This article has been accepted for publication in 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). This is the author's version which has not been fully edited and content may change prior to final publication.

Whole Body Control Formulation for Humanoid Robots with Closed/Parallel Kinematic Chains: Kangaroo Case Study

Sait Sovukluk, Johannes Englsberger, and Christian Ott

DOI: 10.1109/IROS55552.2023.10341391

Whole Body Control Formulation for Humanoid Robots with Closed/Parallel Kinematic Chains: Kangaroo Case Study

Sait Sovukluk[1], Johannes Englsberger[2], and Christian Ott[1,2]

***Abstract*— This study extends the whole-body control (WBC) formulation for bipedal humanoid robots that include closed (parallel) kinematic chains in their structure. Along with general formulation, we also stress the implementation of this formulation on Kangaroo, which is a highly dynamic humanoid robot developed by PAL Robotics. This 76-DOF robot includes 24 independent closed-kinematic chains in its structure and constitutes a good case study for our approach. We discuss the WBC formulation for various control structures, including inverse dynamics control (IDC) and Modular Passive Tracking Control (MPTC). As a test scenario, we employ a 3D spring-loaded inverted pendulum (SLIP) jumping trajectory with disturbance rejection as the desired CoM trajectory.**

I. INTRODUCTION

In recent literature, humanoid robot locomotion is often carried out by assigning CoM trajectories and other complementary tasks to the robot. Studies in [1]–[4] employ capture point trajectories for walking and disturbance rejection. For walking, a similar but more comprehensive approach called the divergent component of motion (DCM) is employed in [5]. Spring-loaded inverted pendulum (SLIP) based trajectory generation methods also showed successful behaviors on humanoid robots for walking, running, jumping, and disturbance rejection [6]–[11].

Even though CoM trajectory generation is one of the essential aspects of locomotion, CoM dynamics does not cover all complexity of the robot. One must employ specific tasks for all limbs to keep the robot in motion or at a particular posture. These tasks include tracking the CoM trajectories, stepping/swinging the legs, keeping the torso upright, swinging arms, dissipating excessive angular momentum, interacting with the environment, etc. For this purpose, whole-body control frameworks come into place and combine all tasks with different prioritization methods respecting various equality and inequality constraints. Due to its implementation and interpretation simplicity, one of the most popular task formulations is the inverse dynamics control [12]–[15]. Studies in [16]–[19] employ passivity-based approaches to have robust tracking and well-damped behavior despite modeling errors and disturbances. If the task set inside the WBC formulation is overconstrained, the whole-body controller compromises these tasks depending on their task prioritization structure. The approach in [20] projects lower priority tasks into the null space of high priority tasks to achieve strict task prioritization, i.e., conflicting lower priority tasks cannot affect the performance of higher priority tasks. On the other hand, [19] handles task prioritization softly by assigning higher gains to higher priority tasks in their quadratic formulation. One of the most recent studies on whole-body robot formulation [21] extensively discusses corresponding task selections and their prioritization for humanoids.

So far, all the control methods we discuss require an accurate system dynamics model. Even though many software packages provide system dynamics for open-kinematic chains from their system descriptions, for example, URDF (Unified Robot Description Format), with ease, the closed-kinematic chains are not widely supported. Closed-kinematic chains require additional constraint forces to be defined, and such systems are harder to describe and exhibit more complicated dynamics than open kinematic trees. For a more detailed discussion of these complexities, the reader is referred to [22, Chapter 3.6]. Instead of computing reduced order system dynamics and solving for constraining forces in multiple stages as in [23], we implement a unified model that employs complete order system dynamics and solves for constraint forces along with other parameters in a single stage.

This study discusses a whole-body control formulation for the full dynamics of a humanoid robot with closed/parallel kinematic chains and its implementation on the Kangaroo robot [23], [24]. The robot's structure with 24 independent closed linkages and 76-DOF constitutes a suitable case study for our approach. We introduce the constraints inside the WBC formulation and close the open chain ends by additional forces defined in open kinematic chain dynamics. Such an approach allows us to utilize non-reduced and non-simplified full-order system dynamics with all their complexities. We also implement two different task formulations for the whole-body control formulation and show that the formulation works well both in task acceleration and task force spaces. These task formulations are based on inverse dynamics control and modular passive tracking control (MPTC). Performance differences of these task formulations are tested via a 3D SLIP jumping trajectory stabilized by deadbeat control.

Section II presents some preliminary information about the methods employed in this study. Building on this background, Section III discusses the whole-body controller formulation for the systems with closed kinematic chains. Implementation details for the Kangaroo model are discussed

[1]Automation and Control Institute (ACIN), TU Wien, 1040 Vienna, Austria

[2]Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819358)

in Section IV. Finally, section V presents simulation results for different task formulations and discusses their trajectory tracking performance under disturbances. We end the study with a discussion and conclusion and provide a supplemental video that shows the robot performing various maneuvers.

II. BACKGROUND

A. Floating Base Model

The general tree structured floating base robot dynamics for n configuration variables $\boldsymbol{q} \in \mathcal{Q}$ is expressed as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{B}\boldsymbol{\tau} + \boldsymbol{J}^\top(\boldsymbol{q})\boldsymbol{f}, \quad (1)$$

where $\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{g}$, $\boldsymbol{B}$, $\boldsymbol{\tau}$, $\boldsymbol{J}$, and $\boldsymbol{f}$ represent inertia matrix, Coriolis matrix, gravity vector, input mapping matrix, actuation efforts, a combination of all Jacobian matrices for external and constraint torques and forces, combination of all external and constraint torques and forces, respectively.

