC

Decision variables (future ZMP velocities and next footstep locations) are computed by solving the QP formulation of the MPC described in Sect. II, which requires an initial state of the pendulum, defined by its CoM position and velocity $x_\mathrm{c}, \dot{x}_\mathrm{c}$ and its ZMP position x_z. Traditional MPC gait generation typically envolves this state from an initial value (supposed known) by forward integration of its decision variables through a LIP motion model. This assumes that the

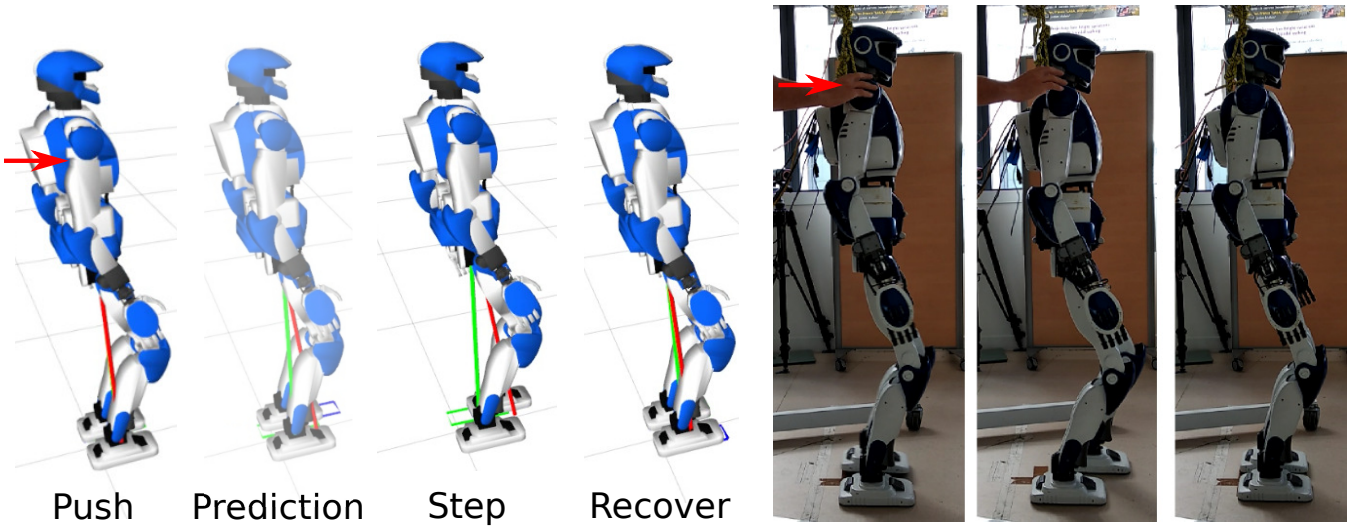

Fig. 3. Reaction to a push during a real experiment with HRP-4. During the push, the ZMP-CoM pendulum estimated from SLAM and force sensor measurements reaches the edge of the foot (red line) while the robot's CoM is displaced forward. As a result, the MPC computes a new optimal footstep (blue square) and computes a footstep trajectory to bring its next swinging foot there. The cartesian-regulator controls the MPC target velocity to walk back to its initial position.

underlying system represented by the reduced LIP model is behaving perfectly as expected. Unfortunately, real physical systems are subject to interactions with their environment, which can easily perturbate the state of the system (imperfect contacts, external forces, poor tracking of the reference pendulum by the underlying system). It is proposed here to estimate the state of the system, and periodically use it to compute an updated MPC solution. The next sections will show how this estimation is achieved, and used to obtain a closed-loop MPC formulation.

A. Estimation of the floating base

The robot's floating base pose ${}^{B}\mathbf{X}_0$ and velocity ${}^{B}\mathbf{V}_0$ is a prerequisite to estimating its CoM and ZMP state. In this work, two simple estimators are considered.

