3-D Dynamic Walking Trajectory Generation for a Bipedal Exoskeleton with Underactuated Legs: A Proof of Concept

A. F. Soliman[1], P. Sendur[1], and B. Ugurlu[1]

***Abstract*— This paper presents a framework to address three dimensional (3-D) dynamic walking for a bipedal exoskeleton with underactuated legs. To achieve this goal, the framework is constructed via a trajectory generator and an optimized inverse kinematics algorithm that can cope with underactuation. In order to feasibly attain task velocities with underactuated legs, the inverse kinematics algorithm makes use of a task prioritization method via the exploitation of null space. In doing so, the tasks with lower priority, e.g., swing foot orientation, are attained as much as possible without disrupting the higher priority tasks, such as, CoM trajectory. Meanwhile, the trajectory generator utilizes the ZMP concept analytically and ensures the acceleration continuity throughout the whole walking period, regardless of the contact and phase changes. The proposed method is verified via a lumped human-bipedal exoskeleton model that is developed and simulated in MSC.ADAMS simulation environment. As the result, we obtained feasible and dynamically balanced 3-D walking motion, in which no oblique foot landing or exaggerated torso orientation variations were observed, despite the underactuated nature of the robot legs.**

I. Introduction

Spinal Cord Injury (SCI) impacts 10.4 to 83.3 per million people worldwide each year [1], and a great majority of these patients lose their lives in case proper rehabilitation treatment is not provided [2]. In this connection, engineers devised several bipedal exoskeleton systems, most of which were clinically tested and are commercially available [3]. In order to provide walking support while satisfying weight and power consumption criteria, most of these systems provide active support solely for hip and knee F/E (Flexion/Extension) joints [4], [5]. Being very useful in its own right, long-term use and applicability of such an approach is deemed to be questionable due to several issues, e.g., limited gait capability and constant need of upper body engagement via crutches to maintain balance [6], [7].

In order to overcome these issues, one may increase the number active joints in an exoskeleton system to achieve more enhanced walking and balancing capabilities. For instance, Ugurlu et al. demonstrated self-balancing and active disturbance rejection capabilities with the addition of an active ankle joint along the D/PF (Dorsi/Plantar Flexion) axis [8].In order to address lateral motion, an active hip A/A (Abduction/Adduction) joint was integrated to the Mindwalker exoskeleton [9]. Further examples include fully-actuated (6 active joints per leg) systems with static and dynamic walking ability in 3-D with no crutches [10], [11]. Increasing the active number of robot joints surely enhances the gait capability; yet, these systems should be as light as possible to improve long-term usability and to meet power consumption requirements. In this paper, we argue that the minimum active joint requirement to address 3-D walking is 4-DoF per leg, namely, a 2-DoF (Degrees of Freedom) hip joint along the F/E and A/A axes, a 1-DoF knee joint along the F/E axis and a 1-DoF ankle joint along the D/PF axis. Therefore, it is technologically possible to provide 3-D dynamic walking support with a robotic exoskeleton using comparatively lower weight and reduced power consumption rates. Therefore, this paper chiefly investigates this research problem.

The main challenge in using the aforementioned 4-DoF per leg joint configuration is the underactuation problem. Conventional gait generation techniques may not be directly applicable to this case since the number of constraints exceeds the number of active robot joints. To that end, we present a framework to generate 3-D walking trajectory while handling the underactuation problem in real-time. The proposed framework includes a novel whole-body inverse kinematics algorithm with task-prioritization capability to handle underactuation. It is then integrated to a zero moment point (ZMP)-based analytical Cartesian trajectory generator that ensures acceleration continuity throughout the whole walking period [12]. The proposed framework is verified via 3D simulations in MSC.ADAMS environment in which a lumped human-robot model with distributed mass and inertia was constructed.

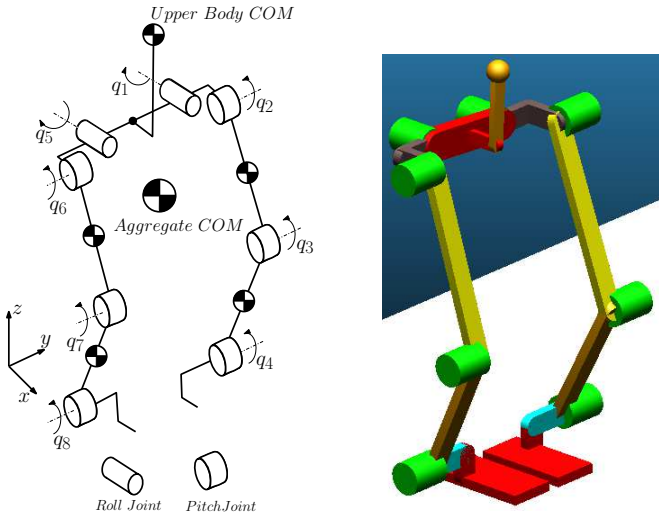
The reminder of this paper is organized as follows. Section 2 discusses the lumped human-robot model structure and its joints configuration, discloses the kinematics of the robot regarding its underactaution structure. The ZMP based analytical gait trajectory generation is discussed and an algorithm is proposed to generate feasible dynamic walking motion at the end of this section. Section 3 demonstrates the simulation results, while the conclusions are stated in Section 4.