B. Inverse Dynamics Control (IDC)

The usage of inverse dynamics in a whole-body control framework is well studied in literature [12]–[15]. For each task space Jacobian $\boldsymbol{J}_i$, the task space velocity can be represented as:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{J}_i\dot{\boldsymbol{q}}. \quad (2)$$

Similarly, the time derivative of (2) yields the task acceleration:

$$\ddot{\boldsymbol{x}}_i = \boldsymbol{J}_i\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}}. \quad (3)$$

The main concept applied in IDC is to cancel the task-related system dynamics and to enforce some desirable task dynamics, e.g. in the form of a stable second-order dynamics:

$$\ddot{\boldsymbol{x}}_{i,d} = \ddot{\boldsymbol{x}}_{i,ref} + \boldsymbol{K}_{D,i}(\dot{\boldsymbol{x}}_{i,ref} - \dot{\boldsymbol{x}}_i) + \boldsymbol{K}_{P,i}(\boldsymbol{x}_{i,ref} - \boldsymbol{x}_i). \quad (4)$$

For $\boldsymbol{e}_i = \boldsymbol{x}_{i,d} - \boldsymbol{x}_i$ the stable closed-loop error dynamics for the particular task turns out to be

$$\ddot{\boldsymbol{e}}_i + \boldsymbol{K}_{D,i}\dot{\boldsymbol{e}}_i + \boldsymbol{K}_P\boldsymbol{e}_i = 0, \quad (5)$$

where $\boldsymbol{K}_{D,i}$ and $\boldsymbol{K}_{P,i}$ are positive definite gain matrices. If the combination of all task mappings $\boldsymbol{J} = [\boldsymbol{J}_1; \boldsymbol{J}_2; \dots]$ is square and invertible, then the control system is well defined, and the solution is unique. If underconstrained, the control system has infinitely many solutions and is thus only partially stabilized. Finally, suppose the control system is overconstrained. In that case, one may solve it via optimization such that depending on weight selection, the controller will compromise between different tasks (soft hierarchy), or one can also map lower-priority tasks to the null space of higher-priority tasks (strict hierarchy).

C. Modular Passive Tracking Control (MPTC)

Modular Passive Tracking Control (MPTC) is a generic passivity-based controller that aims at independently fulfilling several subtask objectives [19]. The control framework combines the advantages of inverse dynamics controllers and passivity-based controllers: ease of implementation and use, task space tracking capabilities, and passivity. For an actual task force definition,

$$\boldsymbol{f}_i = \boldsymbol{J}_i^{M^+}(\boldsymbol{B}\boldsymbol{\tau} + \boldsymbol{J}^\top\boldsymbol{f}), \quad (6)$$

where $\boldsymbol{J}_i^{M^+} = \boldsymbol{\Lambda}_i\boldsymbol{J}_i\boldsymbol{M}^{-1}$ is the dynamically consistent pseudo-inverse of $\boldsymbol{J}_i^\top$, selecting the desired task force to be

$$\begin{aligned}\boldsymbol{f}_{i,d} = {} & \boldsymbol{J}_i^{M^+}\boldsymbol{g} + \boldsymbol{\Lambda}_i(\boldsymbol{J}_i\boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{J}}_i)\dot{\boldsymbol{q}} + \boldsymbol{\Lambda}_i\ddot{\boldsymbol{x}}_{i,ref} \\ & + (\boldsymbol{\Lambda}_i(\boldsymbol{J}_i\boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{J}}_i)\boldsymbol{J}_i^{M+^\top} + \boldsymbol{K}_{D,i})\dot{\boldsymbol{e}}_i \\ & + \boldsymbol{K}_{P,i}\boldsymbol{e}_i.\end{aligned} \quad (7)$$

where $\boldsymbol{\Lambda}_i = (\boldsymbol{J}_i\boldsymbol{M}^{-1}\boldsymbol{J}_i^\top)^{-1}$ is the task inertia matrix, results with the stable closed-loop error dynamics:

$$\boldsymbol{\Lambda}_i\ddot{\boldsymbol{e}}_i + (\boldsymbol{\Lambda}_i(\boldsymbol{J}_i\boldsymbol{M}^{-1}\boldsymbol{C} - \dot{\boldsymbol{J}}_i)(\boldsymbol{J}_i^{M+})^\top + \boldsymbol{K}_{D,i})\dot{\boldsymbol{e}}_i + \boldsymbol{K}_{P,i}\boldsymbol{e}_i = 0, \quad (8)$$

where $\boldsymbol{K}_{D,i}$ and $\boldsymbol{K}_{P,i}$ are the positive definite gain matrices.

The first distinct difference between MPTC and IDC is that the desired MPTC dynamics (8) is denoted in force space, whereas the desired IDC dynamics (5) is written in acceleration space. The motivation behind the MPTC dynamics is the corresponding passive (and thus robust) interaction behavior, as compared to the IDC dynamics, which is motivated through the related nominally stable system eigenvalues. Different from (5), in MPTC error dynamics (8), the damping term $\boldsymbol{K}_{D,i}$ is summed with the task space Coriolis matrix, and the task space inertia matrix remains, which causes the eigenvalues of error dynamics to vary depending on the configuration. The task space Coriolis matrix is kept due to passivity considerations. The reader is referred to [19] for more details.

D. Reactive 3D SLIP Trajectory Generation Yielding Deadbeat Stability

In previous studies, usage of the spring-loaded inverted pendulum (SLIP) dynamics as a humanoid's CoM trajectory objective showed impressive results on walking, running, and jumping [6]–[11]. Similarly, in this study, we employ SLIP trajectory as a CoM tracking objective for periodic stationary jumping and disturbance rejection. Following the Deadbeat control approach suggested in [7], touchdown angle and stiffness are determined based on apex velocity and height errors. The corresponding CoM tracking objective can be implemented using the centroidal Jacobian [25].

