

## Article

# Precise Calculation of Inverse Kinematics of the Center of Gravity for Bipedal Walking Robots

İsmail Hakkı Şanlıtürk \*  and Hikmet Kocabaş

Department of Mechanical Engineering, Faculty of Mechanical Engineering, Istanbul Technical University, 34485 İstanbul, Turkey; kocabash@itu.edu.tr

\* Correspondence: sanliturk18@itu.edu.tr

**Abstract:** The walking of humanoid robots is dependent on the precise tracking of their center of gravity and foot trajectories. Trajectory tracking is achieved by mobilizing their joints to achieve the correct trajectory. Errors occur because of assumptions on tracking the center of gravity and the foot trajectories. In this study, a numerical algorithm was developed that produces an exact and single kinematic solution in which the center of gravity and foot trajectories can be tracked with the desired precision. The effectiveness of this algorithm was examined with a dynamic simulation and compared with a method given in the literature. The main highlight of this study, using the presented algorithm, is that the robot could walk even if the position of its center of gravity was lower than its hips, resulting in a tracking error that was smaller than that reported in the literature.

**Keywords:** bipedal robot; dynamic walking; open loop



**Citation:** Şanlıtürk, İ.H.; Kocabaş, H. Precise Calculation of Inverse Kinematics of the Center of Gravity for Bipedal Walking Robots. *Appl. Sci.* **2024**, *14*, 3706. <https://doi.org/10.3390/app14093706>

Academic Editor: Alessandro Gasparetto

Received: 3 April 2024

Revised: 23 April 2024

Accepted: 24 April 2024

Published: 26 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Humanoid robots have a similar joint structure to humans and are expected to perform actions usually performed by them. Robots with human-like mobility that do not require changes in work environments in the industry, space, service, and health sectors can increase the quality of life of people. Humanoid robots have advanced enough for use in industrial settings [1–3]. These robots mainly consist of a bottom part, which consists of two legs connected to the pelvis that carry the trunk and is responsible for locomotion, and an upper body, which is connected to two arms and contains a central computer, sensors, and a power supply, to perform the tasks assigned to the robot. This study focuses on the locomotion of the bottom part, consisting of two legs and a pelvis piece.

Humanoid robots move via gravity, contact forces, and torques applied to their joints [4]. In the first step, a plan is made for gait and steps in the process of walking [5–7]. In this process, the trajectories of the center of gravity (CoG) of the robot and its feet are obtained. When it is expected to walk based on its CoG, the inverted pendulum method is used [8]. The cart-table model is used for walking based on foot positions [9]. In both methods, the zero-moment point (ZMP) is used as the point of contact between the fixed foot and the floor [10]. Inverse kinematics methods are used to move the limbs to the desired position in the workspace, whereas forward kinematics methods are used to move them in the joint space [11]. In the second step, inverse kinematics calculations are performed so that the robot can follow these trajectories. When the foot and pelvis positions are known, joint angles can be calculated using the inverse kinematics method, either analytically or numerically [12–17].

However, this does not guarantee the tracking of the CoG of the robot at the desired precision level. The CoG is in the middle of the connection points of the two legs of the robot or at a higher position [18–25]. The CoG is calculated before the robot starts its motion. Throughout this motion, the difference between the positions of the pelvis and the CoG is assumed to be fixed [9]. Another assumption is that the CoG of the robot is fixed in reference to the supporting leg [26].

When the CoG is not calculated precisely, the deviations in the CoG caused by contact forces prevent the locomotion of the robot [27]. The methods used in the literature to solve this are listed in the following paragraph.

In a previous study, the feet of the robot were increased in size [28]. To increase the accuracy of calculation, in addition to CoG and foot trajectories, analyses requiring the hip position and trajectory and orientation trajectories were carried out [20,29]. When the weight of the foot constitutes a higher proportion of the total weight of the robot, the foot that is in motion has a greater influence on the locomotive dynamics of the robot. To mitigate this effect, the method of modeling the moving foot as a pendulum was used. Nevertheless, there were still errors in CoG tracking and the movements of the robot were restricted due to the aforementioned assumptions [30].

Applications aim to ensure that the controller moves the robot within the desired tolerances against these errors. The smallest achievable error was found to be 0.18 m with the inverse dynamics control method [31] and 0.03–0.4 m with the non-linear optimization control method [32].

In this study, for the precise tracking of the CoG, an iterative method of calculating joint angles was developed, which can calculate the CoG at a precision of 0.001 m or lower, resulting in a single solution. There is no need to select the optimum solution among multiple; only the CoG and the trajectories of the feet are sufficient.

The novelty of this study is that the precision of the CoG calculation error can be determined before the robot starts moving. Rather than analyzing and eliminating the cause of errors in the CoG trajectory by looking at the trajectory resulting from the control cycle, it would be better to examine the error that occurs according to the predetermined error in establishing cause–effect relationships.

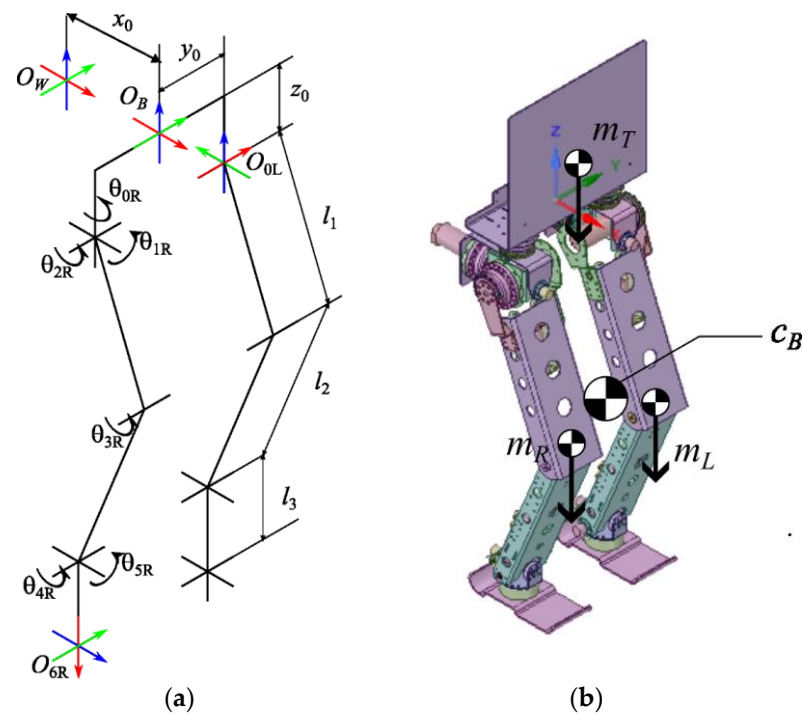
The rest of this paper is organized as follows. Section 2 gives information on the robot used in this study. The kinematics and dynamics of the robot are explained in Sections 2.2 and 2.3, respectively. The calculation of the CoG and zero-moment point equations is explained in Section 2.4. The main subject of the study, the algorithm, is explained in Section 2.5 with equations and pseudo-code. Initial values and result trajectories are also given. In Section 2.6, the simulation model and parameters of this study are presented. The results of this study are discussed in Section 3, and its conclusions are given in Section 4.

