Adaptive Compliance in Post-Impact Humanoid Falls Using Preview Control of a Reduce Model

Vincent Samy, Stéphane Caron, Karim Bouyarmane, Abderrahmane Kheddar

Adaptive Compliance in Post-Impact Humanoid Falls Using Preview Control of a Reduce Model

Vincent Samy[1], Stéphane Caron[1], Karim Bouyarmane[2] and Abderrahmane Kheddar[1,3]

***Abstract*— We present a novel approach to control a humanoid robot in active compliance just after an impact consecutive to a fall. Using linear model predictive control (LMPC), the momentum accumulated by the robot during the falling phase is absorbed, by driving it to zero, until the robot comes to safe rest. The LMPC is written for a reduced center-of-mass model of the robot subject to external contact forces applied on the impact bodies of the robot, each body belonging to an impact limb (arm, leg). Distributing optimally the initial momentum at impact and the total gravity force among all the impacting limbs, we write one LMPC per such limb, each contributing to its own share of the momentum absorption problem. The control vector of each MPC is the contact force applied at the impact body of the limb. We propose a method that allows to encode in a contact polytope both the friction limitations and the actuation torque limits of the impact limb's actuated joints, this polytope models in a synthetic way the linear constraint on the control vector of the LMPC. The approach is validated in full-body dynamics simulation of a humanoid robot falling on a wall.**

I. Introduction

Designing humanoids robots with robust falling and recovery strategies is as important as agile walking or whole-body attitude control. The DARPA Robotics Challenge highlighted how falls that are not properly handled result in substantial hardware damage. Several fall avoidance strategies were thus experimented, including reliable but conservative approaches [1], foothold replacement on the fly [2], planning of additional contacts on the fly [3], etc. There is certainly room for improvement, and such strategies shall be further investigated. Nevertheless, even the most sophisticated legged creatures in nature do end-up in situations were fall is inevitable, and so will humanoid robots. Our research is focused on dealing with falls when they occur and that cannot be recovered from by trying to reduce damage as much as possible.

A fall incurs relatively high impacts on the mechanical structure of the robot, including linkage and actuators. Unfortunately, there is not much to do at the software level to deal with the propagation of mechanical impacts. Our conviction is that the hardware design of humanoid robots must embed impact-absorbing components at any rate, *e.g.* flexible covers [4], compliant contacting interfaces [5] or compliant joint actuators coupled with gains tuning algorithms [6]. They can be complemented by active or semi-active methods to reduce impact intensity (pre-impact prevention) or further absorb its effect (post-impact compliance).

In our previous work [7], we leveraged the brief amount of time available in the pre-impact landing phase to adapt the humanoid's posture based on a predefined taxonomy of postures suited to each fall direction. Then, to deal with the post-impact phase, we suggested reducing PD gains in the low-level actuator controllers to realize a compliant spring-damper-like behavior at the joint level. These post-impact PD gains, first determined *ad hoc* from experiments, were later derived automatically using a gain-adaptive whole-body quadratic-programming controller [8]. We call this control scheme "adaptive QP". An adaptive-QP controller enables the robot to comply during post-impact with PD gains computed in accordance with its state variables. Combining posture selection and gain adaption, we carried out real-world experiments where the HRP-4 humanoid robot sustained front and back falls on a soft mattress, starting from an upright standing posture.

Post-impact recovery is about absorbing undesired momentum, accumulated during the fall, using contact forces. This momentum cannot be absorbed instantly due to friction and actuation limits. However, it must be absorbed *before* position limits (joint limits or collision with the environment) are reached. The adaptive-QP scheme, being an *instantaneous* and reactive control scheme, lacks the level of hindsight necessary to reach an efficient balance between these two kinds of limitations.

The contribution of the present work is to complement adaptive-QP whole-body control with an adaptive-compliance *predictive controller* that satisfies both post-impact actuation and position limits. Due to the size of receding-horizon problems, predictive control requires the use of reduced dynamic models, which usually cannot represent whole-body actuation limits. To overcome this, we introduce a novel mapping of joint-torque limits into actuation polytopes that generalize friction cones. We then distribute momentum between contacts and solve the predictive control problem using parallel computations, resulting in a reduced-model trajectory that is finally tracked by adaptive-QP control. The validity of this approach is demonstrated in simulations with the HRP-4 humanoid robot.

II. Background

A. Gain-adaptive QP-based whole-body control

In our recent work [8], we extended our whole-body QP task-space controller [9] to include PD gains of the actuators as part of the QP decision variables. We also demonstrated that such a scheme applies to position or torque controlled robots. The reason behind this approach was to automate PD-gains tuning so as to make the robot compliant during the post-impact phase, and hence reduce damage by absorbing as much as possible the momentum accumulated while falling. The PD coefficients have been inserted into the decision