The first provides ground-truth measurement obtained from a high-precision VICON mocap system composed of 8 infrared cameras. Reflective markers are placed on the robot's floating base link, forming a frame V, and tracked by the system with high accuracy at a rate of 100Hz.

$$ {}^{B}\mathbf{X}_0 = {}^{B}\mathbf{X}_V \; {}^{V}\mathbf{X}_0. \tag{7} $$

The second is obtained from the dense visual SLAM method of [7], where RGB-D images from the robot's Asus Xtion sensor are used to estimate the pose of the sensor's optical frame S. Contrary to IMU-based estimators usually used in walking scenarios, visual SLAM provides the full position and orientation in a global world-frame.

$$ {}^{B}\mathbf{X}_0 = {}^{B}\mathbf{X}_H \; {}^{H}\mathbf{X}_S \; {}^{S}\mathbf{X}_0. \tag{8} $$

Note that the transformation between the VICON markers frame and the floating base ${}^{B}\mathbf{X}_V$, and between the camera optical frame and the robot's head link ${}^{S}\mathbf{X}_H$ requires calibration. Both are obtained by Hand-Eye calibration [13]

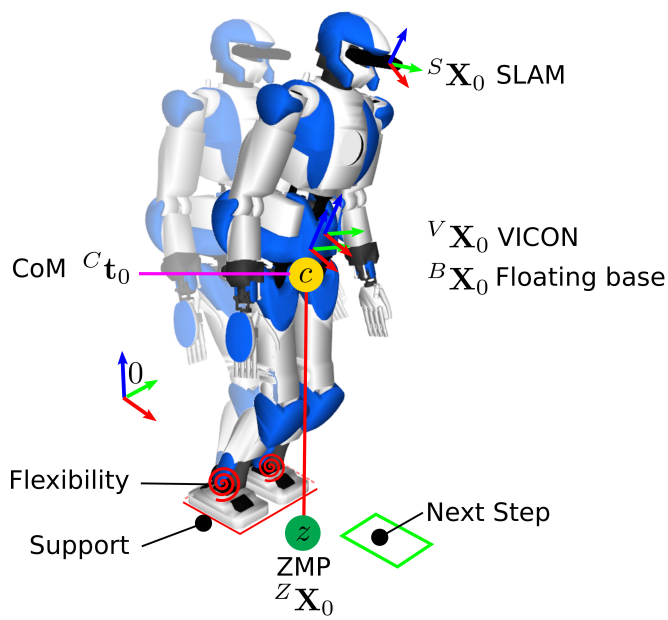

Fig. 4. This HRP-4 is fitted with a passive mechanism between its ankles and feet soles, which provides a spring-damper compliant system. Its state is taken into account by estimating the robot's floating base with dense visual SLAM and validated with a VICON tracking system, with reflective IR markers attached to the floating base link. The ZMP is estimated from ankle force-torque sensors. Note that here it is depicted outside of the robot's support polygon and triggers a step from the MPC.

with *RobCalib* software[1]. The transformation between the head link and the floating base ${}^{B}\mathbf{X}_H$ depends on an accurate kinematic tree calibration, and joint-encoder measurements. HRP-4 is fitted with high-accuracy optical encoders, providing this information at control rate (200Hz).

[1] https://github.com/arntanguy/robcalib

Pose estimates obtained from both VICON and SLAM are noisy, and its velocity is not directly estimated. The use of the Savitzky-Golay filter [14] is proposed, for which an open-source implementation[2] is provided based on Gram-Polynomials [15]. This filter has two main advantages: first, when applied at the end-point of a filtering-window, no additional delays are introduced, and the nth order derivative can be readily obtained. Second, it can be efficiently implemented as a convolution, whose weights depend on the size of the time window, the order of Gram-polynomials, and its derivative order.