## 2. Materials and Methods

### 2.1. Robot Structure

The robot reported in this paper is the numerical model of the ITU Biped bipedal walking robot [33]. The rotational axes of 3 harmonic geared (Harmonic Drive, Frankfurt, Germany) direct-current (DC) motors (Maxon, Sachseln, Switzerland) are placed at the hip joint so that they intersect at a single point, while the rotational axes of 2 DC motors (Maxon, Sachseln, Switzerland) connected to screw nuts (Rexroth, Istanbul, Türkiye) are placed at the ankle joint so that they intersect at a single point. The joint types and positions of the robot are shown in Figure 1.

In Figure 1a,  $O_W$  is the inertial coordinate system,  $O_B$  is the base coordinate system,  $x_0$  is the starting point for movement,  $y_0$  is the pelvis length, and  $z_0$  is the pelvis height of the robot. The  $x$  axis is represented in red, and the  $y$  and  $z$  axes are represented in green and blue, respectively.  $O_{0L}$  is the hip point of the left leg,  $O_{6R}$  is the foot coordinate system, and  $\theta_{nR}$  are the joint angles of the right leg. To increase readability, the hip point of the right leg and the joint angles of the left leg are not shown in Figure 1.  $l_1$ ,  $l_2$ , and  $l_3$  are the link lengths of the robot. Values of the length parameters are given in Table 1.



**Figure 1.** Joint angles and basic dimensions of the robot: (a) joint and length parameters; (b) leg indices and approximate CoGs.

**Table 1.** Length parameters.

Parameter Symbol	Value	Unit
$x_0$	−0.14	m
$y_0$	0.13	m
$z_0$	−0.07	m
$l_1$	0.422	m
$l_2$	0.4	m
$l_3$	0.075	m

The approximate CoGs of the legs are given in Figure 1b. Index  $R$  represents the right leg,  $L$  is the left leg, and  $T$  is the torso.  $c_B$  is the CoG of the robot.  $m_T$  in Figure 1b is the weight of the torso, and  $m_R$  and  $m_L$  are the total weights of the right and left legs, respectively.

The CoGs with respect to the link coordinate system and the masses of the body parts used in the simulation are given in Table 2.

**Table 2.** CoGs and masses of the body parts.

Part Name	$c_x$ (m)	$c_y$ (m)	$c_z$ (m)	$m$ (kg)
Torso	0.000006	0.020549	0.157635	2.387902
Hip roll body	0.000003	0.035534	0.0661	2.139072
Hip yaw body	−0.009912	0.002801	−0.0228	0.71518
Upper leg body	0.186095	−0.004384	−0.003505	3.804756
Lower leg body	0.180778	−0.007	0.000426	2.805996
Ankle roll body	−0.000011	−0.001187	0.001606	0.022225
Foot	0.058582	0.000012	0.018663	1.584761

## 2.2. Robot Kinematics

The method developed in this study requires the separate calculation of the CoG of the legs. Because each leg can be represented as a matrix structure, the Denavit–Hartenberg

(DH) [34] method is used for kinematic calculations. The link transformations are presented in Table 3.

**Table 3.** Link transformations according to the DH method.

Joint No.	$\theta$	$d$	$a$	$\alpha$
1	$\theta_0$	0	0	$\pi/2$
2	$\theta_1$	0	0	$-\pi/2$
3	$\theta_2$	0	$l_1$	0
4	$\theta_3$	0	$l_2$	0
5	$\theta_4$	0	0	$\pi/2$
6	$\theta_5$	0	$l_3$	0

Equations (1)–(4) present the link transformation matrices for an inertial reference frame.

$$O_B = \begin{bmatrix} 1 & 0 & 0 & x_0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

The transformation matrix of  $O_{0R}$  according to  $O_W$  is given in Equation (2). The calculation for the left foot is made by writing  $y_0$  in the second row and fourth column.

$${}^W O_{0R} = O_B \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & -y_0 \\ 0 & 0 & 1 & z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

The joint transformation matrix can be written with Equation (3).

$${}^{n-1}T_n = \begin{bmatrix} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & r_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & r_n \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

After these expressions, the position and orientation of the foot can be calculated based on the 0th coordinate system as given in Equation (4).

$${}^{0R}T_{6R} = {}^{0R}T_{1R} {}^{1R}T_{2R} {}^{2R}T_{3R} {}^{3R}T_{4R} {}^{4R}T_{5R} {}^{5R}T_{6R} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

where the left side of the equation is obtained by the placement of the values in the DH matrix. Its right side includes the position and orientation values of the foot obtained in the 0th coordinate system. When the reciprocal of the left side of the equation is taken, the chain from the foot to the hip is obtained. If both sides are multiplied by the matrix  ${}^{5R}T_{6R}^{-1}$ , the transformation matrix from the ankle to the hip can be obtained [35].