[1] CNRS-UM2 LIRMM, IDH group, UMR5506, Montpellier, France.
[2] Université de Lorraine-INRIA-CNRS LORIA, 54600 Villers-lès-Nancy, France.
[3] CNRS-AIST Joint Robotics Laboratory (JRL), UMI3218/RL, Japan.
* This work is supported in part by H2020 EU project COMANOID http://www.comanoid.eu/, RIA No 645097.
Corresponding author: vincent.samy@lirmm.fr

vector of a QP whole-body control problem as follows:

$$\begin{aligned}\min_{\ddot{\boldsymbol{q}},\boldsymbol{\lambda},\boldsymbol{K},\boldsymbol{B}} & \sum_k \omega_k^{\text{sp}} E_k^{\text{sp}} + \omega_\lambda \|\boldsymbol{\lambda}\|^2 + \omega_G(\|\boldsymbol{K}\|^2 + \|\boldsymbol{B}\|^2)\,, \\ \text{s.t.} \quad & H_S\ddot{\boldsymbol{q}} + \boldsymbol{c}_S - (J^TG)_S\boldsymbol{\lambda} = K\boldsymbol{e}_S - B\dot{\boldsymbol{e}}_S \\ & \underline{\boldsymbol{\tau}}_S \le K\boldsymbol{e}_S - B\dot{\boldsymbol{e}}_S \le \overline{\boldsymbol{\tau}}_S \\ & \underline{\boldsymbol{\tau}}_{\text{NS}} \le H_{\text{NS}}\ddot{\boldsymbol{q}} + \boldsymbol{C}_{\text{NS}} - (J^TG)_{\text{NS}}\boldsymbol{\lambda} \le \overline{\boldsymbol{\tau}}_{\text{NS}} \\ & \underline{\ddot{\boldsymbol{q}}} \le \ddot{\boldsymbol{q}} \le \overline{\ddot{\boldsymbol{q}}},\ \boldsymbol{\lambda} \ge \boldsymbol{0},\ \boldsymbol{K} \ge \boldsymbol{0},\ \boldsymbol{B} \ge \boldsymbol{0}\end{aligned} \tag{1}$$

with $\ddot{\boldsymbol{q}}$ the vector of joint accelerations, $\boldsymbol{\lambda}$ the generator weights of linearized friction cones, $\boldsymbol{K}$ and $\boldsymbol{B}$ the vectors of stiffness and damping coefficients, ω_k the positive weight of task k, E_k^{sp} the residual (error) vector of task k, H the inertia matrix, $\boldsymbol{c}$ the vector of Coriolis and gravity forces, J^TG the mapping from linearized friction cones to the joint space and $\boldsymbol{e} = \boldsymbol{q}^{\text{ref}} - \boldsymbol{q}$ the vector of joint deviations from a reference posture $\boldsymbol{q}^{\text{ref}}$. The subscript S denotes joints that are set to be *compliant*, for which non-zero entries will appear in the vectors $\boldsymbol{K}$ and $\boldsymbol{B}$. The subscript NS denotes joint that are not set to be *compliant*. Generally, $\boldsymbol{q}^{\text{ref}}$ is set as the state at the impact time, while $\dot{\boldsymbol{q}}^{\text{ref}}$ is zero. Finally, variables such as $\overline{\boldsymbol{\tau}}$ (*resp.* $\underline{\boldsymbol{\tau}}$) denote the upper bound (*resp.* lower bound) of the corresponding variables. The lower and upper bound for the free-floating base part of the dynamics equation is zero.

B. Polyhedra and polytopes

Polyhedra are convex sets realized by the intersection of finitely-many halfspaces. They correspond to the inequality constraints found in linear and quadratic programs, as well as the dual constraints such as *friction cones* or *ZMP support areas* that arise in mechanics [10]. A polytope is a bounded polyhedron. It can be represented equivalently as: (*i*) the intersection of finitely-many halfspaces, called the $\mathcal{H}$-representation, or (*ii*) the convex hull of a finite set of vertices, called the $\mathcal{V}$-representation. The latter can be written mathematically as:

$$\mathcal{P} = \left\{ \boldsymbol{x} = \textstyle\sum_{i=1}^N \alpha_i \boldsymbol{v}_i \;\middle|\; \forall i, \alpha_i \ge 0, \sum_{i=1}^N \alpha_i = 1 \right\} \tag{2}$$

where the positive coefficients α_i form a *convex combination* of the N vertices $\boldsymbol{v}_i$ of the polytope. Meanwhile, the $\mathcal{H}$-representation can be concisely written as:

$$A\boldsymbol{x} \le \boldsymbol{b} \tag{3}$$

where the matrix $A = [\boldsymbol{A}_0 \dots \boldsymbol{A}_N]^T$ stacks halfspace normals $\boldsymbol{A}_i$, so that $\boldsymbol{A}_i^T\boldsymbol{x} = b_i$ is the equation of the i^{th} supporting halfspace of the polytope. The conversion from $\mathcal{H}$ to $\mathcal{V}$ (resp. from $\mathcal{V}$ to $\mathcal{H}$) representation of a polytope is known as the *vertex enumeration* problem. Both can be realized by the double-description algorithm [11].

III. Reduced Dynamic Model

We approximate the whole-body dynamics at the center of mass (CoM) of the robot and search for contact forces that compensate for both gravity and post-impact linear momentum. The Newton equation of motion that governs the acceleration of the CoM is:

$$\boldsymbol{F} = M(\ddot{\boldsymbol{s}} - \boldsymbol{g}), \tag{4}$$

with M the total robot mass, $\ddot{\boldsymbol{s}}$ the acceleration of the CoM, $\boldsymbol{g}$ the gravity vector and $\boldsymbol{F}$ the net contact force (sum of all external forces) applied to the robot.

Under the assumption that the initial linear momentum $M\dot{\boldsymbol{s}}^0$ at impact is known, the goal of post-impact absorption is to find a trajectory $\ddot{\boldsymbol{s}}(t)$ such that, at time $t = t_f$, $\dot{\boldsymbol{s}}(t_f) = 0$, meaning that all the linear momentum accumulated while falling has been driven out of the system. Achieving this task requires proper control of the net contact force $\boldsymbol{F}$.