Position can be directly smoothened by the filter. Rotation smoothing for a time window consists of solving the following maximum likelihood

$$\widehat{\mathbf{R}} = \underset{\mathbf{R}\in\mathbb{SO}(3)}{\text{argmin}} \sum_k d_{\text{geo}}(\mathbf{R}, \mathbf{R}_k), \tag{9}$$

where $d_{\text{geo}}(\mathbf{R}_1, \mathbf{R}_2)^2 = \frac{1}{2}||\text{logm}(\mathbf{R}_1^T\mathbf{R}_2)||_{\text{frobenius}}$.
Many methods exist for solving this problem. One of them consists in applying a temporal convolution, followed by an orthogonalization, which can be shown to be a 2nd order Taylor approximation of the geodesic distance [16].

$$\begin{array}{lcl} \widetilde{\mathbf{R}} & = & \sum_k w_k \mathbf{R}_k \\ \mathbf{U}\mathbf{D}\mathbf{V}^T & = & svd(\widetilde{\mathbf{R}}) \\ \widehat{\mathbf{R}} & = & \mathbf{U}\mathbf{V}^T \end{array} \tag{10}$$

This implies that rotation matrices can be directly smoothened at any point in a time window by the Gram-Savitzky-Golay convolution coefficients. Consequently, the filtered floating base position and velocity can be readily obtained.

B. CoM State

Knowing the floating base position and velocity, and the kinematic and inertial model of the robot, the CoM position is trivially obtained as ${}^C\mathbf{t}_0$. However, our MPC formulation expects its state to be expressed *w.r.t.* the current support foot. HRP-4 feet are fitted with a passive mechanical flexibility between its ankles and the feet soles that act as a spring-damper to protect the force sensors by reducing impacts, and provide some mechanical compliance while walking. The state of this mechanism is not measured but needs to be taken into account in order to correctly express the CoM in the support foot frame.

In the proposed MPC, the support foot is assumed to be flat on the floor. Let's denote by $\boldsymbol{p}_0^f = (x^f, y^f, 0)$ the position of the support foot in the inertial frame, and θ_{z}^f its rotation around the floor normal axis $\boldsymbol{n}_{\text{z}} = [0, 0, 1]^T$. This frame assumes that the support foot stays in perfect contact with the floor, and that discrepancies in the CoM state are coming from the robot's passive ankle mechanism. In this frame, the CoM is expressed as

$$\tilde{x}_{\text{c}} = \begin{bmatrix} \boldsymbol{R}(\theta_z^f) & \boldsymbol{p}_0^f \\ 0 & 1 \end{bmatrix} {}^C\mathbf{t}_0. \tag{11}$$

Its velocity is also expressed in the flat support foot frame. This implicitly embeds the state of the flexibility in the measurement of the CoM state, as long as the real robot's foot sole remains in perfect contact with the floor.

C. ZMP State

The ZMP can be computed *w.r.t.* to the CoM as

$$x_{\text{z}} = x_{\text{c}} - \frac{\ddot{x}_{\text{c}}}{\eta^2}. \tag{12}$$

In practice $\ddot{x}_c$ can be obtained from accelerometer measurements, or by differentiating the CoM position obtained from the above estimate twice, which proves unreliable.

Since HRP-4 is fitted with force-torque sensors under its ankles providing measurements at control-rate (200Hz), a more reliable expression of the ZMP can be obtained, without the need for differentiation. The main interest is in the ZMP expressed in the ground plane, passing through a point $\boldsymbol{p}_0$ and orthogonal to the unit vector $\boldsymbol{n}_z$. Let $(\boldsymbol{f}_0, \boldsymbol{\tau}_0)$ be the total contact-wrench measured by the left and right foot force-torque sensors, expressed at the point $\boldsymbol{p}_0$. The ZMP is defined as a point Z where the moment of the contact wrench aligns with $\boldsymbol{n}_{\text{z}}$ [17], that is $\boldsymbol{n}_{\text{z}} \times \boldsymbol{\tau}_Z = 0$. Consequently

$$-\boldsymbol{n}_{\text{z}} \times (\boldsymbol{p}_{\text{z}} \times \boldsymbol{f}) + \boldsymbol{n}_{\text{z}} \times \boldsymbol{\tau}_0 = 0 \tag{13}$$

$$-(\boldsymbol{n}_{\text{z}} \cdot \boldsymbol{f})\boldsymbol{p}_Z + (\boldsymbol{n}_{\text{z}} \cdot \boldsymbol{p}_Z)\boldsymbol{f} + \boldsymbol{n}_{\text{z}} \times \boldsymbol{\tau}_0 = 0 \tag{14}$$