As 2 coordinate systems intersect at the ankle, and 3 intersect at the hip, the value of the knee joint is obtained based on the matrix that is formed. According to this, inverse kinematic calculations can be made using Equations (5)–(10).

$$\theta_4 = \sqrt{\left\{ \frac{1 - \cos \theta_4^2}{\frac{[(p_x + l_3)^2 + p_y^2 + p_z^2 - l_1^2 - l_2^2]}{(2l_1 l_2)}} \right\}} \quad (5)$$

$$\theta_5 = \arctan\left(\frac{-p_z}{\sqrt{[(p_x + l_3)^2 + p_y^2]}}\right) - \arctan\left(\frac{l_1 \sin \theta_4}{l_1 \cos \theta_4 + l_2}\right) \quad (6)$$

$$\theta_6 = \arctan\left(\frac{p_y}{-p_x - l_3}\right) \quad (7)$$

$$\theta_2 = \arctan\left(\frac{-\sqrt{1 - (a_x \sin \theta_6 + a_y \cos \theta_6)^2}}{a_x \sin \theta_6 + a_y \cos \theta_6}\right) \quad (8)$$

$$\theta_1 = \arctan\left(\frac{-o_x \sin \theta_6 - o_y \cos \theta_6}{-n_x \sin \theta_6 - n_y \cos \theta_6}\right) \quad (9)$$

$$\theta_3 = \arctan\left(\frac{a_z}{a_x \cos \theta_6 - a_y \sin \theta_6}\right) - \pi - \theta_4 - \theta_5 \quad (10)$$

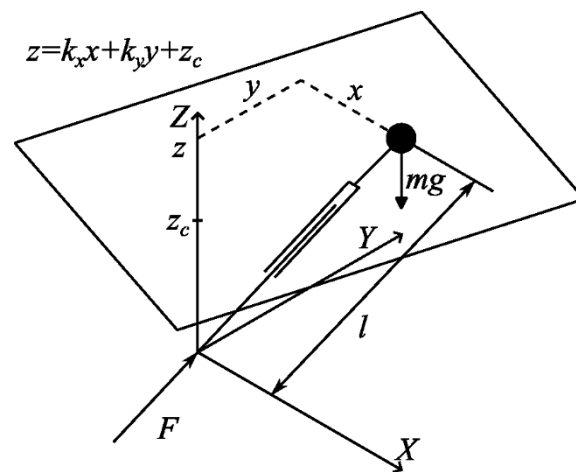
The forward kinematics of the robot can be calculated using Equation (11).

$${}^W O_{0R} \cdot {}^{0R} T_{6R} \quad (11)$$

This expression denotes the orientation and position matrix of the robot in an inertial reference frame according to the joint angles used as inputs.

### 2.3. Robot Dynamics

The method developed in this study uses the reference CoG trajectory. Thus, the dynamics of the robot are obtained using the linear inverted pendulum method. In the 3-dimensional linear inverted pendulum model (3DLIPM) shown in Figure 2, the desired surface equation is reached by setting  $k_x = 0$  and  $k_y = 0$ . As the height of the inverted pendulum,  $z_c$  is taken as a constant.



**Figure 2.** Three-dimensional linear inverted pendulum model (3DLIPM).

After making these arrangements, the differential equations that define the system can be given with Equations (12) and (13).

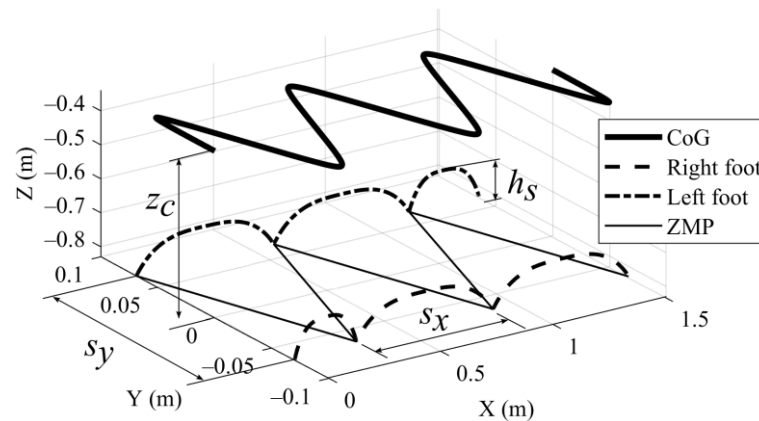
$$\ddot{x} = \frac{g}{z} x \quad (12)$$

$$\ddot{y} = \frac{g}{z} y \quad (13)$$

Step positions are set in a matrix  $p$ , as given in Equation (14). The trajectory of the CoG is obtained as the output.

$$p = \begin{bmatrix} s_{x0} & s_{x1} & \dots & s_{x(n-1)} & s_{xn} \\ s_{y0} & s_{y1} & \dots & s_{y(n-1)} & s_{yn} \\ s_{z0} & s_{z1} & \dots & s_{z(n-1)} & s_{zn} \end{bmatrix} \quad (14)$$

Using Equations (12)–(14), for a step length of 0.3 m ( $s_x$ ), a step width of 0.065 m ( $s_y$ ), and a CoG height of 0.45 m ( $z_c$ ), the trajectory of the CoG is found, as shown in Figure 3. When the foot position is taken as a reference, the height of the hip of the robot in this case is 0.826 m. The step height is 0.075 m ( $h_s$ ). The foot trajectories are continuous up to the second derivation [36].

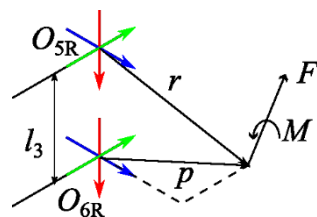


**Figure 3.** Trajectory of the CoG and fixed foot positions.

In Figure 3, the robot starts its movement by taking a step with its right leg. It walks for a total of 6 steps and stops.

#### 2.4. CoG and ZMP Calculations

The ITU Biped has a six-axis force–torque sensor at its ankle (ATI Mini85 F/T Transducer, ATI, Apex, NC, USA). ZMP is indirectly measured using the parameters shown in Figure 4.