A. Actuation constraints

Let us consider again the full-body dynamics equation of motion of the system

$$H\ddot{\boldsymbol{q}} + \boldsymbol{c} = J^T\boldsymbol{F} + S^T\boldsymbol{\tau}. \tag{5}$$

$\boldsymbol{F} = (J^TG)\boldsymbol{\lambda}$ is the stacked vector of contact forces and S is a selection matrix of actuated joints among all the degrees of freedom (DoFs) of the robot (hence the matrix that excludes the floating-base DoFs from the total DoFs [12].

A fundamental property of the humanoid kinematic tree topology is that it comprises at least 4 limbs (two legs and two arms), indexed here with the variable $c \in \{\text{lh}, \text{rh}, \text{lf}, \text{rf}\}$, that extend from the root of the kinematic tree to the end-effectors (respectively the two hands and two feet). The contact force applied at the end-effector c is denoted $\boldsymbol{F}_c$ (therefore we have $\boldsymbol{F} = (\boldsymbol{F}_c)_{c\in\{\text{lh},\text{rh},\text{lf},\text{rf}\}}$) The rest of the DoFs, including the floating base are indexed with 0. Eq. (5) can be rewritten as

$$\begin{bmatrix} H_0 \\ H_{\text{lh}} \\ H_{\text{rh}} \\ H_{\text{lf}} \\ H_{\text{rf}} \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} \boldsymbol{c}_0 \\ \boldsymbol{c}_{\text{lh}} \\ \boldsymbol{c}_{\text{rh}} \\ \boldsymbol{c}_{\text{lf}} \\ \boldsymbol{c}_{\text{rf}} \end{bmatrix} = \begin{bmatrix} J_0^T\boldsymbol{F} \\ J_{\text{lh}}^T\boldsymbol{F}_{\text{lh}} \\ J_{\text{rh}}^T\boldsymbol{F}_{\text{rh}} \\ J_{\text{lf}}^T\boldsymbol{F}_{\text{lf}} \\ J_{\text{rf}}^T\boldsymbol{F}_{\text{rf}} \end{bmatrix} + \begin{bmatrix} S_0^T\boldsymbol{\tau}_0 \\ \boldsymbol{\tau}_{\text{lh}} \\ \boldsymbol{\tau}_{\text{rh}} \\ \boldsymbol{\tau}_{\text{lf}} \\ \boldsymbol{\tau}_{\text{rf}} \end{bmatrix}, \tag{6}$$

where J_c is the *reduced* Jacobian of the contact points at the end-effector c to the root of the kinematic tree, with respect only to the set of joints of the considered limb (and not with respect to all the DoFs of the robot). The contact force at limb $c \in \{\text{lh}, \text{rh}, \text{lf}, \text{rf}\}$, $\boldsymbol{F}_c$, only affects the part of the dynamics equation in $\{c, 0\}$:

$$H_c\ddot{\boldsymbol{q}} + \boldsymbol{c}_c = J_c^T\boldsymbol{F}_c + \boldsymbol{\tau}_c. \tag{7}$$

Using the reduced Jacobian, the kinematic contact constraint at the end-effector c is expressed in acceleration-form as:

$$J_c\ddot{\boldsymbol{q}} + \dot{J}_c\dot{\boldsymbol{q}} = \boldsymbol{0}. \tag{8}$$

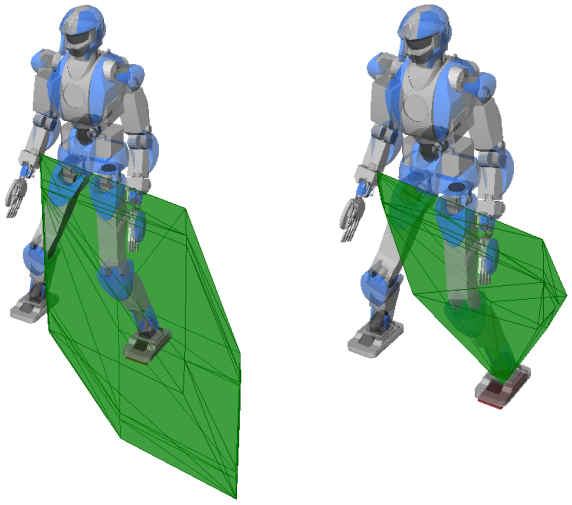
Substituting Eq. (7) in Eq. (8) yields

$$\begin{aligned} J_cH_c^{-1}J_c^T\boldsymbol{F}_c &= J_cH_c^{-1}(\boldsymbol{c}_c - \boldsymbol{\tau}_c) - \dot{J}_c\dot{\boldsymbol{q}} \\ \Lambda_c^{-1}\boldsymbol{F}_c &= -J_cH_c^{-1}\boldsymbol{\tau}_c + J_cH_c^{-1}\boldsymbol{c} - \dot{J}_c\dot{\boldsymbol{q}} \\ \boldsymbol{F}_c &= -\Lambda_cA_c\boldsymbol{\tau}_c + \boldsymbol{d} \\ \boldsymbol{F}_c &= C_c\boldsymbol{\tau}_c + \boldsymbol{d}_c. \end{aligned} \tag{9}$$

The matrix C_c and vector $\boldsymbol{d}_c$ provide an affine mapping from torque limits to contact force limits. The $\mathcal{H}$-representation of the torque-constraint polytope is:

$$\underline{\boldsymbol{\tau}}_c \;\le\; \boldsymbol{\tau}_c \;\le\; \overline{\boldsymbol{\tau}}_c \tag{10}$$

where $\boldsymbol{\tau}_c$ is the vector of selected joint torques for the contact c (corresponding to the limb at hand, *e.g.* left-leg joints for a left-foot contact), $\underline{\boldsymbol{\tau}}_c$ and $\overline{\boldsymbol{\tau}}_c$ are respectively the lower and upper torque limits of the actuators. Each limb of our humanoid model having 6 joints, it is straightforward to compute the $2^6 = 64$ vertices of the $\mathcal{V}$-description for



