A performance comparison between closed form and numerical optimization solutions for humanoid robot walking pattern generation

Samer A Mohamed1, Shady A Maged1 and Mohammed I Awad1,2

Abstract
This article presents the modeling process of the lower part of a humanoid biped robot in terms of kinematic/dynamic states and the creation of a full dynamic simulation environment for a walking robot using MATLAB/Simulink. This article presents two different approaches for offline walking pattern generation: one relying on a closed-form solution of the linear inverted pendulum model (LIPM) mathematical model and another that considers numerical optimization as means of desired output trajectory following for a cart table state-space model. This article then investigates the possibility of introducing solution-dependent modifications to both approaches that could increase the reliability of basic walking pattern generation models in terms of smooth single support–double support phase transitioning and power consumption optimization. The algorithms were coded into offline walking pattern generators for NAO humanoid robot as a valid example and the two approaches were compared against each other in terms of stability, power consumption, and computational effort as well as against their basic unmodified counterparts.

Keywords
Bipedal, model-predictive control, humanoid, cart-table model, cost of transport, linear inverted pendulum model, zero moment point, bio-inspired robotics, trajectories, center of mass

Introduction
A humanoid robot is a robot that resembles human body form and is bipedal in nature. The first human-resembling robot was developed by Ichiro Kato et al. from Waseda University in 1973. That robot was able to communicate and interact with the environment through some sensors and receptors. It could achieve a stable dynamic gait and manipulate some objects. Waseda University has been one of the leading research institutes for anthropomorphic robots. Ten years following that discovery, Waseda University has produced a handfull of humanoid robots like the musician robot WABOT-2 made in 1984, Hadaly-2, which cooperates with a human partner to do specific jobs,1 and the biped robot, WABIAN, which could perform impressive walking maneuvers in 1997.2 Based on the promising fruits of research in that field, Waseda University

1 Mechatronics Engineering Department, Ain Shams University, Cairo, Egypt
2 Mechanical Engineering Department, Galala University, Egypt

Corresponding author:
Samer A Mohamed, Mechatronics Engineering Department, Ain Shams University, 11566 Cairo, Egypt.
Email: samermansour1994@gmail.com
established the Humanoid Robotics Institute in April 2000 to fund research endeavors, which aim to construct an industrial environment in which humans and robots could cooperate with humans lying in the top tier decision-making hierarchy and robots in a lower tier in an advanced internet of things (IOT) structured industrial facility.\textsuperscript{3–5} Surprisingly, it is expected that a robot would perform domestic tasks such as house cleaning and organizing for middle-aged and old individuals all within few years.\textsuperscript{6} The humanoid will be able to integrate sensory data and perform intelligent actions, which include communication with individuals using paradigms such as natural language processing (NLP), facial expression analysis, and high-level motion control.\textsuperscript{7,8} Evidently, it is essential that humanoid robot research go hand in hand with other related research areas such as biology, human psychology, and social studies.\textsuperscript{9,10}

Later, various studies that focused on bipedal robots,\textsuperscript{11–13} control strategies, and stable dynamic walking\textsuperscript{4,15} have been conducted. These studies handled every aspect of human dynamic activities such as dynamic walking, push recovery from an unknown external force, adapting to uneven terrains, and so on. Boston Dynamics is known worldwide for its contribution to these fields through a collection of state-of-the-art biped and quadruped robots, including robots like BigDog, Spot, Handle, and Atlas.\textsuperscript{16,17}

Furthermore, intelligent behavior and autonomy were added to humanoid robots and several studies have been conducted on the topic. Perhaps, the most notable bipedal robot was Asimo developed by Honda, which could perform tasks like walking, running, and even more advanced tasks like playing football.\textsuperscript{18} Several algorithms that basically lie in the domains of environment perception, robot localization, and environment mapping have been explored to develop some autonomous behaviors for humanoid robots.\textsuperscript{19,20} Also, probabilistic approaches helped the robot to construct maps for different environments and localize itself in the environment using simultaneous localization and mapping algorithms and sensor fusion paradigms like extended kalman filter (EKF) and particle filter-based approaches.\textsuperscript{21} One of the most challenging aspects of humanoid robot design is the technique used in walking pattern generation since a biped robot is an inherently unstable structure in essence.\textsuperscript{22} Toward that end, several models have been adopted like the famous LIPM model, which simplifies the robot into a single point mass lying at the end of an inverted pendulum.\textsuperscript{23–26} Some adaptations have been added to the concept like the double inverted pendulum model in which the lower and upper parts are treated like two connected inverted pendulums\textsuperscript{27,28} and the flywheel addition that further complicates the model by considering rotational inertia for the pendulum end-point rather than just mass. A more versatile model for gait generation and perhaps the most widely used is the cart table model that basically simplifies the robot as a cart moving back and forth on a table at a constant height from the ground and introduces the zero moment point (ZMP) concept.\textsuperscript{29–32} This article proposes novel solutions to both basic models, which could help enhance the walking pattern generation process and make the basic models more flexible and considerate of practical walking considerations like smooth single support-double support transitioning and overall power consumption. This article evaluates the performance metrics of the presented closed form and numerical approaches for walking pattern generation and takes the NAO humanoid robot as a valid example. The results were validated using MATLAB/Simulink simulations.

**Robot modeling**

**Study of NAO kinematic structure**

NAO is an autonomous bipedal robot made by Aldebaran robotics. The version under study here is version 4 with 25 degrees of freedom overall. In the work presented here, the main focus is on the robot’s lower part, which includes the legs, pelvis, and has 11 degrees of freedom. Each leg has two degrees of freedom at the ankle, one degree of freedom at the knee, and two degrees of freedom at the hip. The pelvis is equipped with two coupled joints each attached to one of the two robot hips. The rotation axes of the two coupled joints are inclined to the outside at 45°. Normally, in most classical humanoid robot mechanical designs, the pelvis is composed of three separate joints each having a specific degree of freedom. The NAO design defies this convention by introducing coupled joints in the pelvis design, and despite the gains summarized in fewer motors, less power consumption, and optimized space, this design introduces complexities in the kinematic solution as the coupling prevents the upper part from rotating around the yaw axis when both legs are in support, which demands some form of kinematic decoupling to allow for a proper kinematic solution. An illustration for the NAO lower part kinematic structure is shown in Figure 1.

