Received: 29 January 2018 | Revised: 25 April 2018 | Accepted: 14 May 2018

DOI: 10.4218/etrij.2018-0058

SPECIAL ISSUE

WILEY ETRI Journal

Approach toward footstep planning considering the walking period: Optimization-based fast footstep planning for humanoid robots

Woong-Ki Lee | **In-Seok Kim** | **Young-Dae Hong**

Department of Electrical and Computer Engineering, Ajou University, Suwon, Rep. of Korea.

Correspondence
Young-Dae Hong, Department of Electrical and Computer Engineering, Ajou University, Suwon, Rep. of Korea.
Email: ydhong@ajou.ac.kr

Funding information
This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP), Grant/Award Number: (2016R1C1B1006691).

This paper proposes the necessity of a walking period in footstep planning and details situations in which it should be considered. An optimization-based fast footstep planner that takes the walking period into consideration is also presented. This footstep planner comprises three stages. A binary search is first used to determine the walking period. The front stride, side stride, and walking direction are then determined using the modified rapidly-exploring random tree algorithm. Finally, particle swarm optimization (PSO) is performed to ensure feasibility without departing significantly from the results determined in the two stages. The parameters determined in the previous two stages are optimized together through the PSO. Fast footstep planning is essential for coping with dynamic obstacle environments; however, optimization techniques may require a large computation time. The two stages play an important role in limiting the search space in the PSO. This framework enables fast footstep planning without compromising on the benefits of a continuous optimization approach.

KEYWORDS
continuous footstep planning, humanoid navigation, modified rapidly-exploring random tree algorithm, particle swarm optimization, walking period

1 | INTRODUCTION

Now that we are facing the fourth industrial revolution, more attention is being focused on the development of robot technology than ever. Humanoids, which have been developed for helping human beings while minimizing changes in the way of human life, are also subjects of curiosity and interest. The ability to walk is one of the most important factors that affects the practical use of humanoids that are intended to live and work with humans. Footstep planning under several constraints is a critical issue in humanoid walking research.

At the beginning of the study, the research was focused on discrete properties, as a humanoid has two feet. The easiest method of planning footsteps is the A*-based discrete search method. The A*-based discrete search method has been studied to simplify the complexities arising from the discrete properties of humanoids [1–4]. The A*-based discrete search method plans a feasible footstep sequence using a pre-calculated footstep displacement set (transition model). However, there are limitations to footstep planning depending on which transition model and heuristics function are used. Studies have been undertaken to solve this problem using the adaptive transition model [5]. To solve the issue regarding the complexity of the A*-based discrete search method, a random sampling method using a rapidly-exploring random tree (RRT) algorithm has been studied [6–8]. The issue of the limitation with respect to the heuristics function was solved by using the random-sampling-based discrete search method instead of the A*-based discrete search method, but the limitation of the discrete search method, which is dependent on the transition model, was not completely solved. Recently, a continuous optimization approach has been studied to overcome the limitation of the discrete search method [9–13]. In this study, the primary parameters included the stride length and walking direction. As the humanoid actuator's performance improves and humanoids become increasingly similar to humans, the humanoid's walking periods are being studied as well. Their walking period has mainly been studied in terms of walking stability only. The walking period in the walking pattern generation has been recognized as an important factor in relation to the walking stability [14–19]. However, in the footstep planning stage, which is related to humanoid navigation, research on the walking period rarely bore fruit. The walking period should be considered from the viewpoint of navigation as well as walking stability. The need for a humanoid to change its walking period in the footstep planning stage will be questioned. However, there are circumstances wherein the walking period must be considered. More effective and flexible footstep planning becomes possible when the walking period is included in the footstep planning stage. Although research has been conducted to generate a transition model using a simplified state and the walking speed of a robot, it is difficult to produce suitable transition models that take into consideration environmental changes and the walking period [4]. In this study, we use the continuous optimization approach to bypass the limitation of the discrete search method when considering the walking period in footstep planning. The environments in which the walking period should be considered are dynamic environments, and fast footstep planning is necessary to cope with them. Even though the continuous optimization approach has been adopted, it may result in a long computation time during footstep planning. We propose a footstep planning framework for solving the computational complexity problem while maintaining the advantage of the continuous optimization approach when considering the walking period in footstep planning. The proposed footstep planning framework comprises three stages. In the first stage, the walking period is determined using a binary search algorithm, and in the second stage, the path planning algorithm is used to determine the next position of the robot. In the proposed footstep planner, the RRT algorithm is used as a path planning algorithm, and some modifications are made to it to fit the proposed footstep planner. In this study, we refer to it as the modified RRT algorithm. The aforementioned two stages serve to reduce the optimization search space. They also play a role in reducing the computation time. In the last stage, the continuous optimization approach is adopted to simultaneously optimize the variables determined in the previous stages. The footstep command, which is the step information determined in the previous stages, is made feasible by using the particle swarm optimization (PSO) algorithm. The feasibility indicates whether the footstep command generated in the footstep planner can be followed in the walking pattern generator. To verify this, simulations are conducted under various circumstances. Circumstances in real life where the walking period must be considered are described, and the performance of the proposed footstep planner is verified.