With the additional constraint of the ZMP being obtained in the ground plane $\boldsymbol{n}_{\text{z}} \cdot \boldsymbol{p}_Z = 0$, we obtain

$$\boldsymbol{p}_Z = \frac{\boldsymbol{n}_{\text{z}} \times \boldsymbol{\tau}_0}{\boldsymbol{n}_{\text{z}} \cdot \boldsymbol{f}} \tag{15}$$

D. Computing the MPC from its estimated state

Once the estimated state $\tilde{x} = (\tilde{x}_{\text{c}}, \tilde{\dot{x}}_{\text{c}}, \tilde{x}_{\text{z}})$ is computed, the MPC is initialized with it. This means that the prediction of the evolution of the system via the LIP model (1) over the prediction horizon T_h is performed using the estimation as the initial condition. Moreover the cost function (3) and the constraints (4,5,6) are built using the estimated robot state.

However, the robot state estimation $\tilde{x}$ is only available at the frequency of the slowest sensor, here SLAM ($30Hz$) or VICON ($100Hz$), both lower than the control rate ($200Hz$). Hence the state is not available at every control iteration. When no measured state is available, the initial LIP state is computed by integration of (1). The controls $\dot{x}_{\text{z}}^i$, $i = 1, \ldots, N$, are those computed by the optimization performed by the MPC in the previous iteration. The integration leads to

$$\begin{pmatrix} x_{\text{c}}^{k+1} \\ \dot{x}_{\text{c}}^{k+1} \\ x_{\text{z}}^{k+1} \end{pmatrix} = A \begin{pmatrix} x_{\text{c}}^{k} \\ \dot{x}_{\text{c}}^{k} \\ x_{\text{z}}^{k} \end{pmatrix} + B\dot{x}_{\text{z}}^k \tag{16}$$

where the matrices A and B have the form

$$A = \begin{pmatrix} \cosh(\eta\delta) & \frac{\sinh(\eta\delta)}{\eta} & 1 - \cosh(\eta\delta) \\ \eta\sinh(\eta\delta) & \cosh(\eta\delta) & -\eta\sinh(\eta\delta) \\ 0 & 0 & 1 \end{pmatrix}, \tag{17}$$

$$B = \begin{pmatrix} \delta - \frac{\sinh(\eta\delta)}{\eta} \\ 1 - \cosh(\eta\delta) \\ \delta \end{pmatrix}. \tag{18}$$

[2] https://github.com/arntanguy/gram_savitzky_golay

In this way, a reference whole-body control can be provided at every iteration, by propagating the MPC solution, computed either from the estimated robot state, or its integrated internal state.

E. Choice of footstep

When the loop on the MPC is closed, its solution is computed based on the estimated state $(\tilde{x}_{\mathrm{c}}, \tilde{\dot{x}}_{\mathrm{c}}, \tilde{x}_{\mathrm{z}})$. Hence, at any time instant the optimal solution found by the optimization, will in general be different from the previous one, even in the absence of a real perturbation in the state (due to measurement noise and uncertainties). This leads to different optimal footstep target solutions at every MPC iteration. When tracked with a whole-body QP controller, these abrupt changes will be smoothened to some extent depending on the task gains, reducing tracking precision of the swing foot trajectory, and leading to shaky motions. To overcome this issue, two possible solutions are considered.

