

RESEARCH ARTICLE

Variable-time-interval trajectory optimization-based dynamic walking control of bipedal robot

Erman Selim*, Musa Alcı and Mert Altıntaş

Department of Electrical and Electronics Engineering, Ege University, Izmir, Turkey

*Corresponding author. erman.selim@ege.edu.tr

Received: 3 March 2021; **Revised:** 15 August 2021; **Accepted:** 23 August 2021; **First published online:** 27 September 2021

Keywords: bipedal robots, legged locomotion, trajectory optimization, variable time interval, cost of transport.

Abstract

Bipedal robots by their nature show both hybrid and underactuated system features which are not stable and controllable at every point of joint space. They are only controllable on certain fixed equilibrium points and some trajectories that are periodically stable between these points. Therefore, it is crucial to determine the trajectory in the control of walking robots. However, trajectory optimization causes a heavy computational load. Conventional methods to reduce the computational load weaken the optimization accuracy. As a solution, a variable time interval trajectory optimization method is proposed. In this study, optimization accuracy can be increased without additional computational time. Moreover, a five-link planar biped walking robot is designed, produced, and the dynamic walking is controlled with the proposed method. Finally, cost of transport (CoT) values are calculated and compared with other methods in the literature to reveal the contribution of the study. According to comparisons, the proposed method increases the optimization accuracy and decreases the CoT value.

1. Introduction

In human-designed environments, legged robots have the advantage of movement compared to wheeled robots. For this purpose, full-body humanoids have been produced. Asimo by Honda [1], Atlas by Boston Dynamic, [2] and Gazelle by KAIST [3] are the leading robots of this genre. These robots developed by industrial companies and well-established groups have sophisticated electro-mechanical designs. They can successfully mimic human actions such as walking, running, and carrying loads. Providing walking stability is a challenging problem since walking robots have high degrees of freedom as well as mechanical design difficulties and mathematical model discontinuities. Especially, to reveal the potential advantages of two-legged humanoid robots, stable walking must be guaranteed. To simplify the problem, the three-dimensional walking problem is reduced to two-dimensional models [4]. One of the basic walking models which are defined in two dimensions is the compass biped robot with a single control input. Due to its simple mathematical model, it is used in walking control and periodicity analysis. In addition, there are some studies in which passive walking is provided by taking advantage of gravity [5] or applying very little energy [6]. The compass biped robot, which is frequently preferred in theoretical walking studies, can be physically realizable with minor modification by adding a retracting actuator as a knee to avoid the scuffing problem [7, 8]. The Cornell biped robot is one of the pioneers of this genre [9]. Another two-dimensional model that is often preferred is the five-link biped [10]. Despite the lack of movement in the transverse plane, it can exhibit dynamics of walking at the sagittal plane. Therefore, it is an important model for dynamic walking studies from a control perspective [11]. The five-link planar biped robot has five degrees of freedom, four of which have actuators. Since the number of actuators is less than the total number of degrees of freedom, it is classified as an underactuated system. The collision between the foot and the ground at each walking step causes discontinuity in system dynamics. Thus, the mathematical model can be written as a hybrid system [10].

The control of walking robots consists of sequential operations. At the first stage, joint trajectories that provide mechanical constraints and vertical balance can be found as the solution of the optimization problem [12]. The optimization problem, which can be solved in continuous time, is also solved by the finite difference method. The finite difference trajectory optimization can be used in passive walkers where gait cycle determination is important [13]. Solving the optimization problem with a high accuracy contributes to walking control. However, the main weakness of the trajectory optimization method is the long processing time. Online trajectory optimization methods are suggested to shorten the processing time [14]. In addition, methods are proposed in different disciplines to shorten the processing time of trajectory optimization [15]. Instead of finding the desired trajectories with optimization, alternative iterative methods are suggested. A periodic gait modification strategy is proposed to increase walking success [16]. According to the modification-based periodic gait planning method, sufficient conditions are explored with an iterative equation. In this way, the initial trajectories can be converted to more suitable walking trajectories. After calculating the desired trajectories that provide the system dynamics in the first stage, the robot's walking is controlled by trajectory tracking in the second stage.

Hybrid zero dynamic (HZD) is one of the dynamic gait control methods that is applied in underactuated biped robots [4, 16, 17]. In the HZD-based control methods, trajectories of actuated joints are determined by the function based on the underactuated joint. The fact that the trajectories depend on this increasing function instead of time ensures that the motion space of the robot is limited by virtual constraints [18]. In this way, stable walking depending on desired trajectories can be achieved. On the other hand, the discontinuity of the system model can lead to periodic instability. Lyapunov-based methods with HZD control also aim to provide periodic stability for walking [19, 20] and also running [21]. In addition, HZD is combined with feedback linearization, backstepping, and sliding mode methods, making it robust against system uncertainties and disturbances [22–24]. The HZD method can be applied in a wide range from three-link biped robots [23] to humanoid robots [25]. Although remarkable results have been revealed for biped gait control in the literature, studies on new control methods are still ongoing. These methods are mostly supported by simulation [22, 26]. Unfortunately, there is no guarantee that a method that exhibits successful results in the simulation environment would also be successful with the real model.

In this paper, a computationally efficient trajectory optimization method is developed to increase optimization accuracy. The proposed variable time interval trajectory method is compared with the conventional fixed-time interval trajectory optimization. A five-link biped walking robot with four actuators is used in simulations and experiments. The cost of transport (CoT) parameter has been used to compare the contribution of the proposed method with respect to available methods in the literature. It is seen that the presented results are consistent with the proposed methodology.

This paper is organized as follows: Section 2.1 describes mathematical model of five-linked bipedal robot and Section 2.2 explains the prototype design. Section 3 describes the trajectory optimization specific to walking robots. The proposed variable time interval trajectory optimization method is explained in detail in Section 3.2. Section 4 describes how periodic and stable walking is achieved at both global and local controller levels. Section 5 presents simulations, real-time walking results, and walking performances. In particular, Section 5.3 highlights the advantages of the proposed variable time interval trajectory optimization method. Finally, Section 6 summarizes the accomplishments of this work.

2. Five-Link Biped Model

Walking consists of repeating the swing and stance phases of each leg periodically and sequentially. During the transition from swing phase to stance phase, the contact of the swing leg with the ground occurs as a collision. Along the collision, conservation of momentum is valid and recalculation of the velocity vector according to conservation of momentum causes discontinuity in system dynamics. This discontinuity arising from the nature of walking can only be defined with the hybrid system model.

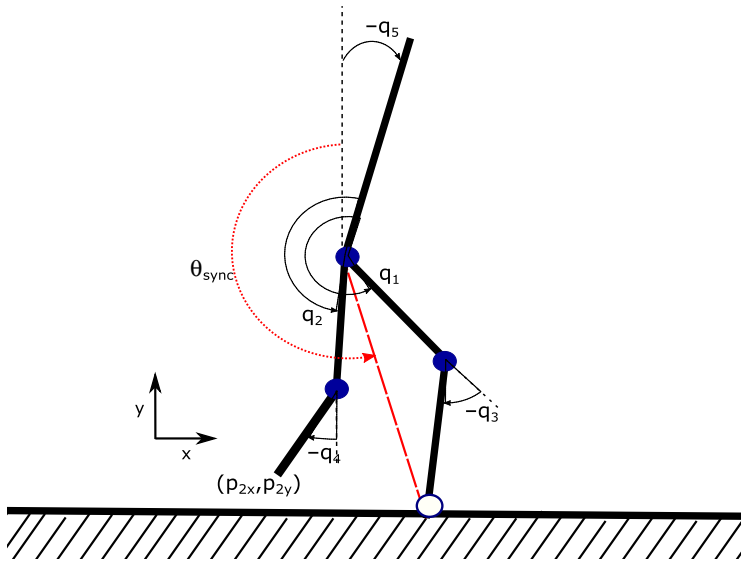


Figure 1. Joint angles of the five-link robot.

2.1 Mathematical model

The mathematical model of the bipedal walking robot can be considered as a hybrid system [27]. This mathematical model, which is frequently used in studies on biped walking control, is also applied as a robot model in this study. Accordingly, the mathematical model consists of two parts named as single support phase and double support phase. In the single support phase section, the swing and stance phases of the legs are defined as a continuous-time ordinary differential equation. In the double support phase section, the collision event is defined as an impact model.

2.1.1. Single support phase model

As shown in Fig. 1, the robot has a point foot and therefore there is no ankle actuator. Joint angles are defined as $q := [q_1, q_2, q_3, q_4, q_5]^T$. In addition to the joint angles, the angle between the hip point and the ground contact point is called θ_{sync} . (See Section 4.1 for the role of the θ_{sync} angle in the gait controller.)

It is assumed that the support leg of the robot does not slip during the single support phase. In this case, the robot can be considered as a serial link manipulator. The center of the gravity of each link can be found via with forward kinematics. Dynamic equations of the biped robot in this phase can be derived in the form of Eq. (1) by Lagrange method [27].