(a) (b)

Fig. 1: Polytopes of feasible contact forces at the left-foot of HRP-4 under (a) leg torque limits and (b) both torque limit and friction cone constraints.

this torque-constraint polytope. Applyting Eq. (9) yields the $\mathcal{V}$-representation of the torque-limited polytope of feasible end-effector forces at the contact point, depicted in Fig.1a. Using the double-description method, we convert it to $\mathcal{H}$-representation:

$$G_c^{\boldsymbol{\tau}} \boldsymbol{F}_c \leq \boldsymbol{h}_c^{\boldsymbol{\tau}}. \tag{11}$$

The next actuation limit at contact comes from friction. Using a linearized friction pyramid, whose $\mathcal{H}$-representation is given by (see *e.g.* [10]):

$$\begin{bmatrix} -1 & 0 & -\mu \\ 1 & 0 & -\mu \\ 0 & -1 & -\mu \\ 0 & 1 & -\mu \end{bmatrix} {}^0E_c \boldsymbol{F}_c \leq \mathbf{0}, \tag{12}$$

where μ is the friction coefficient and ${}^0E_c$ is the rotation matrix from the contact frame to the world frame.

Intersecting polytopes in $\mathcal{H}$-representation is done straightforwardly by concatenation. We thus obtain the $\mathcal{H}$-representation

$$G_c \boldsymbol{F}_c \leq \boldsymbol{h}_c, \tag{13}$$

of the torque- and friction-constrained actuation polytope depicted in Fig. 1b.

IV. Distribution of Gravity and Linear Momentum

Our primary idea is to make each limb $c \in \{\text{lh}, \text{rh}, \text{lf}, \text{rf}\}$ of the robot contribute, in post-impact, to a "share" of the pre-impact linear momentum absorption, summing up all contributions into the resultant equation of motion:

$$\sum_{c \in \{\text{lh,rh,lf,rf}\}} \boldsymbol{F}_c = M(\ddot{\boldsymbol{s}} - \boldsymbol{g}). \tag{14}$$

This can be conceptually interpreted as "splitting" the CoM into four virtual mass points with state vectors $(\boldsymbol{s}_c, \dot{\boldsymbol{s}}_c)$, each going towards one of the limb extremities in contact, and each with a force $\boldsymbol{F}_c$ applied to it. Hence we are solving for the system of four differential equations for the state variables $(\boldsymbol{s}_c, \dot{\boldsymbol{s}}_c)$:

$$\boldsymbol{F}_c = M(\ddot{\boldsymbol{s}}_c - \boldsymbol{g}_c), \tag{15}$$

where $\boldsymbol{g}_c$ is a quantity to be defined below (gravity force distribution problem) and where the initial condition for each component is also to be defined below (initial impact momentum distribution problem). The force $\boldsymbol{F}_c$ is used to decelerate its respective CoM component until full initial momentum absorption, while being contrained to lie inside its respective actuation limit polytope (Section III-A).

The momentum being an additive quantity, when virtually splitting the CoM into four virtual components, the initial momentum has to be distributed among the four components. Similarly, the total gravity force applied on the CoM has also to be split into four components, each applied to one of the four virtual CoM components. Therefore, in the following, we distribute the amount of gravity and initial linear momentum that each contact will support. This distribution must satisfy (Eq. (14)). Additionally, we want to make sure that the distribution is optimal with respect to the actuation constraint polytope at each contact.

Assuming that the linear momentum $\boldsymbol{p}$ after impact is known, we look for a corresponding force quantity $\boldsymbol{f}^a$ that would be physically consistent with the problem of distribution in the respective contact force polytopes. The force being a time-derivative of the linear momentum, we can write:

$$\boldsymbol{f} = \frac{d\boldsymbol{p}}{dt} = M\frac{d\dot{\boldsymbol{s}}}{dt}. \tag{16}$$

In the worst case where the robot is completely rigid (non compliant) and the coefficient of restitution of the impact is zero, the impact force corresponding to the momentum $M\dot{\boldsymbol{s}}^0$ is:

$$\boldsymbol{f} = \lim_{h \to 0} M\frac{\dot{\boldsymbol{s}}^0}{h}. \tag{17}$$

Because this form is non-linear in h, we set $k = 1/h$ (hence k can be seen as a "gain" that maps a linear momentum quantity to a force quantity). The limit becomes

$$\boldsymbol{f} = \lim_{k \to \infty} kM\dot{\boldsymbol{s}}^0. \tag{18}$$

Then we define $\boldsymbol{f}^a$ as the exact applied force

$$\boldsymbol{f}^a \triangleq kM\dot{\boldsymbol{s}}^0 \tag{19}$$

where k is a parameter that approach infinity. $\boldsymbol{f}^a$ corresponds to the *average impact force* for impact time $1/k$.

Being already a force quantity, the gravity force:

$$\boldsymbol{f}^g = M\boldsymbol{g}, \tag{20}$$

can be readily distributed in the polytopes.