III. WHOLE-BODY CONTROL FORMULATION

The whole-body control formulation proposed in this work gathers all tasks and combines them with their corresponding weight matrices in a quadratic form. We utilize a soft prioritization for tasks using weight matrices, i.e., higher priority tasks have higher weights. Including their equations as optimization objectives, the formulation considers state accelerations, actuation inputs, contact forces, and constraint forces as input variables and solves for them. This input vector choice keeps the formulation straightforward and easily generalizable for high DOF and complex systems, as the one considered in this work. Substitution of system dynamics and constraint forces is tedious, and costly [26], especially

for closed-kinematic chain systems. Due to the existence of passive joints in closed-kinematic chains, their open-kinematic chain representation includes much more state variables than regular open-kinematic chain robotics systems. For example, although Kangaroo has only 12 actuated joints, due to its closed kinematic chain construction, it comes with 64 more passive joints, which yields a total of 76-DOF robot [24].

A. Contact Force Formulation

In order to obtain a dynamically consistent contact constraint as in [27], we model contact as a combination of pure forces at each foot corner, i.e., $\boldsymbol{f}_{c,i} \in \mathbb{R}^3$ where $i = 1, \ldots, n_c$ and n_c represents the number of foot corners in contact with the ground. For a stationary feet condition on the ground, following (3), the constraint becomes

$$\boldsymbol{J}_c \ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_c \dot{\boldsymbol{q}} = 0, \tag{9}$$

where $\boldsymbol{J}_c \in \mathbb{R}^{3n_c \times n}$ represents the contact Jacobian. With this constraint, the objective becomes to find a set of contact force combinations such that contact point accelerations are zero in any direction. In order to obtain a realistic solution, one should limit the contact forces to be inside friction limits. A friction pyramid formulation is applied to preserve the linearity of optimization constraints. To keep the forces inside an approximated friction cone, for each contact point force $\boldsymbol{f}_{c,i} = (f_x, f_y, f_z)^\top$, one should ensure that:

$$|f_x| \leq \frac{\mu f_z}{\sqrt{2}}, \;\; |f_y| \leq \frac{\mu f_z}{\sqrt{2}}, \text{ and } \; f_z \geq 0, \tag{10}$$

where μ is the friction constant. The formulation also limits the rotational moment per foot implicitly. Readers interested in conic friction formulation may check [28].

B. Closed Kinematics Constraint Force Formulation

Since most dynamics libraries consider the system an open kinematic chain when facing closed kinematics chains, one should solve for the constraint forces that enforce corresponding endpoints to move together. These constraint forces will drive the open chain ends such that they follow their counterpart link, and the chain is closed. To do so, the movement of open ends must be matched with constraints as in (3). This time, instead of forcing acceleration to be zero as in (9), the acceleration of both ends should be equal[1]. Assuming $\boldsymbol{J}_{k,1}$ and $\boldsymbol{J}_{k,2}$ to be Jacobian mappings of both endpoints of the k^{th} closed chain, the constraint equation becomes

$$\underbrace{(\boldsymbol{J}_{k,1} - \boldsymbol{J}_{k,2})}_{\boldsymbol{J}_{p,k}} \ddot{\boldsymbol{q}} + \underbrace{(\dot{\boldsymbol{J}}_{k,1} - \dot{\boldsymbol{J}}_{k,2})}_{\dot{\boldsymbol{J}}_{p,k}} \dot{\boldsymbol{q}} = 0. \tag{11}$$

[1]To have a valid comparison, frames of both endpoints should share the same orientation.

C. Passive Joint State Estimation

Even though each actuation unit comes with position and force sensors, the passive joints do not provide any sensory information. Since the full-order system dynamics require the state information for all joints, we must solve inverse-kinematics problems for each closed chain. In order to avoid the nonlinearity that comes with the position-based inverse-kinematics approach, we implement velocity-based inverse kinematics with drift compensation. Utilizing $\boldsymbol{J}_{p,k}$ from (11) for the k^{th} closed linkage, the velocity constraint becomes:

$$\begin{bmatrix} \boldsymbol{J}^a_{p,k} & \boldsymbol{J}^p_{p,k} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{q}}^a \\ \dot{\boldsymbol{q}}^p \end{bmatrix} = 0 \tag{12}$$

where $(\cdot)^a$ and $(\cdot)^p$ represent active and passive joint portions[2], respectively. Solving for the passive joint velocity

$$\dot{\boldsymbol{q}}^p = -(\boldsymbol{J}^p_{p,k})^{-1}(\boldsymbol{J}^a_{p,k}\dot{\boldsymbol{q}}^a) \tag{13}$$

and defining the position error for the k^{th} closed linkage to be $\boldsymbol{e}_{p,k} = \boldsymbol{P}^a_k(\boldsymbol{q}) - \boldsymbol{P}^p_k(\boldsymbol{q})$, the position estimation with drift compensation turns out to be:

$$\boldsymbol{q}^p = \int_0^t (\dot{\boldsymbol{q}}^p + \boldsymbol{\alpha}\boldsymbol{e}_{p,k})dt \tag{14}$$

where $\boldsymbol{\alpha}$ is a positive definite gain matrix.

D. Overall Whole-Body Control Formulation

As discussed in Section III, we select $u = [\ddot{\boldsymbol{q}}; \boldsymbol{\tau}; \boldsymbol{f}_c; \boldsymbol{f}_p]$ as optimization parameters and solve for all of them, where $\boldsymbol{\tau}$, $\boldsymbol{f}_c$, and $\boldsymbol{f}_p$ represent actuation input (i.e. joint torques and forces), ground reaction forces, and closed-linkage constraint forces, respectively. To have the system dynamics and all constraints implicitly substituted inside the optimization problem, we introduce them as equality and inequality constraints of the optimization problem. This WBC structure yields 182 optimization variables in the case of the Kangaroo.