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = Bu \tag{1}$$

where $D(q) \in R^{5 \times 5}$ is the positive defined symmetrical inertial matrix, $C(q, \dot{q}) \in R^{5 \times 5}$ centrifugal and Coriolis components matrix, $G(q) \in R^5$ gravity component vector, and $B := [I_4, 0_{4 \times 1}]^T$ is the constant vector indicating whether there is an actuator. It is seen from vector B that there is no actuator on the torso angle q_5 . $u := [u_1, u_2, u_3, u_4]^T$ is the torque to be applied to the actuators between the torso-femur and femur-tibia. System dynamics are defined as state space form in Eq. (2) where $x := [q^T, \dot{q}^T]^T$.

$$\dot{x} = \begin{bmatrix} \dot{q} \\ D^{-1}(q)[-C(q, \dot{q})\dot{q} - G(q) + Bu] \end{bmatrix} \tag{2}$$

According to the control input, the state space representation of the system dynamics is rewritten as the sum of the nonlinear functions:

$$\dot{x} = f(x) + g(x)u \tag{3}$$

where $f(x) = \begin{bmatrix} \dot{q} \\ D^{-1}(q)[-C(q, \dot{q})\dot{q} - G(q)] \end{bmatrix}$ and $g(x) = \begin{bmatrix} 0_{5 \times 1} \\ D^{-1}(q)Bu \end{bmatrix}$.

2.1.2 Double support phase model

The single support phase ends when the swing foot touches on the ground. The collision occurred at the time of contact causes a discontinuity in the state variables. The post-collision values of the state variables are calculated according to the conservation of the angular momentum [28]. To determine the force acting on the second leg during the collision, the generalized coordinate dynamic equation is derived by adding the endpoint coordinates of the second leg to the state variables.

$$D_e(q_e)\ddot{q}_e + C_e(q_e, \dot{q}_e)\dot{q}_e + G_e(q_e) = B_e u + \delta F_{ext} \tag{4}$$

The force acting on the second leg is assumed to be impulsive. When Eq. (4) is integrated during the instantaneous collision time, the external force F_{ext} can be found as in Eq. (5).

$$D_e(q_e^+)\dot{q}_e^+ - D_e(q_e^-)\dot{q}_e^- = F_{ext} \tag{5}$$

The post-collision speed vector \dot{q}_e^+ is calculated based on the pre-collision speed vector \dot{q}_e^- . Although the robot position does not change during the collision time, the swing and the support legs physically switch. According to this alterations, the new position vector can be written as $q_s^+ := [q_2^- q_1^- q_4^- q_3^- q_5^-]^T$. By derivating the collision model according to the state space state variables, the collision transformation map can be obtained as in Eq. (6)

$$x^+ = \Delta(x^-) := \begin{bmatrix} \Delta_q q^- \\ \Delta_{\dot{q}}(q^-)\dot{q}^- \end{bmatrix} \tag{6}$$

where $x^+ := [q^{+T}, \dot{q}^{+T}]^T$.

According to the single and double support phases, the general model of biped walking is defined in Eq. (7) as a hybrid system.

$$\sum : \begin{cases} \dot{x} = f(x) + g(x)u & x^- \notin S \\ x^+ = \Delta(x^-) & x^- \in S \end{cases} \tag{7}$$

where S is the impact surface.

2.2. Physical model

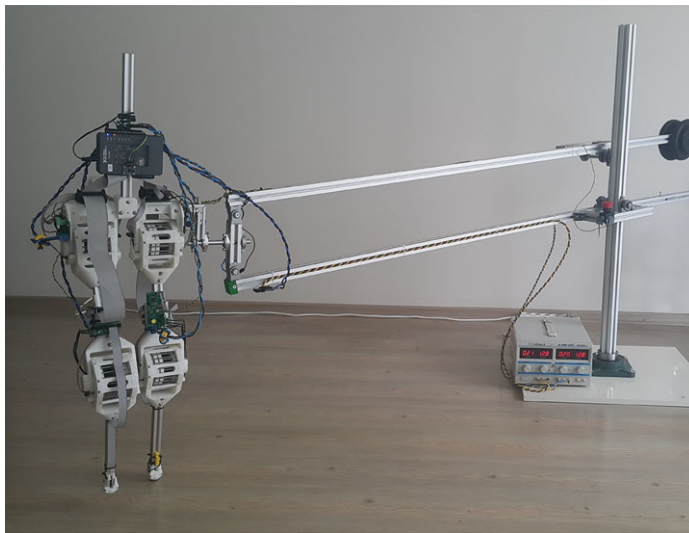
The test platform is designed to exhibit both humanoid walking properties and the mathematical model of the five-link robot. In addition, the carrier platform is used to reduce the three-dimensional motion space to a two-dimensional sagittal plane. The carrier platform consists of two parallel links that can rotate 360° on the tower. At the tip of the carrier platform, there is a connecting rod having a single degree of freedom. The connecting rod always remains horizontal via the parallel link mechanism. Therefore, the workspace of the hip point of the robot is limited to approximately a cylindrical surface (quasi-planar). The designed test robot is shown on the carrier platform as in Fig. 2.

The reference coordinate system of the biped robot is oriented by the torso in accordance with the mathematical model. However, the global angle values of the joints should be known to perform trajectory tracking of the leg endpoints. A BNO055 [29] IMU sensor is used for coordinate transformation where the global position information will be received.

To provide the configuration in Fig. 1, the right and left legs are produced symmetrically. Each leg consists of a series of two links called the femur and tibia whose lengths are 32 cm, and the torso length

Table I. Physical model parameters.

Link definition	Parameter definition	Parameter value
Torso	Mass	$m_T = 1.75 \text{ kg}$
	Total length	$l_T = 0.4 \text{ m}$
	Center of gravity distance	$d_T = 0.18 \text{ m}$
	Moment of inertia	$I_T = 0.0470 \text{ kgm}^2$
Femur	Mass	$m_f = 1.30 \text{ kg}$
	Total length	$l_f = 0.32 \text{ m}$
	Center of gravity distance	$d_f = 0.12 \text{ m}$
	Moment of inertia	$I_f = 0.0158 \text{ kgm}^2$
Tibia	Mass	$m_t = 0.80 \text{ kg}$
	Total length	$l_t = 0.34 \text{ m}$
	Center of gravity distance	$d_t = 0.12 \text{ m}$
	Moment of inertia	$I_t = 0.0115 \text{ kgm}^2$

**Figure 2.** The actual view of the carrier platform and test robot.

is 34 cm. The total weight of the robot is approximately 6 kg. The detailed physical parameters of the robot are shown in Table I.

The robot has four actuators in total, two of them are on the hips and the others are on the knees. To increase the trajectory tracking success of the links, the position of the joints must be precisely controlled. In this way, the joints are designed to apply both high torque and also precise position control. However, conventional reducers are insufficient for precise position control, because a backlash is caused by the gap between the gears.

Cycloid reducers are suitable for precise position control applications due to their low backlash value, compact size, and high conversion rates. According to electro-mechanical properties of the robot, the reducer demand torque is at least 20 Nm. To meet this requirement, the reducer speed reduction rate is determined to be 1:44. The integrated joint is designed taking into consideration the requirements of the biped robot. The motor is positioned inside the gearbox and a direct connection of the input shaft to the eccentric disk is provided. The output disk and cycloid teeth pins are extended to cover the motor. Thus,

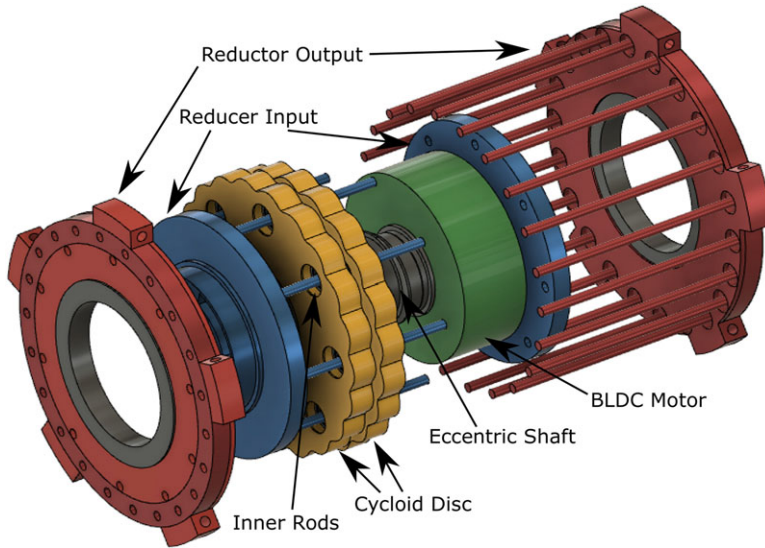


Figure 3. Cad model of the designed joint.

the input and output connections of the joint are positioned symmetrically on both sides. The stable force transfer is provided between the links by a symmetrical input and output connections. A CAD model of the designed joint is shown in Fig. 3.

Reaction forces occurring in applications that interact with the environment can both damage the joint and disrupt trajectory tracking. Therefore, it is advantageous to use series elastic actuators (SEA) in biped applications. SEA offers advantages of low mechanical output impedance, impact forces tolerance, and passive mechanical energy storage [30]. In this design, the torque output of the cycloid reducer is provided with flexible material to the link. The reverse force acting on the output of the reducer by the load creates physical slip in the flexible material. The slip is calculated by measuring the input and output angles of the joint separately, so that the force acting on the joint is estimated depending on the slip. Input and output angles are measured by an As5047 magnetic encoder. A brushless DC (BLDC) motor is preferred as the motion source of the joint due to high speed and high torque advantages as well as precise position control. Those features make the BLDC motor one of the ideal options for robotic applications [31]. In addition, field-oriented control (FOC) is used for joint position control [32]. The driver board is designed and manufactured to provide current up to 30A. According to the experimental measurements, the maximum speed of the joint driven by the BLDC motor is 140 rad/s and the holding torque 20 Nm is observed, within the safe limits. In robotic applications, besides the mechanical structure, the electronic and communication infrastructure affects the system success as well. NI-Myrio is preferred as the central control unit, as it offers a user-friendly interface with LabVIEW. In addition, the coexistence of FPGA and microprocessor increases the processing capability. The central controller communicates with four local controllers and sensor units at high speed. Four magnetic encoders and two motor controllers are connected to one SPI line for one leg. Thanks to the parallel processing capability of the FPGA, the communication period can be completed in less than 150 μ s. The global controller is completed within 400 μ s on the ARM processor. Since the processor and FPGA can work independently, the control and communication periods do not affect each other. Also, each local controller (BLDC motor driver) operates at 170 μ s sampling rate.