The first consists in adding to the cost function (3) a term of the form

$$k_f \left(\left(x_{\mathrm{f}}^j - \bar{x}_{\mathrm{f}}^j \right)^2 + \left(y_{\mathrm{f}}^j - \bar{y}_{\mathrm{f}}^j \right)^2 \right), \tag{19}$$

that penalizes the difference between the predicted footstep $(x_{\mathrm{f}}^j, y_{\mathrm{f}}^j)$ and the previously chosen footstep $(\bar{x}_{\mathrm{f}}^j, \bar{y}_{\mathrm{f}}^j)$. However this also affects the other high-level tasks assigned to the robot, e.g. the reference velocity tracking, because the robot minimizes the difference between two footsteps therefore making steps as short as possible.

The second option relies on the fact that the solution found by the MPC is the optimal one, but it's not the unique solution that satisfies the constraints. Therefore, if the constraints are still satisfied by keeping the previously found footsteps with the new ZMP trajectory, it is chosen to keep it, even if it is not the optimal in terms of cost function. In this way we somewhat filter the continuous change of predicted footsteps, and only the latest optimal solution of the MPC is chosen if the previous one becomes invalid because it violates the constraints, which guarantees robot balance.

F. Quadratic Programming Controller

The desired MPC solution is then tracked by a whole-body quadratic controller (QP), which generates whole-body motion for the real model. A detailed explanation of the QP controller used in this work can be found in [18, 19]. Its decision vector is $\mathbf{z} = (\ddot{\mathbf{q}}, \boldsymbol{\lambda}_{\mathcal{C}})$, where $\ddot{\mathbf{q}}$ gathers the linear and angular acceleration of floating-base coordinates, and the generalized joint velocities. $\boldsymbol{\lambda}_{\mathcal{C}}$ denotes the vector of conic coordinates of linearized Coulomb friction cones, such that the contact forces $\mathbf{f}$ are equal to $\mathbf{S}_f \boldsymbol{\lambda}_{\mathcal{C}}$ with $\mathbf{S}_f$ the span matrix of cone generators. The cost function includes various tasks $\boldsymbol{\mathcal{T}}$ that drive the whole-body state towards a desired configuration. We consider here

- $\boldsymbol{\mathcal{T}}_{CoM}$ tracks the desired CoM position x_{c} and velocity $\dot{x}_{\mathrm{c}}$.
- $\boldsymbol{\mathcal{T}}_{Trajectory}$ tracks the desired swing foot target x_{f}.
- $\boldsymbol{\mathcal{T}}_{CoP}$ tracks the desired ZMP through admittance control, achieved by the embeded stabilizer.
- $\boldsymbol{\mathcal{T}}_{Posture}$ is a reference posture that keeps the robot upright.

The overall QP can be summed up as follows:

$$\begin{aligned} \mathbf{z} = \underset{\mathbf{z}}{\operatorname{argmin}} \; & \sum_{i=1}^{N} w_i \|\boldsymbol{\mathcal{T}}_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\|^2 + w_\lambda \|\boldsymbol{\lambda}_{\mathcal{C}}\|^2 \\ & \text{subject to:} \\ & 1) \text{ dynamic constraints} \\ & 2) \text{ sustained contact positions} \\ & 3) \text{ joint limits} \\ & 4) \text{ non-desired collision avoidance constraints} \\ & 5) \text{ self-collision avoidance constraints} \end{aligned} \tag{20}$$

Here, w_i and w_λ are task weights, and $\boldsymbol{\mathcal{T}}_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ denotes the residual of the ith task. The reader is referred to [20] for details on the formulation of all these constraints. The obtained reference acceleration $\ddot{\mathbf{q}}$ is integrated twice to obtain a desired joint configuration $\mathbf{q}$ which is then executed by the actuators *proportional-derivative (PD)* controllers.

IV. Experiments

The proposed work is based on the MPC formulation presented in [12]. Including footstep generation into the formulation enables an intuitive and flexible way to control the locomotion of a humanoid robot. Walking is achieved by specifying a reference velocity. Footsteps, CoM and ZMP trajectories are chosen accordingly. This work has previously been demonstrated for open-loop control of a NAO humanoid robot, including pursuit avoidance scenarios in the presence of obstacles [21]. However, it has never been demonstrated applied to large-scale humanoids such as HRP-4. In this section, we show that the MPC can be successfully applied to such a robot while validating its closed-loop implementation.

