Maintaining Balance of Mobile Manipulators for Safe Pick-Up Tasks

Francesco D'Orazio[1], Tommaso Belvedere[2], Spyridon G. Tarantos[3], Giuseppe Oriolo[1]

***Abstract*— This paper presents a novel method to maintain the dynamic balance of a Mobile Manipulator (MM) during the pick-up of heavy objects. The approach entails the generation of a preliminary reach-to-grasp trajectory, which is subsequently refined by an Optimization-Based Controller (OBC) formulated as a Quadratic Program (QP). The trajectory is modified in a minimal fashion to ensure that the robot maintains balance during the reaching phase and remains balanced when the payload is grasped. This is accomplished by incorporating a balance constraint into the OBC that predicts the Zero Moment Point (ZMP) position of the robot at the beginning of the pick-up phase. This accounts for the gravitational and inertial effects that the object has on the robot. The method is validated through simulations conducted with the TIAGo robot in Gazebo. The results demonstrate that the proposed approach effectively prevents the robot from tipping over when the payload is considered.**

I. Introduction

In recent decades, fixed-base manipulators have been widely used for pick-and-place operations. However, their limited workspace restricts their range of applications, leading to the development of Mobile Manipulators (MM), where a robotic arm is mounted on a mobile base. These robotic systems typically use kinematic commands to control their End-Effector (EE) movements [1]–[3]. When manipulating heavy objects or performing fast motions, especially with narrow bases designed for tight spaces, the robot may be at risk of tipping over. Moreover, during a pick-up motion the system dynamic parameters change instantaneously, risking a loss of balance if the state of the robot is not appropriate and the object is heavy. Therefore, to maintain safety the robot must adjust its configuration before grasping the object.

The Zero Moment Point (ZMP) method evaluates robot balance by identifying the point where the ground reaction forces act. Originally introduced by Vukobratovic for bipedal locomotion [4], and further developed by Takanishi [5], it has been applied to MMs by Sugano and Huang [6], [7]. Various researchers have addressed MM balance while performing specific tasks [8]–[12], either by generating corrective actions to modify the overall inertia tensor [8], [9] or by formulating balance constraints within optimization problems [10]–[13]. These constraints filter the planned trajectory to keep the ZMP within the balance region. A more robust approach uses the Control Barrier Function (CBF) framework [14], [15], offering a tunable parameter to balance robustness and performance. CBFs define a Safe Set in the state space that can be rendered invariant under feedback where the robot remains balanced. This framework can be extended to discrete-time systems using Discrete-Time Control Barrier Functions (DT-CBFs) [16], [17], and is applicable to both autonomous control and teleoperation [18].

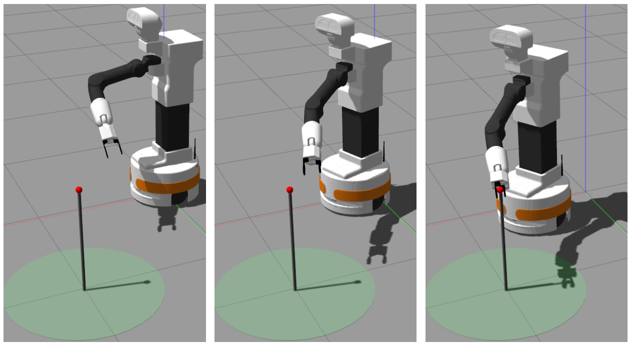

Fig. 1: Snapshots of the motion of the TIAGo robot in Gazebo, obtained implementing the proposed method.

This work proposes a Real-time EE Trajectory Planner (REETP) that generates a reach-to-grasp trajectory, elaborated by an Optimization-Based Controller (OBC) that minimally modifies it to ensure (i) balance during the reaching motion and (ii) stability when lifting the object. The trajectory planner allows the EE to reach the target, while the OBC, formulated as a Quadratic Program (QP), enforces balance via CBF constraints.

The novel contribution is the inclusion of an additional balance constraint in the OBC to predict the ZMP position post-grasping, adjusting the configuration to maintain balance during dynamic parameter changes. As a second contribution, the DT-CBF condition for QP has been manipulated to avoid the explicit dynamic extension of the controller. In fact, these constraints typically require that the time derivative of the CBF is affine with respect to the decision variables. However, in the scenario presented in this paper, the CBF depends directly on the system input, forcing the QP to be defined at a higher differential level, increasing the number of the variables in the state vector.

Simulation tests using the TIAGo robot in the Gazebo environment demonstrate the effectiveness of the method which is, by design, applicable to any MM. Results show that, unlike cases without the additional balance constraint, this approach ensures stability when the robot lifts objects by adjusting its state.

The paper is structured as follows: Section II defines the problem and states the objectives of the proposed method;

[1] Department of Computer, Control and Management Engineering, Sapienza University of Rome, Rome, Italy. E-mail: {dorazio, oriolo}@diag.uniroma1.it.

[2] CNRS, Université de Rennes, Inria, IRISA, Rennes, France. E-mail: tommaso.belvedere@irisa.fr.

[3] Center of AI & Robotics (CAIR) and Engineering Division, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates. E-mail: spyridon.tarantos@nyu.edu.

Section III reports on the modeling and analytical computation of the ground reaction forces; Section IV presents the proposed method; Section V shows the simulation settings and results and Section VI reports some concluding remarks.

II. Problem Formulation

Consider a MM operating in a 3-dimensional workspace $\mathcal{W}$, moving on an horizontal surface. The configuration of the robot takes values in an n-dimensional space and is denoted by $\boldsymbol{q} \in \mathbb{R}^n$. The pose of the EE, namely position and orientation, is denoted by $\boldsymbol{\xi} \in SE(3)$ and is related to the robot configuration via the forward kinematic map $\boldsymbol{\xi} = \boldsymbol{f}(\boldsymbol{q})$. The robot has to pick up a payload with its EE. Let us denote by $\bar{\boldsymbol{\xi}} \in SE(3)$ a desired pose that allows the EE to rigidly grasp the payload.