3. Trajectory Optimization

The main goal in trajectory optimization is to find control signals that will provide physical constraints and system dynamics with minimal effort. In addition to the constraints arising from the physical model of the robot, gait characteristics are also included in the optimization. Walking period is the primary parameter to be determined in gait characteristics. Considering that the leg length is 0.66 m and the step length is between 0.12 and 0.24 m, the step period is aimed to be 1.30 s. The optimization starts at $t_0 = 0$ s and the step period is applied as the final time $t_f = 1.30$ s. By applying the control signals as an input to the mathematical model, the joint trajectories of the biped robot are generated [12]. Trajectory optimization is formulated in Eq. (8). To eliminate the notation confusion, the system dynamics is written as a single function $\dot{x}(t) = \mathbf{F}(x, t, u) = f(x(t)) + g(x(t))u(t)$

$$\begin{aligned} \min_{u(\tau), x(\tau)} \quad & \int_{t_0}^{t_f} u(\tau)^2 d\tau \\ \text{s.t.} \quad & \dot{x}(\tau) = \mathbf{F}(x, t, u) \\ & x_{min} \leq x(\tau) \leq x_{max} \\ & u_{min} \leq u(\tau) \leq u_{max} \end{aligned} \tag{8}$$

Trajectory optimization is basically considered using two different approaches, namely direct and indirect. Although, indirect methods offer high accuracy, feasibility decreases as the degree of freedom increases. Due to the ease of calculation, direct collocation methods are often preferred [12, 33]. With the help of direct collocation method, continuous trajectory optimization turns into a nonlinear programming (NLP) problem.

3.1. Fixed-time optimization

In any collocation method, collocation points are used to reduce the decision variables to an array with N values instead of infinite values $x(t) \rightarrow [x_0, x_1, \dots, x_N]$, $u(t) \rightarrow [u_0, u_1, \dots, u_N]$. Transcription from continuous dynamics to discrete dynamics is achieved by applying the Simpson quadrature method between each consecutive pair of collocation points. Transcription requires the transformation of both system dynamics and integral form. Simpson’s rule for integration is given in Eq. (9) and system dynamics transformation is given in Eq. (12).

$$\int_{t_0}^{t_f} w(\tau) d\tau \rightarrow \sum_{k=0}^{N-1} \frac{\Delta t}{6} (w_k + 4w_{k+1/2} + w_{k+1}) \tag{9}$$

$$\dot{x} = \mathbf{F} \tag{10}$$

$$\int_{t_k}^{t_{k+1}} \dot{x} dt = \int_{t_k}^{t_{k+1}} \mathbf{F} dt \tag{11}$$

$$x_{k+1} - x_k = \frac{(\mathbf{F}_{k+1} + 4\mathbf{F}_{k+1/2} + \mathbf{F}_k)}{6} \Delta t \tag{12}$$

The Hermite–Simpson method requires a midpoint for each segment. To calculate the $\mathbf{F}_{k+1/2}$ which is derived from the known system dynamic functions $f(x_{k+1/2})$ and $g(x_{k+1/2})$, it is necessary to find the $x_{k+1/2}$. Therefore, a second collocation equation is required. In Eq. (13), midpoint system dynamics are calculated according to Hermite interpolation. Detailed formulation of the Hermite–Simpson direct collocation method can be found in Betts’s book [34].

$$x_{k+1/2} = \frac{1}{2}(x_k + x_{k+1}) + \frac{\Delta t}{8}(\mathbf{F}_k - \mathbf{F}_{k+1}) \tag{13}$$

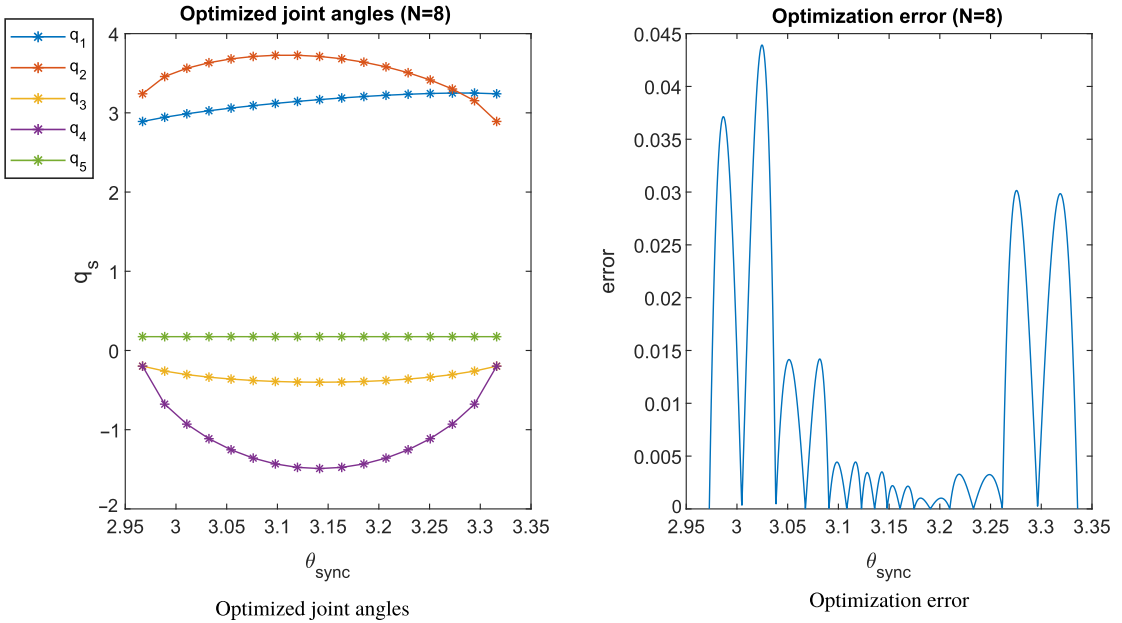


Figure 4. Optimized trajectories (N = 8).

Continuous-time optimization problem written in Eq. (8) is discretized by direct collocation method then rewritten in Eq. (14) as a NLP problem [33].

$$\begin{aligned}
 \min_{\substack{u_0, u_1, \dots, u_N \\ x_0, x_1, \dots, x_N}} & \sum_{k=0}^{N-1} \frac{\Delta t}{6} (u_k^2 + 4u_{k+1/2}^2 + u_{k+1}^2) \\
 \text{s.t.} & \frac{(\mathbf{F}_{k+1} + 4\mathbf{F}_{k+1/2} + \mathbf{F}_k)}{6} \Delta t = x_{k+1} - x_k \\
 & x_{k+1/2} = \frac{1}{2}(x_k + x_{k+1}) + \frac{\Delta t}{8}(\mathbf{F}_k - \mathbf{F}_{k+1}) \\
 & x_{\min} \leq x_k \leq x_{\max} \\
 & u_{\min} \leq u_k \leq u_{\max}
 \end{aligned} \tag{14}$$

Solutions of NLP for N points are obtained with *fmincon* function in Matlab toolbox. To clarify the biped walking robot trajectory design problem, eight collocation points in Fig. 4a are drawn as an example. Graphics are shown in terms of normalized phase variable. Normalization details are given in Eq. (18).

As a result of optimization, state ($\hat{x}(t_k)$), derivative of state ($\dot{\hat{x}}(t_k)$), and control ($\hat{u}(t_k)$) values that provide system dynamics for N points are obtained. Let $\hat{x}(t)$, $\dot{\hat{x}}(t)$, and $\hat{u}(t)$ be the interpolated state, derivative of state, and control values that are calculated according to the optimization result at an arbitrary time t . By evaluating $\hat{x}(t)$ and $\hat{u}(t)$ with actual system dynamics, $\mathbf{F}(\hat{x}, t, \hat{u})$ can be calculated. The difference between $\dot{\hat{x}}(t)$ and $\mathbf{F}(\hat{x}, t, \hat{u})$ gives transcription error $\xi(t)$. Transcription error indicates the optimization solution accuracy. Evaluating the optimization for the collocation points ensures that highly accurate optimization solution for $\xi(t_k)$. On the other hand, high error values for other points can be considered as solution failure.

$$\xi(t) = \left| \dot{\hat{x}}(t) - \mathbf{F}(\hat{x}, t, \hat{u}) \right| \tag{15}$$

As can be seen in the optimization error graph in Fig. 4b, the error value is higher than the average at the endpoints. The reason for the high error values is that these points are the starting or ending points of

Table II. Fixed-time trajectory optimization error values and solution times.

Number of samples	Maximum error $\times 10^{-3}$ [rad]	Optimization solution time [s]
8	44.84	6.22
12	22.56	11.23
18	14.28	29.85
21	6.72	45.44
27	4.23	78.42
36	1.16	143.04
54	0.255	339.45

the contact between the robot and the ground. This indicates that the system dynamics are being moved away at the moment of step. The divergence of targeted trajectories from system dynamics decreases walking performance.

The simplest way to overcome the problems caused by a poor optimization accuracy is to increase the sample point number. However, increasing the number of samples increases the NLP solution time exponentially. According to the fixed-time trajectory optimization method, sampling number, maximum error value, and optimization solution time comparisons are given in Table II. As a solution, variable time interval optimization methods are suggested and tested for various applications [35].