The idea now is to find which contact will better handle $\boldsymbol{f}^g$ and which will better handle $\boldsymbol{f}^a$. For example, in the case of Figure 2, the feet of the robot are more suitable for compensating the gravity than the hands. From Eq. (2) we split α_i in two variables α_i^g and α_i^a corresponding respectively to the percentage of $\boldsymbol{f}^g$ and $\boldsymbol{f}^a$ to address to the contact.

$$\sum_i \alpha_i = \sum_i \alpha_i^g + \alpha_i^a. \tag{21}$$

This is done for each contact c. Using Eq. (2), (19), (20) and (21) we can write the following constraints

$$\sum_c \sum_i \alpha^g_{c,i} \boldsymbol{v}_{c,i} = -M\boldsymbol{g} \tag{22}$$
$$\sum_c \sum_i \alpha^a_{c,i} \boldsymbol{v}_{c,i} = -kM\dot{\boldsymbol{s}}^0 \tag{23}$$
$$\sum_i \alpha^g_{c,i} + \alpha^a_{c,i} \leq 1 \tag{24}$$
$$\alpha^g_{c,i} \geq 0 \tag{25}$$
$$\alpha^a_{c,i} \geq 0 \tag{26}$$
$$k \geq 0. \tag{27}$$

Constraints Eq. (22) and Eq. (23) distribute forces considering the polytope of each contact. Constraint Eq. (24) is a generalization of Eq. (2). Substituting the equality to an inequality does not change the definition of the polytope and relax the constraint. And finally, Eq. (25)-(27) make the variables consistent with the polytope definition (2) and the force generated at the CoM definition (19).

To have a good distribution over the feet and the hands (*i.e.* to avoid having the robot's weight supported by one foot), we design the following cost functions

$$\begin{cases} \| \sum_i \alpha^g_{\text{rf},i} \boldsymbol{v}_{\text{rf},i} - \sum_i \alpha^g_{\text{lf},i} \boldsymbol{v}_{\text{lf},i} \|^2 \\ \| \sum_i \alpha^a_{\text{rh},i} \boldsymbol{v}_{\text{rh},i} - \sum_i \alpha^a_{\text{lh},i} \boldsymbol{v}_{\text{lh},i} \|^2 \\ -k \end{cases} \tag{28}$$

where rf (resp. lf) stands for right foot (resp. left foot) and rh (resp. lh) for right hand (resp. left hand).

As we want the variable k to approach infinity, a cost function is set to maximize its value in the problem.

The overall system, with linear constraints Eq. (22)-(27) and quadratic cost functions Eq. (28) is a Quadratic Programming (QP) problem. Later on, we call it the Force Distribution QP (FDQP). Figure 2 shows a possible repartition of the forces over two contact points. The FDQP looks for a solution that compensates the force from gravity and maximizes the force generated against linear momentum.

Once the solution is found, α^g_c, α^a_c and k can be computed using the FDQP output:

$$-M\boldsymbol{g}_c = \sum_i \alpha^g_{c,i} \boldsymbol{v}_{c,i}\,, \tag{29}$$
$$-kM\dot{\boldsymbol{s}}^0_c = \sum_i \alpha^a_{c,i} \boldsymbol{v}_{c,i}\,. \tag{30}$$

These two latter equations make Eq. (15) fully specified, with the specification of the constant vector $\boldsymbol{g}_c$ and the initial condition $\dot{\boldsymbol{s}}^0$.

V. CoM Trajectory Solution

We solve the system (Eq. 15) using a Linear Model Predictive Control (LMPC). Let $\boldsymbol{x}_c = [\boldsymbol{s}_c^T \ \dot{\boldsymbol{s}}_c^T]^T$ be the parameter vector. Eq. (15) can be put into matrix form:

$$\dot{\boldsymbol{x}}_c = \begin{bmatrix} 0_3 & I_3 \\ 0_3 & 0_3 \end{bmatrix} \boldsymbol{x}_c + \begin{bmatrix} 0_3 \\ 1_3 M^{-1} \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} \boldsymbol{0}_3 \\ \boldsymbol{g}_c \end{bmatrix}$$
$$\dot{\boldsymbol{x}}_c = \mathcal{A}\boldsymbol{x}_c + \mathcal{B}\boldsymbol{u}_c + \boldsymbol{\mathcal{E}}_c.$$

where $\dot{\boldsymbol{x}}_c = [\dot{\boldsymbol{s}}_c^T \ \ddot{\boldsymbol{s}}_c^T]^T$. This is a continuous time system with $\boldsymbol{u} = \boldsymbol{F}_c$, the control vector, M the total mass of the system and $\boldsymbol{g}_c$ the contact distributed part of the gravity (computed with Eq. (29)). After discretization with a sampling time T, we get

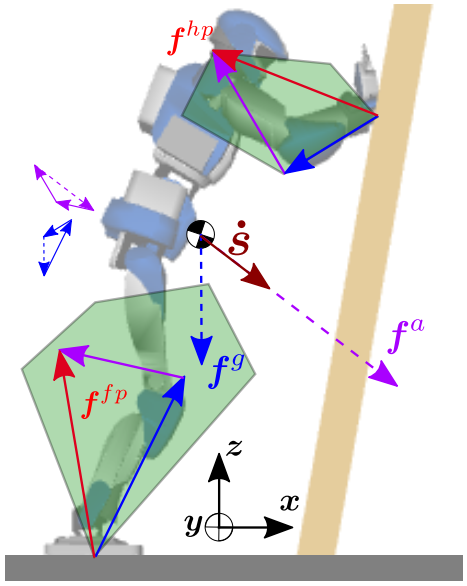


Fig. 2: 2D illustration of a potential force distribution made by the QP. The dotted arrows f^g and f^a are forces to be distributed. Plain arrows are their repartition in the polytopes. The vectors f^{fp} and f^{hp} have to remain inside their respective polytope. The dotted vector f^a is taken so it maximizes the linear momentum coming from the CoM velocity $\dot{s}$.