The objective is to generate in real time a whole-body motion that:

R1. starts from the robot initial configuration $\boldsymbol{q}_s$ and brings the robot EE in pose $\bar{\boldsymbol{\xi}}$;

R2. is kinodynamically feasible, in the sense that it is compatible with the robot dynamics and respects constraints in joint, velocity and torque level;

R3. maintains the robot balance, in the sense that the robot wheels remain in contact with the ground at all times;

R4. permits the robot to maintain its balance immediately after the object is grasped.

Note that R2-R4 must be satisfied in order to ensure that the robot maintains a safe behavior.

III. Robot Model

Consider a MM comprised of a mobile base with n_w driving wheels and a manipulator arm with n_m degrees of freedom. Following the same approach for modeling presented in [13], it is assumed that the wheels are in point contact with the ground and the friction is large enough to prevent the driving wheels from slipping. As for the caster wheels, they only account for robot balance, i.e. their point of contact with the ground is considered fixed with respect to the mobile base and their dynamics are neglected. To model the pose of the mobile base, $n_b = 6$ fictitious joints are considered to connect the base with the world frame. The first three are prismatic joints representing the displacement along the x, y and z axes and the last three represent the base rotation along roll, pitch and yaw axes $(\theta_x, \theta_y, \theta_z)$. These variables are collected in a vector $\boldsymbol{q}_b = (x_b, y_b, z_b, \theta_x, \theta_y, \theta_z)$. The generalized variables of the manipulator are represented by $\boldsymbol{q}_m = (\theta_1, \dots, \theta_{n_m})$, whereas $\boldsymbol{q}_w$ is the vector associated with the driving wheels. The vector of generalized coordinates of the robot is $\boldsymbol{q} = (\boldsymbol{q}_b, \boldsymbol{q}_w, \boldsymbol{q}_m)$ of dimension $n = n_b + n_w + n_m$.

The equations of motion [19] are expressed as

$$\begin{gathered} \boldsymbol{M}_{\boldsymbol{q}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{n}_{\boldsymbol{q}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{E}_q(\boldsymbol{q})\boldsymbol{\tau} + \boldsymbol{A}(\boldsymbol{q})\boldsymbol{\lambda} \\ \boldsymbol{A}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}, \end{gathered} \tag{1}$$

where $\boldsymbol{M}_{\boldsymbol{q}}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the mass matrix, $\boldsymbol{n}_{\boldsymbol{q}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^n$ contains the centrifugal, Coriolis and gravitational terms, $\boldsymbol{E}_{\boldsymbol{q}}(\boldsymbol{q}) \in \mathbb{R}^{n \times n_\tau}$ maps the inputs $\boldsymbol{\tau} \in \mathbb{R}^{n_\tau}$ to the generalized forces that produce work on the generalized coordinates with $n_{\boldsymbol{\tau}} = n_w + n_m$. The model is subject to κ kinematic constraints collected in a matrix $\boldsymbol{A}(\boldsymbol{q}) \in \mathbb{R}^{n \times \kappa}$, whereas $\boldsymbol{\lambda} \in \mathbb{R}^\kappa$ is the vector of the contact forces. A matrix $\boldsymbol{S}(\boldsymbol{q}) \in \mathcal{N}(\boldsymbol{A}^T(\boldsymbol{q}))$ can be defined that relates the generalized velocities and accelerations $\dot{\boldsymbol{q}}$, $\ddot{\boldsymbol{q}} \in \mathbb{R}^n$ to the pseudo velocities $\boldsymbol{\nu} = (\dot{\boldsymbol{q}}_w, \dot{\boldsymbol{q}}_m) \in \mathbb{R}^{(n-\kappa)}$ and their time derivative $\dot{\boldsymbol{\nu}}$,

$$\begin{aligned} \dot{\boldsymbol{q}} &= \boldsymbol{S}(\boldsymbol{q})\boldsymbol{\nu} \\ \ddot{\boldsymbol{q}} &= \boldsymbol{S}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \dot{\boldsymbol{S}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{\nu}. \end{aligned} \tag{2}$$

Premultiplying (1) by $\boldsymbol{S}^T(\boldsymbol{q})$ leads to

$$\boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{M}_{\boldsymbol{q}}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{n}_{\boldsymbol{q}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{E}_{\boldsymbol{q}}(\boldsymbol{q})\boldsymbol{\tau}. \tag{3}$$

The reduced equation of motion [19] is obtained substituting (2) into (3)

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{\nu}} + \boldsymbol{n}(\boldsymbol{q}, \boldsymbol{\nu}) = \boldsymbol{S}^T(\boldsymbol{q})\boldsymbol{E}_{\boldsymbol{q}}(\boldsymbol{q})\boldsymbol{\tau}, \tag{4}$$

where

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q}) &= \boldsymbol{S}(\boldsymbol{q})^T \boldsymbol{M}_{\boldsymbol{q}}(\boldsymbol{q})\boldsymbol{S}(\boldsymbol{q}) \\ \boldsymbol{n}(\boldsymbol{q}, \dot{\boldsymbol{q}}) &= \boldsymbol{S}^T(\boldsymbol{q})(\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{S}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{\nu} + \boldsymbol{n}_{\boldsymbol{q}}(\boldsymbol{q}, \dot{\boldsymbol{q}})). \end{aligned}$$

Since the MM moves of a flat surface, $\boldsymbol{q}_f = (z_b, \theta_x, \theta_y) \in \boldsymbol{q}_b$ are constant. The balance of the system is determined by the reaction forces, which can be analytically computed through the dynamic model due to the presence of $\boldsymbol{q}_f$.