3.2. Variable time optimization

In this study, a method is suggested to increase optimization accuracy by preserving NLP solution time. In the proposed method, collocation points are based on variable time intervals rather than fixed time intervals. While few collocation points are sufficient for low error rate regions, new collocation points should be added in regions with high error rates. According to the applied collocation method, determining the location of the new point increases the accuracy. Increasing the accuracy by adding a minimum number of collocation points is provided by metaheuristic search-based optimization methods.

In this study, finding the critical points to increase the accuracy is acquired by the tree-based breadth-first search optimization method. The tree structure is a graph that branches off from a single point and does not create a loop (). Each node in the tree structure shows a time interval change in trajectory optimization. However, since the optimum sampling time cannot be the same all over the trajectory, there are different levels of solution points on the tree. A new sampling time is determined for each solution point on the tree and new collocation points are found according to the applied collocation method. The tree-based breadth-first search optimization method is illustrated in Fig. 5. The iteration level is denoted by i and the parent number by j . Possible solution points are shown in green since the solution points are not child nodes, there may be unused j numbers at some levels (e.g., $i = 3$).

In this study, Hermite–Simpson method is used as the collocation method. There is a midpoint between the two endpoints in the Hermite–Simpson method. In other words, the sampling time at the starting node of the tree structure is determined as half of the period $\Delta t = (t_f - t_0)/2$. The branching rule has been developed to descend from a node to sub-nodes. According to the branching rule:

- The first collocation point used on the old node will be used as the first collocation point for the new node.
- The midpoint of the old node which is determined according to Hermite–Simpson will be the endpoint of the new node.
- A new midpoint will be calculated according to Hermite–Simpson for the new node.

The time resolution is increased by $3/2$ by this branching rule. Each new node is created according to the branching rule, and the error value is calculated and evaluated whether it is acceptable or not.

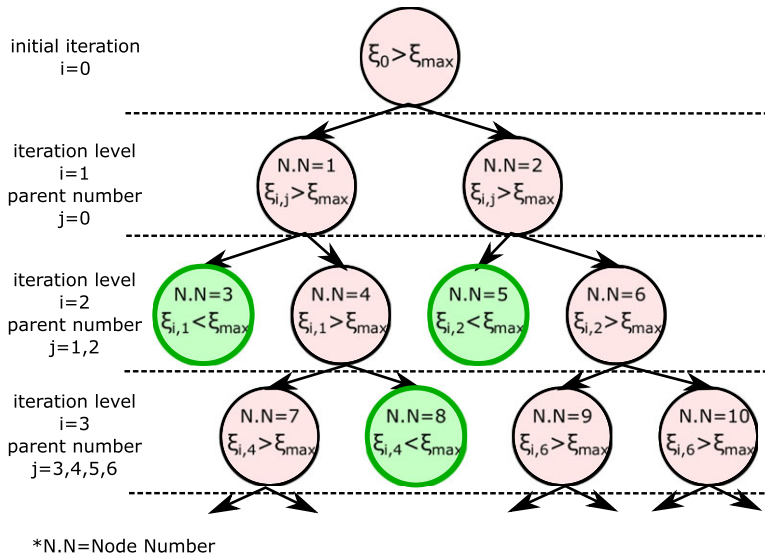


Figure 5. The tree-based breadth-first search optimization illustration.

The branching rule is repeated iteratively until it reaches the acceptable error value. Through iterative breadth-first search, solution points can be found on the tree structure. Figure 6 illustrates how to add a new point. In here, black, blue lines, red, and green dots indicate the system dynamics, optimized trajectories, collocation points, and newly added points, respectively.

As can be seen in Fig. 4 (q4 – swing leg knee joint), the relative joint angle value range found as a result of optimization is less than 1.2 radians. It is aimed that the optimization error does not exceed 0.1% within the specified range; therefore, the maximum acceptable error limit is limited to 1.2×10^{-3} . According to the optimization results by the proposed method, the number of collocation points to achieve this goal is $N = 21$. (In the Hermite–Simpson method, the number of segments is seen to be twofold since there is always a midpoint.) The trajectory optimization within the same number of collocation points is tested to compare the performance of the proposed method. In optimization with the same number of collocation points without changing the time resolution, it is seen that the segment errors extremely exceed the limit error value. The error rates of the proposed variable time trajectory optimization and the fixed-time trajectory optimization method consisting of the same number of segments are compared in Fig. 7. Fixed-time interval trajectory optimization results are shown by blue dashed lines and the proposed method by red lines.

As seen in Fig. 7, the optimization accuracy increases in the moments when the foot touches on the ground and leaves from the ground. It is clear that the proposed variable time interval trajectory optimization method can reduce the error below the threshold level in each segment. In addition, new points added to minimize the error and increase the processing time by a limited amount. The error value which is below the limit value can be reduced at 36 sampling points with fixed-time interval trajectory optimization methods, and in this case the optimization solution time is 143 s. However, with the proposed variable time interval trajectory optimization method, the same limit error value can be reached with 21 sampling points at 51.62 s. Although the same number ($N=21$) of fixed-time trajectory optimization methods can solve it in a relatively short time (45.44 s), it exceeds the error limit considerably (nearly four times more). Measurements performed with *Matlab 2020a* on Intel i7700HQ CPU. The performance comparison of the fixed-time trajectory optimization method and the proposed variable time trajectory optimization method is given in Fig. 8.

In the performance comparison, it can be clearly seen that the variable time trajectory optimization takes less time than the fixed-time trajectory optimization for the same error limit. Reducing the trajectory optimization time from 143.04 to 51.62 s provides a valuable improvement. Unfortunately, both

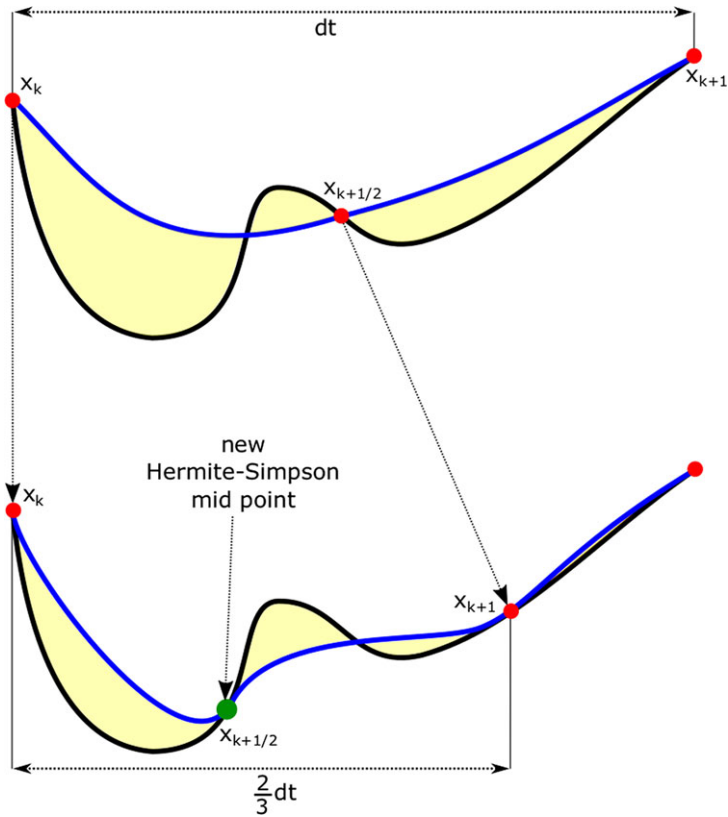


Figure 6. New collocation point illustration.

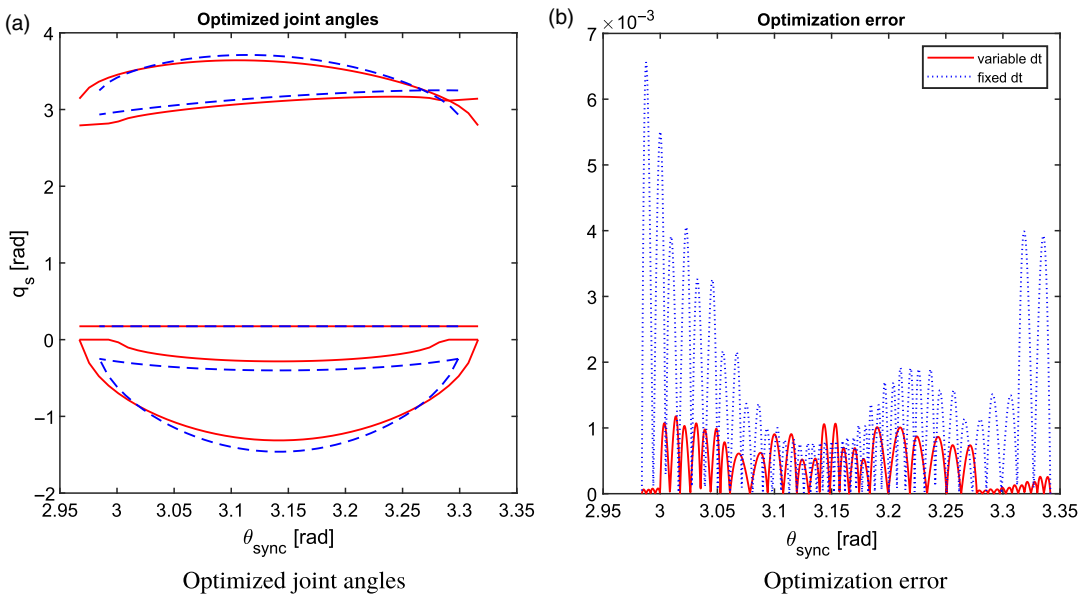


Figure 7. Comparison of the proposed method with fixed-time interval trajectory optimization.

