Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 59-18 (2025) 409–414

Extending humanoids DCM-based walking to push heavy objects while walking

Riccardo K. Simi * **Grazia Zambella** ** **Tobias Egle** **
Antonio Bicchi *** **Christian Ott** **,****

* *Centro di Ricerca "E. Piaggio", Università di Pisa, 56122 Pisa, and Istituto Italiano di Tecnologia, Genova 16163, Italy (e-mail: riccardo.simi@iit.it)*

** *Institut für Automatisierungs, Technische Universität Wien, 1040 Wien, Austria (e-mails: grazia.zambella@tuwien.ac.at, tobias.egle@tuwien.ac.at, christian.ott@tuwien.ac.at)*

*** *Centro di Ricerca "E. Piaggio" and the Dipartimento di Ingegneria dell'Informazione, Università di Pisa, 56122 Pisa, and Istituto Italiano di Tecnologia, Genova 16163, Italy (e-mail: antonio.bicchi@unipi.it)*

**** *Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Weßling, Germany*

Abstract:
Pushing large and heavy objects while walking is an important skill for humanoid robots to effectively assist humans with daily tasks. However, friction, high interaction forces, and the robot's intrinsic limitations make performing this task challenging. This paper presents a DCM-based walking planning algorithm designed to account for the external forces the robot must exert while pushing a heavy object along a desired trajectory. The proposed algorithm is then combined with an inverse dynamics whole-body control to allow the humanoid robot to perform the task. Results obtained in simulation on the humanoid BRUCE prove the effectiveness of the proposed framework.

Keywords: Humanoids, Bipedal locomotion, Whole-body motion planning and control, Pushing, Divergent component of motion (DCM), Loco-manipulation

1. INTRODUCTION

Humanoid robots allow for more natural movement and adaptability due to their human-like design. They have the potential to replicate human skills in both locomotion and manipulation tasks. They can help humans do repetitive and exhausting work, not only in industrial applications, but also in domestic settings, healthcare, and rescue missions. One of the most common tasks expected to be left to humanoids is moving large and heavy objects. While humans effortlessly master this kind of task in everyday situations, easily adapting their skills to handle different objects, replicating this ability in robots is still an open challenge due to the high interaction forces between the robot and the environment involved. Balancing these forces requires advanced and demanding control algorithms to achieve the desired motion. Indeed, the dynamics of pushing are highly non-linear and sensitive to external factors such as friction. It plays a crucial role and is often difficult to predict or measure accurately. In addition, the robot's limitations in sensing and physical constraints should be considered, such as inaccurate feedback from sensors or restricted range of motion, strength, and actuator precision.

Many researchers have focused on developing frameworks for walking and multi-contact pushing, i.e., Xie et al. (2020). In literature, loco-manipulation is seen as a special case of multi-contact motion. Previous research has focused on controlling the pushing locomotion by using a built-in walking pattern generator to produce the desired Center of Mass (CoM) trajectory that follows the specified Zero Moment Point (ZMP) trajectory and using a stabilizer to follow the target ZMP with a force-feedback method which uses data from state sensor measurements or estimators, as in Takubo et al. (2005). Murooka et al. (2015) proposed the use of the ZMP to generate different key postures for manipulation. Instead, Li and Nguyen (2023) suggested combining kinodynamics-based pose optimization and loco-manipulation Model Predictive Control (MPC) to track the optimal pose for pushing without limiting only to hands. Nozawa et al. (2012) introduced a dual-arm force controller to account for friction and compensate for the ZMP by controlling reaction forces. Other approaches used the ZMP-control strategy for pushing heavy loads on a cart (see Vaz and Oh (2020)). However, it is a common practice to use a cart-table or a Linear Inverted Pendulum (LIP) model to approximate the robot's dynamics.

Recent advancements have shifted toward Divergent Component of Motion (DCM)-based methods as suggested by Takenaka et al. (2009), and Englsberger et al. (2013). These DCM-based approaches allow to use the exact

10.1016/j.ifacol.2025.10.255

robot's CoM dynamics and not a simplified model, leading to more efficient control strategies in dynamic environments, as shown by Mesesan et al. (2019), and Vedadi et al. (2021). Morisawa et al. (2019) proposed to modify the desired ZMP with the position of the DCM error with respect to the CoM motion to locomote while leaning to a non-flat surface. A more recent work made by Murooka et al. (2021) presented a DCM control strategy that considers external forces resulting from humanoid-object contact interactions and a DCM feedback control to compensate for the error between desired and actual manipulation forces. Additionally, Kobayashi et al. (2022) implemented a force-reactive walking controller where the interaction forces are included in the definition of the Virtual Repellent Point (VRP) and, consequently, of the DCM. These interaction forces were estimated using robotic skin covering the whole humanoid body. However, this leads to computational complexity of the algorithm and data overload. Moreover, with the rise of machine learning, Deep Reinforcement Learning has been explored for humanoid loco-manipulation tasks, enabling robots to learn optimal policies for complex tasks through trial and error, i.e., Saeedvand et al. (2021). Besides, the lack of data and long training times make using machine learning approaches challenging. However, all the presented methods rely on knowing *a priori* the weight of the object to push and, thus, the minimum force to apply during the pushing task or use sensors to estimate it.