As in [13], two constant selection matrices $\boldsymbol{Q}_a \in \mathbb{R}^{(n-3)\times n}$ and $\boldsymbol{Q}_f \in \mathbb{R}^{3 \times n}$ can be defined, dividing the dynamics variables into $\boldsymbol{q}_f = \boldsymbol{Q}_f \boldsymbol{q}$ and $\boldsymbol{q}_a = \boldsymbol{Q}_a \boldsymbol{q}$, representing the fixed and active parts of the dynamics respectively. Since $\ddot{z}_b$, $\ddot{\theta}_x$ and $\ddot{\theta}_y$ are identically zero, the relation $\ddot{\boldsymbol{q}} = \boldsymbol{Q}_a^T \ddot{\boldsymbol{q}}_a$ holds. For simplicity, the dependencies are omitted from the following equations. Premultiplying (1) by $\boldsymbol{Q}_a$ and $\boldsymbol{Q}_f$ yields to

$$\boldsymbol{Q}_a \boldsymbol{M}_{\boldsymbol{q}} \boldsymbol{Q}_a^T \ddot{\boldsymbol{q}}_a + \boldsymbol{Q}_a \boldsymbol{n}_{\boldsymbol{q}} = \boldsymbol{Q}_a \boldsymbol{E}_{\boldsymbol{q}} \boldsymbol{\tau} + \boldsymbol{Q}_a \boldsymbol{A} \boldsymbol{\lambda} \tag{5}$$

$$\boldsymbol{Q}_f \boldsymbol{M}_{\boldsymbol{q}} \boldsymbol{Q}_a^T \ddot{\boldsymbol{q}}_a + \boldsymbol{Q}_f \boldsymbol{n}_{\boldsymbol{q}} = \boldsymbol{Q}_f \boldsymbol{A} \boldsymbol{\lambda}. \tag{6}$$

Consider a matrix $\boldsymbol{G} \in \mathcal{N}(\boldsymbol{A}^T \boldsymbol{Q}_a^T)$ such that

$$\begin{aligned} \dot{\boldsymbol{q}}_a &= \boldsymbol{G}\boldsymbol{\nu} \\ \ddot{\boldsymbol{q}}_a &= \boldsymbol{G}\dot{\boldsymbol{\nu}} + \dot{\boldsymbol{G}}\boldsymbol{\nu}. \end{aligned} \tag{7}$$

Premultiplying (5) by $\boldsymbol{G}^T$, together with (7) yields to the reduced equation of motion of the active dynamics

$$\boldsymbol{M}_a \dot{\boldsymbol{\nu}} + \boldsymbol{n}_a = \boldsymbol{E}_a \boldsymbol{\tau}, \tag{8}$$

where $\boldsymbol{M}_a = \boldsymbol{G}^T \boldsymbol{Q}_a \boldsymbol{M}_{\boldsymbol{q}} \boldsymbol{Q}_a^T \boldsymbol{G}$, $\boldsymbol{n}_a = \boldsymbol{G}^T \boldsymbol{Q}_a \boldsymbol{M}_{\boldsymbol{q}} \boldsymbol{Q}_a^T \dot{\boldsymbol{G}} \boldsymbol{\nu} + \boldsymbol{G}^T \boldsymbol{Q}_a \boldsymbol{n}_{\boldsymbol{q}}$ and $\boldsymbol{E}_a = \boldsymbol{G}^T \boldsymbol{Q}_a \boldsymbol{E}_{\boldsymbol{q}}$.

The reaction forces at the generalized coordinate level can be computed analytically from (6) using (7) and (8). Since $(f_z \, \mu_x \, \mu_y)^T = \boldsymbol{Q}_f \boldsymbol{A} \boldsymbol{\lambda}$

$$\begin{pmatrix} f_z \\ \mu_x \\ \mu_y \end{pmatrix} = \boldsymbol{\Sigma}\boldsymbol{\tau} + \boldsymbol{\alpha}, \tag{9}$$

with $\boldsymbol{\Sigma} = \boldsymbol{Q}_f \boldsymbol{M}_{\boldsymbol{q}} \boldsymbol{Q}_a^T \boldsymbol{G} \boldsymbol{M}_a^{-1} \boldsymbol{E}_a$ and $\boldsymbol{\alpha} = \boldsymbol{Q}_f \boldsymbol{M}_{\boldsymbol{q}} \boldsymbol{Q}_a^T \dot{\boldsymbol{G}} \boldsymbol{\nu} + \boldsymbol{Q}_f \boldsymbol{n}_{\boldsymbol{q}} - \boldsymbol{Q}_f \boldsymbol{M}_{\boldsymbol{q}} \boldsymbol{Q}_a^T \boldsymbol{G} \boldsymbol{M}_a^{-1} \boldsymbol{n}$.

Remark 1: The relation (9) shows that the ground reaction forces are linear with respect to the torques and can be computed from the system dynamics knowing the robot state.

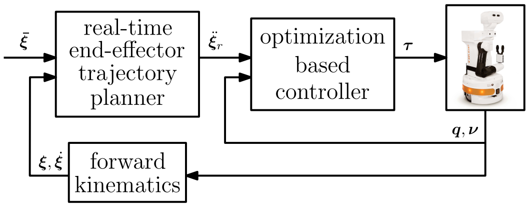

Fig. 2: The block scheme of the proposed method. The planner takes the desired pose $\bar{\boldsymbol{\xi}}$ as input and generates acceleration commands $\ddot{\boldsymbol{\xi}}_r$, which are then used by the OBC to obtain the torque command $\boldsymbol{\tau}$.