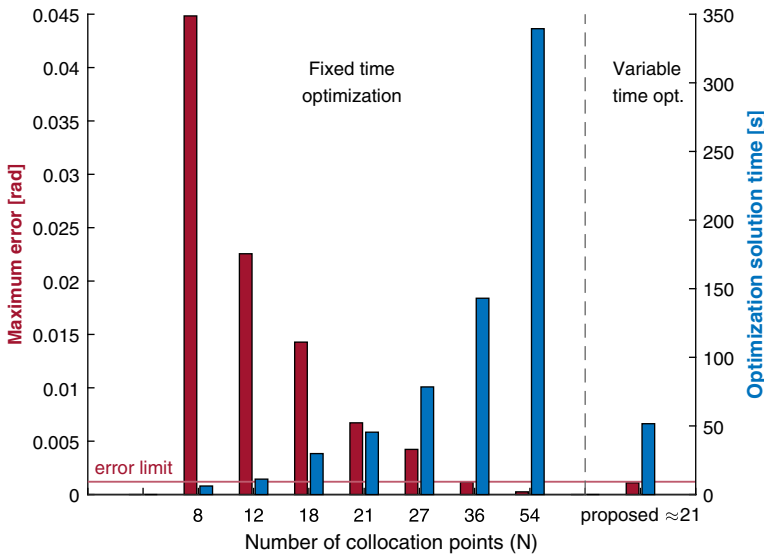


Figure 8. Performance comparison of the proposed method with fixed-time interval trajectory optimization.

classical and proposed trajectory optimization methods cannot perform trajectory optimization within a step period. In addition, since periodic walking is provided, it is not necessary to re-optimize for each step. Trajectory optimization needs to be recalculated only when walking parameters change.

4. Walking Controller

One of the reasons why biped robots cannot be controlled like classic manipulators is that the dynamic model defined in Eq. (1) provides inequality $rank[B(q)] < dim[q]$. Therefore, biped robots are underactuated systems which are not stable and controllable at every point of joint space. It is only controllable on certain fixed equilibrium points and some trajectories that are periodically stable between these points. Accordingly, the controller consists of two parts. The first one is the global controller that performs the virtual constraints method using optimized trajectories previously obtained. The second part is the local controller that provides each joint to follow the desired trajectory. The general schematic of the controller as shown in Fig. 9 consists of both global and local controllers.

4.1 Global controller

In the first stage of walking control, virtual constraints method is applied. By the virtual constraints method, the motion space of joints is limited to desired trajectories. The constraint trajectories are defined by the synchronization angle (θ_{sync}) which is written based on joint angles. The synchronization angle that determines the rhythmic movement of the robot is defined as the angle between the hip and the support foot. In this way, all the joints can perform synchronous movement within the target trajectories.

θ_{sync} is shown with the red line in Fig. 1. The synchronization angle is calculated by summing the joint angles multiplied by the weights. The weight vector $c = [-1, 0, -1/2, 0, -1] \in R^{1 \times 5}$ is obtained from the robot geometry. Synchronization angle is given in Eq. (16). The range of the synchronization angle can be given by the Bézier polynomial as an input by normalizing it between zero and one. Normalization function is given in Eq. (17). The range of the synchronization angle is set around $\pi \pm 10^\circ$ in radians ($\theta_{max} = 3.31, \theta_{min} = 2.96$). Complex walking trajectories found as a result of the optimization are applied with the Bézier polynomial. With the proposed method, optimization is solved using 21 points without

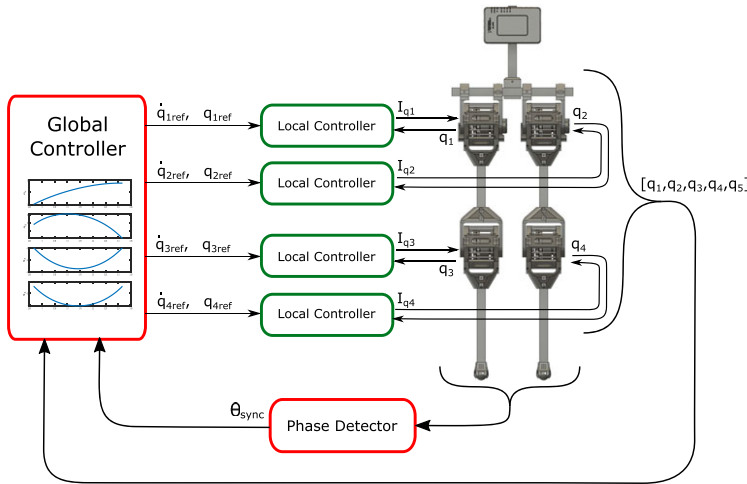


Figure 9. General schematic of the controller.

exceeding the error limit. Each solution point is used as the Bézier coefficient. By connecting solution points with the Bézier polynomial that given at Eq. (18), continuous-time solution of desired trajectories is derived. M is the Bézier polynomial degree. a_k^i is the k numbered polynomial coefficient of the joint i . θ_{norm} is the normalized synchronization angle.

$$\theta_{sync} = cq \tag{16}$$

$$\theta_{norm} = \frac{\theta_{sync} - \theta_{max}}{\theta_{min} - \theta_{max}} \tag{17}$$

$$b_i(\theta_{norm}) = \sum_{k=0}^N a_k^i \frac{M!}{k!(M-k)!} \theta_{norm}^k (1 - \theta_{norm})^{M-k} \tag{18}$$

In the second part, these trajectories are traced by the virtual constraint method. By this method, the motion space of joints is limited to predesigned trajectories [27]. The constraint trajectories are defined by the synchronization angle which is written based on joint angles. The synchronization angle determines the rhythmic movement of the robot. In this way, all joints can perform synchronous movement within the target trajectories.

The robot model is given in Eq. (3) for a single support phase. System outputs are $y \in R^{4 \times 1}$ and desired trajectories are $h_d = [b_1, b_2, b_3, b_4]^T$. The h_d vector is the combination of the Bézier functions that define the desired trajectory of each joint. The system output targeted to be set to zero is written as in Eq. (20).

$$y = h \tag{19}$$

$$y = [q_1 \ q_2 \ q_3 \ q_4]^T - h_d(\theta_{norm}) \tag{20}$$

The feedback that will take the system outputs to zero can be easily acquired by input–output linearization. Relative degree is $r = 2$ due to the structure of mechanical models and since $L_g h = 0$. Output function derivatives are written in Eqs. (21)–(23).

$$\frac{dy}{dt} = \frac{\partial h}{\partial x} \dot{x} \tag{21}$$

$$\frac{dy}{dt} = L_f h(x) \tag{22}$$

$$\frac{d^2y}{dt^2} = L_f^2 h(x) + L_g L_f h(x) u \tag{23}$$

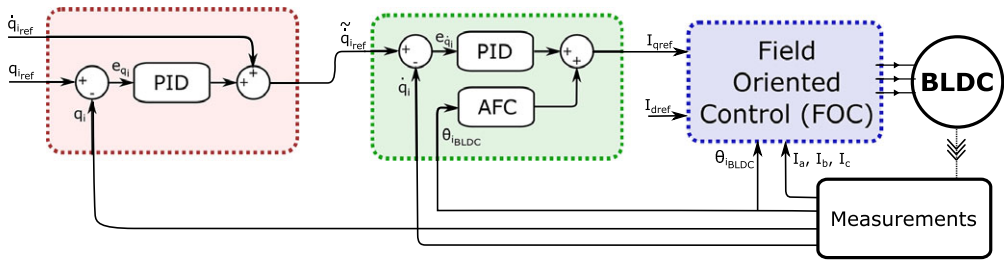


Figure 10. Local controller.

The control signal that can asymptotically take the system output to zero is written as in Eq. (24) according to Eq. (23). The parameter v is the auxiliary control signal and works as a PD controller. K_d and K_p are positive controller gains and ϵ is a tuning parameter which is $0 < \epsilon < 1$. During each step, an impact occurs at the norm of \dot{y} . To reduce this effect, derivative gain should be chosen smaller relative to the proportional gain. Therefore, derivative component of the controller is multiplied by $1/\epsilon$ and proportional component by $1/\epsilon^2$.

$$u = (L_g L_f h)^{-1} (v - L_f^2 h) \tag{24}$$

$$v = -\frac{1}{\epsilon^2} K_p y - \frac{1}{\epsilon} K_d \dot{y} \tag{25}$$

4.2. Local controller

The local controller provides the position control of the joint according to the angle and angular velocity references that are received from the global controls. The local controller function diagram is shown in Fig. 10. The diagram is drawn for joint number i and joint number can vary between one to four. For each local controller, joint angle, joint angular velocity, and BLDC rotor position are denoted by q_i, \dot{q}_i , and θ_{BLDC} , respectively.

With the FOC method, motor currents are divided into a torque component I_q and magnetic flux component I_d . Since the BLDC motor supplies the magnetic flux from the permanent magnets, the I_d current is desired to be zero. Reference I_q current is adjusted according to the demanded torque. To control both joint angle and angular velocity within the certain limits, the cascade controller has been used. The position is controlled at the first stage of the cascade controller as defined in Eq. (28).

$$e_{q_i} = q_{i,ref} - q_i \tag{26}$$

$$\dot{e}_{q_i} = \dot{q}_{i,ref} - \dot{q}_i \tag{27}$$

$$\tilde{q}_{i,ref} = \left(K_p \cdot e_{q_i} + K_I \int_0^t e_{q_i}(\tau) d\tau + K_d \cdot \dot{e}_{q_i} \right) + K_{ff} \cdot \dot{q}_{i,ref} \tag{28}$$

At the second stage of the cascade controller, the torque current is controlled according to the speed. The torque controller stage is tuned to respond approximately ten times faster than the position controller. Although the torque control stage is tuned to be aggressively fast, it is not sufficient to regulate the cogging torque caused by the mechanical structure of the motor [37]. According to the Fourier analysis, the sixth harmonic is observed as the dominant frequency. Since cogging torque cannot sufficiently suppress with the PID, adaptive feedforward cancellation (AFC) is applied. AFC has a significant effect

Table III. Local controller gains.