In this context, this paper aims to develop a new framework that combines a dynamic walking algorithm based on VRP and DCM concepts with inverse dynamics Whole-body Control (WBC) to enable the robot to push heavy objects while it is walking. No force sensors were used to estimate the force the robot should apply in the pushing task. For the validation, the proposed framework is tested on the kid-size humanoid robot BRUCE, Liu et al. (2022).

2. BACKGROUND

Here, we provide some key concepts used in bipedal walking and an overview of the planning algorithm employed to generate reference trajectories that the robot can follow.

2.1 Fundamentals of Bipedal Walking

The DCM separates the second order CoM dynamics into two linear first order dynamics, one naturally stable and the other unstable. The DCM $\boldsymbol{\xi} \in \mathbb{R}^3$ is defined as

$$\boldsymbol{\xi} = \boldsymbol{x} + b\dot{\boldsymbol{x}}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^3$ and $\dot{\boldsymbol{x}} \in \mathbb{R}^3$ are the CoM position and velocity, respectively, and $b > 0$ is the time-constant of the DCM dynamics. Note that (1) is a general linear transformation, and all the equations derived from the DCM theory hold for general free-floating base models and are not restricted to the cart-table, LIP, or other simplified models. Defining $\boldsymbol{F}_{\mathrm{g}} \in \mathbb{R}^3$ as the gravity force and $\boldsymbol{F}_{\mathrm{leg}} \in \mathbb{R}^3$ as the leg force, the total force $\boldsymbol{F}_{\mathrm{TOT}} \in \mathbb{R}^3$ acting on the CoM results as $\boldsymbol{F}_{\mathrm{TOT}} = \boldsymbol{F}_{\mathrm{leg}} + \boldsymbol{F}_{\mathrm{g}}$ and the DCM dynamics can be computed by differentiating (1) and substituting $\dot{\boldsymbol{x}}$ obtained by reordering (1), i.e.,

$$\dot{\boldsymbol{\xi}} = \dot{\boldsymbol{x}} + b\ddot{\boldsymbol{x}} = -\frac{1}{b}\boldsymbol{x} + \frac{1}{b}\boldsymbol{\xi} + \frac{b}{m}\boldsymbol{F}_{\mathrm{leg}} + \frac{b}{m}\boldsymbol{F}_{\mathrm{g}}, \tag{2}$$

with m the robot's total mass. Hence, equation (2) highlights that the external forces directly influence the DCM dynamics. To decouple the DCM dynamics from the CoM dynamics, the enhanced Centroidal Moment Pivot (eCMP) $\boldsymbol{r}_{\mathrm{eCMP}}$ is introduced as a three-dimensional point that encodes all the external forces acting on the CoM except for gravity. Additionally, the VRP ν is defined as a tridimensional point obtained by translating vertically $\boldsymbol{r}_{\mathrm{eCMP}}$ of Δz_{vrp} (the average CoM height relative to the ground surface), i.e.,

$$\boldsymbol{\nu} = \boldsymbol{r}_{\mathrm{eCMP}} + \begin{bmatrix}0 & 0 & b^2 g\end{bmatrix}^{\mathrm{T}} = \boldsymbol{r}_{\mathrm{eCMP}} + \begin{bmatrix}0 & 0 & \Delta z_{vrp}\end{bmatrix}^{\mathrm{T}}. \tag{3}$$

where $b = \sqrt{\frac{\Delta z_{vrp}}{g}} > 0$. Consequently, the VRP encodes the total force $\boldsymbol{F}_{\mathrm{TOT}}$ acting on the CoM as

$$\boldsymbol{F}_{\mathrm{TOT}} = \frac{m}{b^2}(\boldsymbol{x} - \boldsymbol{\nu}). \tag{4}$$

By substituting (4) in (2), the DCM dynamics becomes

$$\dot{\boldsymbol{\xi}} = \frac{1}{b}(\boldsymbol{\xi} - \boldsymbol{\nu}), \tag{5}$$

showing that it is unstable and the DCM diverges away from the VRP.

2.2 DCM Planning Algorithm for Walking

Since the CoM automatically follows the DCM, the focus of the DCM planning algorithm proposed by Englsberger et al. (2013) for bipedal walking is to generate a desired DCM trajectory from a set of chosen feasible steps. The planned DCM trajectory can then be transformed into a reference CoM trajectory that the robot follows while maintaining stable and trackable motions. To reduce complexity, the following assumptions are made:

- robot's feet are point feet corresponding to foot centers
- zero changes in angular momentum
- instantaneous transitions between left and right feet, i.e., no double support phase (both feet in contact with the ground)
- no impact during support transitions.