II. Methods

A. Lumped Human-Robot Model

The lumped model with distributed mass is displayed in Fig. 1a. Each leg has 4-active DoF; namely, a 2-DoF

The authors are with the Dept. of Mechanical Engineering, Ozyegin University, 34794 Istanbul, Turkey. e-mail: fahmy.ahmed@ozu.edu.tr, barkan.ugurlu@ozyegin.edu.tr



(a) Lumped model structure

(b) Lumped model in MSC Adams environment

Fig. 1. Combined human-robot model. Aggregate COM indicates the overall COM of the system. The user upper body was considered as a single rigid body. In the rest of the paper the aggregate COM will be denoted as COM directly.

hip joint rotating along F/E and A/A axes, a 1-DoF knee joint rotating along the F/E axis and a 1-DoF ankle joint rotating along the D/PF axis. The overall upper body of the user is considered as a single rigid body. The decoupling between human and robot limbs is ignored and a basic lumped mass model is used. The inertial parameters of the robot were derived from the CAD model. The inertial parameters concerning the lower limbs of human were taken from [14], [15]. The human-robot coupled model was constructed in MSC.ADAMS; see Fig. 1b.

B. Kinematics

1) Forward Kinematics (FK): Two transformation matrices which represent the swing foot position and orientation with respect to the given support foot frame are derived. It is assumed that there is no relative motion between the support foot and the floor, i.e., the friction is sufficient. The first transformation matrix represents the right foot position and orientation $(P_l^r(q_a), R_l^r(q_a))$ with respect to the left foot inertial frame (LIF) $(T_l^r(q_a))$, where $(q_a = [q_1\ q_2\ q_3\ q_4\ q_5\ q_6\ q_7\ q_8])$ is a vector of the actuated joint variables. The second transformation matrix represents the left foot position and orientation $(P_r^l(q_a), R_r^l(q_a))$ with respect to the right foot inertial frame (RIF) $(T_r^l(q_a))$. The support foot frame position is modified while switching from one supporting foot to another.

$$P_{li} = P_g^l(q_a^-), \qquad P_{ri} = P_g^r(q_a^-). \tag{1}$$

In (1), P_{li} is the LIF position vector, while P_{ri} is RIF position vector. The superscript $(-)$ represents the prior value of the joints just before the switching instant. The subscripts (r, l, g) are the right, left, and global reference frames, respectively. The superscripts (r, l) are the right and left coordinate frames, respectively. The swing foot position obtained from the FK is modified by the support foot updating position vector,

$$P_g^r(q_a) = P_l^r(q_a) + P_{li},\ \ P_g^l(q_a) = P_r^l(q_a) + P_{ri} \tag{2}$$

referring its position with respect to the global inertial frame. The COM and torso position and rotation $(P_r^b(q_a), R_r^b(q_a), P_l^b(q_a), R_l^b(q_a))$ are derived using two distinct transformation matrices with respect to the two inertial frames $(T_r^b(q_a), T_l^b(q_a))$.The modified COM position with respect to the inertial frame is as below:

$$P_g^b(q_a) = \begin{cases} P_r^b(q_a) + P_{ri} & : I \in RIF \\ P_l^b(q_a) + P_{li} & : I \in LIF, \end{cases} \tag{3}$$

where, $P_r^b(q_a)$ is the COM position with respect to the RIF, $P_l^b(q_a)$ is the COM position with respect to the LIF, and I is the inertial frame indicator. The superscript b is the COM coordinate frame.

2) Inverse Kinematics (IK): The mapping between Cartesian $\dot{X}(q_a) : \Re^n \Rightarrow \Re^m$ and joint $\dot{q}_a(t) \in \Re^n$ velocities is as follows:

$$\dot{X}(q_a) = J(q_a(t))\dot{q}_a(t), \tag{4}$$

where $J(q_a(t)) \in \Re^{m\times n}$ is a Jacobian matrix, (n) is the number of the actuated joints, and (m) is the number of the required task variables. Since the robot kinematics are solved with respect to LIF and RIF, two Jacobian matrices are required for the mapping task.

$$J(q_a(t)) = \begin{cases} J_r(q_a(t)) & : I \in RIF \\ J_l(q_a(t)) & : I \in LIF \end{cases} \tag{5}$$