IV. Proposed Approach

The proposed motion generation approach consists of two modules, namely the Real-time End-Effector Trajectory Planner (REETP) and the Optimization-Based Controller (OBC) as illustrated in Fig. 2. The first module considers the desired pose $\bar{\boldsymbol{\xi}}$ that the EE must assume together with the current EE pose, which is related to the state of the robot $(\boldsymbol{q}, \boldsymbol{\nu})$ by the forward kinematics map. As an output, it provides a reference end-effector acceleration $\ddot{\boldsymbol{\xi}}_r$. The second module takes as an input the current state of the robot and the reference $\ddot{\boldsymbol{\xi}}_r$. Through the solution of a QP generates appropriate control commands for the robot $\boldsymbol{\tau}$ that track $\ddot{\boldsymbol{\xi}}_r$ while satisfying requirements R2-R4.

The task can be divided into two phases to highlight the different parameters of the dynamic model of the robot: *reach*, in which the nominal parameter of the robot are considered, and *pick up*, where the model also accounts for the payload, rigidly grasped in the EE.

A. Real-Time End-Effector Trajectory Planner

The trajectory planner implemented in this work is based on a PD control law, which imposes exponential convergence of the EE to the desired pose

$$\ddot{\boldsymbol{\xi}}_r = \boldsymbol{K}_P \boldsymbol{e} - \boldsymbol{K}_D \dot{\boldsymbol{\xi}}, \tag{10}$$

where $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ are diagonal positive definite matrices of appropriate dimensions. The error $\boldsymbol{e}$ is composed of a linear and an angular component $\boldsymbol{e} = (\boldsymbol{e}_p^T, \boldsymbol{e}_\phi^T)^T$. The position error $\boldsymbol{e}_p$ is simply defined as the difference between the desired and actual position of the EE, $\boldsymbol{e}_p = \bar{\boldsymbol{\xi}}_p - \boldsymbol{\xi}_p$, while $\boldsymbol{e}_\phi$ depends on the representation used to describe its orientation. In this paper, the error is defined using rotation matrices [20]. Given the desired EE orientation $\bar{\boldsymbol{R}}(\bar{\boldsymbol{\xi}})$ and the current one $\boldsymbol{R}(\boldsymbol{\xi})$, defined with respect to the same reference frame, $\boldsymbol{e}_\phi$ can be obtained by

$$\boldsymbol{e}_\phi = \frac{1}{2}\begin{pmatrix}[\bar{\boldsymbol{r}}_1]_\times^T & [\bar{\boldsymbol{r}}_2]_\times^T & [\bar{\boldsymbol{r}}_3]_\times^T\end{pmatrix}\begin{pmatrix}\boldsymbol{r}_1\\ \boldsymbol{r}_2\\ \boldsymbol{r}_3\end{pmatrix} \in \mathbb{R}^3,$$

where $\boldsymbol{r}_i$ and $\bar{\boldsymbol{r}}_i$ are the i-th column of matrices $\boldsymbol{R}$ and $\bar{\boldsymbol{R}}$ respectively, and $[\bar{\boldsymbol{r}}_i]_\times$ represents the skew-symmetric matrix constructed from $\bar{\boldsymbol{r}}_i$.

B. Optimization-Based Controller

The OBC is formulated as a QP problem that elaborates the planned acceleration $\ddot{\boldsymbol{\xi}}_r$ to enforce kinodynamic feasibility and two sets of constraints: the *Balance Constraint* (BC), designed to enforce that the robot ZMP is within its Support Polygon (SP), and the *Preemptive Balance Constraint* (PBC), aimed at maintaining balance after pick up. The decision variables are the joint torques of the robot. For the sake of clarity, the cost function and the QP constraints are expressed in terms of joint accelerations which can be mapped to generalized forces via (4).

The cost function to be minimized takes into account the squared error norm between the desired and actual task acceleration plus a regularization term

$$\mathcal{I}_{\text{cost}} = ||\boldsymbol{J}\dot{\boldsymbol{\nu}} + \dot{\boldsymbol{J}}\boldsymbol{\nu} - \ddot{\boldsymbol{\xi}}_r||^2 + ||\boldsymbol{\tau}||^2_{\boldsymbol{\Gamma}}. \tag{11}$$

To ensure the existence of an optimal solution, it is essential to confirm that the Hessian of the optimization problem is positive definite [21]. The Hessian, defined as the second derivative of the Lagrangian function with respect to the decision variables, is expressed as $\boldsymbol{H} = \boldsymbol{J}^T\boldsymbol{J} + \boldsymbol{\Gamma}$. For a redundant robot, such as those considered in this paper, the matrix $\boldsymbol{J}^T\boldsymbol{J}$ is inherently rank deficient. The regularization term $||\boldsymbol{\tau}||^2_{\boldsymbol{\Gamma}}$ is added to guarantee the convexity of $\boldsymbol{H}$ utilizing a diagonal positive definite weighting matrix $\boldsymbol{\Gamma}$, which penalizes the norm of the vector of decision variables.

The commanded torque $\boldsymbol{\tau}$ has to be compatible with the joint limits on $\boldsymbol{q}_m$ and with the velocity limits on the entire vector $\boldsymbol{\nu}$. Since the state $(\boldsymbol{q}, \boldsymbol{\nu})$ is already determined at the current time t_k, these constraints are imposed on the predicted state at time instant $t_{k+1} = t_k + \delta$, where $\delta \in \mathbb{R} > 0$ is the sampling time of the discrete-time controller:

$$\boldsymbol{\nu}_{k+1} = \boldsymbol{\nu} + \delta\,\dot{\boldsymbol{\nu}} \tag{12}$$

$$\boldsymbol{q}_{m,k+1} = \boldsymbol{q}_m + \delta\,\dot{\boldsymbol{q}}_m + \frac{\delta^2}{2}\,\ddot{\boldsymbol{q}}_m. \tag{13}$$