**Figure 4.** Parameters used in ZMP calculations.

Because no torque is generated in parallel with the foot plane according to the definition of the ZMP [10], the position of the ZMP based on the reaction on the sensor created by the force and moment at the ZMP is calculated using Equations (15) and (16).

$$p_z = \frac{M_y + l_3 F_z}{F_x} \quad (15)$$

$$p_y = \frac{l_3 F_y - M_z}{F_x} \quad (16)$$

Individual CoGs of the robot bodies are calculated with Equation (18). Before using Equation (18), Equation (17) must be used for calculating positions of the bodies with respect to the inertial coordinate system.

$${}^0T_n = \prod_{n=0}^{n-1} {}^nT_{n+1} \quad (17)$$

$$\mathbf{c}_n = {}^0T_n \begin{bmatrix} c_x \\ c_y \\ c_z \\ 1 \end{bmatrix} \quad (18)$$

In the algorithm, the CoG of the robot is calculated using the CoGs given in Figure 1b in Equation (19).

$$\mathbf{c}_B = \frac{\sum_{n=1}^6 (m_n \mathbf{c}_n)_R + \sum_{n=1}^6 (m_n \mathbf{c}_n)_L + m_T \mathbf{c}_T}{m_R + m_L + m_T} \quad (19)$$

### 2.5. Converging Center of Gravity Algorithm

For the precise calculation of the CoG, joint angles are first calculated using the current positions. Using these angles, the CoG of the robot is calculated using Equation (19). The vector from the current to the reference CoG is found using Equation (20).

$$(\mathbf{G}_{\text{error}})_n = \mathbf{c}_{B\text{ref}} - (\mathbf{c}_{B\text{robot}})_{n-1} \quad (20)$$

Equation (21) is utilized to calculate the unit vector of this vector.

$$(\hat{\mathbf{G}}_{\text{error}})_n = \frac{(\mathbf{G}_{\text{error}})_n}{|(\mathbf{G}_{\text{error}})_n|} \quad (21)$$

Using Equation (21), the hip connected to the foot that is in contact with the floor is moved to a new position by multiplying the unit vector by a certain coefficient  $\beta$  (0.001 in this study).

$$({}^0\mathbf{O}_3)_n = ({}^0\mathbf{O}_3)_{n-1} + \beta \cdot (\hat{\mathbf{G}}_{\text{error}})_n \quad (22)$$

The hip connected to the foot in motion is also moved accordingly. The joint angles and the CoG of the robot are calculated again based on these new hip positions. This process continues until the difference between the reference and current CoGs becomes smaller than the desired value. In our study, the error margin is taken as 0.001 m. The calculation is first made for the fixed foot. Afterward, the indices of the fixed and moving feet are changed, the foot in motion is assumed to be fixed, and the hips are moved. In this way, the masses and positions of all robot components are utilized.

The pseudo-code of the process is given in Algorithm 1.

In this method, the error between the reference and current CoGs is first calculated.

Figure 5 displays the graphical representation of the legs of the robot. The right leg is shown in blue, and the left leg is shown in red. The CoG at the beginning of the trajectory calculations is called the current CoG and is shown as a red circle. The reference CoG is called the target CoG and is shown as a green circle.

The result of running the algorithm given in Algorithm 1 is shown as the blue track in Figure 5. The walking trajectory is obtained by running the algorithm for all values on the trajectory.

**Algorithm 1.** Converging center of gravity algorithm (CCG)

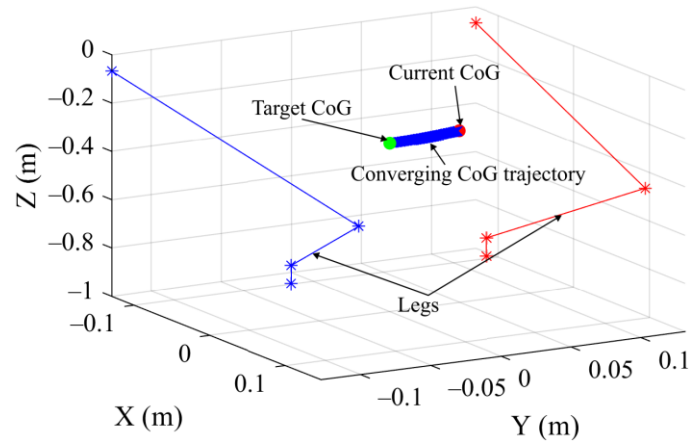
---

```

1:  fixed  $\leftarrow 1$ 
2:  moving  $\leftarrow 2$ 
3:  For each point in CoG_Traj
4:     $\mathbf{G}_{\text{ref}} \leftarrow \text{point (1)}$ 
5:     $\mathbf{p}_{\text{feet}} \leftarrow \text{point (2)}$ 
6:     $\mathbf{G}_{\text{error}}$ 
7:    While  $|(\mathbf{G}_{\text{error}})_n| > \epsilon$ 
8:       $\mathbf{c}_B$  (Equation (19))
9:       $\mathbf{G}_{\text{error}}$  (Equation (20))
10:      $\hat{\mathbf{G}}_{\text{error}}$  (Equation (21))
11:      $({}^0\mathbf{O}_{3\text{fixed}})_n = ({}^0\mathbf{O}_{3\text{fixed}})_{n-1} + \beta \cdot (\hat{\mathbf{G}}_{\text{fixed}})_n$  (Equation (22))
12:      $({}^0\mathbf{O}_{3\text{moving}})_n = ({}^0\mathbf{O}_{3\text{fixed}})_n - (-1)^{\text{moving}} 2[0 \ y_0 \ 0]$ 
13:      $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$  (for left and right legs) (Equations (5)–(10))
14:      $\mathbf{c}_B$  (Equation (19))
15:      $\mathbf{G}_{\text{error}}$  (Equation (20))
16:     If  $|(\mathbf{G}_{\text{error}})_n| < \epsilon$ 
17:       Save joint angles
18:     Else
19:        ${}^B\mathbf{O}_{0R}, {}^{0R}\mathbf{T}_{6R}$  (Equation (11))
20:       Continue cycle
21:     End if
22:     transitory  $\leftarrow j$ 
23:      $j \leftarrow i$ 
24:      $i \leftarrow \text{transitory}$ 
25:   End while
26: End for