	Hip joint			Knee joint		
Current PID gains	$K_p = 0.4$	$K_i = 80$	$K_d = 0.1$	$K_p = 0.4$	$K_i = 80$	$K_d = 0.1$
Position PID gains	$K_p = 200$	$K_i = 150$	$K_d = 8$	$K_p = 800$	$K_i = 600$	$K_d = 8$
Speed PID gains	$K_p = 0.0025$	$K_i = 0.01$	$K_d = 0.0001$	$K_p = 0.001$	$K_i = 0.004$	$K_d = 0.0001$
AFC gains (6th harmonic)		$K_a = 400$			$K_a = 400$	

on suppressing periodic disturbances based on the position information [38]. The torque controller stage is written in Eq. (30). Local controller PID and AFC gains are shown in Table III.

$$e_{q_i} = \dot{q}_{i\text{ref}} - \dot{q}_i \tag{29}$$

$$I_{q_{\text{ref}}} = \left(K_p e_{q_i} + K_I \int_0^t e_{q_i}(\tau) d\tau + K_d \dot{e}_{q_i} \right) \tag{30}$$

$$+ \cos(6\theta_{BLDC}(t)) \int_0^t K_{afc} \cos(6\theta_{BLDC}(\tau) + \phi) d\tau$$

$$+ \sin(6\theta_{BLDC}(t)) \int_0^t K_{afc} \sin(6\theta_{BLDC}(\tau) + \phi) d\tau$$

5. Results

The controller that will ensure the dynamic walking of the biped robot developed as a test platform has been studied. The controller is simulated on the mathematical model before being applied to the mechanical model. In this way, the conformity of the real model with the simulation is perceived.

5.1. Simulation results

Dynamic walking is simulated according to the robot mathematical model in the *Matlab-Simulink* environment with a fixed sampling interval of 400 us. Walking surface function defined in Eq. (7) is applied as a flat surface at zero level. After the synchronization angle is determined, the joint angles are tracked to the desired trajectories. To show that the walking period is determined according to the optimized trajectory, no external disturbance is applied in the first 10 s of the simulation. Step period can be measured between 7.82 and 9.14 s, where the walking period converges to 1.31 s. At the tenth second, the situation in which the robot is pushed forward is examined. Therefore, the 10° impulse signal is added to the torso angle. Similarly, at the sixteenth second, the situation where the robot is pushed in the reverse direction is examined. Therefore, the -10° impulse signal is added to the torso angle. Figure 11a shows the reference joint angles and simulated joint angles. It is seen that the trajectory tracking has been successfully provided. Trajectory tracking errors are shown in Fig. 11b. In addition to the external disturbances at the tenth and sixteenth seconds, the disturbance effect occurs at every step. As seen that the walking controller can exponentially reduce the tracking error to zero.

Tracking the reference trajectories is not enough to guarantee dynamic walking. In addition, the presence of the limit cycle must be determined on the phase portrait and Poincaré map. Figure 12a shows the support leg phase portraits, and Fig. 12b shows the Poincare map according to the walking speed.

Poincare map is obtained by drawing the walking speeds at the time of impact according to the speeds at the previous impact. As seen that the simulations started with two different initial walking speeds converge to the walking period that determined by trajectory optimization. The walking period determined by trajectory optimization is 1.3 s. According to the Poincare chart, a stable limit cycle occurs; therefore, dynamic walking is provided.

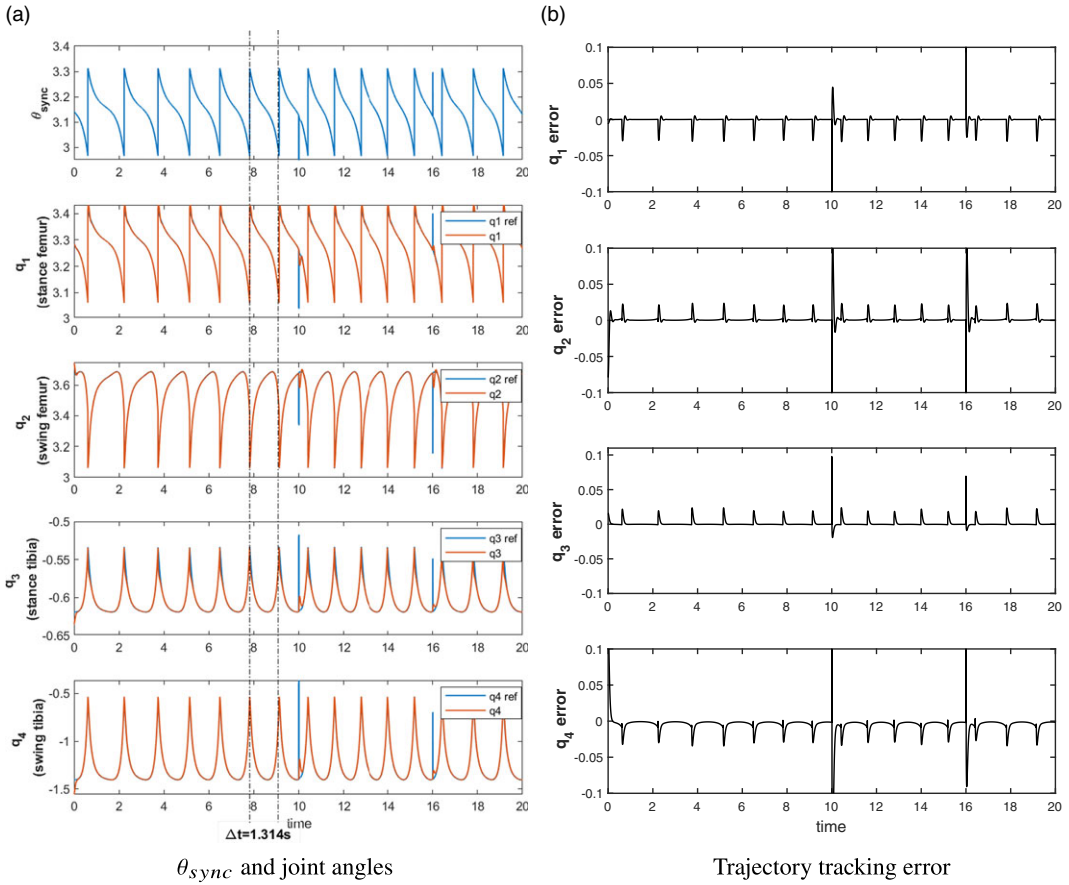


Figure 11. Simulation results.

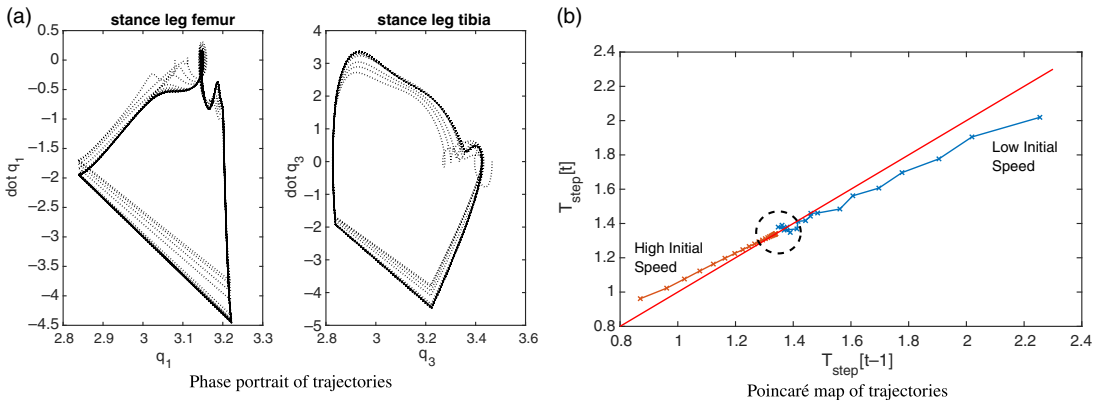


Figure 12. Trajectory analysis.

5.2 Experimental results

The simulation shows that the controller provides dynamic walking. Unfortunately, the simulation demonstration only cannot guarantee that the controller is successful. To support the simulation results, the experimental study is a remarkable factor in proving the success of the proposed controller. The controller can be implemented to the mechanical model due to its model-free trajectory tracking and starting point independent limit cycle features.