After simple manipulation, (12) becomes

$$\begin{aligned}-\delta\,\dot{\boldsymbol{\nu}} &\le \boldsymbol{\nu} - \boldsymbol{\nu}_{\min}\\ \delta\,\dot{\boldsymbol{\nu}} &\le \boldsymbol{\nu}_{\max} - \boldsymbol{\nu},\end{aligned}$$

whereas (13) reads as

$$\begin{aligned}-\frac{\delta^2}{2}\,\ddot{\boldsymbol{q}}_m &\le \boldsymbol{q}_m - \boldsymbol{q}_{\min} + \delta\,\dot{\boldsymbol{q}}_m\\ \frac{\delta^2}{2}\,\ddot{\boldsymbol{q}}_m &\le \boldsymbol{q}_{\max} - \boldsymbol{q}_m - \delta\,\dot{\boldsymbol{q}}_m.\end{aligned}$$

The QP to be solved is:

$$\begin{aligned}\min_{\boldsymbol{\tau}\in[\boldsymbol{\tau}_{\min}, \boldsymbol{\tau}_{\max}]} \quad & ||\boldsymbol{J}\dot{\boldsymbol{\nu}} + \dot{\boldsymbol{J}}\boldsymbol{\nu} - \ddot{\boldsymbol{\xi}}_r||^2 + ||\boldsymbol{\tau}||^2_{\boldsymbol{\Gamma}}\\ \text{subject to} \quad & \text{joint bounds}\\ & \text{balance constraints}\\ & \text{reconfiguration constraints}\end{aligned} \tag{14}$$

C. Balance Constraint

The goal of the BC is to enforce balance throughout the motion of the robot. A mechanical system is in balance if its ZMP lies within the SP of the structure.

The ZMP position $\boldsymbol{p}_{\text{zmp}}$ can be easily derived using the Newton-Euler equations. From the principle of action and reaction between forces and torques exchanged by consecutive joints, it is possible to compute the wrench $\boldsymbol{\mathcal{F}} = (\boldsymbol{\mu},\, \boldsymbol{f})$ that the base exchanges with the ground [19]. The wrench $\boldsymbol{\mathcal{F}}$ can

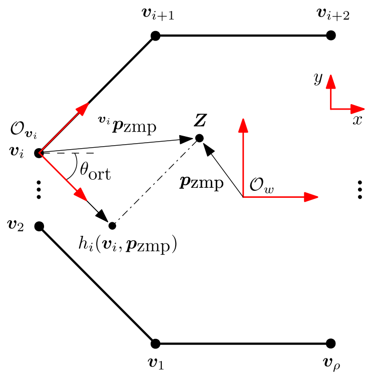

Fig. 3: The i-th DT-CBF represents the signed distance between the ZMP position and the edge connecting $\boldsymbol{v}_i$ and $\boldsymbol{v}_{i+1}$. It is computed as the x-component of the $\boldsymbol{p}_{\mathrm{zmp}}$ expressed in the reference frame $RF_{\boldsymbol{v}_i}$.

be computed either using (9) by including the base variables $\boldsymbol{q}_b$ in the model (Section III), or by extending the algorithm in $SE(3)$ to compute the generalized forces exchanged in all directions [8], [9], [22], [23]. Having computed the ground reaction wrench, the ZMP position can be expressed as

$$\boldsymbol{p}_{\mathrm{zmp}} = \begin{pmatrix} x_{\mathrm{zmp}} \\ y_{\mathrm{zmp}} \\ z_{\mathrm{zmp}} \end{pmatrix} = \begin{pmatrix} -\mu_y/f_z \\ \mu_x/f_z \\ 0 \end{pmatrix}. \tag{15}$$

The SP of a MM can be modeled with linear inequalities that connect the vertices of the polyhedron obtained from the wheel positions $\boldsymbol{v}_i \in \mathcal{V}$ with $i = 1, \ldots, \rho$ (see Fig. 3). To evaluate the balance condition, the signed distance between the ZMP position and the i-th edge connecting two vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_{i+1}$ is considered. Referring to Fig. 3, the ZMP position in the world frame $\boldsymbol{p}_{\mathrm{zmp}}$, is expressed in a reference frame $RF_{\boldsymbol{v}_i}$ that has the y-axis aligned with the edge, and the x-axis directed towards the center of the SP. The origin of $RF_{\boldsymbol{v}_i}$ is placed at the i-th vertex, so ${}^{\boldsymbol{v}_i}\boldsymbol{p}_{\mathrm{zmp}}$ is obtained as

$${}^{\boldsymbol{v}_i}\boldsymbol{p}_{\mathrm{zmp}} = \boldsymbol{R}^T(\theta_{\mathrm{ort}})(\boldsymbol{p}_{\mathrm{zmp}} - \boldsymbol{v}_i), \tag{16}$$

where θ_{ort} represents the angle between $RF_{\boldsymbol{v}_i}$ and the world frame, and $\boldsymbol{R}(\theta_{\mathrm{ort}})$ is the associated rotation matrix.

The x-component of (16) can be written as

$$h_i(\boldsymbol{v}_i, \boldsymbol{p}_{\mathrm{zmp}}) = \cos\theta_{\mathrm{ort}}(p_{\mathrm{zmp},x} - v_{i,x}) + \sin\theta_{\mathrm{ort}}(p_{\mathrm{zmp},y} - v_{i,y}). \tag{17}$$

By construction, $h_i > 0$ whenever the ZMP is inside the SP of the robot, $h_i = 0$ whenever it is on the i-th edge, while $h_i < 0$ whenever it is outside. Therefore, enforcing $h_i \geq 0$ is a necessary and sufficient condition to maintain balance.