$$\begin{cases} A = \text{discrete}(\mathcal{A}) = \begin{bmatrix} I_3 & I_3 T \\ 0_3 & I_3 \end{bmatrix} \\ B = \text{discrete}(\mathcal{B}) = M^{-1} \begin{bmatrix} I_3 \frac{T^2}{2} \\ I_3 T \end{bmatrix} \\ \boldsymbol{E}_c = \text{discrete}(\boldsymbol{\mathcal{E}}_c) = \begin{bmatrix} \boldsymbol{g}_c \frac{T^2}{2} \\ \boldsymbol{g}_c T \end{bmatrix} \end{cases} \tag{31}$$

The sampling time is not necessarily set to the robot control loop time. Indeed the greater the MPC's sampling time the wider is the horizon of time. Then an interpolation is done to get the values at each control loop time. By recursion we can compute a prediction of the behavior of the simplified system [13]. We only need to know the initialization parameter of the system.

After N step, we get:

$$\boldsymbol{X}_c^N = \Phi \boldsymbol{x}_c^0 + \Psi \boldsymbol{U}_c + \boldsymbol{\zeta}_c \tag{32}$$

with

$$\boldsymbol{X}_c^N = [\boldsymbol{x}_c^{0\,T} \ \boldsymbol{x}_c^{1\,T} \ \dots \ \boldsymbol{x}_c^{N\,T}]^T \tag{33}$$
$$\boldsymbol{U}_c = [\boldsymbol{u}_c^{0\,T} \ \boldsymbol{u}_c^{1\,T} \ \dots \ \boldsymbol{u}_c^{N-1\,T}]^T \tag{34}$$

and where Φ, Ψ and $\boldsymbol{\zeta}_c$ are defined as in Eq. (35). Note that $\boldsymbol{x}_c^0 = [\boldsymbol{s}^{0\,T} \ \dot{\boldsymbol{s}}_c^{0\,T}]^T$ where $\dot{\boldsymbol{s}}_c^0$ is the CoM distributed velocity (Eq. (30)) and $\boldsymbol{s}^0$ is the intial position of the CoM (which is the inital position of all four virtual CoMs).

The objective function of the LMPC is

$$\begin{aligned} t_c^d &= (\boldsymbol{x}_c^N - \boldsymbol{x}_c^{\text{target}}) W_x (\boldsymbol{x}_c^N - \boldsymbol{x}_c^{\text{target}}) + \boldsymbol{U_c^T} W_U \boldsymbol{U}_c \\ &= \boldsymbol{U}_c^T Q_c \boldsymbol{U}_c + 2 l_c^T \boldsymbol{U}_c. \end{aligned} \tag{36}$$

$$\Phi = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}, \quad \Psi = \begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}, \quad \boldsymbol{\zeta}_c = \begin{bmatrix} \boldsymbol{E}_c \\ A\boldsymbol{E}_c + \boldsymbol{E}_c \\ \vdots \\ \sum_{i=0}^{i=N-1} A^i \boldsymbol{E}_c \end{bmatrix} \tag{35}$$

The target $\boldsymbol{x}_c^{\text{target}} = [\boldsymbol{s}_c^{\text{target T}} \ \dot{\boldsymbol{s}}_c^{\text{target }T}]^T$ is such that $\boldsymbol{s}_c^{\text{target}} = \boldsymbol{s}_c^0$ and $\dot{\boldsymbol{s}}_c^{\text{target}} = \boldsymbol{0}$. The weights W_x and W_U are chosen so that the priority is to reach a zero velocity, then maintain the position and lastly minimize the effort. Briefly, this means that $W_x(3..6) \gg W_x(0..2) > W_U$. The force limit constraint (computed via Eq. (13)) is added to the system constraints. We then solve the LMPCs corresponding to each contact to get the vectors $\boldsymbol{U}_c$ and deduce the forces to apply at the contact points. As the adaptive-QP of the robot allows us to set a trajectory task for the CoM, the forces are converted into CoM acceleration.

$$\ddot{\boldsymbol{s}}(t) = \frac{1}{M} \sum_c \boldsymbol{F}_c(t) + \boldsymbol{g}. \tag{37}$$

Integrating forward returns $\dot{\boldsymbol{s}}(t)$ and $\boldsymbol{s}(t)$. Those three trajectories are sent to the adaptive-QP.

Finally, since the LMPCs are launched again and again and that it targets a zero velocity about $T \times N_{\text{steps}}$ seconds, the robot can never reach it completely so a stopping criterion is necessary. This criterion can simply be based on the current CoM velocity norm.

Discussion: when extracting the sub-equation Eq. (15) from the complete equation Eq. (14), we have assumed that each generated mass-point are controlled by a dedicated limb of the robot (the kinematic chain from the contact point to its root). This means that the trunk of the robot, which closes kinematic chains between pairs of links and the environment, is not taken into account. As such, the polytope at Eq. (13) does not represent the full set of feasible forces at the contact point, but only the subset of forces that do not rely on closed kinematic chains. Requiring the computation of the whole-body torques limits polytope is out of the reach of state-of-the-art polyhedral libraries [14].