With these hypotheses, $\boldsymbol{r}_{\mathrm{eCMP}}$ can be designed to coincide with the foot point $\boldsymbol{r}_f$, and the trajectory planning is reduced to choosing N reference foot-positions. Consequently, (3) is rewritten as $\boldsymbol{\nu} = \boldsymbol{r}_f + [0\ 0\ \Delta z_{vrp}]^{\mathrm{T}}$ and a backward recursive iteration is used to compute the desired DCM locations at the end of each step $\boldsymbol{\xi}_{eos,d}$ assuming that in the final step the robot will stop, i.e., $\boldsymbol{\xi}_{d,eos,N-1} = \boldsymbol{\nu}_{d,N}$ and from (1) and (4) both coincide with the final CoM position. Hence, the desired DCM trajectory in time $\boldsymbol{\xi}_d(t)$ is computed by integrating (5) w.r.t. time t, with $t \in [0, t_{step}]$ so that time is reset at the beginning of each step. Summarizing, the final DCM algorithm to generate reference trajectories includes the following steps:

(1) Define a set of N reference foot-positions $\boldsymbol{r}_{f,i} \in \{1, .., N\}$;

(2) Set a desired VRP height Δz_{vrp} and compute the desired VRP as $\boldsymbol{\nu}_{d,i} = \boldsymbol{r}_{f,i} + [0\ 0\ \Delta z_{vrp}]^{\mathrm{T}}$;

(3) Assuming $\boldsymbol{\nu}_d$ constant, calculate the desired DCM at the end of each step as

$$\boldsymbol{\xi}_{eos,d,i-1} = \boldsymbol{\xi}_{ini,d,i} = \boldsymbol{\nu}_{d,i} + e^{-\frac{t_{step}}{b}}(\boldsymbol{\xi}_{eos,d,i} - \boldsymbol{\nu}_{d,i})$$

starting from the final step until the first one;

(4) Known the DCM end of step, compute the whole DCM trajectory as

$$\boldsymbol{\xi}_d(t) = \boldsymbol{\nu}_{d,i} + e^{\frac{t-t_{step}}{b}}(\boldsymbol{\xi}_{eos,d,i} - \boldsymbol{\nu}_{d,i})$$

where $t < t_{step}$.

From the planned DCM trajectory, the CoM reference trajectory $\boldsymbol{x}_d(t)$ is computed by integrating its velocity $\dot{\boldsymbol{x}}$ obtained by reordering (1). Both DCM and CoM trajectories thus obtained have been proven to be convex by Mesesan et al. (2018).

3. PROPOSED METHOD

3.1 Problem Definition

During the pushing task, an interaction force $\boldsymbol{F}_{\text{int}} \in \mathbb{R}^3$ is generated between the robot that is pushing and the object being pushed. If $\boldsymbol{F}_{\text{int}}$ is known, it can be explicitly included in the robot's CoM dynamics and, hence, in its DCM dynamics as follows

$$\dot{\boldsymbol{\xi}} = \dot{\boldsymbol{x}} + b\ddot{\boldsymbol{x}} = \frac{1}{b}(\boldsymbol{\xi} - \boldsymbol{x}) + b\left(\boldsymbol{g} + \frac{\boldsymbol{F}_{\text{leg}}}{m} + \frac{\boldsymbol{F}_{\text{int}}}{m}\right) = \\ = \frac{1}{b}\left(\boldsymbol{\xi} \underbrace{-\boldsymbol{x} + b^2\boldsymbol{g} + \frac{b^2}{m}\boldsymbol{F}_{\text{leg}}}_{-\tilde{\boldsymbol{\nu}}} + \frac{b^2}{m}\boldsymbol{F}_{\text{int}}\right), \tag{6}$$

where $\tilde{\boldsymbol{\nu}}$ is the VRP computed if no interaction forces would act on the CoM (named from now on the unshifted VRP). Explicitly including $\boldsymbol{F}_{\text{int}}$ in (3), the VRP can be adjusted accordingly

$$\bar{\boldsymbol{\nu}} = \boldsymbol{r}_{\text{eCMP}} + [0\ 0\ \Delta z_{vrp}]^{\text{T}} - \frac{b^2}{m}\boldsymbol{F}_{\text{int}} = \tilde{\boldsymbol{\nu}} - \frac{b^2}{m}\boldsymbol{F}_{\text{int}}, \tag{7}$$

where the negative sign in front of $\boldsymbol{F}_{\text{int}}$ implies that the VRP is shifted towards the interaction force while the DCM is pushed in the same direction as $\boldsymbol{F}_{\text{int}}$. Consequently, the CoM will move on the back of the robot, as shown in Figure 1a.