However, a more robust approach consists in using DT-CBFs, which enforces safety for a generic nonlinear system, and that is briefly introduced before being applied to the problem. Safety is defined through a function $h(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$, used to generate a Safe Set $\mathcal{C}$ that can be rendered invariant under feedback [16], [17]:

$$\mathcal{C} = \{\boldsymbol{x} \in \mathcal{X} : h(\boldsymbol{x}) \geq 0\}$$
$$\partial\mathcal{C} = \{\boldsymbol{x} \in \mathcal{X} : h(\boldsymbol{x}) = 0\}.$$

If the relation

$$\Delta h(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq -\gamma h(\boldsymbol{x}_k), \qquad \gamma \in (0, 1] \tag{18}$$

where $\Delta h(\boldsymbol{x}_k, \boldsymbol{u}_k) = h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}_k)$ and $\boldsymbol{u}$ represents the input of the system. If (18) holds for all time instants $t_k \in \mathbb{Z}^+$, the set $\mathcal{C}$ is forward invariant, i.e. the evolution of the state remains confined in the set starting from $x(0) \in \mathcal{C}$, and $h(\boldsymbol{x})$ is a DT-CBF. Condition (18) can be equivalently written as

$$h(\boldsymbol{x}_{k+1}) \geq (1 - \gamma)h(\boldsymbol{x}_k). \tag{19}$$

Remark 2: Condition (19) fixes the exponential rate of the function, in fact at $t_k \in \mathbb{Z}^+$, $h(\boldsymbol{x}_k) \geq (1-\gamma)^k h(\boldsymbol{x}_0)$.

Remark 3: If $\boldsymbol{x}_k \in \mathcal{C}$, condition (19) is more restrictive than the simple constraint $h(\boldsymbol{x}_{k+1}) \geq 0$; whereas if $\boldsymbol{x}_k \notin \mathcal{C}$, it imposes exponential convergence toward the safe set.

The DT-CBF in (19) is a function of the current state only, whereas (17) is a function of both state and input since (9) shows that the ZMP depends directly on the torques. Therefore, to apply DT-CBFs for constraining the ZMP position, the controller would have to be defined at a higher differential level including $\boldsymbol{u}$ as a part of the state. Considering a dynamically extended system of the form

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{u}_k)$$
$$\boldsymbol{u}_{k+1} = \boldsymbol{v}_k,$$

condition (19) becomes

$$h(\boldsymbol{x}_{k+1}, \boldsymbol{u}_{k+1}) \geq (1 - \gamma)h(\boldsymbol{x}_k, \boldsymbol{u}_k),$$

which is nonlinear and therefore cannot be used inside a QP. However, at time t_{k-1}, the same condition must have been satisfied for some $\boldsymbol{u}_k = \boldsymbol{v}_{k-1}$ for the system to be safe:

$$h(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq (1 - \gamma)h(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}). \tag{20}$$

Then, one can enforce (20) using the system without dynamic extension to compute the feasible $\boldsymbol{u}_k^*$, solution of the QP, for which the equivalent condition on $\boldsymbol{v}$ is satisfied at time t_{k-1}, implying safety.

This can be applied to impose a safety condition directly on (17), as the resulting constraint is linear. Substituting (17) into (20), together with (15), yields

$$-\cos\theta_{\mathrm{ort}}\left(\frac{\mu_y}{f_z} + v_{i,x}\right) + \sin\theta_{\mathrm{ort}}\left(\frac{\mu_x}{f_z} - v_{i,y}\right) \geq (1-\gamma)h_i^{\mathrm{prev}},$$

where $h_i^{\mathrm{prev}} = h_i(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}^*)$. Multiplying both sides by f_z, allows to collect the ground reaction forces in a vector

$$\begin{pmatrix} \cos\theta_{\mathrm{ort}} v_{i,x} + \sin\theta_{\mathrm{ort}} v_{i,y} + (1-\gamma)h_i^{\mathrm{prev}} \\ -\sin\theta_{\mathrm{ort}} \\ \cos\theta_{\mathrm{ort}} \end{pmatrix}^T \begin{pmatrix} f_z \\ \mu_x \\ \mu_y \end{pmatrix} \leq 0.$$

Denoting as $\boldsymbol{\beta}$ the vector multiplying the ground reaction forces, the BC can be expressed linearly in terms of the generalized forces using (9):

$$\boldsymbol{\beta}^T \boldsymbol{\Sigma} \boldsymbol{\tau} \leq -\boldsymbol{\beta}^T \boldsymbol{\alpha}. \tag{21}$$

D. Preemptive Balance Constraint

The PBC represents the main contribution of this work and its purpose is to force the *Virtual ZMP* (VZMP) to remain within the SP. The VZMP represents the ZMP position computed with the augmented dynamic model of the robot, namely the model obtained when the payload is rigidly attached to the gripper. After the pick-up phase, the VZMP becomes the ZMP due to the change in the model parameters, meaning that the PBC affects the optimization problem only during the reach phase. At that moment, balance is affected by the configuration of the robot and by the mass of the payload. However, the BC does not consider these influences, lacking critical information for the optimization problem, which can lead to a potential tip-over of the robot.

The BC and the PBC are both designed to maintain balance, however, the PBC cannot replace the BC in general due to the mismatch between the models before grasping the object. Consequently, relying solely on the PBC in the OBC may result in the generation of commands that destabilize the system when the robot is in certain configurations. For this reason, the OBC incorporates both the BC and the PBC constraints during the reach phase.

The procedure for constructing the PBC is the same as presented in Section IV-C, but it accounts for the augmented dynamic model. However, considering the full mass of the object from the start of the motion would result in a more conservative condition than needed. To avoid this, we define a *transition region*, i.e., a cylindrical area of radius r_{safe} around the payload, and let the mass increase linearly in this region:

$$m_{\text{PBC}} = m\left(1 - \min\left(1, \frac{||\boldsymbol{e}_p||}{r_{\text{safe}}}\right)\right), \tag{22}$$

where m is the actual mass of the object. Outside the region, $m_{\text{PBC}} = 0$ and the PBC coincides with the BC.

V. Simulations

To highlight the effectiveness of the proposed method, two simulations were performed using TIAGo robot. All simulations were run in Gazebo, on an Intel Core i9-9900K CPU at 3.60 GHz. The QP is solved using the Eigen-QuadProg library. The control loop takes 5 ms on average, allowing for real time implementation.