This paper is organized as follows: Section 2 introduces the necessity of the walking period and the continuous optimization-based footstep planner. Section 3 validates the necessity of the walking period and the proposed footstep planner using several simulations. Section 4 discusses the limitations of the proposed footstep planner and future works and concludes the paper.

2 | PROPOSED FOOTSTEP PLANNER

2.1 | Necessity of the walking period

When observing the walking of the elderly, we can see changes in their stride and walking direction, but the walking period hardly changes. Elderly people are vulnerable in situations in which they must suddenly change their walking period, for example, in a slippery environment or owing to a sudden disturbance. They are vulnerable in such situations because it is difficult for them to cope with changes in the walking period. Similarly, humanoids are vulnerable in such situations if they do not take the walking period into consideration. For this reason, a study was conducted on footstep planning in a slippery environment with regard to the walking period [19], and research on push recovery corresponding to a sudden disturbance was also conducted [14]. As such, the walking period is focused on mainly in terms of walking stability.

In contrast, research has been rarely conducted on a humanoid's walking period in view of navigation. Sometimes, we come across news about the elderly having accidents in revolving doors. They are exposed to risks in navigation because they cannot easily cope with variations in the walking period. Humanoids face this problem too when the walking period is not considered in footstep planning. There are three advantages of considering the walking period with respect to navigation.

First, the range of the responses to dynamic obstacles increases. Humanoid footstep planning research has mainly dealt with changes that take place in the stride and walking direction to cope with dynamic obstacles. However, the walking speed of humanoids is determined by the stride and the walking period. Therefore, by considering the walking period, it is possible to cope with dynamic obstacles in a more flexible manner. Figure 1 comprises a graph that shows the range of the obstacle environment that can be handled by the humanoid robot according to the stride and walking period. The straight line "A" refers to the conventional range of obstacles that can be dealt with only when considering the stride. If the humanoid robot changes only the stride while maintaining the walking period constant, the obstacle environment that the humanoid robot can cope with is limited as shown in Figure 1. In contrast, if humanoid robot considers the walking period additionally, it becomes possible to move at a higher speed, which indicates that the range of the obstacles that can be dealt with is widened. The area of the obstacle that can be handled with respect to the walking period is the "B" area. As shown in Figure 1, it can be confirmed that the humanoid robot can cope with the obstacle environment at a higher speed in the same stride by considering the walking period to be shorter than the walking period fixed at "A."