1) Inverse Dynamics Based WBC: The combination of all inverse dynamics-based task controllers (4) into a quadratic problem constitutes the cost function of the optimization problem:

$$\underset{\ddot{\boldsymbol{q}}, \boldsymbol{\tau}, \boldsymbol{f}_c, \boldsymbol{f}_p}{min} \sum_i (\ddot{\boldsymbol{x}}_{i,d} - \ddot{\boldsymbol{x}}_i)^\top \boldsymbol{W}_i (\ddot{\boldsymbol{x}}_{i,d} - \ddot{\boldsymbol{x}}_i) \tag{15}$$

Such that:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g} = \boldsymbol{B}\boldsymbol{u} + \boldsymbol{J}_c^\top \boldsymbol{f}_c + \boldsymbol{J}_p^\top \boldsymbol{f}_p \tag{15a}$$

$$\boldsymbol{J}_c\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_c\dot{\boldsymbol{q}} = 0 \tag{15b}$$

$$|f_{x,l}| \leq \frac{\mu f_{z,l}}{\sqrt{2}}, \;\; |f_{y,l}| \leq \frac{\mu f_{z,l}}{\sqrt{2}}, \text{ and } \; f_z \geq 0 \;\; \forall l \tag{15c}$$

$$\boldsymbol{J}_p\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_p\dot{\boldsymbol{q}} = 0 \tag{15d}$$

$$\boldsymbol{\tau_{min}} \leq \boldsymbol{\tau} \leq \boldsymbol{\tau_{max}} \tag{15e}$$

where $\ddot{\boldsymbol{x}}_i = \dot{\boldsymbol{J}}_i\dot{\boldsymbol{q}} + \boldsymbol{J}_i\ddot{\boldsymbol{q}}$ and $\boldsymbol{\tau_{min}}$ and $\boldsymbol{\tau_{max}}$ represent minimum and maximum limits of input torques and forces. The cost function (15) is constituted by a combination of

[2]In our case: $\dot{\boldsymbol{q}}^a \in \mathbb{R}^{12}$ and $\dot{\boldsymbol{q}}^p \in \mathbb{R}^{64}$

different tasks and their corresponding weight matrices $\boldsymbol{W}_i$. In this formulation, since task errors with higher weights have more effect on total cost, they are prioritized during the solution process.

2) Modular Passive Tracking Based WBC: The WBC for MPTC-based task formulation shares the same constraints with (15). The difference appears in the cost function formulation. In this case, the cost function is the combination of modular passive tracking task controllers (7) in a quadratic problem form. The resultant WBC formulation turns out to be:

$$\underset{\ddot{\boldsymbol{q}},\boldsymbol{\tau},\boldsymbol{f}_c,\boldsymbol{f}_p}{min} \sum_i (\boldsymbol{f}_{i,d} - \boldsymbol{f}_i)^\top \boldsymbol{W}_i (\boldsymbol{f}_{i,d} - \boldsymbol{f}_i) \tag{16}$$

Such that: (15a) (15b) (15c) (15d) (15e)

where $\boldsymbol{f}_i = \boldsymbol{J}_i^{M^+}(\boldsymbol{B}\boldsymbol{\tau} + \boldsymbol{J}^\top \boldsymbol{f})$ and $\boldsymbol{B}\boldsymbol{\tau}$ and $\boldsymbol{J}^\top \boldsymbol{f}$ comes from the system dynamics (1) representing the combination of all actuation efforts along with all external and constraint force effects. In this case $\boldsymbol{J}^\top \boldsymbol{f} = \boldsymbol{J}_c^\top \boldsymbol{f}_c + \boldsymbol{J}_p^\top \boldsymbol{f}_p$.

IV. Implementation Details and Task Formulations: Kangaroo

This section includes some vital implementation details, such as parameter selection for MuJoCo's approximate constraint space dynamics, Kangaroo's description, WBC parameter selections, and 3D SLIP control implementation details.

A. Kangaroo Robot Description

Kangaroo is a 76-DoF bipedal robot. Among them, due to 24 closed linkages in its structure, 64 are passive, and 12 are actuated [24]. We illustrate the closing points of these parallel kinematics in Fig. 1.

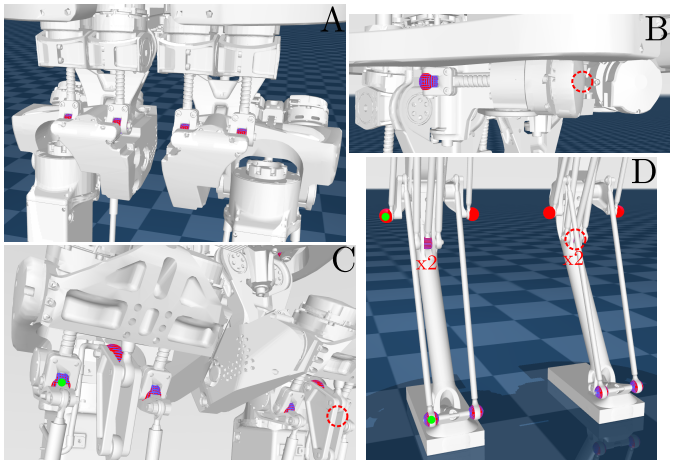

Fig. 1. The linkage closing points of Kangaroo. A: hip differential, B: hip yaw, C: ankle differential and knee support bar, D: ankle transmission and knee length drive, which also coincides with the knee support bar constraint. The dashed circles represent the invisible closing points due to the perspective, and the green dots highlight one of the ankle transmission lines for later discussion. The complete robot can be seen in Fig. 4.