In (5), each Jacobian $(J_r(q_a(t)), J_l(q_a(t)))$ is obtained based on the given transformation matrices $(T_r^l(q_a),\ T_r^b(q_a),\ T_l^r(q_a),\ T_l^b(q_a))$ that are assigned for the specific inertial frame. Consider a generic function over RIF and LIF in the following form:

$$f_h(q_a) = \begin{bmatrix} \dot{P}_h^b(q_a) & \omega_h^b(q_a) & \dot{P}_h^{\overline{h}}(q_a) & \omega_h^{\overline{h}}(q_a) \end{bmatrix}^T. \tag{6}$$

In (6), the angular velocity vectors $(\omega_h^b(q_a), \omega_h^{\overline{h}}(q_a))$ are obtained by using the skew symmetric matrices $\dot{R}_h^{\overline{h}} R_h^{\overline{h}^T}(\omega_h^{\overline{h}})$ and $\dot{R}_h^b R_h^{b^T}(\omega_h^b)$. The superscript and subscript $(h \in L, R)$ indicates the frame; $\overline{h}$ and h are complement of each other. Equation (6) implies that the FK solution can be expressed as:

$$\dot{X}(q_a) = \begin{cases} f_R(q_a) & : I \in RIF \\ f_L(q_a) & : I \in LIF \end{cases}; \quad \forall q_a \in [\underline{q_a}, \overline{q_a}],\ (\underline{q_a}, \overline{q_a}) \in [-\tfrac{\pi}{2}, \tfrac{\pi}{2}], \tag{7}$$

where $(\underline{q_a}, \overline{q_a})$ are the physical joint limits. Since the robot model is underactuated $(m > n)$, there is no closed-form analytical solution for (4).

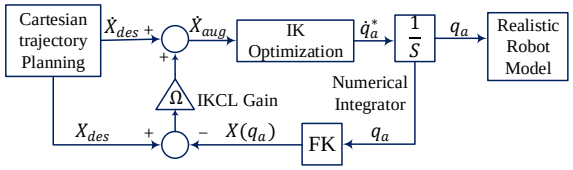


Fig. 2. The block diagram of the kinematics algorithm.

3) Kinematically Feasible IK Solution: Kinematically feasible joint velocities for given task velocities can be generated in terms of the best fitted task velocities with respect to the desired gait trajectories using an optimization algorithm. The optimization is performed taking into account of the following: i) It minimizes the residual terms representing the difference between the optimized solution and the desired gait trajectories, ii) the optimization problem is solved in multiple priority levels such that the tasks with lower priority are solved without disrupting the tasks with higher priority [16], [17], iii) during the optimization process joint limitations were taken into account, iv) due to numerical integration drift, a closed loop term is added.

4) Holonomic Constraints: These constraints represent the desired foot, torso, and COM trajectories $X_{des}(t) : \Re \in \Re^{12}$ in the task space which are defined between the robot coordinate frames and the ground. The objective of the optimization problem is to minimize the deviation between robot coordinates and gait trajectories. Since the optimization problem is solved in terms of differential kinematics, it is required to impose the assigned constraint as a differential expression: $\dot{X}_{des}(t)$. Hence, the error norm between the desired and actual trajectories is calculated as below:

$$\delta(\dot{q}_a) = ||\dot{X}_{des} - J(q_a)\dot{q}_a|| \tag{8}$$

The term $\delta(\dot{q}_a) : Q \in \Re^{12}$ represents the residual function between the desired and actual task trajectories in terms of actuated joint velocities $\dot{q}_a$. Q is the allowable joint configuration set in terms of joint velocity limits.

5) Task Prioritization: The proposed approach depends on dividing the task trajectories into two sets, based on their priority levels. The first set is responsible for achieving the generated gait that is based on ZMP; see section II-C. The second set is responsible for reducing the variations of COM height, torso and swinging foot orientation. The first set is solved in a higher priority level. The joint velocity set $\dot{q}_a$ is represented as the summation of two terms that depict the prior solution and the null space motion with respect to the prior tasks:

$$\dot{q}_a = J_{p-1}^{\dagger}\dot{X}_{des_{p-1}} + (I - J_{p-1}^{\dagger}J_{p-1})b \tag{9}$$

The subscript p indicates the priority level, e.g, $p-1$ is the prior level. The $\dagger$ superscript indicates matrix pseudo-inverse, while the variable $b \in \Re^n$ is a set of variables optimized with respect to the less prior constraints. The operator $P_{p-1} = (I - J_{p-1}^{\dagger}J_{p-1})$ is used to point the joint velocity modification variable b to the null space of the prior constraint Jacobian J_{p-1}. The term $\dot{q}_{a_{p-1}}^* = J_{p-1}^{\dagger}\dot{X}_{des_{p-1}}$ represents the optimized joint velocities with respect to the prior constraints. The term $(I - J_{p-1}^{\dagger}J_{p-1})b$ stands for the joint modification term that prevents less prior constraints to interfere with higher prior constraints.