Forward kinematics is derived based on a chain rule of homogeneous transformations, which is later interpreted as a series of closed-form equations. First, it is crucial to assign both a global (world frame) and a robot local frame and then obtain the necessary transformations to create the closed-form solution.

The translation and orientation of joint \( j \) with respect to adjacent joint \( i \) in the 3D space\textsuperscript{33} is described using a \( 4 \times 4 \) transformation matrix \( T_{ij} \)

\[
T_{ij} = \begin{bmatrix}
\text{rot} & \text{tr}^+ \\
[0 \ldots 0] & 1
\end{bmatrix}
\] (1)

where \( \text{rot} \in R^{3 \times 3} \) is the rotational matrix describing the attitude of joint \( j \) frame relative to joint \( i \) frame and \( \text{tr}^+ \in R^3 \) is the translation vector of joint \( j \) frame origin.
A point $p_j = [p_x \ p_y \ p_z \ 1]^T$ described in frame $j$ can be transformed to a point $p_i$ in another frame $i$ by cascading the transformations for all intermediate frames

$$T_i = T_i^{i+1} T_i^{i+2} \ldots T_i^{j-1}$$

Applying these formulas altogether with Denavit–Hartenberg method yields the following kinematic equations

$$X = F_x + l_{tg} \times \sin(\theta_3) \times \left( \cos(\theta_4 + \theta_3) + \frac{l_{th}}{l_{tg}} \times \cos(\theta_4) \right)$$

$$Y = F_y + l_{tg} \times \sin(\theta_3) \times \left( \cos(\theta_4 + \theta_3) + \frac{l_{th}}{l_{tg}} \times \cos(\theta_4) \right) + h_m$$

$$Z = F_z + l_{tg} \times \cos(\theta_3) \times \left( \cos(\theta_4 + \theta_3) + \frac{l_{th}}{l_{tg}} \times \cos(\theta_4) \right) + \frac{h_m \cos(\theta_3)}{\cos(\theta_3)} + l_{fz}$$

where $X$, $Y$, and $Z$ represent the pelvis link center coordinates in global $x$, $y$, and $z$ axes, respectively, while $F_x$, $F_y$, and $F_z$ represent the global coordinates of the reference point on the foot sole.

As for inverse kinematics, a closed-form solution is considered to be the optimal solution since it requires minimal computational effort. Based on this fact, a geometrical solution of inverse kinematics was developed using trigonometric relations. The inverse kinematics closed-form solution can be used to generate the trajectories of the joint angles and help track the desired center of mass trajectory.

$$\theta_5 = \tan^{-1} \left( \frac{Y - F_y - h_m}{Z - F_z - l_{fz} - \frac{h_m}{\cos(\theta_3)}} \right)$$

$$\theta_3 = \cos^{-1} \left( \frac{(X - F_x)^2 + A^2 - l_{th}^2 - l_{tg}^2}{2 \times l_{th} \times l_{tg}} \right)$$

$$\theta_4 = -\tan^{-1} \left( \frac{A \times \sin(\theta_3) + (X - F_x) \times \cos(\theta_3) + \frac{l_{th}}{l_{tg}} \times (X - F_x) \times \sin(\theta_3)}{A \times \cos(\theta_3) + \frac{l_{th}}{l_{tg}} \times A - (X - F_x) \times \sin(\theta_3)} \right)$$

$$\theta_1 = -\theta_5$$

$$\theta_2 = -\theta_4 - \theta_3$$

Locating the center of mass location relative to the global frame is a crucial step in the walking pattern generation algorithm especially in the joint angle trajectory correction step because this step helps track the exact center of mass trajectory and make it follow the desired center of mass (COM) motion profile for a stable gait. First, it is required to locate the center of mass of each link relative to the global frame, then knowing all links’ center of mass coordinates and mass properties; it is possible to locate the robot COM coordinates as a whole. In the analysis presented here, it is assumed that the upper part of the robot is a rigid body whose center of mass is determined according to the current pose of the upper part, which has to do with the manipulation task the robot is performing like raising hands, objects, head turning, and so on. The COM transforms are computed as follows.
\[ T_1 = Trans(I_f, 0, 0) \]  \hspace{1cm} (12)

\[ T_2 = Trans(I_f, 0, 0) \]  \hspace{1cm} (13)

\[ T_3 = Trans(0, 0, I_f).Rot(x, -\theta_4).Trans(0, 0, 0.00684) \]  \hspace{1cm} (14)

\[ T_4 = Trans(0, 0, I_f).Rot(x, -\theta_4).Trans(0, 0, 0.00684) \]  \hspace{1cm} (15)

\[ T_5 = Trans(0, 0, I_f).Rot(x, -\theta_4).Rot(y, -\theta_3).Trans(0.0036, -0.0021, 0.053) \]  \hspace{1cm} (16)

\[ T_6 = Trans(0, 0, I_f).Rot(x, -\theta_4).Rot(y, -\theta_3).Trans(0.0036, 0.0021, 0.053) \]  \hspace{1cm} (17)

\[ T_7 = Trans(0, 0, I_f).Rot(x, -\theta_4).Rot(y, -\theta_3).Trans(0, 0, 0.1).Rot(y, -\theta_2). Trans(-0.00165, -0.00225, 0.046) \]  \hspace{1cm} (18)

\[ T_8 = Trans(0, 0, I_f).Rot(x, -\theta_4).Rot(y, -\theta_3).Trans(0, 0, I_{ig}). Rot(y, -\theta_2). Trans(-0.00165, 0.00225, 0.046) \]  \hspace{1cm} (19)

\[ T_9 = Trans(0, 0, I_f). Rot(x, -\theta_4).Rot(y, -\theta_3).Trans(0, 0, I_{ib}).Rot(y, -\theta_2).Trans(0, 0, I_{ig}).Rot(y, \theta_3) \]  \hspace{1cm} (20)

\[ T_{10} = Trans(0, 0, I_f). Rot(x, -\theta_4).Rot(y, -\theta_3).Trans(0, 0, I_{ib}).Rot(y, -\theta_2).Trans(0, 0, I_{ig}).Rot(y, \theta_3) \]  \hspace{1cm} (21)

where \( T_i \) represents the \( i \)th link center of mass frame transformation matrix.