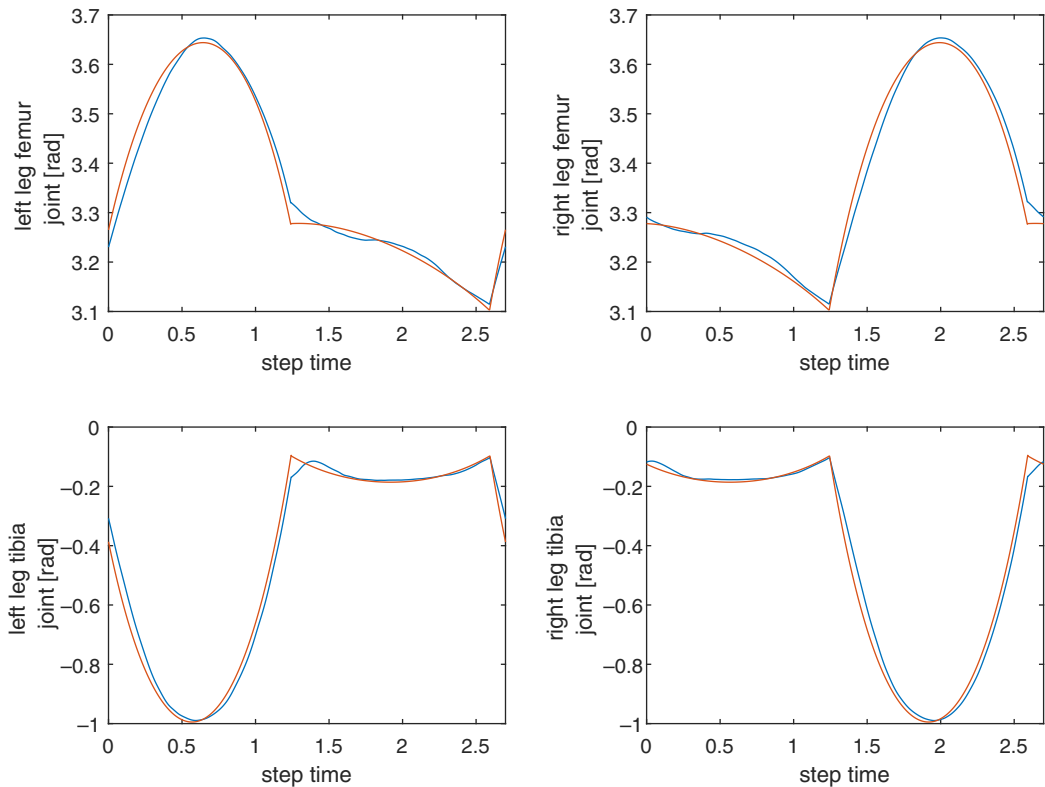


Figure 13. Experimental results logged from LabVIEW interface (desired trajectory: red line, joint trajectory: blue line).

As explained in Section 2.2, a biped robot is designed and manufactured. The designed joint structure meets the mechanical requirements and thus enables precise position control. The control and communication periods are performed at high speed due to the advantage of FPGA and ARM processors. While the main controller period is 400 μ s and the local controller period is 170 μ s, the communication cycle is provided less than 150 μ s. The planar biped robot that mounted to the carrier platform on the flat ground can walk a full turn at the circular path. Although the walking robot is designed as a mobile system, a wired power source was used because of the lack of the proper battery during the tests. Therefore, the range of motion is unfortunately limited. During the walking, pressure sensor information, joint, and torso pitch angles can be easily observed via LabVIEW user interface. According to the simulation results, it is seen that the robot walking period is converged to 1.3 s at the experimental study. The angles of the joints for each leg in both the support and swing phase are given in Fig. 13 for two steps. As it can be seen that the experimental results match the simulation results quite well. Although the simulation results (Fig. 11a) are shown according to the swing/stance angles, which are the state variables, the experimental results (Fig. 13) are shown as the right/left leg angles which turn into swing and stance state variables at each step.

Each joint has a physical connection with other joints. Therefore, each joint also causes a disturbance on other joints. However, the local controller can successfully provide trajectory tracking in a limited error range. The joint trajectory tracking errors are given in Fig. 14.

The video of the movement of the robot is recorded and each frame at 0.4 s intervals is shown as a series in Fig. 15. During this series of frame, the front leg takes two steps and the rear leg takes one step. The entire walking video can be seen in this link

(https://drive.google.com/drive/folders/1_9rLTi9R29Nq305tPOPcDR_gWIKWYg0F?usp=sharing).

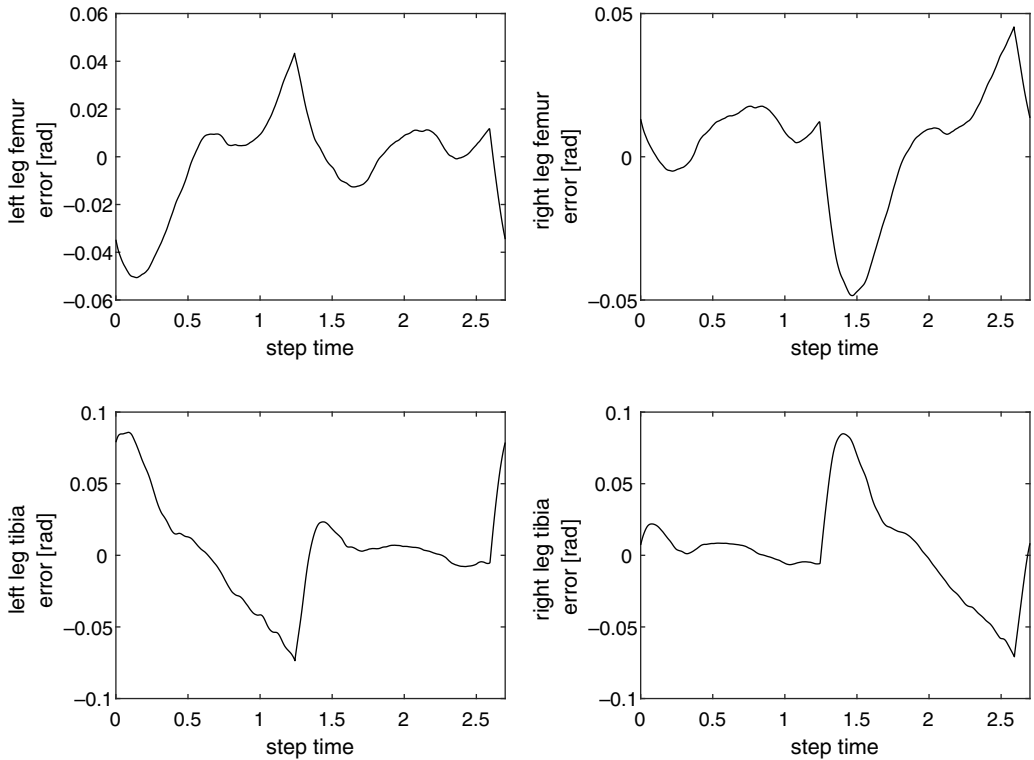


Figure 14. Experimental study joint trajectory tracking error.

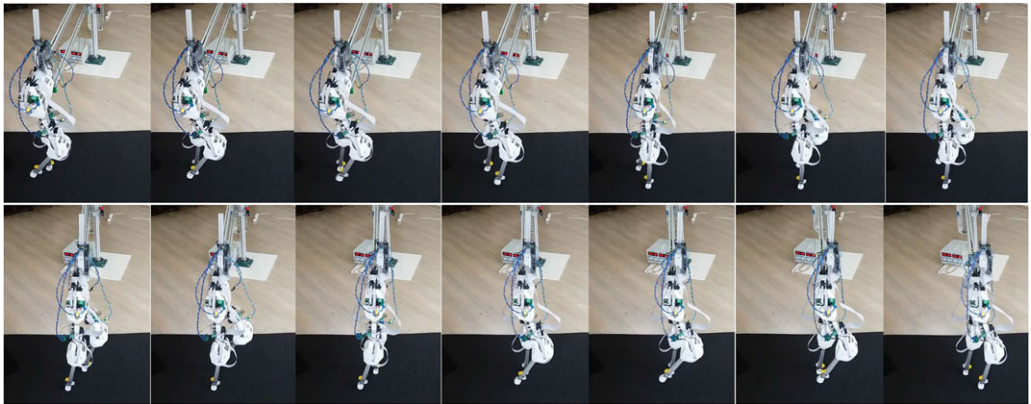


Figure 15. Experimental walking results (frame interval is 0.4 s).

5.3 Walking performance results

The most fundamental parameter that determines walking performance for walking robots is the CoT. This parameter shows the total energy required to transport unit mass per unit distance. CoT calculation is defined in Eq. (31) [39].

$$C_t = \frac{1}{mg |hip^h(t_f) - hip^h(t_0)|} \int_{t_0}^{t_f} |\dot{q}_s^T B u| dt \quad (31)$$

Table IV. *CoT values for various biped robots.*

Robot	Total weight	Leg length	Walking characteristics	CoT (C_T)
RABBIT [40]	32 kg	0.8 m	Step length is 0.3 m	0.38
MABEL [41]	65 kg	1.0 m	walking speed ranges from 0.8 m/s to 1.5 m/s	min: 0.14 max: 0.29
Saurian [39]	5.5 kg	0.37 m	Step length ranges from 0.13 m to 0.2 m	0.129
Cornell Biped [9] (passive-dynamic walker)	12.7 kg	0.8 m	Step length is 0.38 m	0.055
Our robot with fixed-time interval interval (conventional method)	5.95 kg	0.66 m	Step length ranges from 0.12 m to 0.24 m	0.121
Our robot with variable time interval (proposed method)	5.95 kg	0.66 m	Step length ranges from 0.12 m to 0.24 m	0.094

$hip(t)$ represents the horizontal position of the robot's hip point and m represents the total robot mass. The energy transferred to the robot is estimated according to the product of the angular velocity and the applied torque value.

CoT values of fixed and proposed variable time interval direct collocation methods are calculated. According to the 5th and 6th rows of Table IV, it is seen that the CoT value can be reduced on the same robot only by improving the trajectory optimization method. Also, CoT values are compared with other planar five-link bipedal robots known in the literature as presented in Table IV. To compare different robots and different methods, besides the CoT value, the physical and walking characteristics of the robots are added to the comparison table.