6) Joint Range of Motion (ROM): The ROM represents the feasible limit for the gait motion in the joint space. As the exoskeleton robot is coupled with the human, the feasible joint limits are dictated by the human joint limits [18]. The joint ROM in this work is proposed in terms of joint angular displacement limits: $\{\underline{\dot{q}_a} = \frac{\underline{q_a} - q}{T}, \overline{\dot{q}_a} = \frac{\overline{q_a} - q}{T}\}$, where T is the optimization algorithm sampling time. The joint ROM is imposed to the objective function in the form of a barrier function $\psi(\dot{q}_a)$. It should satisfy the following conditions to preserve differentiability and convexity of the objective function: i) $\psi(\dot{q}_a)$ is a convex function in $\dot{q}_a$, ii) $\psi(\dot{q}_a)$ is continuously differentiable with $\dot{q}_a$, iii) $\psi(\dot{q}_a) \to \infty, \forall \quad \dot{q}_a \in \{\underline{\dot{q}_a}, \overline{\dot{q}_a}\}$, iv) $\psi(\dot{q}_a) \to C^n, \forall \quad \dot{q}_a \notin \{\underline{\dot{q}_a}, \overline{\dot{q}_a}\}$.

The set C^n is a constant valued set, where the barrier function $\psi(\dot{q}_a)$ tends to output constant values where the joint variables are not close to the joint limits $\{\underline{\dot{q}_a}, \overline{\dot{q}_a}\}$. The barrier function acts as an indicator function. Once the objective function gradient increases dramatically with respect to one of the actuated joints, the algorithm halts the optimization loop.

7) Auxiliary Functions: To tackle the numerical drift problem a Closed Loop Inverse Kinematics (CLIK) term is embedded to the optimization objective function. It adds a correction factor to optimization term that accounts for numerical approximation error:

$$\tilde{X} = \Omega(X_{des} - X), \tag{10}$$

In (10) X_{des} is the desired task trajectories, X is the forward kinematics solution obtained from (7), and $\Omega \in \Re^{m \times m}$ is a positive definite gain matrix to tune the convergence rate of the closed loop. The gain matrix in this work is tuned in a way so as to guarantee fast convergence and recovery of the joint trajectories. The objective function obtained from (8) and (9) constitutes the following unconstrained optimization problem,

$$\begin{aligned} b^* &= \arg\min_{\forall b \in R^n} (||\dot{X}_{aug_p} - J_p(\dot{q}_{a_{p-1}} + P_{p-1}b)||^2 \\ &+ \alpha\psi(\dot{q}_{ap})) \\ \dot{q}_{a_p}^* &= \dot{q}_{a_{p-1}}^* + P_{p-1}b^*, \end{aligned} \tag{11}$$

where $\dot{X}_{aug_p}$ is an augmented variable: $\dot{X}_{aug_p} = \dot{X}_{des_p} + \tilde{X}_p$. It is used to embody the closed loop correction term $\tilde{X}_p$ and the desired task trajectories vector $\dot{X}_{des_p}$ at the priority level p. The coefficient matrix α approximates the interference level between the barrier function $\psi(\dot{q}_a)$

and the optimization algorithm. See Fig. 2 for the joint trajectories generator block diagram.

C. ZMP-based Trajectory Generation

The COM trajectories were analytically synthesized based on the ZMP concept, which can be expressed as follows when considering the classical point mass model with a massless rod and a foot with a finite size. [13].

$$y_{zmp} = y_b - \frac{\ddot{y}_b}{g} z_b, \qquad x_{zmp} = x_b - \frac{\ddot{x}_b}{g} z_b \tag{12}$$

In (12), the COM height is kept constant: z_b. The Cartesian coordinates of the COM are denoted as (x_b, y_b, z_b), while the Cartesian coordinates of ZMP are denoted as $(x_{zmp}, y_{zmp}, 0)$. To synthesize COM trajectories for given ZMP inputs, we adopted the analytical trajectory generator presented in [12] due to its favorable features, such as smooth and continuous CoM trajectories for all position, velocity and acceleration levels and experimental applicability in real-time.

1) Single Support Phase: Fig. 3 displays the COM trajectory in blue and ZMP trajectory in red with respect to the world frame for a single walking period. During the left SSP (Single Support Phase), the footprints on the ground in Fig. 3 are given by the supporting polygon $(ABCDA)$, while the right foot swings in the air. In this phase, the ZMP trajectory is fixed at the point $(1')$, while the COM position is moving from point (1) to (2).