Link center of mass can be obtained by the following relation

\[ C_i = \begin{bmatrix} 0 \\ T_i \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{bmatrix} + F \] \hspace{1cm} (22)

where \( C_i \) is the \( i \)th link center of mass global coordinates and is the foot sole reference point coordinates relative to the global frame. The robot center of mass is obtained by the following equation

\[ \text{COM} = \sum_{i=1}^{10} C_i \times m_i + c_{\text{upper}} \times m_{\text{upper}} \] \hspace{1cm} (23)

where \( m_i \) is the mass of the \( i \)th link. The kinematic model was constructed on MATLAB/Simulink and validated by inserting a desired trajectory and validating the output trajectory by comparing it to the input/desired trajectory. The COM tracking algorithm showed convergence with a reasonable root mean square error (RMSE) that reached 0.000382 m for \( x \) coordinates, 0.000484 m for \( y \) coordinates, and 0.000148 m for \( z \) coordinates (Figure 2).

**Dynamic model**

A contact model was developed for ZMP measurement based on placing four force sensors (FSRs) at each foot sole to measure the resultant vertical force and calculate the corresponding ZMP location using SimMechanics’ ground contact library. To confirm the validity of the suggested contact model, simulations were carried out to test both the friction and FSR models. The FSR model steady-state response was tested by statically placing an object of known weight on the ground and recording the average reading of an FSR. The offset of the reading for a 50 kg object was found to be 7.95 N and the variance of the reading was found to be 0.4023 N\(^2\), as shown in Figure 3(a), and for a 10 kg object, the offset of the reading was found to be 2.1 N and the variance was found to be 0.4011 N\(^2\), as shown in Figure 3(b).

It is important to test the frequency response of an FSR since the walking cycle is repetitive and the impact forces on the FSR will follow a periodic pattern. A vertical sinusoidal force was made to act on the object to view the
Figure 2. COM trajectory versus time (a) x-coordinate, (b) y-coordinate, and (c) z-coordinate.

Figure 3. Normal force measurement versus time for an FSR (a) 50 kg object and (b) 10 kg object.
results in terms of attenuation and phase delay. A 10-Hz force of amplitude 5 N had a gain of −1.93 dB and a phase delay of 0.63 rad, as shown in Figure 4(c).

The friction model was tested by moving a 1 kg object horizontally on the ground through a linearly varying horizontal force with slope 1 N/s, varying the static/dynamic coefficients of friction values and measuring the resultant resistance force, as shown in Figure 5(a), in which the static coefficient of friction was declared to be 0.7 and dynamic coefficient of friction to be 0.6 and Figure 5(b) in which the static coefficient of friction was declared to be 0.9 and dynamic coefficient of friction to be 0.8.

Walking pattern generation

Analytical approach

The 3D LIPM dynamics (Figure 6) is first analyzed to formulate the basic walking pattern generation algorithm followed by introducing modifications.

By resolving the components of the “push” force, the following equations are obtained

\[
\begin{align*}
f_x &= \frac{x - p_x}{r} f \\
f_y &= \frac{y - p_y}{r} f \\
f_z &= \frac{z}{r} f
\end{align*}
\]

The motion equations of the COM are given by Newton’s second law as follows

\[
\begin{align*}
x'' &= \frac{x - p_x}{r} f \\
y'' &= \frac{y - p_y}{r} f \\
z'' &= \frac{z}{r} f - mg
\end{align*}
\]

By realizing that motion in the vertical direction is constrained to a plane parallel to the x-y plane and lies at a constant distance \( z_c \) from it, the following is deduced

\[
\begin{align*}
x'' &= \frac{g}{z_c} (x - p_x) \\
y'' &= \frac{g}{z_c} (y - p_y)
\end{align*}
\]

By taking Laplace and inverse Laplace transforms for equation (30), the following time domain solution is obtained

\[
\begin{align*}
x(t) &= \left(x(t_i^{(n)}) - p_x\right) \cosh \frac{t}{T_c} + T_c \dot{x}(t_i^{(n)}) \sinh \frac{t}{T_c} + p_x \\
x(t) &= \frac{x(t_i^{(n)}) - p_x}{T_c} \sinh \frac{t}{T_c} + \dot{x}(t_i^{(n)}) \cosh \frac{t}{T_c}
\end{align*}
\]
The same solution can be derived for y-coordinate differential equation (31). In this context, a “walking primitive” needs to be explored. A walking primitive represents the unit upon which the entire walking pattern is based. After a support time \( T_s \) is assumed, a walking primitive can be imagined, as shown in Figure 7, as a 3D LIPM, whose base is fixed to the foot placement point (physically, a predetermined point on the foot sole selected by the gait designer). It can easily be shown that a desired walk primitive could be fully described by its terminal states \((x_f_{\text{desired}}, y_f_{\text{desired}})\) since the initial and final velocities are desired to be equal in magnitude (initial and final \(x\)-component velocities have the same sign while initial and final \(y\)-component velocities have opposite signs) and the initial and final positions are desired to be symmetric about some local axis parallel to the \(y\)-axis and passing through the foot placement point to assure gait consistency.

Using the analytical solution developed for an inverted pendulum earlier and by observing the symmetric nature of the desired walk primitive, the following equation can be deduced

\[
x_f = (x_i - p_x) \cosh \frac{T_s}{T_c} + T_c \dot{x}_i \sinh \frac{T_s}{T_c} + p_x
\]  

(34)
Equation (34) can be replicated easily for $y$-axis. To achieve the desired criteria of the walk primitive mentioned earlier, the initial values can be eliminated from equation (34) and its $y$-replica and replaced with their mathematical desired equivalents as follows

$$x_f \text{ desired } = (p_x - x_f \text{ desired }) \cosh \frac{T_x}{T_c} + T_c \dot{x}_f \text{ desired } \sinh \frac{T_x}{T_c} + p_x$$

$$y_f \text{ desired } = (y_f \text{ desired } - p_y) \cosh \frac{T_y}{T_c} - T_c \dot{y}_f \text{ desired } \sinh \frac{T_y}{T_c} + p_y$$

(35)

(36)