B. MuJoCo Closed Linkage Constraints

We performed the simulation via MuJoCo dynamic simulator [29]. MuJoCo employs approximate elastic constraint space dynamics to support constraints such as closed linkages. This feature enables us to define *equality* constraints for link closing points. Since it is an elastic model, we needed to adjust *solimp* and *solref* settings and change the solver's effective stiffness, damping, time constant, and impedance parameters to get more consistent behavior and reduce modeling differences between the elastic simulation and rigid WBC constraints. Setting the time constant lower and the impedance parameter higher than their default values result in stronger and harder constraints. We address readers to the discussion about computation and solver parameters sections of [29] for a more detailed discussion.

C. Deadbeat 3D SLIP Control

To generate CoM trajectory and reject disturbances, we adjust a Deadbeat controlled 3D SLIP model from [7] for our jumping implementation. The SLIP model (see Fig. 2) includes four different input parameters, which are $\boldsymbol{u}_{SLIP} = [\theta, \phi, k_{s1}, k_{s2}]^\top$, where, to add or remove energy, k_{s1} and k_{s2} are used to make k_s differ in compression and relaxation phases, respectively. Depending on horizontal[3] and lateral velocity components and maximum jumping height, the Deadbeat controller modifies these parameters such that

$$\boldsymbol{u}_{SLIP} = \boldsymbol{u}^*_{SLIP,0} + \boldsymbol{K}(\boldsymbol{x}_{SLIP,0} - \boldsymbol{x}^*_{SLIP,0}) \tag{17}$$

where $\boldsymbol{u}^*_{SLIP,0}$, $\boldsymbol{K}$, $\boldsymbol{x}_{SLIP,0}$, and $\boldsymbol{x}^*_{SLIP,0}$ represent periodic input parameters, control gain, current apex state, and periodic apex state, respectively. For undisturbed condition $\boldsymbol{u}^*_{SLIP,0} = [0; 0; k_s; k_s]$ and since the vertical velocity at the apex is zero, the apex state can be represented using only the horizontal velocity, lateral velocity, and jumping height: $\boldsymbol{x}_{SLIP,0} = [\dot{p}_{c,1}; \dot{p}_{c,2}; p_{c,3}]$. For undisturbed condition $\boldsymbol{x}^*_{SLIP,0} = [0; 0; p^*_{c,3}]$ where $p^*_{c,3}$ represents periodic desired jumping height. For details of the Deadbeat gain matrix $\boldsymbol{K}$, we address readers to [7]. In order to reject velocity disturbances, this gain adjusts touchdown angles such that the SLIP model bounces in the reverse direction. Similarly, suppose the jumping height is higher than the desired value. In that case, it increases compression stiffness k_{s1} and reduces relaxation stiffness k_{s2} such that a required amount of energy is removed from the system. The reverse behavior happens when the jumping height is lower. We present the control diagram of the whole system in Fig. 3.

Since the SLIP controller (17) doesn't include any positional information of CoM in forward and lateral directions and controls only for velocities and jump height, a position drift throughout the aerial phases may happen due to slight non-zero velocity components in these directions. In order to prevent this position drift, at each touchdown phase of the robot, that is also, when the SLIP simulation starts, we locate the CoM point of the SLIP model above the origin, that is $[p_{c1}, p_{c2}]^\top = [0, 0]^\top$ (see Fig. 2). Since

[3]We refer horizontal as the forward direction.

the position command for non-vertical directions always stays around zero, the WBC prevents position drift and makes the robot return to its initial position. Equivalently, changing $\boldsymbol{x}^*_{SLIP,0}$ in (17) to be $[-\alpha_1 p_{c,1}; -\alpha_2 p_{c,2}; p^*_{c,3}]$, where α_1 and α_2 represent positive constants, also results in additional velocity commands in case of a position error, i.e., $[p_{c1}, p_{c2}]^\top \neq [0,0]^\top$, such that the robot jumps back to its initial position. Since we observed more robust behavior, instead of modifying $\boldsymbol{x}^*_{SLIP,0}$, we utilize WBC to prevent position drift of the robot.

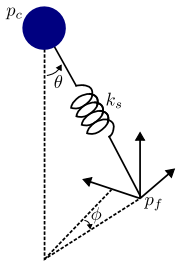

Fig. 2. 3D SLIP template model where p_c and p_f represent CoM and touchdown points, respectively. If there is no disturbance, angles are zero, and stiffness is constant for a fixed-height periodic jumping.

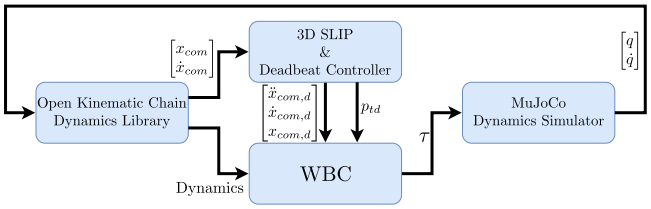

Fig. 3. Control schematic of the system. The WBC receives CoM trajectory and foot touchdown position ($\boldsymbol{p}_{td}$) commands from the Deadbeat-controlled 3D SLIP controller and tracks them.