A first experiment validates the walking performance with a simple cartesian regulator based on SLAM localization that makes the robot walk to specified targets in its environment.

The second experiment validates the closed-loop aspect of the MPC by reacting to perturbations that would otherwise make the robot fall, even in the presence of a stabilizer.

For all experiments, ground-truth floating base position is obtained from a VICON motion capture system, and SLAM estimations are obtained from Asus Xtion's RGB-D frames with *D6DSLAM* [7] software. A video is provided for each experiment respectively.

A. Stabilization

The desired ZMP computed by the MPC is sent to the embedded stabilizer of HRP-4 [22]. Its role is to compensate deviations from a reference ZMP trajectory by accelerating the torso in the opposite direction. This is achieved by modifying the desired ankle joint reference. In case of a perturbation too large to be recovered without taking a step, the robot would normally fall. Instead, with our proposed closed-loop MPC the robot takes a step forward, and the stabilizer tracks the new reference ZMP (see Figure 5, and the video).

B. Drift-free cartesian space control

An additional benefit to the proposed closed-loop control is its ability to robustly reach target position specified

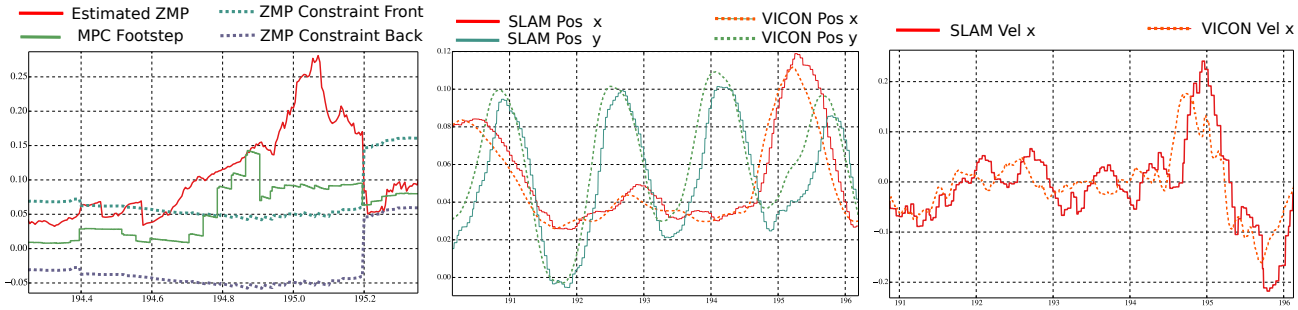

Fig. 5. A strong push forward causes the MPC to step forward to prevent loss of balance. Left - the estimated ZMP from force-sensor measurements is leaving the stability constraint area at time 194.7, the MPC plans a step forward to compensate. Middle - Estimation of the floating base position from SLAM and VICON measurements along the x and y direction. Right - Estimation of the floating base velocity.

in the environment. To illustrate this, we implemented a simple cartesian regulator that computes reference velocities (v_x, v_y, ω) for the MPC according to its position and orientation relative to a desired world position (x_d, y_d, θ_d). The floating base pose $(\tilde{x}_b, \tilde{y}_b, \tilde{\theta}_b)$ is chosen as the robot's reference surface, and is estimated with SLAM. Ground-truth measurements are obtained as (x_v, y_v, θ_v) with the VICON tracking system.

$$\begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} = \begin{pmatrix} \lambda_x(x_d - \tilde{x}_b) \\ \lambda_y(y_d - \tilde{y}_b) \\ \lambda_\theta(\theta_d - \tilde{\theta}_b) \end{pmatrix} \tag{21}$$

Figure 6 shows the result of HRP-4 repeatedly walking from an initial position $t_0 = (0, 0, 0)$ towards a target $t1 = (1, 0.5, 0)$, then coming back to its initial position before walking towards a second target $t_2 = (1, -0.5, 0)$. Without feedback on the regulator, considerable drift is observed, and the targets are not reached accurately. Using the estimated robot state, the regulator can however lead the MPC to walk accurately towards each target.