By reorganizing the equations, the following final expression for desired final velocities is obtained

$$[\dot{x}_f \text{ desired }^{(n)}] = \begin{bmatrix}
    (x_f \text{ desired } - p_x)^{(n)} \left( \cosh \frac{T_x}{T_c} + 1 \right) \\
    T_c \sinh \frac{T_x}{T_c}
\end{bmatrix}$$

$$[\dot{y}_f \text{ desired }^{(n)}] = \begin{bmatrix}
    (y_f \text{ desired } - p_y)^{(n)} \left( \cosh \frac{T_y}{T_c} - 1 \right) \\
    T_c \sinh \frac{T_y}{T_c}
\end{bmatrix}$$

(37)

Several walking primitives are connected back to back by reversing the sign of the $y$-component after each support exchange to account for the exchange between right and left feet. It is crucial to determine the relationship between the robot’s foot placements at each instant of time in terms of $x$ and $y$ spacing as follows

$$[p_x^{(n)}] = \begin{bmatrix}
    p_x^{(n-1)} + s_x^{(n)} \\
    p_y^{(n-1)} - (-1)^n s_y^{(n)}
\end{bmatrix}$$

(38)

Equation (38) simply indicates how the sequence of $N$ footsteps can be represented mathematically, having the $x$-coordinate of the $n$’th footstep equal to the summation of the $x$-coordinate of the last footstep and the step length $s_x$ taken by the robot. The same can be deduced for the $n$’th $y$-coordinate with the only difference being the step sign flipping at each instant to represent the support exchange between the right and left legs, respectively

$$[y_f \text{ desired }^{(n)}] = \begin{bmatrix}
    p_x^{(n)} + \frac{s_x^{(n+1)}}{2} \\
    p_y^{(n)} + \frac{(-1)^n s_y^{(n+1)}}{2}
\end{bmatrix}$$

(39)

Equation (39) shows that the desired final displacement of the center of mass trajectory along local $x$-axis is equal to half the magnitude of the step plus the $x$-coordinate of the foot placement point. It also shows a similar expression for the desired final displacement of COM trajectory along local $y$-axis with the exception of the sign, which flips at each instant of time to reflect support exchange. The equations derived for the LIPM show that the desired situation is not guaranteed to happen and the best hope would be to try to optimize the trajectory to follow the desired behavior as much as possible. The only variable that could be used to minimize the error between the desired and actual final conditions is the foot placement location. By modifying the foot placement position and considering it as an optimization variable, a better solution can be obtained that could achieve the requirements of a consistent gait. An objective function can be formulated as follows

$$x(t) = \left( x(t^{(n)}) - p_{x,\text{opt}} \right) \cosh \frac{t}{T_c} + T_c \dot{x} \left( t^{(n)} \right) \sinh \frac{t}{T_c} + p_{x,\text{opt}}$$

$$\dot{x}(t) = \left( x \left( t^{(n)} \right) - p_{x,\text{opt}} \right) \sinh \frac{t}{T_c} + \dot{x} \left( t^{(n)} \right) \cosh \frac{t}{T_c}$$

$$A = q \left( x_f \text{ desired }^{(n)} - x_f^{(n)} \right)^2 + r \left( \dot{x}_f \text{ desired }^{(n)} - \dot{x}_f^{(n)} \right)^2$$

(40)

(41)

(42)

The optimal foot placement can be computed by obtaining the partial derivative of the objective function $A$ with respect to the foot placement $p_{x,\text{opt}}$ and equating the partial derivative with zero then finally solving for $p_{x,\text{opt}}$. The standard LIPM approach is very computationally cost efficient and is ideal for many walking scenarios, but there are some considerations that limit the proposed algorithm’s success rate such as

- Double support phase is ignored and support exchange is jerky
- The proposed algorithm does not consider the locomotion cost of transport, the biped robot has a limited capacity battery, and therefore, energy consumption optimization is an important factor that is ignored in the formulation.

The model assumes an ideal condition in which support exchange occurs instantly, which is an impossible situation physically speaking. In a real-world situation, even with joint angle trajectory smoothening, such instantaneous exchange would cause the robot to highly decelerate at instants of support exchange and most likely fall. This problem can be solved by introducing a double support region in the COM trajectory with a predetermined double support period between each two walk primitives. Analyzing this would lead to the conclusion that a COM profile has to be inserted into the gap between two walk primitives whose initial conditions are the final position and final velocity of the walk primitive number $(n-1)$ and whose terminal conditions are the initial position and velocity of the walk primitive number $(n)$, which means we could have
four equations. This suggests using a fourth order polynomial having four constant coefficients. By solving four equations in four unknowns, these coefficients can easily be determined. The addition of the double support phase profile in between the original walk primitives would increase the gait time since the total gait time will be divided between single support and double support phases. To obtain a natural gait, the double support phase period should not be higher than 0.1 times of the single support phase period.

\[
xt(t) = a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \quad (43)
\]

By taking the derivative of the polynomial, the velocity profile can also be obtained. The second limitation to the LIPM approach is ignoring the robot’s power consumption. The target in the standard LIPM model is to make the base of the pendulum (also known as the ZMP) centered at a specific point on the foot sole. In truth, this is not the case for human natural walking. The natural ZMP trajectory is not so stiff and the foot placement point on the foot sole does not have to stick to one place during the walk primitive as long as it remains within the area of the support polygon. Therefore, a reasonable approach toward reducing the torques exerted by the robot motors (and hence power consumption) would suggest relaxing the constraints on the foot placement point. The foot placement point represents the point through which the kick force acts on the COM (pendulum end). Therefore, manipulating the position of the foot placement would also mean shifting the point of intersection of the kick force with the horizontal ground plane. One way to achieve this without modifying the pendulum concept is shifting the base of the pendulum beneath the ground, as shown in Figure 8, which represents the pendulum base shifted beneath the ground by \(Z_p\), while maintaining kick force line of action within ground support polygon (i.e. kick force intersection point with the ground plane changes location within the support polygon).

Due to the base shift, the mathematical model of the inverted pendulum will be slightly modified as follows:

\[
\dot{x} = \frac{g}{z + Z_p} x \quad (44)
\]

Taking Laplace/inverse Laplace transforms yields as follows:

\[
x(t) = x(0) \cosh \frac{t}{T_c} + T_c \dot{x}(0) \sinh \frac{t}{T_c} \quad (45)
\]

\[
\dot{x}(t) = \frac{x(0)}{T_c} \sinh \frac{t}{T_c} + \dot{x}(0) \cosh \frac{t}{T_c} \quad (46)
\]

\[
T_c = \sqrt{\frac{z + Z_p}{g}} \quad (47)
\]

The value of \(Z_p\) must be restrained to obtain a reasonable trade-off between stability and energy optimization. To do this, a suitable area contained within the support polygon is to be chosen and has to have its boundaries slightly far away from the support polygon boundaries to create a safety margin because if the ZMP approaches the support polygon boundaries the robot will be marginally stable. The chosen area is called the “safe zone,” as shown in Figure 9.