D. Whole Body Control Parameter Selection

Both WBC formulations ((15) and (16)) we employ in this study require parameter selection for their desired damping and stiffness terms. In order to have well-defined damping behavior, the study in [30] designs a damping matrix $\boldsymbol{K}_{D,i}$ that depends on the desired stiffness $\boldsymbol{K}_{P,i}$ and the actual cartesian mass matrix $\boldsymbol{\Lambda}_i(q)$. The aim is to obtain a configuration-independent damping behavior for each task, which might preferably be critically damped behavior. Following their study, at each time step (or configuration), selecting $\boldsymbol{K}_{D,i}$ to be

$$\boldsymbol{K}_{D,i} = \boldsymbol{\Lambda}_i^{1/2} \boldsymbol{D}_{\xi,i} \boldsymbol{K}_{P,i}^{1/2} + \boldsymbol{K}_{P,i}^{1/2} \boldsymbol{D}_{\xi,i} \boldsymbol{\Lambda}_i^{1/2} \qquad (18)$$

results in a well-damped closed-loop error dynamics for each task. In the equation, $\boldsymbol{D}_{\xi,i}$ is a diagonal matrix with repeating or varying ξ_i elements and $0 \leq \xi_i \leq 1$ (0 for undamped behavior and 1 for real eigenvalues with critical damping).

Based on our observations, we manually tune weight and stiffness parameters for the best jumping behavior without disturbance. During the tuning process, we consider states at the moment of touchdown, the required amount of torso orientation and CoM state corrections, and the remaining non-vertical velocity components of CoM at the moment of the lift-off phase, along with overall tracking performance. We list all tasks and their corresponding parameter selections along with phase-dependent constraints in Table I. We employ OSQP [31] to solve the optimization problems in (15) and (16).

TABLE I

LIST OF TASKS, CONSTRAINTS, WEIGHT, AND CONTROLLER PARAMETER SELECTIONS FOR THE STANCE AND FLIGHT PHASES, RESPECTIVELY. DURING THE FLIGHT PHASE, THE CONTACT CONSTRAINTS ARE DEACTIVATED, AND THE CONTACT FORCES ARE CONSTRAINED TO ZERO. THE VALUES REPRESENT REPEATING ELEMENTS OF DIAGONAL MATRICES.

Stance		$\boldsymbol{W}$	$\boldsymbol{K}_{P,IDC}$	$\boldsymbol{K}_{P,MPTC}$	ξ
	CoM trajectory	2	300	5000	1.0
	Torso orientation	5	40	50	1.0
	Constraints	(15a), (15b), (15c) (15d), (15e)			
Flight		$\boldsymbol{W}$	$\boldsymbol{K}_{P,IDC}$	$\boldsymbol{K}_{P,MPTC}$	ξ
	Foot position	2	1500	750	1.0
	Foot orientation	2	2500	15	1.0
	Constraints	(15a), (15d), (15e) $\boldsymbol{f}_c = 0$			

V. SIMULATION RESULTS

We conduct simulations for both inverse dynamics and modular passive tracking based whole body controllers. This section discusses behavioral differences between these controllers by observing their undisturbed and disturbed performances. Snapshots from the robot's jumping can be seen in Fig. 4.

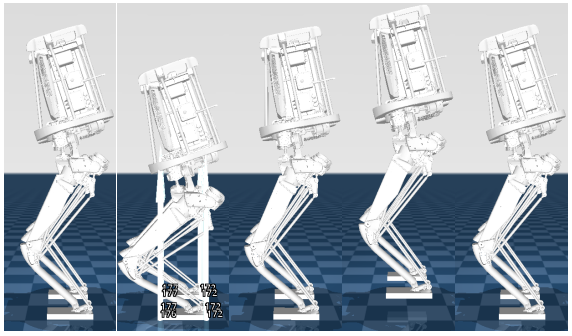

Fig. 4. Snapshots of jumping simulation. From left to right, the images show touchdown, bottom, lift-off, apex, and touchdown moments for 10cm periodic jumping. The bottom phase snapshot also includes ground reaction forces. Contact state detection relies on absolute foot height.

Since the SLIP model does not cover the torso dynamics of the robot, the torso tends to disturb the robot's horizontal CoM position and velocity as it oscillates back and forward. It tends to fall forward, especially at the impact moment, and this repeating torso regulation effort causes periodic jumps in horizontal CoM velocity. This effect can be seen in undisturbed simulation results with both controllers in Fig. 5 and 6. Due to their characteristic differences in the error

dynamics, we observe that MPTC shows a more damped behavior and follows a smoother trajectory. The figures also show that the Deadbeat-controlled SLIP model generates a nice and smooth path back to zero when a velocity error exists. Since the aerial phase is a basic free-fall motion, we activate the SLIP trajectory commands when the robot touches the ground.

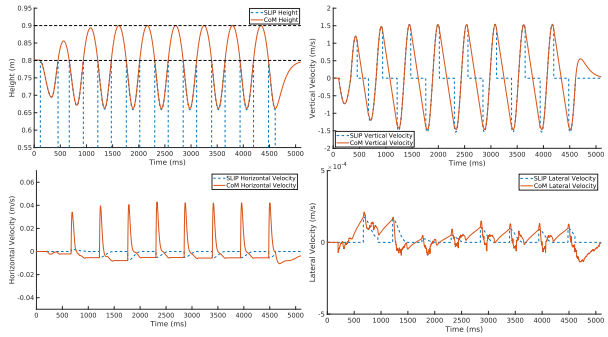

Fig. 5. Standing-jumping-standing simulation result of undisturbed MPTC-based WBC. Vertical dashed lines indicate activation of SLIP trajectory from touchdown to liftoff phase.