The TIAGo robot is equipped with a differential drive base, a prismatic joint in the torso, and an arm with $n_m = 7$ degrees of freedom. The mobile base is fitted with four passive caster wheels and $n_w = 2$ driving wheels. Throughout this work, the position of the torso is held constant.

The base of TIAGo is velocity-controlled and its torque commands are obtained through the relation

$$\tau_i = k_{\omega_i}(\omega_i - \dot{q}_i), \qquad i = \{L, R\}, \tag{23}$$

where k_{ω_i} is a fixed gain chosen by the manufacturer, ω_i is the velocity command to each wheel and $(\dot{q}_L, \dot{q}_R)$ are the current velocities of the left and right wheels respectively. Since the relation is affine with respect to ω_L and ω_R, they can be included in the QP presented in Section IV-B as decision variables, including (23) as additional equality constraints.

TIAGo has a heavy and large base, and even in simulation it is difficult to get the ZMP of the robot at the edge of the SP with realistic movements. To highlight the effects of the balance constraints, it is then treated as having a smaller SP.

Animations of the simulations can be found in the accompanying video (`https://youtu.be/7601Cj4qSuU`).

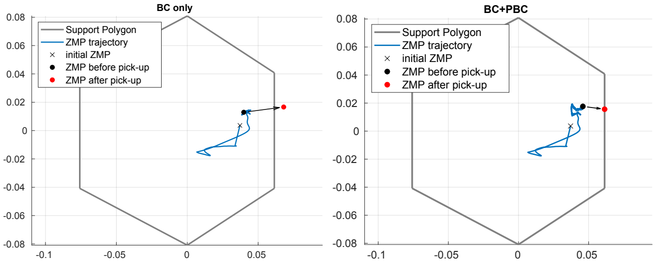

Fig. 4: Simulation 1: ZMP profiles obtained with BC only vs. BC plus PBC (in m). With BC only, the ZMP jumps outside the SP during the pick-up phase.

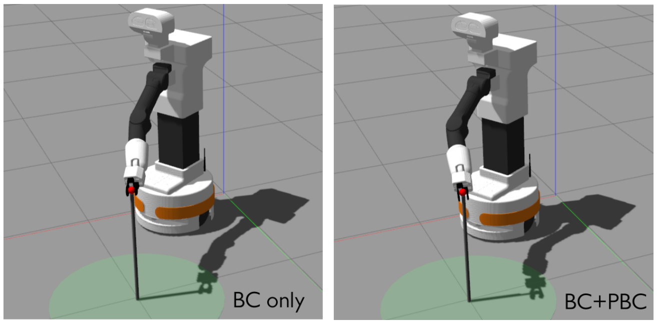

Fig. 5: Simulation 1: final robot configuration.

In the first simulation, the proposed method (BC+PBC) is compared with a version of it that does not consider the PBC in the OBC (BC only). Figure 4 shows the ZMP trajectories of the robot in the two cases. At the outset of the motion, the profiles are identical until the EE enters the transition region. Subsequently, the parameters of the dynamic model change due to the consideration of the external payload according to (22). At this point, the PBC starts affecting the solution of the QP, modifying the decision variables. After the pick up, the ZMP becomes the predicted VZMP, jumping instantaneously to its new position. The plots clearly demonstrate that in the simulation run without the PBC, the ZMP jumps outside the SP, whereas in the simulation with the PBC, balance is preserved since the new ZMP position is at the edge of the SP and the horizontal moments are compensated by the structure. A careful observation of the video reveals the dynamic effects of the PBC. When the PBC is taken into account, the optimization procedure finds a local minimum, affecting the velocity rather than altering the configuration of the robot. This results in a smoother deceleration with a slight overshoot instead of an abrupt stop. Fig. 5 shows the final configurations assumed by the robot at the end of the reach phase in both cases.

In the second simulation, both the BC and the PBC are considered. Figure 6 illustrates the evolution of the ZMP and the VZMP, along with the profiles of their respective CBFs, while Fig. 7 provides snapshots of the motion. The plots clearly shows that both the ZMP and VZMP remain within the SP throughout the entire motion. In this scenario, the ZMP remains within the SP by robot reconfiguration rather than by break modulation therefore, no overshoot can be observed in the video at the end of the motion.

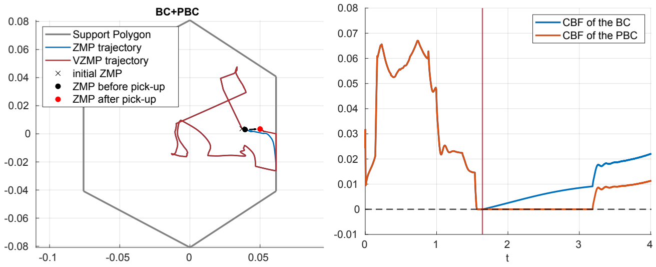

Fig. 6: Simulation 2. Left: ZMP profile obtained with the proposed method (in m); both the ZMP and the VZMP remain within the SP. Right: the CBF profiles show the distance from the closest edge for both ZMP and VZMP. The vertical red line in the CBF plot marks the instant when the EE enters the transition region.

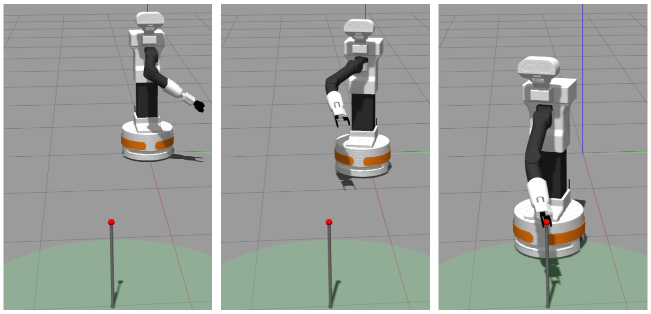

Fig. 7: Simulation 2: snapshots of the robot motion.