```

---

**Figure 5.** Graphical presentation of CCG algorithm.**2.6. Simulation Model**

A simulation model is used to avoid mechanical rigidity, gearbox backlash, and signal delay issues and to obtain the effectiveness of the CCG algorithm. A dynamic model in which the limb angles calculated using this algorithm are used as inputs is obtained as in Figure 6 using MATLAB R2023b SimScape MultiBody [37]. The foot positions in the gait of the robot are predetermined. To define contact with the floor, the “Spatial Contact Force” [38] block of SimScape MultiBody that defines contact between spherical and planar surfaces is used. A total of 4 contact spheres are added at the corners on both feet of the robot with a radius of 0.01 m. The “Inertia Sensor” component in MATLAB Multibody [39] is used to measure the actual CoG of the mechanism in the simulation. The parameters that are used in this study are presented in Table 4.



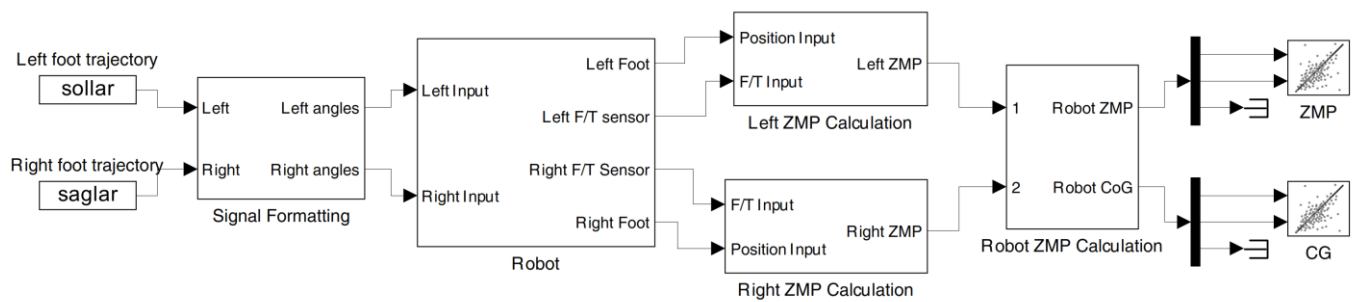


Figure 6. Simulink model diagram.

Table 4. Study parameters.

Parameter	Value	Unit	Explanation
$s_x$	0.3	m	Step length
$s_y$	0.065	m	Distance between feet
$h_s$	0.075	m	Step height
$r_{sphere}$	0.01	m	Contact sphere radius
$k_{contact}$	$6.00 \times 10^4$	N/m	Contact rigidity
$b_{contact}$	$6.00 \times 10^3$	Ns/m	Contact dampening ratio
$G_{prec}$	0.001	m	Precision of calculation
$\beta$	0.001		Unit vector magnitude coefficient

The joint angles of the left and right legs are input to the model. The model applies trajectories to the robot. ZMP and CoG are calculated in the model. Since the floor is flat and the CoG height of the robot is constant in the simulation, only the  $x$  and  $y$  components of the ZMP and CoG are plotted.

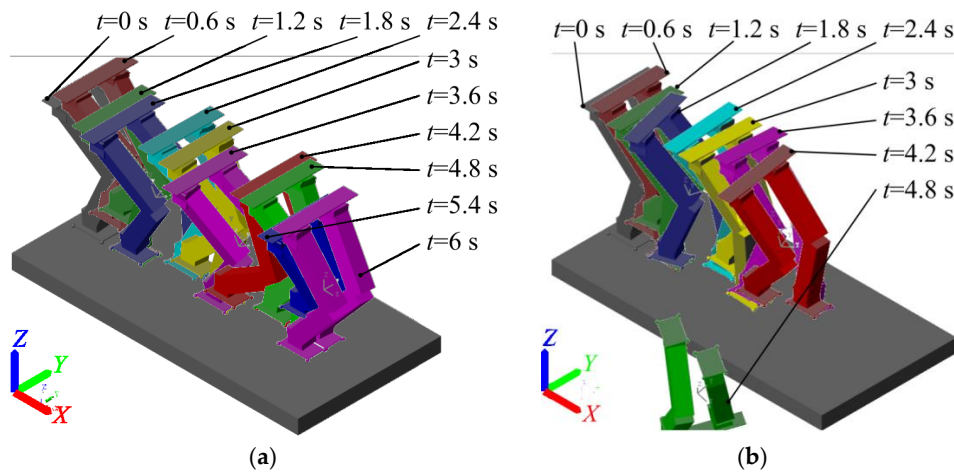
### 3. Results

Two simulations that use the same walking parameters are run. The CCG algorithm is used in the first simulation. Joint angles are calculated with the CCG algorithm for precise CoG tracking in each position in the trajectory. In the second simulation, the CoG is calculated before the movement starts. The vector is assumed to be constant between the pelvis and the CoG. The CCG algorithm is compared to the method given in [9]. Other methods restrict the movement of the robot and require additional trajectories. Both simulations are run as an open loop. Neither a balancer nor a controller is used.

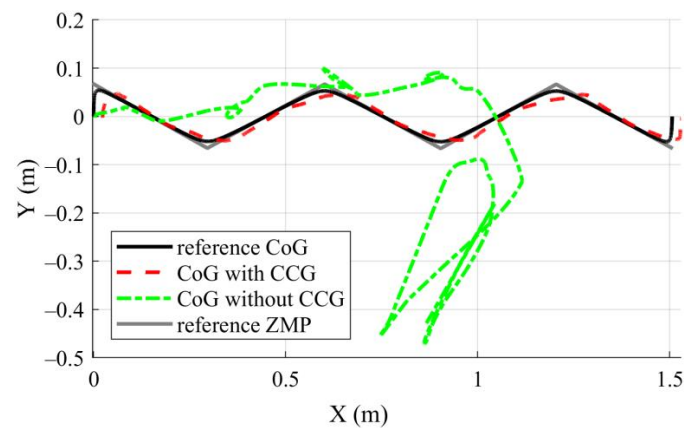
The timespan of the simulation is divided into 10 equal intervals, as shown in both Figure 7a,b. A screenshot of the simulation is taken and overlaid on the previous moment at the start of the intervals. The leftmost pose is the starting pose of the simulation in each panel in Figure 7. The robot moves forward in the positive  $x$  direction, which is represented by the red arrow in Figure 7.