Second, footstep planning is possible in a more efficient manner by considering the walking period in the navigation stage. The efficiency can vary depending on the criterion. If energy consumption is a criterion, then reaching the destination without collisions with the obstacles while lowering energy consumption would be considered to be effective footstep planning. If the planned path length is a criterion, then planning to navigate shorter paths would be considered effective footstep planning. In a real dynamic environment, humans unconsciously lay the foundation for an efficient walking strategy. They decide to adjust speed when dynamic obstacles disturb the walking path. As the walking speed is adjusted considering both the stride and the walking period, humans slow down to select a short walking distance by passing a dynamic obstacle first. They do this while adjusting the walking period more than the stride or speeding up while taking quick steps (a short stride with a short walking period) to reach the destination quickly without any obstacle collision. In such cases, if the speed is adjusted using only the stride, unnecessary motions, such as walking on the spot or stopping to avoid a collision, may occur. In addition, humans may collide with obstacles if they cannot use quick steps. It is also essential for a humanoid to change the walking period to maintain the efficiency of footstep planning rather than simply aiming for the destination without colliding with dynamic obstacles. The area "C" in Figure 1 represents the obstacle environment that the humanoid robot can cope with when both the stride and walking period are considered simultaneously. The area "C" includes all the remaining regions.

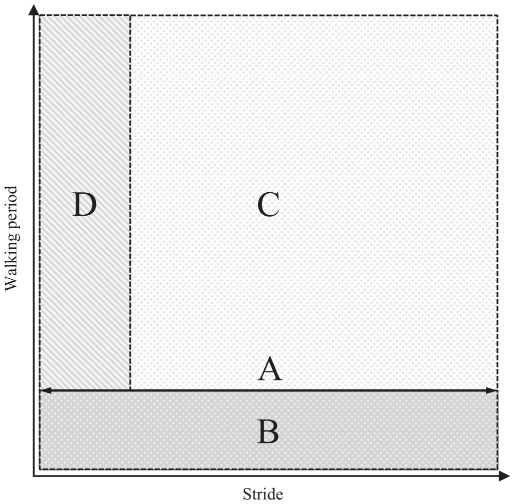


FIGURE 1 Range of the obstacle environment that can be handled by the humanoid robot according to the stride and walking period. Straight line "A" refers to when only the stride is considered, the "B" area indicates that the humanoid robot can cope with obstacles at a higher speed while considering the additional walking period, the "C" area indicates that the humanoid can cope with various obstacles by controlling the stride and the walking period, and the "D" area indicates cases wherein the obstacle is handled by adjusting the walking period when the stride is restricted

Third, it is possible to deal with dynamic obstacles that cannot be pass without considering the walking period. In this study, we define an environment that cannot be navigated without changing the walking period as a closed environment, which is an environment wherein the robot is in an enclosed space owing to the obstacles surrounding it. To cope with dynamic obstacles, it is necessary to adjust the speed of the robot. However, the obstacles surrounding the robot limit its stride. Therefore, if the speed is not adjusted by changing the walking period, a collision occurs with the obstacle. In such an environment, the walking period must be taken into consideration in order to cope with the dynamic obstacles. The "D" area in Figure 1 shows the range of the obstacle environment that the humanoid robot can cope with if the walking period is taken into consideration when the stride is limited. When the stride is limited below a certain value, it can be confirmed that the humanoid robot should consider an appropriate walking period that is not too short. As such, there are circumstances where it is possible to cope with dynamic obstacles more flexibly and efficiently and in which humanoids must take into consideration the walking period. Therefore, it is necessary to consider the walking period in terms of navigation as well as walking stability.