3.2 DCM Planning Algorithm with Interaction Forces

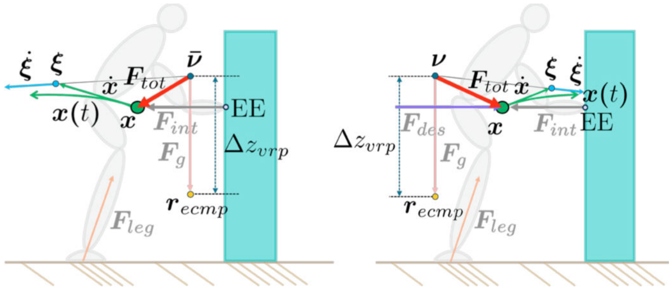
To obtain the robot's desired behavior, the DCM must be placed in front of the actual CoM position, with the DCM dynamics pointing in the desired moving direction. To this end, a desired force $\boldsymbol{F}_{\text{des}} \in \mathbb{R}^3$ with $||\boldsymbol{F}_{\text{des}}|| \geq ||\boldsymbol{F}_{\text{int}}||$ is introduced to compensate the effects of the interaction force so that the VRP shifts again on the back of the robot's torso. Thus, the desired DCM can be redefined as

$$\dot{\boldsymbol{\xi}}_{\boldsymbol{d}} = \frac{1}{b}(\boldsymbol{\xi} - \boldsymbol{\nu}_{\boldsymbol{d}} + \frac{b^2}{m}\boldsymbol{F}_{\text{int}}), \tag{8}$$

where $\boldsymbol{\nu}_{\boldsymbol{d}} = \tilde{\boldsymbol{\nu}} + \boldsymbol{\nu}_{\text{shift}}$ is the desired VRP computed as the sum of the unshifted VRP, i.e. $\tilde{\boldsymbol{\nu}} = \boldsymbol{x} - b^2\boldsymbol{g} - (b^2/m)\boldsymbol{F}_{\text{leg}}$, and the VRP-shift

$$\boldsymbol{\nu}_{\text{shift}} = -\frac{b^2}{m}\boldsymbol{F}_{\text{des}}, \tag{9}$$

which encodes the effect of the desired force. Note that to exactly compensate the interaction force, $\boldsymbol{F}_{\text{des}}$ should be chosen to be equal but opposite to $\boldsymbol{F}_{\text{int}}$. However, extra forces can be added to consider the desired motion of the object. Figure 1b shows the robot's behavior once the desired force is introduced.



(a) Without $\boldsymbol{F}_{\text{des}}$ (b) With $||\boldsymbol{F}_{\text{des}}|| \geq ||\boldsymbol{F}_{\text{int}}||$

Fig. 1. Effects on VRP, DCM, and CoM of the interaction force (a) and relative compensation through the introduction of the desired force (b).

The desired force $\boldsymbol{F}_{\text{des}}$ can be considered directly inside the DCM planning algorithm by $\boldsymbol{\nu}_{\text{shift}}$. The new DCM algorithm follows:

(1) Define a set of N reference foot-positions $\boldsymbol{r}_{f,i} \in \{1, \ldots, N\}$;

(2) Set a desired VRP height Δz_{vrp} and compute the desired VRP as

$$\boldsymbol{\nu}_{d,i} = \boldsymbol{r}_{f,i} + [0\ 0\ \Delta z_{vrp}]^{\text{T}} - \frac{b^2}{m}\boldsymbol{F}_{\text{des}},$$

and $||\boldsymbol{F}_{\text{des}}|| \geq ||\boldsymbol{F}_{\text{int}}||$;

(3) Assuming the VRP constant, from (8), calculate the desired DCM at the end of each step as

$$\boldsymbol{\xi}_{eos,d,i-1} = \boldsymbol{\xi}_{ini,d,i} = \\ = \boldsymbol{\nu}_{d,i} - \frac{b^2}{m}\boldsymbol{F}_{\text{int}} + e^{-\frac{t_{step}}{b}}(\boldsymbol{\xi}_{eos,d,i} - \boldsymbol{\nu}_{d,i} + \frac{b^2}{m}\boldsymbol{F}_{\text{int}})$$

starting from the final step until the first one;

(4) Known the DCM end of step, compute the whole DCM trajectory as

$$\boldsymbol{\xi}_d(t) = \boldsymbol{\nu}_{d,i} - \frac{b^2}{m}\boldsymbol{F}_{\text{int}} + e^{\frac{t-t_{step}}{b}}(\boldsymbol{\xi}_{eos,d,i} - \boldsymbol{\nu}_{d,i} \\ + \frac{b^2}{m}\boldsymbol{F}_{\text{int}}),$$

where $t < t_{step}$.