The poses of the robot with and without CCG are presented in Figure 7. As seen in (a), the robot is able to follow the trajectory without falling when CCG is used. When CCG is not used, as shown in (b), the robot is unable to preserve its stance and falls by losing its balance after taking its fourth step. The fall of the robot in the fourth step can be seen as an inverted robot pose in (b).

Figure 8 shows the reference ZMP and CoG trajectory, as well as a comparison of the trajectories of the CoG obtained with and without CCG. While the result of the simulation run with CCG is compatible with the reference CoG, the trajectory calculated without CCG starts to deviate after the second step and results in the robot falling in the fourth step. The starting points of motion are very close to each other. There is sliding in the feet during motion as only four points contact the floor. This explains the errors observed in the trajectories along the  $x$  and  $y$  axes.



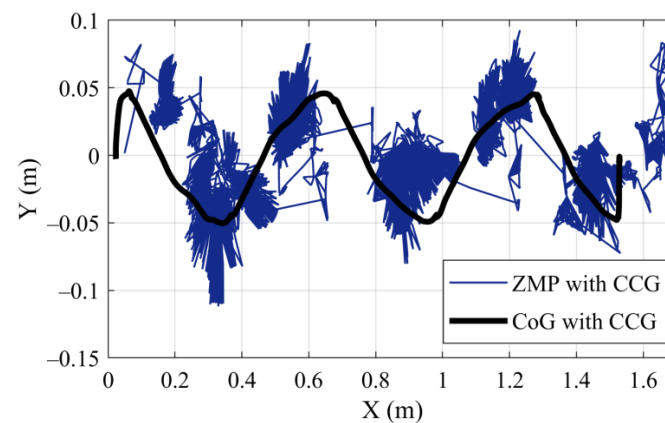
**Figure 7.** Simulation steps: (a) the CCG, (b) constant pelvis to reference CoG.



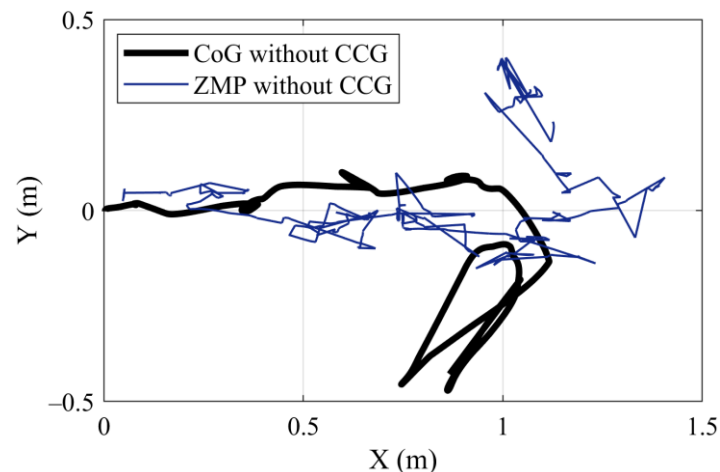
**Figure 8.** Calculated and simulated trajectories.

The calculation times of this method are also obtained. In a computer equipped with Windows 10 Pro 22H2, Core i7 2700, 16 GB of memory, and 240 GB SSD hardware, running MATLAB R2023b, the calculation time for the first value on the trajectory is 0.001662 s, while it is 0.000414 s for the following calculations. This shows that it is possible to make calculations 2415 times per second. Accordingly, the calculation method that is developed here can calculate joint angles according to a given CoG and foot trajectories at the desired level of precision in the real-time control of bipedal walking robots.

The results of the simulation with and without CCG are given in Figures 9 and 10, respectively.



**Figure 9.** Result of simulation with CCG.

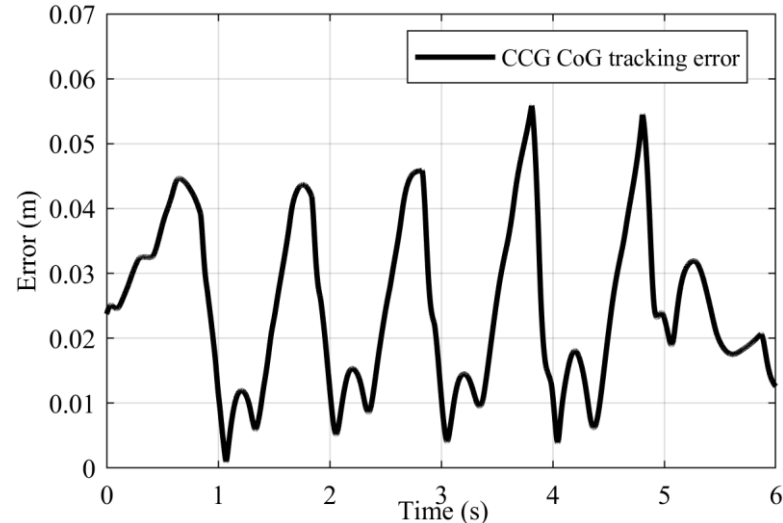


**Figure 10.** Result of simulation without CCG.

The ZMP values and CoG trajectory are in good agreement, as shown in Figure 9, which presents the result of the simulation run with CCG. Due to the hysteresis in contact forces, there are deviations in the ZMP values. However, because these values remain within the support geometry, the robot does not fall and can complete its walk.

On the other hand, in the result of the simulation run without CCG in Figure 10, the CoG and ZMP values are compatible during the first two steps but deviate afterwards, and the trajectory showing the robot fall is seen in the fourth step as a result of the disruption of this compatibility in the third step.