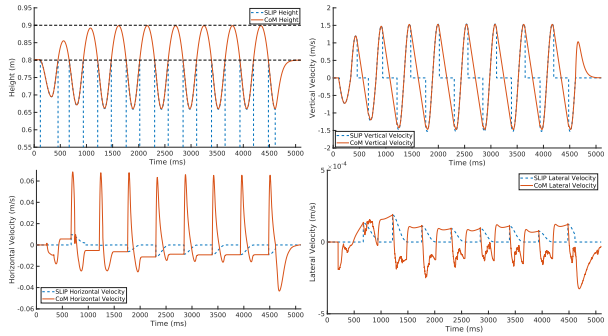

Fig. 6. Standing-jumping-standing simulation result of undisturbed IDC-based WBC. Vertical dashed lines indicate activation of SLIP trajectory from touchdown to liftoff phase.

In case of a disturbance, the controllers show similar characteristic behaviors with their undisturbed counterparts, as seen in Fig. 7 and 8. When we push the robot, the SLIP controller changes the touchdown angle and rejects the disturbance. Inverse dynamics-based control shows more oscillatory behavior with higher absolute error. Overall, both controllers behave well enough to reject the disturbances. The figures also show that the SLIP control generates nice and smooth velocity paths back to zero in case of disturbances. Since the forward and lateral CoM positions are not zero after the disturbances, the WBC causes velocities to fall below zero and returns the robot to its initial position.

As an example, in Fig. 9, we illustrate force transmission from one of the ankle drive motors towards the foot through the closed-linkage points we mark with green dots in Fig. 1. The green dot in subfigure C shows the connection point of one of the ankle motors. The movement of the motor is then transferred into the second closed-linkage connection point (on the top left corner of subfigure D). This

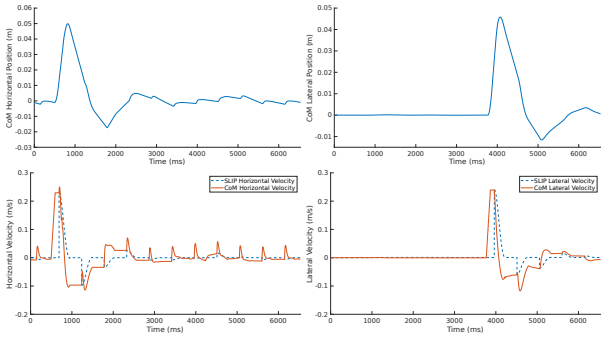

Fig. 7. Disturbed simulation result of MPTC-based WBC. Starting 50ms before apex, we apply 100N force from torso for 100ms in both directions.

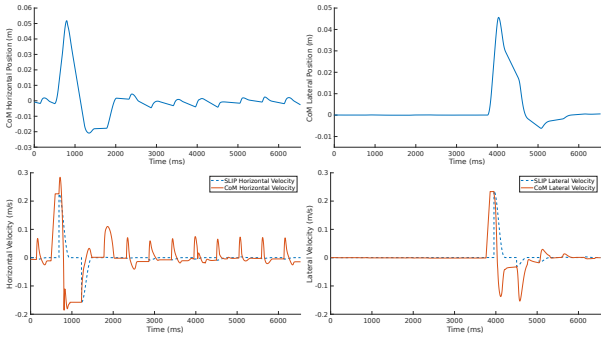

Fig. 8. Disturbed simulation result of IDC-based WBC. Starting 50ms before apex, we apply 100N force from torso for 100ms in both directions.

movement continues to the foot through the third closed-linkage connection point. For distinction, we named these points left ankle differential 1, left knee transmission 1, and left foot ball joint 1. This force analysis may also be helpful during a mechanical design process. The joint force magnitudes for particular robot movements may be used for bearing selections. Even the open kinematic chain joints can be broken and analyzed using our approach. The slight deformations in the figure originate from our stiff parameter selection for MuJoCo's constraint dynamics, as discussed in part B of section IV.

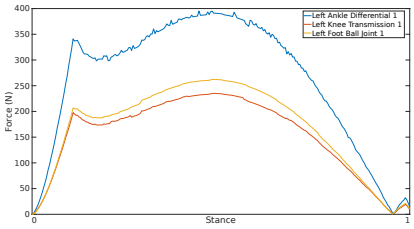

Fig. 9. Force transmission through the closed-linkage points marked with green dots in Fig. 1 during the stance phase, i.e., from touchdown to liftoff.

A. *Comments on Real-Time Capability*

Since the full-order Kangaroo model includes 76-DoF, our non-optimized implementation environment takes around 3-5 ms to compute a solution. So far, we haven't conducted a detailed solution time analysis with an optimized environment. In case of real-time implementation concerns for

higher frequencies, for example, 1kHz, changing the system structure, one can reduce the system model and eliminate some closed linkages which have a neglectable amount of inertial effects on the system. For example, the model in [23] reduces Kangaroo's dynamics into 22-DoF and utilizes only two closed linkages. This simplified model removes all linear actuation units and differential drive systems and assumes rotational actuation in all active joints other than knee length. With this method, one must again map actuation efforts back to the full-order system considering closed linkage kinematics. For this simplified model, we compute a solution in less than one millisecond in the same simulation environment. One can also introduce a less radical simplification to preserve some characteristic features of the robot.

VI. Conclusion

This study proposes a convenient and easy-to-implement whole-body control formulation for systems containing many closed kinematic chains and joints. We also discuss the implementation and control details for Kangaroo. Its 76-DoF design, with 24 independent closed linkages, provides an interesting usage case for our study and proves the effectiveness and usability of our approach. We also employ two different task controllers and show that this method works both in task acceleration and task force space.

References

[1] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, "Capture point: A step toward humanoid push recovery," in *6th IEEE-RAS international conference on humanoid robots*, 2006, pp. 200–207.