3.3 Empirical Estimation of the Interaction Force

The proposed method requires the knowledge of the minimum force the robot should apply to the object in order to compute the VRP-shift. Since the object's weight is not always known in advance, the following approach estimates the minimum force needed to push the object without relying on a sensor, but by taking inspiration from human behavior. When humans push a heavy object, they initially place both feet some distance away from the object and try to push. If the object does not move after trying to push it, humans will take steps backward to gain mechanical advantage and be able to apply more force, repeating this process until the object starts to move. To replicate this human behavior, the algorithm is shown in the flow diagram presented in Figure 2.

According to the latter, once the robot's hands come in contact with the object, the current VRP position $\boldsymbol{\nu}_{d,ini}$ is saved and the algorithm plans to do two steps on the back. Then, the robot tries to push the object by moving

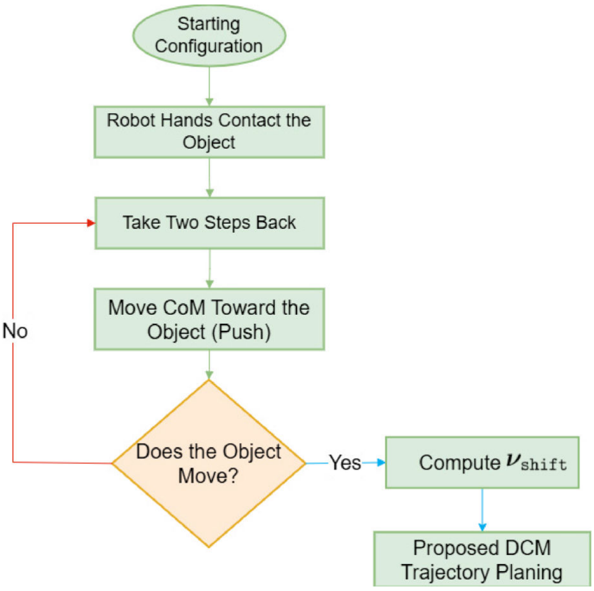

Starting Configuration
Robot Hands Contact the Object
Take Two Steps Back
Move CoM Toward the Object (Push)
No
Does the Object Move?
Yes
Compute $\nu_{\texttt{shift}}$
Proposed DCM Trajectory Planing

Fig. 2. Flow diagram for $\boldsymbol{\nu}_{\text{shift}}$ empirical computation, extended with online DCM trajectory planning.

its DCM and, consequently, its CoM, towards the object. If the object does not move, it means that the robot has not yet reached the force needed to push so its CoM position is reset to the previous one. Two more steps are taken backward, and then the robot tries to push again. This human-like procedure is repeated iteratively until the heavy object starts to move. Since the ground projection of the VRP is always on the feet, the VRP-shift $\boldsymbol{\nu}_{\text{shift}}$ is then computed considering the offset between the saved starting VRP $\boldsymbol{\nu}_{d,ini}$ and the last VRP obtained as soon as the object started to move, $\boldsymbol{\nu}_{d,obj\_move}$.

4. VALIDATION

To validate the proposed approach, the framework is applied to BRUCE, a kid-size humanoid robot with an approximate height of $0.7m$ and a weight of $4.8kg$, developed by Westwood Robotics. Tests were conducted in simulation in the physics engine MuJoCo created by Todorov et al. (2012).

4.1 BRUCE Implementation

First, we implemented the inverse dynamics WBC algorithm to compute the torques commanded to the actuated joints to accomplish the tasks presented below. Inspired by Englsberger (2016), instead of requiring the WBC to track the desired acceleration of the CoM as typically done, the DCM has been used to compute the reference quantity to track. In particular, the DCM Task commands $\boldsymbol{F}_{TOT,des} = \frac{m}{b^2}(\boldsymbol{x} - \boldsymbol{\nu}_{ref} - (1 + bk_{\xi})(\boldsymbol{\xi} - \boldsymbol{\xi}_{ref}))$ as the reference desired total force acting on the CoM and aims at reducing the error between the contact forces expressed in the world frame and $\boldsymbol{F}_{TOT,des}$. The other desired tasks we considered are:

(1) Angular Momentum Task, to keep the angular momentum equal to zero;
(2) Arm Task, which is formulated in the joint-space if the contact has not yet occurred to track reference joint positions and velocities that bring the robot's hands into contact with the object to push, and in the Cartesian-space to track the desired hand positions and velocities ($\boldsymbol{p}_{h,des}$ and $\boldsymbol{v}_{h,des}$, respectively) if arms are no longer in free motion;
(3) Base-Link Rotation Task, to maintain robot's orientation constant;
(4) Contact Task, that requires zero velocity to be applied to the feet contacting point;
(5) Swing Leg Position, which tracks the planned swing foot trajectory;
(6) Swing Leg Orientation, which keeps the orientation of the foot constant during the execution of the step.