Algorithm 1. Determination of the walking period

```
/* initialization*/
$T_0 \leftarrow 0$;
  $T_1 \leftarrow T_n$;
  /* start binary search algorithm */
  For $n = 1$ to $n = N$
    $T_m \leftarrow (T_0 + T_1)/2$;
    If ObstacleCollision($T_1$) then
      $T_1 \leftarrow T_m$;
    else then
      $T_0 \leftarrow T_m$;
      If $n == N$ then
        return $T_1$;
      end If
    end If
  end For
  return $T_1$;
```

2.2 | Determination of the walking period

The walking period T is determined in the first stage. The determination of T is related to the determination of the position of the predicted obstacle before extending a node of the modified RRT. In the proposed footstep planner, T is additionally considered for coping with the dynamic obstacle. It is defined as follows:

$$T \equiv T_{\mathrm{SS}} + T_{\mathrm{DS}}, \tag{1}$$

here, T_{SS} is the single support time, and T_{DS} is the double support time. A binary search is performed to determine the walking period. One of the important characteristics of the proposed footstep planning framework is the small computation time. Therefore, a binary search is used as a search algorithm, which is known to be fast for finding a specific value in a continuous data set. In addition, as the value to be searched in this stage is a continuous time, the binary search is performed using the array without using a data structure such as a tree. In the stage wherein