In addition, CoG tracking error values of the CCG algorithm are illustrated in Figure 11.



**Figure 11.** CoG tracking error values of CCG algorithm.

Figure 11 displays the errors between the calculated and simulated CoG values. The greatest error value was 0.0559 m, the smallest error value was  $9.6353 \times 10^{-4}$  m, and the average error value was 0.0238 m.

#### 4. Conclusions

The variation in limb positions using the unit vector of the vector between the target and reference centers of gravity, rather than joint angles, allows for the precise determination of the CoG. Using this method, the robot can walk on an even surface without needing any feedback or control.

There is no need for the CoG of the robot to be higher than the hip level. Therefore, there is no need to design the legs as very light components and the trunk as a heavy component. Using the CCG algorithm, the robot can walk even if its CoG is lower than its hips.

This method has advantages over those in the literature mentioned in this article. The model developed in the study allowed for the tracking of the CoG trajectory with a 0.0238 m error in the numerical simulation. This value is lower than 0.18 in [31] and lower than 0.03–0.4 m in [32].

The CCG algorithm can be used with any position-controlled bipedal robot. The only inputs required are the reference feet and CoG trajectories.

The CCG algorithm can also be used in situations where the CoG trajectory needs to be regenerated precisely before the movement of the robot starts.

**Author Contributions:** Conceptualization, İ.H.Ş.; methodology, İ.H.Ş.; software, İ.H.Ş.; validation, İ.H.Ş.; formal analysis, İ.H.Ş.; investigation, H.K.; resources, İ.H.Ş.; data curation, İ.H.Ş.; writing—original draft preparation, İ.H.Ş.; writing—review and editing, H.K.; visualization, İ.H.Ş.; supervision, H.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Optimus (Robot). Available online: [https://en.wikipedia.org/wiki/Optimus\\_\(robot\)](https://en.wikipedia.org/wiki/Optimus_(robot)) (accessed on 15 March 2024).
2. Agility Robotics. Available online: <https://Agilityrobotics.Com/> (accessed on 15 March 2024).
3. Kaneko, K.; Kaminaga, H.; Sakaguchi, T.; Kajita, S.; Morisawa, M.; Kumagai, I.; Kanehiro, F. Humanoid Robot HRP-5P: An Electrically Actuated Humanoid Robot with High-Power and Wide-Range Joints. *IEEE Robot. Autom. Lett.* **2019**, *4*, 1431–1438. [\[CrossRef\]](#)
4. Sugihara, T.; Morisawa, M. A Survey: Dynamics of Humanoid Robots. *Adv. Robot.* **2020**, *34*, 1338–1352. [\[CrossRef\]](#)
5. Rokbani, N.; Cherif, A.; Alimi, A.M. Toward Intelligent Biped-Humanoids Gaits Generation. In *Humanoid Robots*; Choi, B., Ed.; Intech: Rijeka, Croatia, 2009; pp. 259–272. [\[CrossRef\]](#)
6. Chevallereau, C.; Razavi, H.; Six, D.; Aoustin, Y.; Grizzle, J. Self-Synchronization and Self-Stabilization of 3D Bipedal Walking Gaits. *Robot. Auton. Syst.* **2018**, *100*, 43–60. [\[CrossRef\]](#)
7. Yin, C.; Zhu, J.; Xu, H. Walking Gait Planning And Stability Control. In *Humanoid Robots*; Choi, B., Ed.; Intech: Rijeka, Croatia, 2009; pp. 297–332. [\[CrossRef\]](#)
8. Tang, Z.; Er, M.J. Humanoid 3D Gait Generation Based on Inverted Pendulum Model. In Proceedings of the 22nd IEEE International Symposium on Intelligent Control, ISIC 2007. Part of IEEE Multi-conference on Systems and Control, Singapore, 1–3 October 2007; pp. 339–344. [\[CrossRef\]](#)
9. Kajita, S.; Kanehiro, F.; Kaneko, K.; Fujiwara, K.; Harada, K.; Yokoi, K.; Hirukawa, H. Biped Walking Pattern Generation by Using Preview Control of Zero-Moment Point. In Proceedings of the 2003 IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; pp. 1620–1626. [\[CrossRef\]](#)
10. Vukobratovic, M.; Borovac, B. Zero-Moment Point—Thirty Five Years of Its Life. *Int. J. Humanoid Robot.* **2012**, *1*, 157–173. [\[CrossRef\]](#)
11. Lator, C.; Xambó-Descamps, S.; Zaplana, I. Robot Kinematics. In *SpringerBriefs in Mathematics*; Springer: Cham, Switzerland, 2018; pp. 75–100. [\[CrossRef\]](#)
12. Maalouf, N.; Elhajj, I.H.; Shammas, E.; Asmar, D. Biomimetic Energy-Based Humanoid Gait Design. *J. Intell. Robot. Syst.* **2020**, *100*, 203–221. [\[CrossRef\]](#)
13. Kobayashi, T.; Sekiyama, K.; Hasegawa, Y.; Aoyama, T.; Fukuda, T. Virtual-Dynamics-Based Reference Gait Speed Generator for Limit-Cycle-Based Bipedal Gait. *Robomech J.* **2018**, *5*, 18. [\[CrossRef\]](#)
14. Alba, M.; Prada, J.C.G.; Meneses, J.; Rubio, H. Center of Percussion and Gait Design of Biped Robots. *Mech. Mach. Theory* **2010**, *45*, 1681–1693. [\[CrossRef\]](#)
15. Lim, I.S.; Kwon, O.; Park, J.H. Gait Optimization of Biped Robots Based on Human Motion Analysis. *Robot. Auton. Syst.* **2014**, *62*, 229–240. [\[CrossRef\]](#)