Note that along the x-axis, the desired motion of the object being pushed is used as the reference quantity to be tracked by the controller in the arm task when hands are in contact. Thus, assuming that the hands are always in contact with the object during the push, $\boldsymbol{v}_{h,des}$ is equal to the constant velocity $\boldsymbol{v}_{obj,des}$ we would like the object to have and $\boldsymbol{p}_{h,des}$ is obtained by translating the desired position of the middle of the object in the point in which the hand is in contact.

Since the robot interacts with its surrounding environment, feet are modeled as two-point contacts, corresponding to the robot's toe and heel, whereas hands are defined as single-point contacts. For each contact point, unilateral contact ($(\boldsymbol{f}_{c_i} \cdot \hat{\boldsymbol{z}}_i) > 0$) and friction constraints were considered ($|\boldsymbol{f}_{c_i} \cdot \hat{\boldsymbol{x}}_i| \leq \mu\,(\boldsymbol{f}_{c_i} \cdot \hat{\boldsymbol{z}}_i)$ and $|\boldsymbol{f}_{c_i} \cdot \hat{\boldsymbol{y}}_i| \leq \mu\,(\boldsymbol{f}_{c_i} \cdot \hat{\boldsymbol{z}}_i)$).

Additionally, to ensure that the robot can follow a feasible trajectory even if the whole-body control does not perfectly track the one planned before, the planner presented in Section 3.2 is executed online at each iteration, i.e., the N foot placement references are updated at each iteration starting from the current foot position and all the steps of the algorithm are executed again.

4.2 Results

In this simulation, we tested the framework considering all the phases of the pushing while walking task from the moment when BRUCE makes contact with the object until it stops pushing and balances in the final configuration. As shown in Figure 3a, in the beginning, BRUCE stands in front of the object. Then, a desired arm's joint trajectory is commanded to the whole-body controller in order to bring the hand in contact with the object. Once the contact is reached (Figure 3b), the robot tries to push the object by moving its CoM toward it (Figure 3c). Since the robot can not move the object, it takes two steps backward, each of length $l_{step} = 0.038m$, and tries to push again. Thus, the object moves and the VRP-shift is empirically computed as described in Section 3.3. Consequently, by reordering (9), the desired force is equal to $\boldsymbol{F}_{\text{des}} = 10N$ considering that both hands push the object and the DCM algorithm starts to compute the references for the WBC considering $N = 4$ footsteps at each iteration. With this method, the robot is successfully able to push the heavy object for a total of 12 steps (Figures 3d and 3e show the first and last steps taken, respectively). Afterward, considering both friction and the fact that the WBC controller is performing multiple tasks simultaneously, the robot does not execute all the required tasks perfectly. Therefore, to

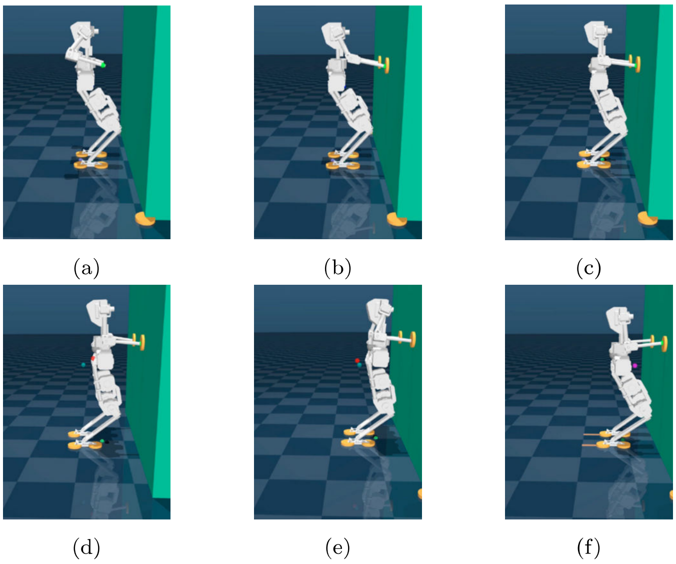


Fig. 3. Snapshot of the simulation in which BRUCE robot makes contact with the object and tries to push (a)-(c), pushes (first step, d)-(last step, e), and stops in the current safe configuration (f). The object is $0.7m \times 0.5m \times 0.5m$ with $m_{obj} = 1.6kg$. Friction coefficient is set to $\mu = 0.5$. Robot's mass is $m = 4.8kg$ with $l_{step} = 0.04m$ and $z_{apex} = 0.03m$.