C. Push reaction

In the second experiment, the robot is stepping in place with a low-weight cartesian-regulator ($\lambda_x = \lambda_y = \lambda_\theta = 0.2$) when it is pushed from behind. The MPC is computed from the estimated robot state $(\tilde{x}_c, \tilde{\dot{x}}_c, \tilde{x}_z)$, obtained by the combination of SLAM and force-torque sensors. When perturbed, the MPC decides to change it's desired footstep, and computes a new corresponding ZMP velocity trajectory. This behavior emerges as a result of the optimization: in order to find an optimal solution that satisfies the constraints (bringing the future ZMP trajectory back inside the support foot polygon while respecting kinematic feasibility), the MPC has to move the ZMP forward thus performing a step.

Figure 5 shows this behavior on a real perturbation. As can be seen from the first plot, around time 194.7s, the perturbation becomes large enough for the ZMP to exit the balance constraint area (defined as a sub-rectangle of the real foot) with sufficient velocity, and the MPC has no choice but to alter its desired footstep. The shaky behavior of the footstep solution can be seen on the MPC footstep plot, as the experiment was performed without penalizing any change of footstep. Good estimation of the floating base position is achieved with SLAM, although additional delays are introduced, and the velocity estimates remain quite noisy.

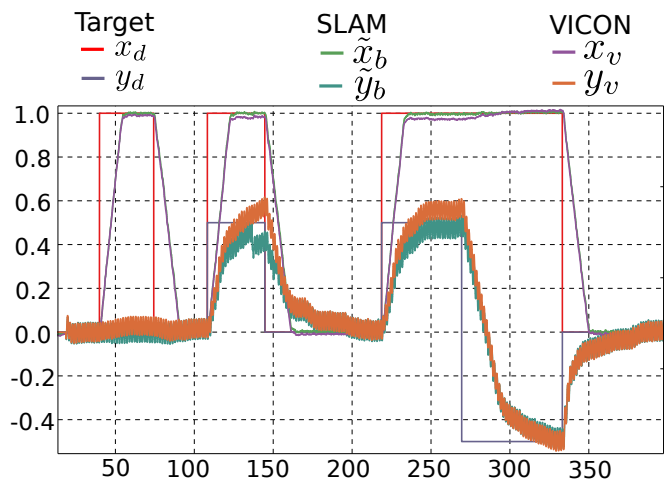

Fig. 6. HRP-4 walks to specific places in the environment. A cartesian-regulator continuously computes MPC reference velocities based on the robot's position *w.r.t.* its target as estimated from SLAM measurements. It's trajectory is validated by both VICON ground-truth data, and visual markers on the floor. In the video, a comparative test of openloop and closed-loop walking is performed.

V. Conclusion

In this work a closed-loop formulation of the intrinsically stable MPC for humanoid gait generation [12] has been presented, where the robot CoM position and velocity are estimated with visual SLAM, while the ZMP position is computed from force/torque sensor measurements. The closed-loop implementation of the MPC allows the robot to react to external disturbances through stepping, enhancing the overall robustness of the control scheme.

Moreover, the visual localization of the robot in the environment, allows it to navigate with precision to specific locations, recovering from the drift generated by uncertainties and unmodeled interactions with the environment (e.g. foot slipping). As extension of this work, the dense SLAM map can be better exploited, in order to take into account obstacles on the robot path to avoid them.

References

[1] S. Kajita, H. Hirukawa, K. Harada, and K. Yokoi, *Introduction to Humanoid Robotics*. Springer, 2014.