16. Atmeh, G.; Subbarao, K. A Neuro-Dynamic Walking Engine for Humanoid Robots. *Robot. Auton. Syst.* **2018**, *110*, 124–138. [CrossRef]
17. Hildebrandt, A.-C.; Ritt, K.; Wahrman, D.; Wittmann, R.; Sygulla, F.; Seiwald, P.; Rixen, D.; Buschmann, T. Torso Height Optimization for Bipedal Locomotion. *Int. J. Adv. Robot. Syst.* **2018**, *15*. [CrossRef]
18. Khomariah, N.E.; Pramadihanto, D.; Dewanto, R.S. FLoW Bipedal Robot: Walking Pattern Generation. In Proceedings of the 2015 International Electronics Symposium: Emerging Technology in Electronic and Information, IES 2015, Surabaya, Indonesia, 29–30 September 2015; pp. 73–78. [CrossRef]
19. Park, I.W.; Kim, J.Y.; Lee, J.; Oh, J.H. Online Free Walking Trajectory Generation for Biped Humanoid Robot KHR-3(HUBO). In Proceedings of the 2006 IEEE International Conference on Robotics and Automation, Orlando, FL, USA, 15–19 May 2006; pp. 1231–1236. [CrossRef]
20. Mandava, R.K.; Vundavilli, P.R. Forward and Inverse Kinematic Based Full Body Gait Generation of Biped Robot. In Proceedings of the International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016, Chennai, India, 3–5 March 2016; pp. 3301–3305. [CrossRef]
21. Yang, L.; Liu, Z.; Chen, Y. Bipedal Walking Pattern Generation and Control for Humanoid Robot with Bivariate Stability Margin Optimization. *Adv. Mech. Eng.* **2018**, *10*, 2018. [CrossRef]
22. Olcay, T.; Özkurt, A. Design and Walking Pattern Generation of a Biped Robot. *Turk. J. Electr. Eng. Comput. Sci.* **2017**, *25*, 761–769. [CrossRef]
23. Carpentier, J.; Tonneau, S.; Naveau, M.; Stasse, O.; Mansard, N. A Versatile and Efficient Pattern Generator for Generalized Legged Locomotion. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 3555–3561. [CrossRef]
24. Chae, H.-S. Development of the Modeling for Biped Robot Using Inverse Kinematics. *WSEAS Trans. Syst.* **2004**, *3*, 2788–2792.
25. Tevatia, G.; Schaal, S. Inverse Kinematics for Humanoid Robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Symposia Proceedings, San Francisco, CA, USA, 24–28 April 2000; pp. 294–299. [CrossRef]
26. Efrain, O.; Ponce, R.; Mansard, N.; Souères, P.; Ramos, O.E.; Soû, P. Whole-Body Motion Integrating the Capture Point in the Operational Space Inverse Dynamics Control. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Madrid, Spain, 18–20 November 2014; pp. 707–712. [CrossRef]
27. Shibata, M.; Natori, T. Impact Force Reduction for Biped Robot Based on Decoupling COG Control Scheme. In Proceedings of the 6th International Workshop on Advanced Motion Control, Proceedings (Cat. No.00TH8494), Nagoya, Japan, 30 March 2000–1 April 2000; pp. 612–617. [CrossRef]
28. Sugihara, T. Solvability-Unconcerned Inverse Kinematics by the Levenberg–Marquardt Method. *IEEE Trans. Robot.* **2011**, *27*, 984–991. [CrossRef]
29. Bajrami, X.; Dermaku, A.; Likaj, R.; Demaku, N.; Kikaj, A.; Maloku, S.; Kikaj, D. Trajectory Planning and Inverse Kinematics Solver for Real Biped Robot with 10 DOF-s. *IFAC-PapersOnLine* **2016**, *49*, 88–93. [CrossRef]
30. Buschmann, T.; Lohmeier, S.; Bachmayer, M.; Ulbrich, H.; Pfeiffer, F. A Collocation Method for Real-Time Walking Pattern Generation. In Proceedings of the IEEE-RAS International Conference on Humanoid Robots, Pittsburgh, PA, USA, 29 November 2007–1 December 2007; pp. 1–6. [CrossRef]
31. Nakanishi, J.; Mistry, M.; Schaal, S. Inverse Dynamics Control with Floating Base and Constraints. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 1942–1947. [CrossRef]
32. Clever, D.; Hu, Y.; Mombaur, K. Humanoid Gait Generation in Complex Environments Based on Template Models and Optimality Principles Learned from Human Beings. *Int. J. Robot. Res.* **2018**, *37*, 1184–1204. [CrossRef]
33. Bayraktaroğlu, Z.Y.; Acar, M.; Gerçek, A.; Tan, N.M. Design and Development of the I.T.U. Biped Robot. *Gazi Univ. J. Sci.* **2018**, *31*, 251–271.
34. Denavit, J.; Hartenberg, R.S. A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. *J. Appl. Mech.* **1955**, *22*, 215–221. [CrossRef]
35. Ali, M.A.; Park, H.A.; Lee, C.S.G. Closed-Form Inverse Kinematic Joint Solution for Humanoid Robots. In Proceedings of the IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS 2010—Conference Proceedings, Taipei, Taiwan, 18–22 October 2010; pp. 704–709. [CrossRef]
36. Yapıcı, K.O. 14 Serbestlik Dereceli İki Ayaklı Bir Robotun Dinamik Yürüme Hareketinin Kontrolü. Master Thesis, ITU Science and Technology Institute, İstanbul, Turkey, 2008.
37. Simscape Multibody. Available online: <https://www.mathworks.com/products/simscape-multibody.html> (accessed on 15 March 2024).
38. Spatial Contact Force. Available online: [https://www.mathworks.com/help/sm/ref/Spatialcontactforce.html?SearchHighlight=s%20contact&s\\_tid=srchtitle\\_support\\_results\\_1\\_spatial%20contact](https://www.mathworks.com/help/sm/ref/Spatialcontactforce.html?SearchHighlight=s%20contact&s_tid=srchtitle_support_results_1_spatial%20contact) (accessed on 15 March 2024).
39. Inertia Sensor. Available online: [https://www.mathworks.com/help/sm/ref/Inertiasensor.html?S\\_tid=srchtitle\\_site\\_search\\_1\\_inertia%20sensor](https://www.mathworks.com/help/sm/ref/Inertiasensor.html?S_tid=srchtitle_site_search_1_inertia%20sensor) (accessed on 15 March 2024).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.