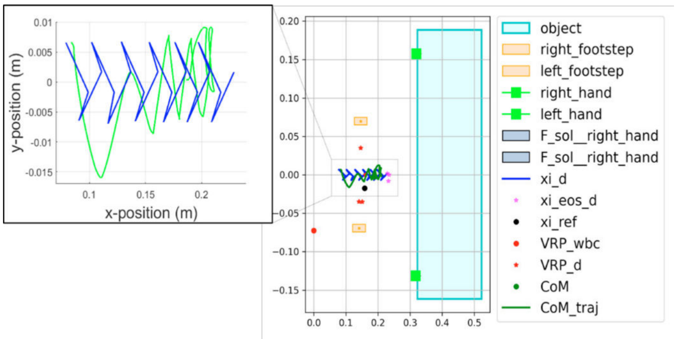


Fig. 4. Top view (XY-plane) of the DCM (blue line) and CoM (green line) trajectories generated. Here, $l_{step} = 0.01m$, $w_{step} = 0.07m$, $z_{apex} = 0.01m$, $\Delta z_{vrp} = 0.386m$, and $t_{step} = 0.05s$.

prioritize its balancing, the robot is unable to move the DCM forward and make it follow its reference exactly, causing the DCM to shift $0.03m$ backward relative to the CoM position. For this reason, the planning algorithm is forced to stop, and the WBC will stabilize the robot in the current safe configuration (Figure 3f). To improve this limitation, the DCM could be brought in front of the robot CoM again with a footstep adaptation or a reactive replanning approach. We leave this to future work.

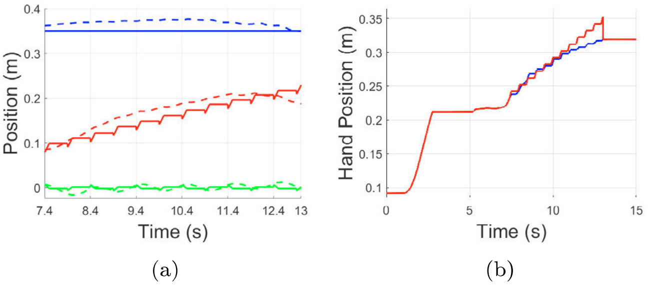


Fig. 5. (a) DCM (solid lines) and CoM (dashed lines) trajectories over time generated by the proposed method where the red, green, and blue lines indicate the x, y, and z components, respectively; (b) Actual (blue line) and desired (from the object, red line) x-trajectories of the robot's right hand over time.

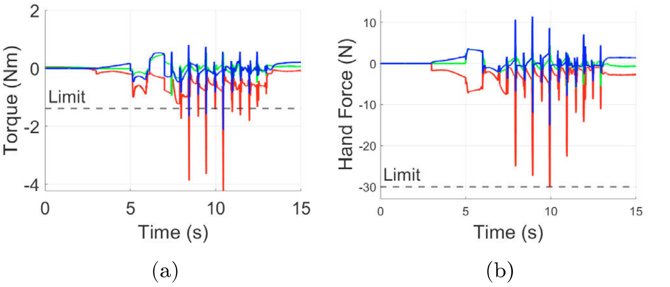


Fig. 6. (a) Torque control inputs of the robot's three actuators of the right arm over time. The black dashed line indicates the actuator's torque limit of $1.4Nm$; (b) Total force exerted by the robot's right hand on the heavy object over time in the x (red line), y (green line), and z (blue line) directions.

Figure 4 shows the results of the implementation in the XY plane, and Figure 5a displays the time trajectories of DCM and CoM, where a distance of $0.128m$ is covered. Note that at each step taken, the DCM reference is paused for $0.2s$ to simulate a double-support step phase. Figure 5b shows the x trajectories of the actual and desired robot hand positions over time. The latter is calculated considering $\boldsymbol{v}_{obj,des} = 0.2m/s$ and is re-planned at every footstep, considering the current hand position as the starting point for the reference trajectory for the robot's hands. The plot shows that the desired hand trajectory is well tracked in terms of root mean square error with $\varepsilon_{hand,x} = 0.0022m$ in the first 7 steps. After that, the computed control torques exceed the arm actuators' limit of $1.4Nm$ and the BRUCE's DCM begins not to follow perfectly its reference since the robot can exert less thrust. Figures 6a and 6b display, respectively, the evolution of the right arm control input and the total force exerted by the robot's right hand on the heavy object throughout the task. The hand contact is achieved at $t = 3s$, and the object starts to move from $t \geq 7.4s$. Note that the maximum force that the hands can exert is set to $30N$ to prevent any damage to the arm actuators. The x-axis has a negative force value due to the way the hand reference system has been chosen.

5. CONCLUSION

This paper presents a new framework that allows humanoids to push a heavy object while walking by combining the concepts of DCM and VRP with Inverse Dynamics WBC. The key innovation of this method is to include the required pushing force directly into the trajectory planning algorithm, allowing for smoother movements and better adaptation to changing conditions during motion. Additionally, the proposed framework is capable of both empirically estimating the desired force for pushing from the VRP and re-planning the desired generated DCM trajectory online considering the current feet positions. The proposed method is versatile and adaptable to different robots since it is DCM-based and uses the exact CoM dynamics. The algorithm was validated in simulation in the MuJoCo environment on the humanoid robot BRUCE. Results show the proposed method's efficiency in pushing the object, although the robot did not know the mass or friction parameter of it.