After selecting the \(x\) and \(y\) boundaries of the safe zone relative to the local frame of the walk primitive, a suitable \(Z_p\) value can be obtained by triangle symmetry as follows:

\[
\frac{Z_p}{x_{\text{bound}}} = \frac{z}{|x(0)|} \quad (48)
\]

\[
Z_p = \frac{z \times x_{\text{bound}}}{|x(0)| - x_{\text{bound}}} \quad (49)
\]

The same equation can be used for \(y\)-axis in the same manner. The upgraded LIPM algorithm can be formulated, as shown in Algorithm 1.
Numerical approach

Another way of interpreting the biped robot dynamics is by representing it in terms of a simpler model known as the cart table model. In this model, a cart with mass runs back and forth on top of a massless table. The table can keep its balance by pushing the cart forward and backward. The cart responds by an equal and opposite force causing the table to maintain balance by a positive counter-moment that overcomes the negative moment caused by the cart weight. The mass represents the robot’s mass and its location defines the location of the robot COM. The kinematic states of the table also represent the kinematics states of the COM (Figure 10).

The cart table model case resembles that of a point mass located at a specific height from the ground, the table tries to balance itself by applying a force on the cart as follows

\[ F_{\text{push}} = m \ddot{x} \]  

\[ \tau_{\text{reaction}} = -F_{\text{push}} \times z_c = -m \ddot{x} \times z_c \]  

\[ \tau_{\text{weight}} = mg \times (x - p_x) \]  

The ZMP is the point at which all moments vanish, therefore

\[ \tau_{\text{zmp}} = -m \ddot{x} \times z_c + mg \times (x - p_x) = 0 \]  

By eliminating common factors and rearranging equation variables, the ZMP location can be calculated as follows

\[ p_x = x - \frac{z_c}{g} \]  

A very valuable advantage of the cart table model is that it does not need to cut down the walking pattern into a series of walking primitives and even does not need to treat single support and double support phases as separate cases. The walking pattern flow can be continuous without breakdowns. The cart table model in its pure mathematical form is not useful and needs to be treated in a different way. To
do this, the principles of modern control theory are used to create a state-space model instead of solving the model using classical methods.

\[
\frac{d}{dt} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \quad \text{(55)}
\]

\[
p = \begin{bmatrix} 1 & 0 & -z_c \\ g & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \quad \text{(56)}
\]

It can be easily shown that the above state-space system is controllable and observable by evaluating the controllability and observability matrices. A servo control system is inefficient in terms of time response. This initiates the need for a dynamic programming approach to optimize a linear system under definite constraints. Model predictive control (MPC) is chosen since MPC can adapt to sudden future set-point changes, which happen to be the case during the support feet exchange. A loop in which a set of arbitrary control actions are fed to the cart table model and the output ZMP trajectory is compared to desired ZMP trajectory continues until the error (in the form of an objective function) is minimized globally. Only then, the resultant set of control actions is then deemed optimal (Figure 11).

First, the relationship between the present and future values of the states of the system is to be determined by discretizing the state-space model.

\[
\begin{bmatrix} x_{t+1} \\ \dot{x}_{t+1} \\ \ddot{x}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ \dot{x}_t \\ \ddot{x}_t \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^3}{6} \\ \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} u \quad \text{(57)}
\]

Algorithm 2. CTM_WPG1 \((\Delta t, N, \dot{x}_{com}, \dot{x}_{com}, \ddot{x}_{com}, \ddot{x}_{com}, T, p^{\text{ref}})\):

1: \(U_{\text{optimal}} = [], X_0 = [x_{com}, \dot{x}_{com}], Q = 5000000, R = 1\)

2: FOR \(k\) IN \(1\) TO \(\frac{T}{\Delta t}\)
   \(F = [], U = [], X = X_0\)
   FOR \(i\) IN \(k\) TO \(k + N\)
   
   \[
   X = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ \dot{x}_t \\ \ddot{x}_t \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^3}{6} \\ \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} u_t
   \]
   \(f_t = [1, 0, \frac{-\Delta t}{g}] X\)
   \(U = [U, u_t]\)
   \(F = [F, f_t]\)
   END

\[
J = \sum_{i=k}^{N} (Q(p_t^{\text{ref}} - F)^2 + RU_t^2)
\]

\(u_{\text{optimal}} = \arg\min_u J\)

s.t. \(x_{\text{min}} \leq x \leq x_{\text{max}}, |u| \leq u_{\text{max}}\)

\(U_{\text{optimal}} = [U_{\text{optimal}}, u_{\text{optimal}}(0)]\)

\[
X_0 = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} X_0 + \begin{bmatrix} \frac{\Delta t^3}{6} \\ \frac{\Delta t^2}{2} \\ \Delta t \end{bmatrix} u_{\text{optimal}}(0)
\]

END

![Figure 12](image-url)

**Figure 12.** ZMP reference trajectory: (a) without double support and (b) with double support.
The second step is to formulate the optimization problem’s objective function. The optimization problem’s goal would be minimizing the deviation of the actual ZMP trajectory from the desired ZMP trajectory as well as minimizing the overall input jerk to the system

$$ J = \sum_{i=1}^{N} \left\{ Q(p_i^{\text{ref}} - p_i^{\text{actual}})^2 + Ru_i^2 \right\} $$

where $Q$ and $R$ are the performance and effort weight matrices, respectively, and $N$ is the prediction horizon. The last step is to determine the optimization problem constraints. The only constraints on the system are state constraints and input jerk constraints. The states have no specific constraints except the margins of the walking area (room or building) and since the robot workspace is naturally considered very small relative to room space, these constraints can be ignored and left for the gait designer to evade. Also, the input jerk to the system has an upper bound in terms of magnitude. This constraint can be easily described as follows

$$ |u| \leq u_{\text{max}} $$

The solution strategy would be to use an optimization toolbox to minimize the objective function $J$ by producing an optimal series of control actions (jerks) that would make the system output (ZMP trajectory) follow the desired pattern. The linear optimization parametric problem was formulated symbolically on MATLAB and CasADi library for linear/nonlinear optimization was used for solving the objective function problem under maximum allowable jerk constraint (Algorithm 2).