VI. Conclusions

This paper presents a novel approach to maintain balance using MM that performs pick-up tasks, where dynamic changes in the system parameters pose significant stability challenges. By leveraging the ZMP and CBF frameworks, an OBC is deployed to adjust the trajectory of the robot to ensure balance both during the reach-to-grasp motion and at the moment when the robot starts lifting the object.

The method incorporates the PBC into the OBC, which involves the prediction of the ZMP position after grasping the object. This allows to anticipate and counteract the destabilizing effects of the inertial and gravitational forces that the payload exerts on the system, guiding the robot with a smooth transition of the considered payload mass. The OBC can be formulated as a QP providing a method suitable for real-time control. The simulation results using the TIAGo robot demonstrate the effectiveness of the approach in maintaining stability. However, the dynamic effects caused by incorporating of the PBC can be observed at the velocity level, resulting in smoother wheel deceleration. This corresponds to a local solution of the optimization problem, whereas a global solution would involve changing the configuration of the robot.

The findings underscore the critical role of dynamic balance control in enhancing the operational safety and flexibility of MMs, particularly in environments requiring precise and stable object handling. Future work could explore extending this approach to more complex tasks and real-world scenarios, further improving the robustness and autonomy of MMs.

References

[1] E. Brzozowska, O. Lima, and R. Ventura, “A generic optimization based cartesian controller for robotic mobile manipulation,” in *2019 Int. Conf. on Robotics and Automation*, 2019, pp. 2054–2060.

[2] S. Stavridis, P. Falco, and Z. Doulgeri, “Pick-and-place in dynamic environments with a mobile dual-arm robot equipped with distributed distance sensors,” in *2020 IEEE-RAS 20th Int. Conf. on Humanoid Robots*, 2021, pp. 76–82.

[3] M. Sorour, A. Cherubini, and P. Fraisse, “Motion control for steerable wheeled mobile manipulation,” in *2019 European Conf. on Mobile Robots*, 2019, pp. 1–7.

[4] M. Vukobratovic and B. Borovac, “Zero-moment point—thirty five years of its life,” *Int. J. of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[5] A. Takanishi, M. Tochizawa, H. Karaki, and I. Kato, “Dynamic biped walking stabilized with optimal trunk and waist motion,” in *IEEE Int. Conf. on Intelligent Robots and Systems*, vol. 1989, 1989, pp. 187–192.

[6] S. Sugano, Q. Huang, and I. Kato, “Stability criteria in controlling mobile robotic systems,” in *Proceedings of 1993 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2, 1993, pp. 832–838.

[7] Q. Huang, K. Tanie, and S. Sugano, “Coordinated motion planning for a mobile manipulator considering stability and manipulation,” *The Int. J. of Robotics Research*, vol. 19, no. 8, pp. 732–742, 2000.

[8] S.-H. Lee, J. Kim, F. C. Park, M. Kim, and J. E. Bobrow, “Newton-type algorithms for dynamics-based robot movement optimization,” *IEEE Trans. on Robotics*, vol. 21, no. 4, pp. 657–667, 2005.

[9] S. Lee, M. Leibold, M. Buss, and F. C. Park, “Rollover prevention of mobile manipulators using invariance control and recursive analytic zmp gradients,” *Advanced Robotics*, vol. 26, no. 11-12, pp. 1317–1341, 2012.

[10] X. Li, Y. Gu, L. Wu, Q. Sun, and T. Song, “Time and energy optimal trajectory planning of wheeled mobile dual-arm robot based on tip-over stability constraint,” *Applied Sciences*, vol. 13, no. 6, p. 3780, 2023.

[11] S. Furuno, M. Yamamoto, and A. Mohri, “Trajectory planning of mobile manipulator with stability considerations,” in *2003 IEEE Int. Conf. on Robotics and Automation*, vol. 3, 2003, pp. 3403–3408.

[12] K. Alipour, P. Daemi, A. Hassanpour, and B. Tarvirdizadeh, “On the capability of wheeled mobile robots for heavy object manipulation considering dynamic stability constraints,” *Multibody System Dynamics*, vol. 41, pp. 101–123, 2017.

[13] S. G. Tarantos and G. Oriolo, “Real-time motion generation for mobile manipulators via NMPC with balance constraints,” in *2022 30th Mediterranean Conf. on Control and Automation*, 2022, pp. 853–860.

[14] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Trans. on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[15] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *2019 18th European Control Conf.*, 2019, pp. 3420–3431.

[16] A. Agrawal and K. Sreenath, “Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation,” in *Robotics: Science and Systems*, vol. 13, 2017, pp. 1–10.

[17] J. Zeng, B. Zhang, and K. Sreenath, “Safety-critical model predictive control with discrete-time control barrier function,” in *2021 American Control Conf.*, 2021, pp. 3882–3889.

[18] B. Xu and K. Sreenath, “Safe teleoperation of dynamic uavs through control barrier functions,” in *2018 IEEE International Conference on Robotics and Automation*, 2018, pp. 7848–7855.

[19] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer, 2008.

[20] R. Campa and H. De La Torre, “Pose control of robot manipulators using different orientation representations: A comparative review,” in *2009 American Control Conf.*, 2009, pp. 2855–2860.

[21] S. J. Wright, *Numerical optimization*. Springer, 2006.

[22] F. C. Park, J. E. Bobrow, and S. R. Ploen, “A lie group formulation of robot dynamics,” *The Int. J. of Robotics Research*, vol. 14, no. 6, pp. 609–618, 1995.

[23] S. R. Ploen and F. C. Park, “Coordinate-invariant algorithms for robot dynamics,” *IEEE Trans. on Robotics and Automation*, vol. 15, no. 6, pp. 1130–1135, 1999.