[2] A. Herdt, N. Perrin, and P. B. Wieber, "Walking without thinking about it," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 190–195.

[3] M. Naveau, M. Kudruss, O. Stasse, C. Kirches, K. Mombaur, and P. Souères, "A reactive walking pattern generator based on nonlinear model predictive control," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 10–17, Jan 2017.

[4] N. A. Villa and P. Wieber, "Model predictive control of biped walking with bounded uncertainties," in *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, Nov 2017, pp. 836–841.

[5] S. Feng, X. Xinjilefu, C. G. Atkeson, and J. Kim, "Robust dynamic walking using online foot step optimization," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5373–5378.

[6] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended KinectFusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, Jul 2012.

[7] M. Meilland and A. I. Comport, "On unifying key-frame and voxel-based dense visual SLAM at large scales," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 3-7 November 2013, pp. 3677–3683.

[8] M. F. Fallon, P. Marion, R. Deits, T. Whelan, M. Antone, J. Mcdonald, and R. Tedrake, "Continuous Humanoid Locomotion over Uneven Terrain using Stereo Fusion," *IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, pp. 881–888, 2015.

[9] M. F. Fallon, M. Antone, N. Roy, and S. Teller, "Drift-free humanoid state estimation fusing kinematic, inertial and lidar sensing," in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*. IEEE, 2014, pp. 112–119.

[10] R. Scona, S. Nobili, Y. R. Petillot, and M. Fallon, "Direct visual SLAM fusing proprioception for a humanoid robot," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1419–1426, 2017.

[11] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "ElasticFusion: Dense SLAM without a pose graph," in *International Journal of Robotics Research (IJRR)*, 2015.

[12] N. Scianca, M. Cognetti, D. De Simone, L. Lanari, and G. Oriolo, "Intrinsically stable MPC for humanoid gait generation," in *16th IEEE-RAS Int. Conf. on Humanoid Robots*, 2016, pp. 101–108.

[13] A. Tanguy, A. Kheddar, and A. I. Comport, "Online Eye-Robot Self-Calibration," *Proceedings - IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2018.

[14] A. Savitzky and M. J. Golay, "Smoothing and differentiation of data by simplified least squares procedures." *Analytical chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.

[15] P. A. Gorry, "General Least-Squares Smoothing and Differentiation by the Convolution (Savitzky-Golay) Method," *Analytical Chemistry*, vol. 62, no. 6, pp. 570–573, 1990.

[16] C. Gramkow, "On averaging rotations," *Journal of Mathematical Imaging and Vision*, vol. 15, no. 1-2, pp. 7–16, 2001.

[17] S. Caron, Q.-C. Pham, and Y. Nakamura, "Zmp support areas for multicontact mobility under frictional constraints," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 67–80, 2017.

[18] K. Bouyarmane and A. Kheddar, "Humanoid robot locomotion and manipulation step planning," *Advanced Robotics*, no. 26, pp. 1099–1126, July/September 2012.

[19] A. Escande, A. Kheddar, and S. Miossec, "Planning contact points for humanoid robots," *Robotics and Autonomous Systems*, vol. 61, no. 5, pp. 428–442, 2013.

[20] J. Vaillant, A. Kheddar, H. Audren, F. Keith, S. Brossette, A. Escande, K. Bouyarmane, K. Kaneko, M. Morisawa, P. Gergondet, E. Yoshida, S. Kajita, and F. Kanehiro, "Multi-contact vertical ladder climbing with an HRP-2 humanoid," *Autonomous Robots*, vol. 40, no. 3, pp. 561–580, 2016.

[21] D. De Simone, N. Scianca, P. Ferrari, L. Lanari, and G. Oriolo, "MPC-based humanoid pursuit-evasion in the presence of obstacles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 5245–5250.

[22] K. Yokoi, F. Kanehiro, K. Kaneko, S. Kajita, K. Fujiwara, and H. Hirukawa, "Experimental study of humanoid robot hrp-1s," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 351–362, 2004.