During SSP, ZMP reference input stays constant; $x_{zmp} = p_x$ and $y_{zmp} = p_y$, where p_x and p_y are constants. The COM trajectory can be obtained by solving (12).

$$x_b = p_x + \frac{\dot{x}_{so}}{\omega} sinh(\omega t) + (x_{so} - p_x) cosh(\omega t), \tag{13}$$

$$y_b = p_y + \frac{\dot{y}_{so}}{\omega} sinh(\omega t) + (y_{so} - p_y) cosh(\omega t), \tag{14}$$

where $\omega = \sqrt{\frac{z_b}{g}}$ and $t = t' - t_o$. t' represents the time span since the walking motion started, and t_o is the initial time at the beginning of the corresponding SSP. $(x_{so}, \dot{x}_{so}, y_{so}, \dot{y}_{so})$ are COM SSP initial conditions, and g is the gravitational acceleration.

2) Double Support Phase: In DSP (Double Support Phase), both feet are on the ground; hence, the supporting polygon is $(ABCGHEA)$. The ZMP is moved from $(1')$ to $(2')$ linearly, while the COM position is transferred from (2) to (3). The ZMP trajectory is used in this phase shifts it from one foot to another: $x_{zmp} = p_x + k_x t'$ and $y_{zmp} = p_y + k_y t'$. The parameters (k_x, k_y) represent the time slope of linear ZMP trajectories. In this case, the analytical solution of ZMP equations are as below:

$$x_b = p_x + k_x t' + \frac{\dot{x}_{do} - k_x}{\omega} sinh(\omega t') + (x_{do} - p_x) cosh(\omega t') \tag{15}$$

$$y_b = p_y + k_y t' + \frac{(\dot{y}_{do} - k_y)}{\omega} sinh(\omega t') + (y_{do} - p_y) cosh(\omega t') \tag{16}$$

where $(x_{do}, y_{do}, \dot{x}_{do}, \dot{y}_{do})$ are COM DSP initial conditions, i.e., previous SSP terminal conditions.

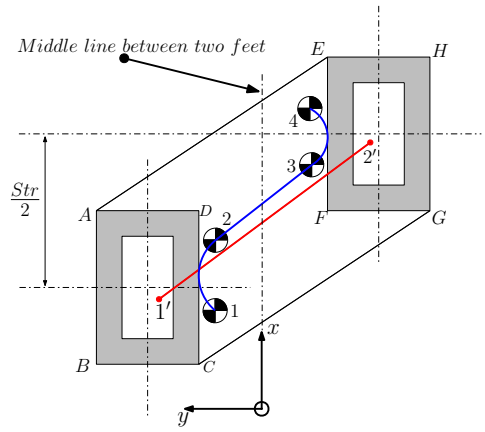


Fig. 3. COM trajectory in blue line vs. ZMP trajectory in red line with respect to the footprints. Left SSP: COM moves from 1 to 2 while ZMP stays constant at 1'. DSP: COM moves from 2 to 3 while ZMP linearly shifts from 1' to 2'. Right SSP: COM moves from 3 to 4 while ZMP stays constant at 2'.

D. Boundary Conditions and ZMP References

The boundary conditions and ZMP references were obtained in a way so as guarantee the continuity of COM acceleration regardless of the gait phases [12]. Fig. 4, 5 displays the x-axis and y-axis position, velocity and acceleration of COM trajectories for 10 seconds. The variables t_s, t_{ms}, t_d, t_{md} are the SSP, middle SSP, DSP, and middle DSP time span, respectively. Observing Fig. 4, 5, one may realize the following features: i) In SSP, y-axis COM trajectory is symmetrical, i.e., it bounces back to the initial position, ii) because of this reason, the COM velocity is zero in the middle of SSP, ii) y-axis COM position and acceleration crosses over zero in the middle of SSP, iv) an average forward velocity term, v_m can be used to define terminal x-axis position, v) x-axis COM acceleration crosses over zero in the middle of DSP. Using these symmetry conditions, one can compute the necessary parameters in eqs. (13)-(16) as follows.

$$x_{ts} = v_m t_s + x_{so} \tag{17}$$

$$p_x = \frac{x_{ts} sinh(\omega t_{ms}) + x_{ts} sinh(\omega(t_s - t_{ms}))}{sinh(\omega(t_s - t_{ms}) + sinh(\omega t_{ms})} \tag{18}$$

$$\dot{x}_{so} = (p_x - x_{so}) \omega coth(\omega t_{ms}) \tag{19}$$

$$k_x = \dot{x}_{do} + (x_{do} - p_x) \omega coth(\omega t_{md}) \tag{20}$$

$$Str = 2 k_x t_d \tag{21}$$

$$\dot{y}_{so} = \omega (p_y - y_{so}) tanh(\omega t_{ms}) \tag{22}$$

$$\dot{y}_{do} = (p_y - y_{do}) \omega coth(\frac{\omega t_d}{2}) + k_y \tag{23}$$

$$k_y = \frac{-2 p_y}{t_d} \tag{24}$$

$$p_y = \frac{y_{do} \omega coth(\omega t_{md}) + \omega tanh(\omega(t_{ms}))}{\omega coth(\omega t_{md}) + \omega tanh(\omega(t_{ms})) - \frac{2}{t_d}}. \tag{25}$$