The proposed method reduces the energy loss during the collision, as it reduces the optimization error, especially in the moments when the foot lands on the ground and taking off the ground. In this way, more efficient walking is provided.

6. Conclusions

In this study, the trajectory optimization method is proposed to increase optimization accuracy while preserving NLP solution time. The effect of reference trajectories in the control of underactuated systems cannot be denied. Therefore, trajectory optimization accuracy is just as important as the controller design. Although it is effective to increase the number of collocation points to increase optimization accuracy, it is not feasible because the problem-solving time increases exponentially. The proposed method which is based on variable time interval is aimed to keep the optimization accuracy below the threshold value in each segment.

In addition, dynamic walking simulation of five-link planar robot, design of test platform, and experimental study are explained. The designed robot model offers an uncomplicated and affordable test platform for dynamic walking experiments. The dynamic walking algorithm has been run both in the simulation environment and on the real-time model. According to the walking tests, simulation results of the proposed robot model are proved to highly match the experimental results.

To reveal the contribution of the proposed trajectory optimization method, it has been compared with the fixed-time interval trajectory optimization method and those of other robots in the literature. The CoT parameter, which clearly shows the efficiency of walking robots, is used as the comparison criterion. According to comparisons, it can be seen that the proposed method decreases the CoT value approximately 22%. In this way, the proposed method contributes to the bipedal robot trajectory optimization problem.

References

- [1] Y. Sakagami, R. Watanabe, C. Aoyama, S. Matsunaga, N. Higaki and K. Fujimura, "The Intelligent Asimo: System Overview and Integration," In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3 (IEEE, 2002) pp. 2478–2483.
- [2] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonom. Robots* 40(3), 429–455 (2016).
- [3] H. Jeong, K. K. Lee, W. Kim, I. Lee and J.-H. Oh, "Design and control of the rapid legged platform gazelle," *Mechatronics* 66, 102319 (2020).
- [4] E. R. Westervelt, J. W. Grizzle and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE Trans. Automat. Contr.* 48(1), 42–56 (2003).
- [5] W. Znegui, H. Gritli and S. Belghith, "Design of an explicit expression of the poincaré map for the passive dynamic walking of the compass-gait biped model," *Chaos, Solitons & Fractals* 130, 109436 (2020).
- [6] W. Znegui, H. Gritli and S. Belghith, "Stabilization of the passive walking dynamics of the compass-gait biped robot by developing the analytical expression of the controlled poincaré map," *Nonlinear Dyn.* 101(2), 1061–1091 (2020).
- [7] M. Iribe, R. Hirouji, D. Ura, K. Osuka and T. Kinugasa, "Experimental verification of the characteristic behaviors in passive dynamic walking," *Artif. Life Robot.* 26(2), 187–194 (2021).
- [8] C. Vasileiou, A. Smyrli, A. Drogosis and E. Papadopoulos, "Development of a passive biped robot digital twin using analysis, experiments, and a multibody simulation environment," *Mech. Mach. Theory* 163, 104346 (2021).
- [9] S. Collins, A. Ruina, R. Tedrake and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science* 307(5712), 1082–1085 (2005).
- [10] J. W. Grizzle, J. Hurst, B. Morris, H.-W. Park and K. Sreenath, "Mabel, A New Robotic Bipedal Walker and Runner," In: 2009 *American Control Conference* (IEEE, 2009) pp. 2030–2036.
- [11] R. Singh, H. Chaudhary and A. K. Singh, "Sagittal Position Analysis of Gait Cycle for a Five Link Biped Robot," In: *CAD/CAM, Robotics and Factories of the Future* (Springer, 2016) pp. 387–396.
- [12] M. Posa, C. Cantu and R. Tedrake, "A direct method for trajectory optimization of rigid bodies through contact," *Int. J. Robot. Res.* 33(1), 69–81 (2014).
- [13] M. Safartoobi, M. Dardel and H. M. Daniali, "Gait cycles of passive walking biped robot model with flexible legs," *Mech. Mach. Theory* 159, 104292 (2021).
- [14] T. Apgar, P. Clary, K. Green, A. Fern and J. W. Hurst, "Fast Online Trajectory Optimization for the Bipedal Robot Cassie," In: *Robotics: Science and Systems*, vol. 101 (2018).
- [15] J. Zhao and T. Shang, "Dynamic optimization using local collocation methods and improved multiresolution technique," *Appl. Sci.* 8(9), 1680 (2018).
- [16] H. Yuan, Y. Ge and C. Gan, "Modification-based periodic gait planning for a 3-d bipedal robot with two underactuated DOFs," *Asian J. Cont.* (2020).
- [17] K. Sreenath, H.-W. Park, I. Poulakakis and J. W. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on MABEL," *Int. J. Robot. Res.* 32(3), 324–345 (2013).
- [18] A. D. Ames, K. Galloway, K. Sreenath and J. W. Grizzle, "Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics," *IEEE Trans. Automat. Cont.* 59(4), 876–891 (2014).
- [19] H. Wang, H. Zhang, Z. Wang and Q. Chen, "Finite-time stabilization of periodic orbits for under-actuated biped walking with hybrid zero dynamics," *Commun. Nonlinear Sci. Numer. Simul.* 80, 104949 (2020).
- [20] H. Sadeghian, C. Ott, G. Garofalo and G. Cheng, "Passivity-Based Control of Underactuated Biped Robots within Hybrid Zero Dynamics Approach," In: 2017 *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017) pp. 4096–4101.
- [21] W.-L. Ma, S. Kolathaya, E. R. Ambrose, C. M. Hubicki and A. D. Ames, "Bipedal Robotic Running with Durus-2D: Bridging the Gap between Theory and Experiment," In: *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control* (2017) pp. 265–274.
- [22] M. M. Kakaei and H. Salarieh, "New robust control method applied to the locomotion of a 5-link biped robot," *Robotica* 38(11), 2023–2038 (2020).
- [23] W. M. L. T. Weerakoon, D. H. S. Maithripala and J. M. Berg, "Feedback Regularization based PD Controller for a Planar Three-Link Biped Robot," In: 2017 *IEEE International Conference on Industrial and Information Systems (ICIIS)* (IEEE, 2017) pp. 1–6.
- [24] M. Fevre, B. Goodwine and J. P. Schmiedeler, "Terrain-blind walking of planar underactuated bipeds via velocity decomposition-enhanced control," *Int. J. Robot. Res.* 38(10–11), 1307–1323 (2019).
- [25] A. Hereid, C. M. Hubicki, E. A. Cousineau and A. D. Ames, "Dynamic Humanoid Locomotion: A Scalable Formulation for HZD Gait Optimization," *IEEE Trans. Robot.* 34(2), 370–387 (2018).
- [26] R. Heydari and M. Farrokhi, "Robust model predictive control of biped robots with adaptive on-line gait generation," *Int. J. Cont. Automat. Syst.* 15(1), 329–344 (2017).
- [27] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi and B. Morris, *Feedback Control of Dynamic Bipedal Robot Locomotion* (CRC Press, 2018).
- [28] Y. Hurmuzlu, F. Génot and B. Brogliato, "Modeling, stability and control of biped robots—A general framework," *Automatica* 40(10), 1647–1664 (2004).
- [29] B. Sensortec, "Intelligent 9-axis absolute orientation sensor," BNO055 datasheet (November 2014).

- [30] J. W. Hurst and A. A. Rizzi, “Series compliance for an efficient running gait,” *IEEE Robot. Automat. Mag.* **15**(3), 42–51 (2008).
- [31] S. Mefoued and D. E. C. Belkhiat, “A robust control scheme based on sliding mode observer to drive a knee-exoskeleton,” *Asian J. Cont.* **21**(1), 439–455 (2019).
- [32] A. Fayazi, N. Pariz, A. Karimpour and S. H. Hosseinnia, “Robust position-based impedance control of lightweight single-link flexible robots interacting with the unknown environment via a fractional-order sliding mode controller,” *Robotica* **36**(12), 1920–1942 (2018).
- [33] M. Kelly, “An introduction to trajectory optimization: How to do your own direct collocation,” *SIAM Rev.* **59**(4), 849–904 (2017).
- [34] J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming* (SIAM, 2010).
- [35] Y. Yang, J. Pan, P. Long, X. Song and L. Zhang, “Time variable minimum torque trajectory optimization for autonomous excavator,” arXiv preprint arXiv:2006.00811 (2020).
- [36] B. Ghoghogh, S. Sharifian and H. Mohammadzade, “Tree-based optimization: A meta-algorithm for metaheuristic optimization,” arXiv preprint arXiv:1809.09284 (2018).
- [37] M. S. Islam, S. Mir and T. Sebastian, “Issues in reducing the cogging torque of mass-produced permanent-magnet brushless DC motor,” *IEEE Trans. Ind. Appl.* **40**(3), 813–820 (2004).
- [38] N. A. Qamar and C. J. Hatziaodoniu, “Cancellation of selected stator harmonics in BLDC by using an adaptive feedforward controller,” *Electr. Power Syst. Res.* **154**, 88–94 (2018).
- [39] J. Arcos-Legarda, J. Cortes-Romero and A. Tovar, “Robust compound control of dynamic bipedal robots,” *Mechatronics* **59**, 154–167 (2019).
- [40] J. W. Grizzle and C. Chevallereau, “Virtual constraints and hybrid zero dynamics for realizing underactuated bipedal locomotion,” arXiv preprint arXiv:1706.01127 (2017).
- [41] K. Sreenath, *Feedback Control of a Bipedal Walker and Runner with Compliance*. PhD thesis (2011).