[2] J. Pratt, T. Koolen, T. De Boer, J. Rebula, S. Cotton, J. Carff, M. Johnson, and P. Neuhaus, "Capturability-based analysis and control of legged locomotion, part 2: Application to m2v2, a lower-body humanoid," *The international journal of robotics research*, vol. 31, no. 10, pp. 1117–1133, 2012.

[3] J. Englsberger, C. Ott, M. A. Roa, A. Albu-Schäffer, and G. Hirzinger, "Bipedal walking control based on capture point dynamics," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. -, 2011, pp. 4420–4427.

[4] M. Krause, J. Englsberger, P.-B. Wieber, and C. Ott, "Stabilization of the capture point dynamics for bipedal walking based on model predictive control," *IFAC Proceedings Volumes*, vol. 45, no. 22, pp. 165–171, 2012.

[5] J. Englsberger, C. Ott, and A. Albu-Schäffer, "Three-dimensional bipedal walking control based on divergent component of motion," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 355–368, 2015.

[6] P. M. Wensing and D. E. Orin, "Control of humanoid hopping based on a slip model," in *Advances in Mechanisms, Robotics and Design Education and Research*. Springer, 2013, pp. 265–274.

[7] ——, "High-speed humanoid running through control with a 3d-slip model," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. -, 2013, pp. 5134–5140.

[8] ——, "3d-slip steering for high-speed humanoid turns," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4008–4013.

[9] ——, "Development of high-span running long jumps for humanoids," in *IEEE International Conference on Robotics and Automation (ICRA)*. -, 2014, pp. 222–227.

[10] G. Garofalo, C. Ott, and A. Albu-Schäffer, "Walking control of fully actuated robots based on the bipedal slip model," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1456–1463.

[11] Y. Liu, P. M. Wensing, J. P. Schmiedeler, and D. E. Orin, "Terrain-blind humanoid walking based on a 3-d actuated dual-slip model," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1073–1080, 2016.

[12] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt, "Design of a momentum-based control framework and application to the humanoid robot atlas," *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1650007, 2016.

[13] S.-H. Lee and A. Goswami, "A momentum-based balance controller for humanoid robots on non-level and non-stationary ground," *Autonomous Robots*, vol. 33, no. 4, pp. 399–414, 2012.

[14] T. Koolen, J. Smith, G. Thomas, S. Bertrand, J. Carff, N. Mertins, D. Stephen, P. Abeles, J. Englsberger, S. Mccrory *et al.*, "Summary of team ihmc's virtual robotics challenge entry," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2013, pp. 307–314.

[15] L. Righetti, J. Buchli, M. Mistry, M. Kalakrishnan, and S. Schaal, "Optimal distribution of contact forces with inverse-dynamics control," *The International Journal of Robotics Research*, vol. 32, no. 3, pp. 280–298, 2013.

[16] B. Henze, M. A. Roa, and C. Ott, "Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios," *The International Journal of Robotics Research*, vol. 35, no. 12, pp. 1522–1543, 2016.

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

[18] B. Henze, R. Balachandran, M. A. Roa-Garzon, C. Ott, and A. Albu-Schäffer, "Passivity analysis and control of humanoid robots on movable ground," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3457–3464, 2018.

[19] J. Englsberger, A. Dietrich, G.-A. Mesesan, G. Garofalo, C. Ott, and A. Albu-Schäffer, "Mptc-modular passive tracking controller for stack of tasks based control frameworks," *Robotics: Science and Systems (RSS)*, 2020.

[20] L. Sentis and O. Khatib, "Synthesis of whole-body behaviors through hierarchical control of behavioral primitives," *International Journal of Humanoid Robotics*, vol. 2, no. 04, pp. 505–518, 2005.

[21] O. Khatib, M. Jorda, J. Park, L. Sentis, and S.-Y. Chung, "Constraint-consistent task-oriented whole-body robot formulation: Task, posture, constraints, multiple contacts, and balance," *The International Journal of Robotics Research*, vol. 41, no. 13-14, pp. 1079–1098, 2022.

[22] B. Siciliano, O. Khatib, and T. Kröger, *Springer handbook of robotics*. Springer, 2008, vol. 200.

[23] A. Roig, S. Kishor Kothakota, N. Miguel, P. Fernbach, E. Mingo Hoffman, and L. Marchionni, "On the hardware design and control architecture of the humanoid robot kangaroo," in *6th Workshop on Legged Robots during the International Conference on Robotics and Automation*, May 2022.

[24] E. Mingo Hoffman, S. Kothakota, A. Moreno, A. Curti, N. Miguel, and L. Marchionni, "Whole-Body Kinematics Modeling in presence of Closed-Linkages: application to the Kangaroo Biped Robot," in *International Conference on Robotics and Automation, Workshop on "New frontiers of parallel robotics" (second edition)*, May 2022.

[25] D. E. Orin, A. Goswami, and S.-H. Lee, "Centroidal dynamics of a humanoid robot," *Autonomous robots*, vol. 35, no. 2, pp. 161–176, 2013.

[26] N. Mansard, "A dedicated solver for fast operational-space inverse dynamics," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 4943–4949.

[27] J. Park and O. Khatib, "Contact consistent control framework for humanoid robots," in *IEEE International Conference on Robotics and Automation*, 2006, pp. 1963–1969.

[28] P. M. Wensing and D. E. Orin, "Generation of dynamic humanoid behaviors through task-space control with conic optimization," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3103–3109.

[29] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[30] A. Albu-Schaffer, C. Ott, U. Frese, and G. Hirzinger, "Cartesian impedance control of redundant robots: Recent results with the dlr-light-weight-arms," in *2003 IEEE International conference on robotics and automation (Cat. No. 03CH37422)*, vol. 3. IEEE, 2003, pp. 3704–3709.

[31] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.