In (17)-(25), t_s and t_d periods are determined beforehand. Initial values of x_{so} and y_{so} are assigned based on the geometric properties and initial joint configuration of

the robot. Furthermore, we have the following equalities to ensure seamless connection between a SSP and a DSP: $x_{do} = x_{ts}$, $\dot{x}_{do} = \dot{x}_{so}$, $\dot{y}_{do} = -\dot{y}_{so}$, and $y_{do} = y_{so}$. Where x_{ts} is the COM forward displaced distance during SSP. Swing leg trajectories are synthesized via polynomials. Refer to [12] for a detailed explanation.

III. Results

The walking motion is simulated using a realistic model developed in MSC.Adams software. The simulation parameters are listed in Table I.

The generated COM trajectories for x-axis and y-axis can be viewed in Figs. 4 and 5, respectively, in position, velocity and acceleration levels. In these figures, solid red lines indicate COM variations, while solid blue and black lines stand for feet trajectories, all with respect to the global frame. Furthermore, solid red and black lines in Fig. 6 point out the actual and estimated COM trajectories, with respect to global frame.

The actual ZMP response is displayed in Fig. 7 concerning the x-y plane. In this figure, solid red lines indicate the actual ZMP, while solid blue and black lines stand for the support polygon. Torso oscillation during the entire walking period is shown in Fig. 8.

IV. Discussion and Conclusion

Observing Fig. 6, the efficacy of the optimization algorithm can be verified as it shows convergence to the designated COM trajectories for the forward and lateral trajectories. Yet, it also indicates a slight deviation between the desired and estimated COM trajectories due to uncompensated yaw moment. We argue that it occurs due to the lack of closed-loop trajectory control, which will be addressed in our next work.

Fig. 7 displays that the actual ZMP always remained within the support polygon, thus adequately confirming that the system maintained its dynamic balance within the whole walking period. Furthermore, torso oscillation is inevitable because of the underactuated nature of the system; however, the proposed algorithm well-contained its variation as the maximum value was $\pm 14.1^o$.

The combined-human robot model weighed at 123 kg, so as to see if the proposed method can achieve balanced walking within this challenging condition. Therefore, it led to a walking behavior with a relatively low velocity profile. We argue that this issue can be improved with the addition of feedback controllers.

To conclude, this work indicated that a robot with 4-DoF per leg joint configuration can address 3D walking support with the help of a novel algorithm with task prioritization. Therefore, it is possible to minimize the total number of actuators for a lighter and less energy-demanding exoskeleton prototypes. In our future work, the proposed algorithm will be experimentally implemented to our lower body exoskeleton.

TABLE I

Simulation parameters

Mean Velocity	v_m	$42(mm/s)$
SSP Initial Conditions	(x_{so}, y_{so})	$(0, 8.28)(mm)$
SSP Period	t_s	$1(s)$
DSP Period	t_d	$0.18(s)$
Optimization Constants		
	Ω	$50\ I_{12\times 12}$
	T	$0.01(s)$
	α	10^{-5}
High priority tasks	$(\dot{x}_b, \dot{y}_b, \dot{P}_h^{\overline{h}}(q_a)$	
Low priority tasks	$(\dot{z}_b, \omega_h^b(q_a), \omega_h^{\overline{h}}(q_a))$	

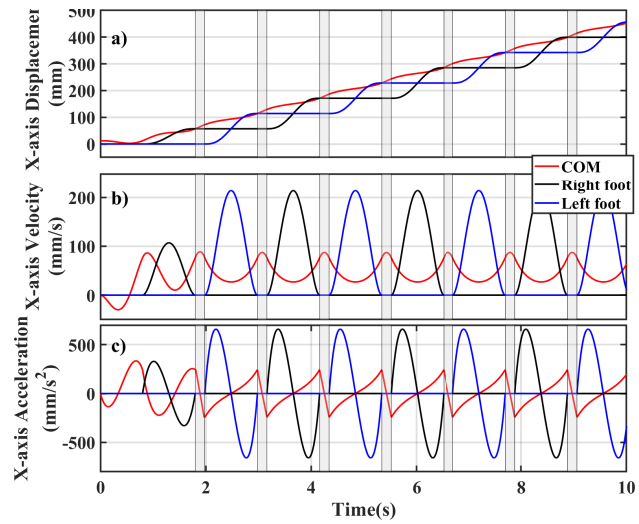


Fig. 4. x-axis COM trajectory and feet trajectories with respect to the global frame. Gray regions represent double support phases.