Future works aim at improving algorithm robustness with reinforcement learning by training in simulation various object masses, friction coefficients, and external disturbances, or with step and timing adaptation during online DCM trajectory generation as proposed by Egle et al. (2023), enabling empirical re-computation of the VRP-shift when the robot can no longer push in the current configuration, and validating the proposed algorithm by conducting experiments with the real-life BRUCE robot.

ACKNOWLEDGEMENTS

Funded by the European Union - NextGenerationEU and by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5, project "RAISE - Robotics and AI for Socio-economic Empowerment" (ECS00000035). Riccardo K. Simi is part of RAISE Innovation Ecosystem.

REFERENCES

Egle, T., Englsberger, J., and Ott, C. (2023). Step and timing adaptation during online dcm trajectory generation for robust humanoid walking with double support phases. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, 1–8. IEEE.

Englsberger, J. (2016). *Combining reduced dynamics models and whole-body control for agile humanoid locomotion*. Ph.D. thesis, Technische Universität München.

Englsberger, J., Ott, C., and Albu-Schäffer, A. (2013). Three-dimensional bipedal walking control using divergent component of motion. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2600–2607. IEEE.

Kobayashi, T., Dean-Leon, E., Guadarrama-Olvera, J.R., Bergner, F., and Cheng, G. (2022). Whole-body multicontact haptic human–humanoid interaction based on leader–follower switching: A robot dance of the "box step". *Advanced Intelligent Systems*, 4(2), 2100038.

Li, J. and Nguyen, Q. (2023). Kinodynamics-based pose optimization for humanoid loco-manipulation. *arXiv preprint arXiv:2303.04985*.

Liu, Y., Shen, J., Zhang, J., Zhang, X., Zhu, T., and Hong, D. (2022). Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors. In *2022 International Conference on Robotics and Automation (ICRA)*, 8547–8553. doi: 10.1109/ICRA46639.2022.9811790.

Mesesan, G., Englsberger, J., Garofalo, G., Ott, C., and Albu-Schäffer, A. (2019). Dynamic walking on compliant and uneven terrain using dcm and passivity-based whole-body control. In *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*, 25–32. IEEE.

Mesesan, G., Englsberger, J., Ott, C., and Albu-Schäffer, A. (2018). Convex properties of center-of-mass trajectories for locomotion based on divergent component of motion. *IEEE Robotics and Automation Letters*, 3(4), 3449–3456.

Morisawa, M., Benallegue, M., Cisneros, R., Kumagai, I., Escande, A., Kaneko, K., and Kanehiro, F. (2019). Multi-contact stabilization of a humanoid robot for realizing dynamic contact transitions on non-coplanar surfaces. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2252–2258. IEEE.

Murooka, M., Chappellet, K., Tanguy, A., Benallegue, M., Kumagai, I., Morisawa, M., Kanehiro, F., and Kheddar, A. (2021). Humanoid loco-manipulations pattern generation and stabilization control. *IEEE Robotics and Automation Letters*, 6(3), 5597–5604.

Murooka, M., Nozawa, S., Kakiuchi, Y., Okada, K., and Inaba, M. (2015). Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 5682–5689. IEEE.

Nozawa, S., Kakiuchi, Y., Okada, K., and Inaba, M. (2012). Controlling the planar motion of a heavy object by pushing with a humanoid robot using dual-arm force control. In *2012 IEEE International Conference on Robotics and Automation*, 1428–1435. IEEE.

Saeedvand, S., Mandala, H., and Baltes, J. (2021). Hierarchical deep reinforcement learning to drag heavy objects by adult-sized humanoid robot. *Applied Soft Computing*, 110, 107601.

Takenaka, T., Matsumoto, T., and Yoshiike, T. (2009). Real time motion generation and control for biped robot-1 st report: Walking gait pattern generation. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1084–1091. IEEE.

Takubo, T., Inoue, K., and Arai, T. (2005). Pushing an object considering the hand reflect forces by humanoid robot in dynamic walking. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 1706–1711. IEEE.

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 5026–5033. IEEE.

Vaz, J.C. and Oh, P. (2020). Material handling by humanoid robot while pushing carts using a walking pattern based on capture point. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 9796–9801. IEEE.

Vedadi, A., Sinaei, K., Abdolahnezhad, P., Aboumasoudi, S.S., and Yousefi-Koma, A. (2021). Bipedal locomotion optimization by exploitation of the full dynamics in dcm trajectory planning. In *2021 9th RSI International Conference on Robotics and Mechatronics (ICRoM)*, 365–370. IEEE.

Xie, Y., Lou, B., Xie, A., and Zhang, D. (2020). A review: Robust locomotion for biped humanoid robots. In *Journal of Physics: Conference Series*, volume 1487, 012048. IOP Publishing.