VI. Simulations

The simulation sets up HRP-4 pushed against a wall, as depicted in Fig. 2. Here, it is assumed that the pre-impact phase (treated in [8]) is already done and only the post-impact phase remains to be controlled. The control loop of HRP-4 is 5 ms, it is pushed on the back with a force of 300 N for 0.15 s. The wall is 10° inclined and we use an LMPC[1] with variables set as $N_{steps} = 15$ and $T = 15$ ms. The detection of the impact is done through the distance between the hands and the wall. As the algorithm computation time is higher than the control loop time, it launched is an estimated position of the hand touch the wall. This estimation is done with a simple Euler integration over 15 ms of the nearest point of a hand to the wall. To ensure that the adaptive-QP does not output high torque at the impact time, all tasks are removed but the minimization of the stiffness and damping coefficient. As soon as the LMPCs output the first results,

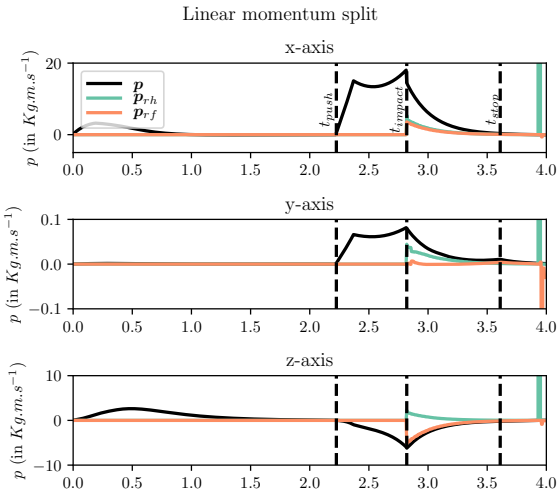


Fig. 3: Linear momentum and its splitted part

a CoM trajectory task and posture task (with relatively low weight) is added. With an Intel(R) Core(TM) i7-4900MQ CPU at 2.8GHz (up to 3.8GHz) with 4 cores and 8 running threads, the time required to compute the polytopes, the distribution force QP and the LMPCs is up to 15 ms. To improve performance, the contact polytopes[2] and the LMPCs are computed in parallel.

The stopping criterion has been set so that it is triggered when the norm of CoM velocity reached 1 cm.s^{-1}. From there, the weight of the posture task is increased and the CoM trajectory is updated to a fixed set-point task at the current CoM position. We then stop the adaptive-QP and revert to the standard QP whole-body controller.

As the problem is symmetric the FDQP gives arms and legs the same amount of linear momentum and gravity hence Fig. 3 to Fig. 6 only display right side contacts/joints. The linear momentum of the CoM (Fig. 3) smoothly decreases, on each axis, and reaches zero after roughtly 1.5 s. Notwithstanding the symmetrical properties of the problem, we can denote a deviation of the linear momentum on the y-axis. In Fig. 4, the FDQP choses to compensate the z-axis of the gravity with only the foot which is coherent since the maximal forces of the feet are on this axis. It is also interesting to note that the FDQP may give one or multiple contacts the ability to go along with the whole-body linear momentum, rather than against it. This is quite visible on Fig. 3 on the z-axis where after the impact time, the sum of the two feet is greater than the robot linear momentum. Thus the feet alone have to compensate more than needed. This is due to the fact that the FDQP gives to the hand

[1] https://github.com/vsamy/Copra

[2] The polyhedral library (cddlib) itself is not multi-threadable.

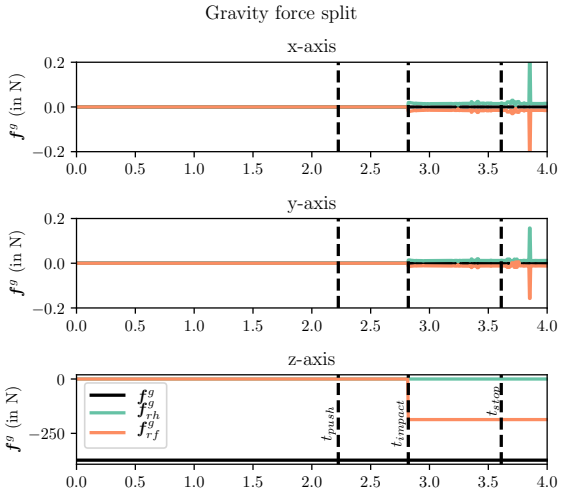


Fig. 4: Gravity force and its splitted part

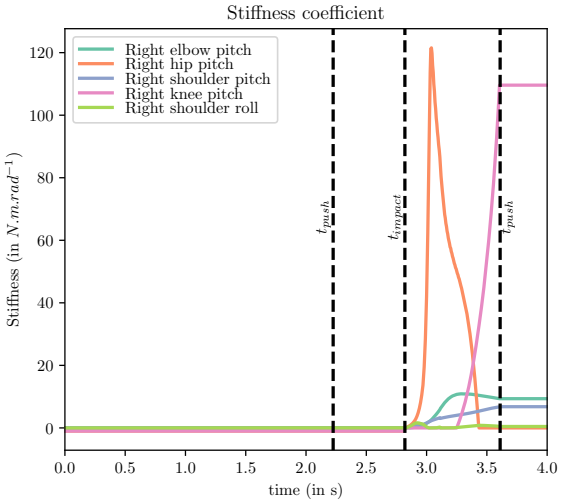


Fig. 5: Stiffness coefficient output of the adaptive-QP for several joints

linear momentum a sign opposite to the whole-body one. So, LMPCs attached to the hands are seeing the robot going upward while the feet sees the robot going downward with higher velocity.