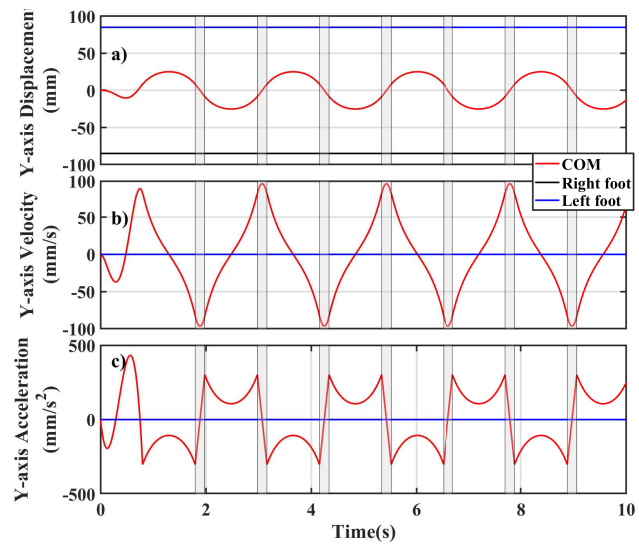


Fig. 5. y-axis COM trajectory and feet trajectories with respect to the global frame. Gray regions represent double support phases.

Acknowledgement

This work is supported by the Scientific and Technological Research Council of Turkey (TUBITAK) via the project numbered 215E138. The authors thank M. C. Yildirim for his support.

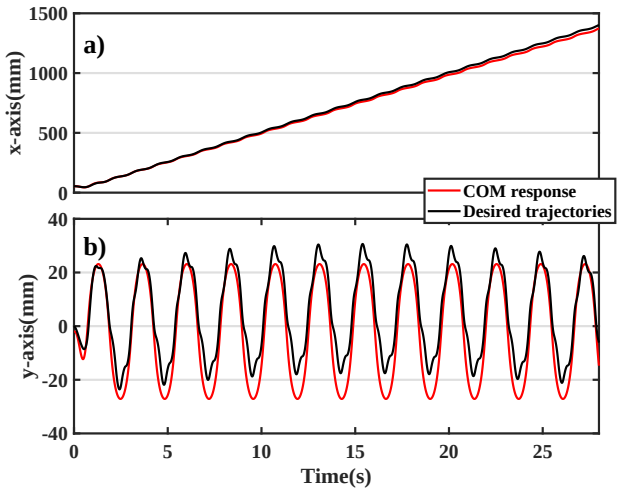


Fig. 6. Desired vs response of COM trajectories with respect to the global frame.

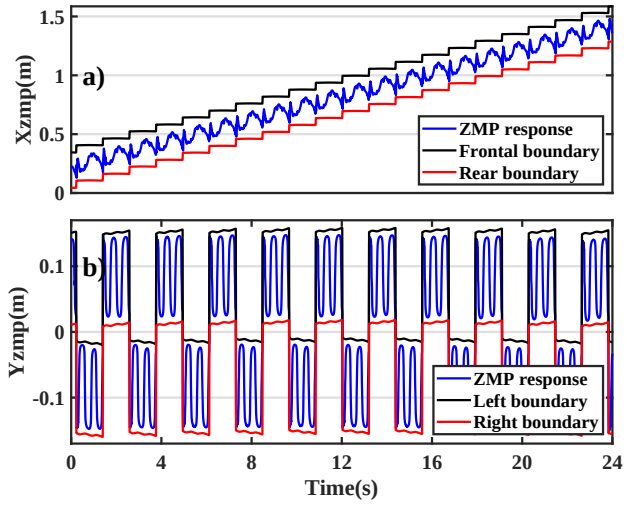


Fig. 7. ZMP vs. feet prints boundary trajectories over the $x - y$ Cartesian plan. a) Actual forward ZMP and feet prints trajectories. b) Actual lateral ZMP and feet prints trajectories

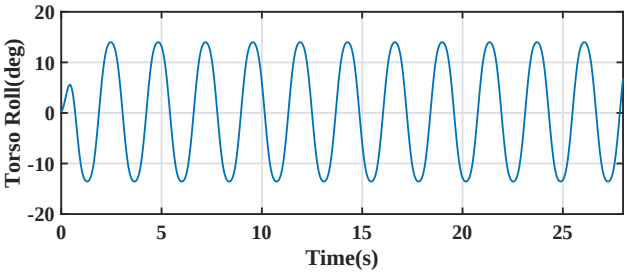


Fig. 8. Torso roll angle variation.

References

[1] M. Wyndaele and J.J, Wyndaele, *"Incidence, prevalence and epidemiology of spinal cord injury: what learns a worldwide literature survey?"* Int. Spinal Cord Society, vol. 44, 2006.