Unlike the first approach, the numerical approach is flexible when it comes to double support phase insertion, which can be done by inserting smooth ZMP reference trajectory transitions with finite slopes, as shown in Figure 12.

The criterion that is worth considering is power consumption optimization. The best way to investigate the optimal strategy for this in the present numerical approach is by observing the human gait pattern, such gait is the product of thousands of years of evolutionary natural selection. A study made by the Department of Neurorehabilitation Sciences, Istituto Auxologico Italiano\(^3\)\(^5\) suggests that the COM of the human body follows an almost sinusoidal curve during walking (Figure 13). Indeed, a human body tends to stretch the knee in the middle of the walk and then flex the knees back by the end of the walking cycle. It is therefore necessary to investigate the effect of such motion on the energy consumption and stability. This means that the COM height constraint that was presented in the standard formulation has to be canceled and instead COM height is to be defined as a function of time.

Based on the above, the mathematical model of the cart table can be modified by including a second lift force in the $Z$-direction

![COM path in the sagittal plane.](image)
F\text{\_lift} = m\ddot{z} \quad (61)

The moment due to the reaction of the cart to the lift force becomes

\[ \tau_{\text{reaction2}} = F_{\text{lift}} \times (x - p_x) = m\ddot{z} \times (x - p_x) \quad (62) \]

Therefore, the summation of moments acting about the ZMP becomes

\[ \tau_{\text{zmp}} = mg(x - p_x) + m\ddot{z}(x - p_x) - m\ddot{z}x = 0 \quad (63) \]

By eliminating common factors and rearranging the equation

\[ p_z = x - \frac{\ddot{z}x}{g + \ddot{z}} \quad (64) \]

The state-space representation of the updated model can now be formulated as follows

\[
\begin{bmatrix}
  x_{t+1} \\
  \dot{x}_{t+1} \\
  \ddot{x}_{t+1} \\
  z_{t+1} \\
  \ddot{z}_{t+1}
\end{bmatrix} =
\begin{bmatrix}
  1 & \Delta t & \frac{\Delta t^2}{2} \\
  0 & 1 & \Delta t \\
  0 & 0 & 1 \\
  1 & \Delta t & \frac{\Delta t^2}{2} \\
  0 & 1 & \Delta t
\end{bmatrix}
\begin{bmatrix}
  x_t \\
  \dot{x}_t \\
  \ddot{x}_t \\
  z_t \\
  \ddot{z}_t
\end{bmatrix} + \begin{bmatrix}
  \frac{\Delta t^3}{6} \\
  \frac{\Delta t^2}{2} \\
  \Delta t \\
  \frac{\Delta t^3}{6} \\
  \frac{\Delta t^2}{2}
\end{bmatrix} \begin{bmatrix}
  u_x \\
  u_z
\end{bmatrix}\]

\[ p_z = x_t - \frac{\ddot{x}_t z_t}{g + \ddot{z}_t} \quad (66) \]
To satisfy the energy optimization criteria, the work done by the push forces is to be minimized. This can be done by introducing an effort weight matrix that may have many forms. Here, the overall work done is to be minimized by minimizing the input jerk to the system. The performance weight is assigned to the minimization of the ZMP error as in the previous standard cart-table walking algorithm. Therefore, the objective function can be reformulated as follows

\[
J = \sum_{i=1}^{N} \left( Q(p_i^{\text{ref}} - p_i^{\text{actual}}) \right)^2 + [u_{xi} \ u_{zi}] R \begin{bmatrix} u_{xi} \\ u_{zi} \end{bmatrix} \]  

(67)

New constraints need to be added to the algorithm, these will include restrictions on the allowable COM height range based on robot reach, also restrictions on the maximum allowable jerk in z-direction. Total system constraints can be introduced as follows

Figure 16. (a–e) Joint torques for closed-form approach walking scenario.
\[ z_{\min} \leq z \leq z_{\max} \]  
\[ |u_{z,z}| \leq u_{\text{max}_{z,z}} \]

The numerical walking pattern generator is upgraded by considering these findings (Algorithm 3). The walking pattern generation algorithm is used as part of a fully developed walking engine, where the obtained COM trajectory required to keep the robot dynamically stable during walking is fed to the inverse kinematics module and verified by the COM tracking forward kinematics module. An iterative optimization technique in which the engine moves back and forth between inverse and forward kinematics is used to reduce the error between desired and actual COM trajectory below a certain threshold till the threshold is no longer violated. The optimized angular trajectory is then fed to the robot motors after second-order filtration.

### Results

To validate the efficiency of the proposed algorithms against their standard counterparts and to compare the two approaches suggested in this article against each other in terms of stability, power consumption, and computational effort. A simulation scenario for NAO walking straight forward on an even terrain was conducted using SimMechanics. The walking pattern used is shown in Figure 14, which displays the footprints of the robot’s support foot during walking. Simulation parameters are given in Table 1.

The closed-form algorithms were used to generate the walking pattern. The RMSE for \( x \)-coordinate ZMP reached 0.0648 m for the standard closed-form approach and 0.0388 m for the upgraded closed-form approach, as shown in Figure 15(a), while RMSE for \( y \)-coordinate ZMP reached 0.0321 m for the standard closed-form approach and 0.0253 m for the upgraded closed-form approach.

### Table 2. Closed-form joint torques.

| Joint | \( \theta_1 \) | \( \theta_2 \) | \( \theta_3 \) | \( \theta_4 \) | \( \theta_5 \) |
|-------|------------|------------|------------|------------|------------|
| \( T_{\text{rms}} \) (N.m) - Standard | 2.4727 | 2.8505 | 0.5887 | 1.0524 | 1.5107 |
| \( T_{\text{rms}} \) (N.m) - Upgraded | 2.4145 | 2.9411 | 0.6475 | 1.3769 | 1.6843 |
| \( T_{\text{peak}} \) (N.m) - Standard | 10.7939 | 12.1434 | 2.8151 | 6.0327 | 9.2043 |
| \( T_{\text{peak}} \) (N.m) - Upgraded | 10.1144 | 11.3820 | 2.9342 | 7.6316 | 10.0117 |

---

Figure 17. (a–c) COM trajectories for closed-form approach walking scenario.
m for the upgraded approach, as shown in Figure 15(b) \((Z_p = 0.5, \quad T_{up} = 1)\).