The adaptive-QP is able to find the stiffness (Fig. 5) and damping (Fig. 6) coefficient for arms and legs. At impact time, the damping coefficient dominates the stiffness one since the error term is almost zero whereas velocity is high. It is very clear that the hip joint is making most of the effort to fulfill the tasks. From common sense, the elbow joint should have particapated more since it is the one that can directly damp the linear momentum but it is not. This can be explained by the fact that the hip motor has a torque limit 3.5 times superior to the elbow's one. Around $t = 3$ s, the knee joint takes over and is mainly set so that it maintains the height of the CoM and compensates gravity.

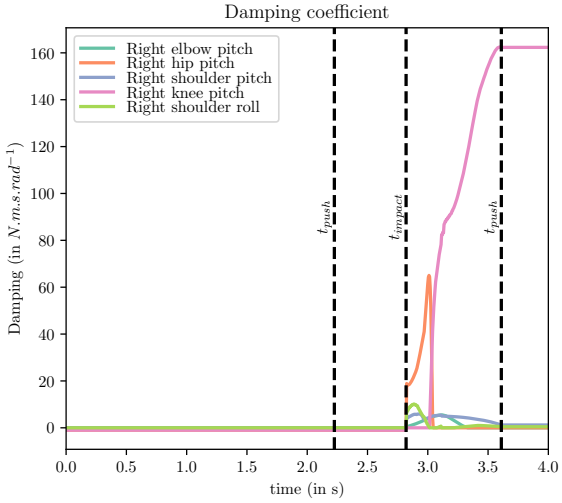


Fig. 6: Damping coefficient output of the adaptive-QP for several joints

VII. Conclusion and Future Work

We have presented an approach that allows a humanoid robot to actively control its compliance after falling and impacting its environment with its arms and/or legs. Parallel model predictive controllers are run at each of the four contacting limbs, the joint action of which is to absorb the undesired linear momentum accumulated during fall. These predictive controllers are instantiated by a force distribution quadratic-program (FDQP) that optimally distributes the initial momentum and gravity force across contacting limbs. Despite relying on reduced dynamic models, all predictive controllers are constrained by contact polytopes that encode both friction and joint torque limitations.

The approach was validated in full-body simulations with a model of the HRP-4 humanoid robot in falling scenarios. The simulation showed corroborating results making the linear momentum vanish after few seconds, in an active compliance way.

With our previous work [7] [8], we have now a complete framework for fall handling from pre-impact posture adoption, to impact-time motor PD-gains automatic adaptation, and with the present work, post-impact active compliance with MPC to optimally absorb the impact and bring the robot to a safe rest.

Although it is a very challenging experiment to be carried out on real hardware, we plan in a near future to test the method on HRP-4. Future work will also need to address sliding of contacts at impact points, reliable linear momentum estimation, visual estimation of the robot's surroundings for impact surfaces, and impact time detection are other open topics in fall control that are challenging and that we are actively researching.

References

[1] C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, M. Gennert, J. P. Graff, P. He, A. Jaeger, J. Kim, K. Knoedler, L. Li, C. Liu, X. Long, T. Padir, F. Polido, G. G. Tighe, and X. Xinjilefu, "No falls, no resets: Reliable humanoid behavior in the darpa robotics challenge," in *IEEE-RAS Int. Conf. on Humanoids*, Nov 2015, pp. 623–630.

[2] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *IEEE-RAS Int. Conf. on Humanoid Robots*, 2006, pp. 200–207.
[3] A. Sherikov, D. Dimitrov, and P. B. Wieber, "Balancing a humanoid robot with a prioritized contact force distribution," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*, Nov 2015, pp. 223–228.
[4] M. Battaglia, L. Blanchet, A. Kheddar, S. Kajita, and K. Yokoi, "Combining haptic sensing with safe physical interaction," in *IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, Saint Louis, MO, USA, 11-15 October 2009, pp. 231–236.
[5] G. De Magistris, A. Pajon, S. Miossec, and A. Kheddar, "Optimized humanoid walking with soft soles," *Robotics and Autonomous Systems*, vol. 95, pp. 52–63, 09 2017.
[6] N. G. T. Enrico Mingo Hoffman, Nicolas Perrin and D. G. Caldwell, "Upper limb compliant strategy to avoid falls exploiting external contacts for humanoid robots," 2013, int. Workshop on Human-Friendly Robotics.
[7] V. Samy and A. Kheddar, "Falls control using posture reshaping and active compliance," in *IEEE-RAS Int. Conf. on Humanoids*, 2015, pp. 908–913.
[8] V. Samy, K. Bouyarmane, and A. Kheddar, "QP-based Adaptive-Gains Compliance Control in Humanoid Falls," Feb. 2017, working paper or preprint. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01365108
[9] K. Bouyarmane and A. Kheddar, "Using a multi-objective controller to synthesize simulated humanoid robot motion with changing contact configurations," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 4414–4419.
[10] S. Caron, Q.-C. Pham, and Y. Nakamura, "Zmp support areas for multi-contact mobility under frictional constraints," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 67–80, Feb 2017.
[11] K. Fukuda and A. Prodon, *Double description method revisited*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 91–111.
[12] K. Bouyarmane and A. Kheddar, "On the dynamics modeling of free-floating-base articulated mechanisms and applications to humanoid whole-body dynamics and control," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*. IEEE, 2012, pp. 36–42.
[13] H. Audren, J. Vaillant, A. Kheddar, A. Escande, K. Kaneko, and E. Yoshida, "Model preview control in multi-contact motion-application to a humanoid robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 4030–4035.
[14] D. Avis, D. Bremner, and R. Seidel, "How good are convex hull algorithms?" *Computational Geometry*, vol. 7, no. 5-6, pp. 265–301, 1997.