[2] D, Dryden, L, Saunders, B, Rowe, L, May, N, Yiannakoulias, L, Svenson, and D, Schopflocher *"Utilization of health services following spinal cord injury: a 6-year follow-up study,"* International Spinal Cord Society, vol. 42, pp. 513-525, 2004.

[3] A. J. Young and D. P. Ferris, *"State of the Art and Future Directions for Lower Limb Robotic Exoskeletons,"* IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 25, no. 2, pp. 171-182, 2017.

[4] B. Ugurlu, H. Oshima, and T. Narikiyo, *"Lower Body Exoskeleton-Supported Compliant Bipedal Walking for Paraplegics: How to Reduce Upper Body Effort?"* in Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1354-1360, Hong Kong, 2014.

[5] L. Benner, *"Exploring the Psychosocial Impact of Ekso Bionics Technology,"* Archives of Physical Medicine and Rehabilitation, vol. 97, no. 10, 2016.

[6] I. Benson, K. Hart, D. Tussler, J. J. van Middendorp *"Lower-limb exoskeletons for individuals with chronic spinal cord injury: findings from a feasibility study"*, Clinical Rehabilitation, vol. 30, no. 1, pp. 73-84, 2016.

[7] A. E. Palermo, J. L. Maher, C. B. Baunsgaard, M. S. Mash, *"Clinician-Focused Overview of Bionic Exoskeleton Use After Spinal Cord Injury,"* Topics in Spinal Cord Injury Rehabilitation, vol. 23, no. 3, pp. 234-244, 2017.

[8] B. Ugurlu, C. Doppmann, M. Hamaya, P. Forni, T. Teramae, T. Noda, and J. Morimoto, *"Variable Ankle Stiffness Improves Balance Control: Experiments on a Bipedal Exoskeleton,"* IEEE/ASME Trans. on Mechatronics, vol. 21, no. 1, pp. 79-87, 2016.

[9] S. Wang, L. Wang, C. Meijneke, E. vanAsseldonk, T. Hoellinger, G. Cheron, Y. Ivanenko, V. La Scaleia, F. Sylos-Labini, M. Molinari, F. Tamburella, I. Pisotta, F. Thorsteinsson, M. Ilkovitz, J. Gancet, Y. Nevatia, R. Hauffe, F. Zanow, and H. van der Kooij, *"Design and Control of the MINDWALKER Exoskeleton,"* IEEE Trans. on Neural Systems and Rehabilitation Engineering, vol. 23, no. 2, pp. 277-286, 2015.

[10] G. Barbareschi, R. Richards, M. Thornton, T. Carlson and C. Holloway, *"Statically vs dynamically balanced gait: Analysis of a robotic exoskeleton compared with a human,"* 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 6728-6731, 2015.

[11] A. Agrawal, O. Harib, A. Hereid, S. Finet, M. Masselin, L. Praly, A. D. Ames, K. Sreenath, and J. W. Grizzle, *"First Steps Towards Translating HZD Control of Bipedal Robots to Decentralized Control of Exoskeletons,"* IEEE Access, vol. 5, pp. 9919-9934, 2017.

[12] B. Ugurlu, I. Havoutis, C. Semini, K. Kayamori, D. G. Caldwell, and T. Narikiyo, *"Pattern Generation and Compliant Feedback Control for Quadrupedal Dynamic Trot-Walking Locomotion: Experiments on RoboCat-1 and HyQ,* Autonomous Robots, vol. 38. no. 4, 2015, pp. 415-437.

[13] M. Vukobratovic, B. Borovac, *"zero Moment Point - Thirty Five Years of its Life,"* Int. Journal of Humanoid Robots, vol. 1, no. 1, pp. 157-173, 2004.

[14] P. De Leva, *"Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters,"* in Journal of Biomechanics, vol. 29, no. 9, 1223-1230, 1996.

[15] R. Dumas, L. Cheze, and J.P. Verriest, *"Adjustments to McConville et al. and Young et al. body segment inertial parameters,"* in Journal of Biomechanics, vol. 40, no. 3, 543-553, 2007.

[16] A. Escande, N. Mansard and P. Wieber, *"Fast resolution of hierarchized inverse kinematics with inequality constraints,"* 2010 IEEE Int. Conf. on Robotics and Automation, Anchorage, pp. 3733-3738, 2010.

[17] P. Baerlocher and R. Boulic, *"An inverse kinematic architecture enforcing an arbitrary number of strict priority levels,"* The Visual Computer, 6(20):402-417, 2004.

[18] A. Zoss, H. Kazerooni and A. Chu, *"Biomechanical design of the Berkeley lower extremity exoskeleton (BLEEX),"* in IEEE/ASME Trans. on Mechatronics, vol. 11, no. 2, pp. 128-138, 2006.