The joint torques of both the standard and upgraded closed-form approaches were evaluated and compared, as shown in Figure 16. Torque RMS and peak for each joint are given in Table 2.

The results show, contrary to expectations, that the ZMP following accuracy increased in the upgraded approach because relaxing the ZMP constraints reduced ZMP spikes.

Also, the upgraded approach did not contribute much to power consumption optimization since only the first joint RMS and peak torques decreased while all the other joint RMS and peak torques increased slightly. This shows that ZMP relaxation did not necessarily reduce the power consumption in the actual multibody model but rather reduced the effort of some joint motors at the expense of increasing the effort of other joint motors, all while increasing the accuracy of ZMP following and hence enhancing robot stability. The generated COM trajectories of the robot using both approaches are shown in Figure 17.

Next, the numerical generation algorithms were used to generate the same walking pattern separately. The RMSE for x-coordinate ZMP reached 0.0635 m for the standard numerical approach and 0.0164 m for the upgraded numerical approach, as shown in Figure 18(a), while the RMSE for y-coordinate ZMP reached 0.0279 m for the standard numerical approach and 0.0233 m for the upgraded numerical approach, as shown in Figure 18(b). The cart table model parameters are given in Table 3.

These results show that the upgraded numerical approach is relatively superior in terms of ZMP following accuracy and that the upgrades have no stability compromise side effect; on the contrary, the ZMP following was improved.

The joint torques of both the standard and upgraded numerical approaches were evaluated and compared, as shown in Figure 19. The torque RMS and peak for each joint are given in Table 4.

Observations show that the upgraded numerical approach is far more superior to the standard one in terms of power consumption optimization since the torque RMS values recorded are a strong indicator of RMS current draw and hence electrical power. The torque RMS value was reduced by almost 15\% for the first joint, 18.7\% for the second, 23\% for the third, 34.4\% for the fourth, and 60.39\% for the fifth. Also, the peak torques were reduced, which suggests that the upgraded approach would help optimize motor sizing process for bipedal humanoids in general. The generated COM trajectories of the robot using both approaches are shown in Figure 20. The upgraded approach allows more freedom of motion in the vertical direction as assumed. Also, the vertical motion of the COM in the upgraded approach resembles a sinusoid as modern clinical

| Table 3. Cart table model parameters. |
|--------------------------------------|
| Parameter                             | Range          |
| Prediction horizon (N)                | 100            |
| Performance weight \((Q - diagonal element value)\) | 50,000,000    |
| Effort weight \((R - diagonal element value)\) | 1              |
| Maximum absolute jerk \((u)\)         | 100            |
| Maximum COM height \((z_{max})\)      | 0.1978 m       |
| Minimum COM height \((z_{min})\)      | 0.1484 m       |
| Sampling time \((\Delta t)\)          | 0.05 s         |
research predicted, which provides further assurance regarding the validity of the assumptions presented earlier.

Finally, the closed-form and numerical upgraded approaches are compared against each other. The simulations show that the upgraded numerical approach is superior to the upgraded closed-form approach in terms of ZMP trajectory following since it has lower RMSE in both axes. Also, the upgraded numerical approach is superior to both the standard and upgraded closed-form approaches in terms of power consumption optimization since it managed to

---

**Table 4. Numerical approach joint torques**

| Joint | \( \theta_1 \) | \( \theta_2 \) | \( \theta_3 \) | \( \theta_4 \) | \( \theta_5 \) |
|-------|----------------|----------------|----------------|----------------|----------------|
| \( T_{\text{rms}} \) (N.m)-Standard | 2.4631 | 2.6666 | 0.6578 | 1.7014 | 1.7691 |
| \( T_{\text{rms}} \) (N.m)-Upgraded | 2.0958 | 2.1674 | 0.5059 | 1.1156 | 0.70066 |
| \( T_{\text{peak}} \) (N.m)-Standard | 10.4377 | 11.8453 | 3.0402 | 7.1833 | 9.7538 |
| \( T_{\text{peak}} \) (N.m)-Upgraded | 4.1693 | 4.6581 | 1.1440 | 5.3107 | 3.4652 |

---

**Figure 19.** Joint torques for numerical approach walking scenario.
achieve less torque values for joints 1, 2, 3, 5 and almost equal torque values for joint 4, proving, overall, that it can offer better power optimization and stabilization capabilities than a closed-form solution. The only drawback for the upgraded numerical solution is its computational expense. The computational time for the upgraded numerical approach reached 77.388005 s, while the computational time for the upgraded closed-form approach reached 6.723688 s (both approaches were tested using the same computer and same MATLAB software version).

### Conclusion

Solution-dependent modifications were introduced for the basic walking pattern generation model to enhance their performance. The results for different comparisons made between closed form and numerical algorithms as well as their suggested modifications were displayed. The numerical approach shows general dominance in terms of stability and ease of formulation as well as energy consumption optimization. Introducing biologically verified modifications to the algorithm formulation related to COM height manipulation during walking drastically improved the performance and effort minimization. The closed-form solution used a ZMP relaxation strategy that managed to increase stability but did not contribute much to power consumption, which gives the upper hand to the upgraded numerical approach in that aspect. The numerical approach makes it easier for the gait designer to generate a gait since no separate phases need to be handled differently unlike the closed-form approach which separates single and double support phases. However, the superiority of the numerical approach comes at the expense of a lengthy computational time (almost 12 times that of the closed-form approach), which suggests its usage for robots with heavy-duty expensive computer boards and whose link masses and speed are likely to put weight on the energy optimization criteria. The lighter upgraded closed-form approach can be used for robots with small masses and sizes with relatively humble computational capabilities.

Figure 20. COM trajectories for numerical approach walking scenario.
Declaration of conflicting interests
The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs
Samer A Mohamed https://orcid.org/0000-0002-5721-8866
Shady A Maged https://orcid.org/0000-0001-8641-9985

References
1. Ortigoza RS, Marcelino-Aranda M, Ortigoza GS, et al. Wheeled mobile robots: a review. IEEE Lat Am Trans 2012; 10(6): 2209–2217.
2. Hashimoto S, Narita S, Kasahara H, et al. Humanoid robots in Waseda University—Hadaly-2 and WABIAN. Auton Robot 2002; 12(1): 25–38.
3. Chen Y and Hu H. Internet of intelligent things and robot as a service. Simul Model Pract Theory 2013; 34: 159–171.
4. Al-Taae MA, Al-Nuaimy W, Muhsin ZJ, et al. Robot assistant in management of diabetes in children based on the Internet of things. IEEE Internet Things J 2016; 4(2): 437–445.
5. Kim BC. A Internet of Things (IoT) based exploration robot design for remote control and monitoring. J Digit Converg 2015; 13(1): 185–190.
6. Sung J, Griner RE, and Christensen HI. Domestic robot ecology. Int J Soc Robot 2010; 2(4): 417–429.
7. Albani D, Youssef A, Suriani V, et al. A deep learning approach for object recognition with NAO soccer robots. In: Robot world cup. Cham: Springer. 2016. pp. 392–403.
8. Mota T and Sridharan M. Commonsense reasoning and knowledge acquisition to guide deep learning on robots. In: Bicchi A, Kress-Gazit H, and Hutchinson S (Eds), Robotic science and systems 2019. https://doi.org/10.15607/RSS.2019.XV.077
9. Kanda T, Hirano T, Eaton D, et al. The experimental humanoid robot H7: a research platform for autonomous behaviour. Philos Trans Royal Soc A: Math Phys Eng Sci 2007; 365(1850): 11–19.
10. Nishiwaki K, Kuffner J, Kagami S, et al. The experimental humanoid robot WABIAN—walkin by the biped walking robot WL-10 R. In: Proceedings of the second international conference on information systems and electrical engineering (ICITSEE), Yogyakarta, Indonesia, 1 November 2017. pp. 217–222. IEEE.
11. Li TH, Ho YF, Kuo PH, et al. Natural walking reference model. J Theoret Biol 2012; 310: 55–79.
12. Suzuki Y, Nomura T, Casadio M, et al. Intermittent control with ankle, hip, and mixed strategies during quiet standing: a theoretical proposal based on a double inverted pendulum model. Int J Robot Res 2015; 34(11): 159–171.
13. Sovero SE, Saglam CO, and Byl K. Passive frontal plane stabilization in 3D walking. Submitted to IROS. 2015.
14. Kuindersma S, Deits R, Fallon M, et al. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. Auton Robot 2016; 40(3): 429–455.
15. Koolen T, Bertrand S, Thomas G, et al. Design of a momentum-based control framework and application to the humanoid robot atlas. Int J Hum Robot 2016; 13(01): 1650007.
16. Hirose M and Ogawa K. Honda humanoid robots development. Philos Trans Royal Soc A: Math Phys Eng Sci 2007; 365(1850): 79–107.
17. Capi G, Nasu Y, Barolli L, et al. Real time gait generation for autonomous humanoid robots: a case study for walking. Robot Auton Syst 2003; 42(2): 107–116.
18. Feng JC, Gu LZ, and Su JB. The hand tracking for humanoid robot using Camshift algorithm and Kalman filter. J Shanghai Jiaotong Univ-Chin Ed 2006; 40(7): 1161.
19. Kajita S, Kanehiro F, Kaneko K, et al. Biped walking pattern generation by a simple three-dimensional inverted pendulum model. Adv Robot 2003; 17(2): 131–147.
20. Kovalev A, Pavliuk N, Krestovnikov K, et al. Generation of walking patterns for biped robots based on dynamics of 3D linear inverted pendulum. In: International conference on interactive collaborative robotics, Istanbul, Turkey, 20 August 2019, pp. 170–181. Cham: Springer.
21. Pristovani RD, Henri BE, Sanggar D, et al. Walking strategy model based on zero moment point with single inverted pendulum approach in “T-FLoW” humanoid robot. In: 2017 2nd International conferences on information technology, information systems and electrical engineering (ICITISEE), Yogyakarta, Indonesia, 1 November 2017. pp. 217–222. IEEE.
22. Olcay T and Özkurt A. Design and walking pattern generation of a biped robot. Turk J Electr Eng Comput Sci 2017; 25(2): 761–769.
23. Suzuki Y, Nomura T, Casadio M, et al. Intermittent control with ankle, hip, and mixed strategies during quiet standing: a theoretical proposal based on a double inverted pendulum model. J Theoret Biol 2012; 310: 55–79.
24. Ayres RU and Miller SM. Robotics: applications and social implications. Bethesda, MD: World Future Society. 1983.
25. Ayres RU and Miller SM. Robotics: applications and social implications. Bethesda, MD: World Future Society. 1983.
26. Ayres RU and Miller SM. Robotics: applications and social implications. Bethesda, MD: World Future Society. 1983.
27. Ayres RU and Miller SM. Robotics: applications and social implications. Bethesda, MD: World Future Society. 1983.
30. Sun G, Wang H, and Lu Z. A novel biped pattern generator based on extended ZMP and extended cart-table model. *Int J Adv Robot Syst* 2015; 12(7): 94.

31. Park S and Oh J. Real-time continuous ZMP pattern generation of a humanoid robot using an analytic method based on capture point. *Adv Robot* 2019; 33(1): 33–48.

32. Mohamed SA, Awad MI, and Maged SA. Online gait generation for NAO humanoid robot using model predictive control. In: 2020 International conference on innovative trends in communication and computer engineering (ITCE), Aswan, Egypt, 8 February 2020, pp. 205–209. IEEE.

33. Siegler S, Chen J, and Schneck CD. The three-dimensional kinematics and flexibility characteristics of the human ankle and subtalar joints—part i: kinematics. *J Biomech Eng* 1988; 110: 364–373.

34. Danilov I, et al. ZMP trajectory from human body locomotion dynamics evaluated by kinect-based motion capture system. In *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Rome, Italy, pp. 160–166. VISAPP. DOI: 10.5220/0005726001600166

35. Tesio L and Rota V. The motion of body center of mass during walking: a review oriented to clinical applications. *Front Neurol* 2019; 10: 999.

Appendix

Variable A

\[ A = \sqrt{(Y - F_y - h_m)^2 + \left(Z - F_z - l_{fr} - \frac{h_m}{\cos \left(\frac{\theta}{4}\right)}\right)^2} \]