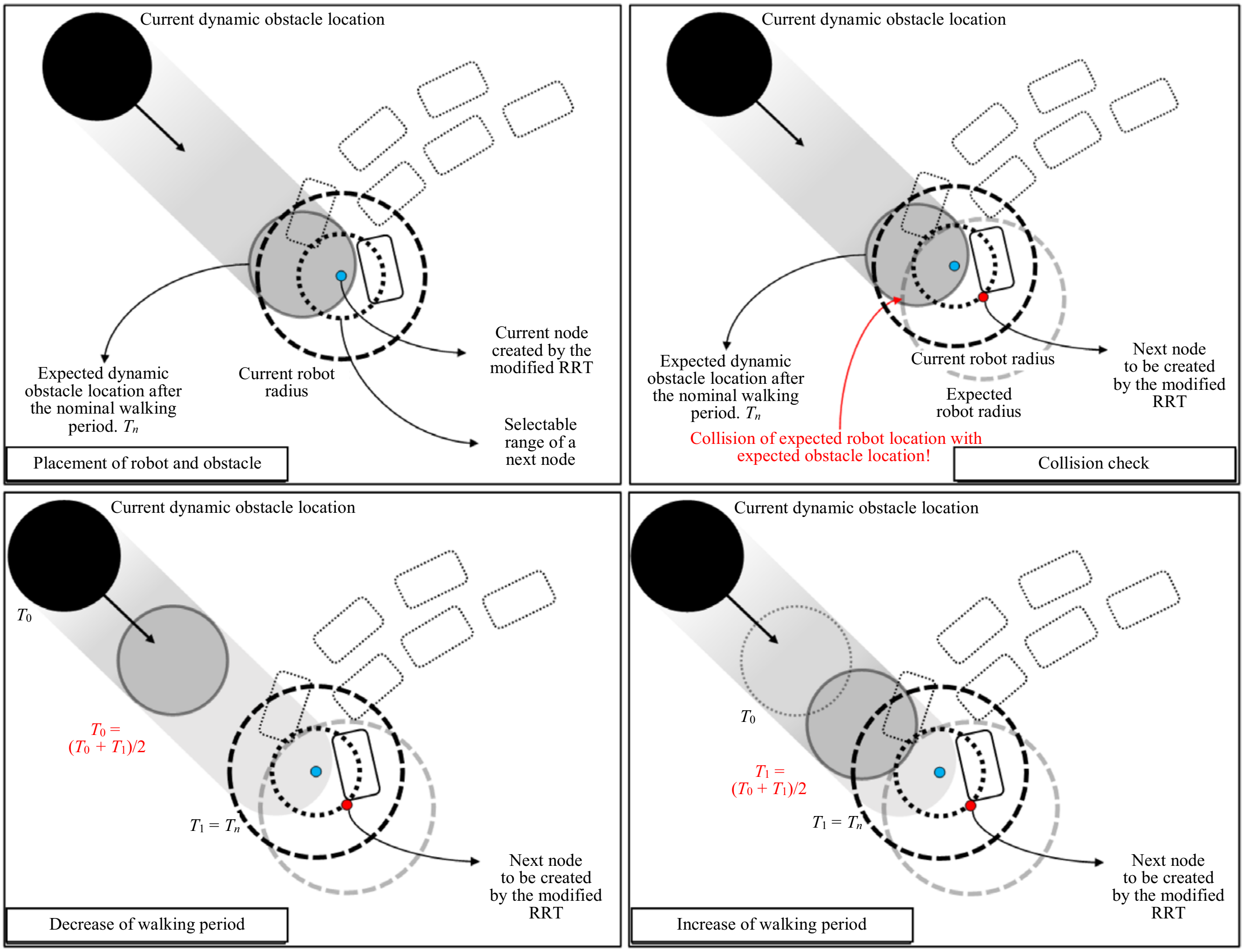


FIGURE 2 Determination of the walking period using binary search and collision check

we determine the walking period, the predicted position of the obstacle after the nominal walking period T_n is calculated based on the position of the obstacle in the current time. It is assumed that the direction and speed of the obstacle are known or estimated. A collision check is used to determine whether the modified RRT algorithm can be used based on the predicted position of the obstacle and the current obstacle position. If it is determined that the modified RRT algorithm can extend a node through the collision check, T_n is determined as the walking period for the next step, or a binary search is used to determine a new walking period in which the modified RRT algorithm can be extended. The collision check is repeated in every loop of the binary search algorithm. This is repeated until we reach the termination condition of the binary search.

Before proceeding with the binary search, we determine the upper limits of the walking period T_1 and the lower limits of the walking period T_0. T_1 is determined to be the nominal walking period T_n, and T_0 is determined to be a short walking period that the robot can use. The procedure of the determination of the walking period is presented in Algorithm 1 and Figure 2.

2.3 | Path planning

The second stage comprises path planning. The path planning algorithm used in the proposed footstep planner is required to meet certain requirements. It must determine the direction to the destination while the considering the configuration of the obstacles at every step in order to cope with the dynamic obstacles, and it should have a small convergence time in the majority of environments. The direction to the destination while considering the obstacle configuration is the direction for realizing the next robot position after a walking period while considering the obstacle configuration and destination instead of the direction from the current position to the destination.

The RRT algorithm has a short convergence time in the majority of environments, but it may not be able to determine the direction for the destination at every step. The paths generated by the RRT algorithm do not guarantee an optimum solution. Therefore, it can be said that the shape of the path from the current position to the destination can be jagged like a tree instead of smooth in Figure 3A. Owing to these factors, we may not be able to achieve the direction from the current position to the destination. To solve this problem, the modified RRT algorithm is used instead of the RRT algorithm.

The modified RRT algorithm uses two methods to deal with this problem.

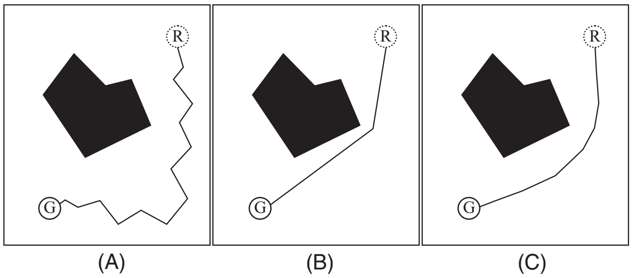


FIGURE 3 Final path of rapidly-exploring random tree (RRT) variants: (A) final path of original RRT algorithm, (B) final path of original RRT algorithm using shortcut method, and (C) final path of modified RRT algorithm

Algorithm 2. Modified RRT

```
Initialize the tree for RRT.
Insert the initial position of the robot into the first node of the tree.
For $i = 1$ to $i = N$
Extract random coordinates for the next node selection.
Calculate the closest node from random coordinates.
  If the angle limit is exceeded, then
    Randomly coordinate re-selection within the angle limit
  end If
  If the distance from the selected node to the nearest node exceeds the step size, then
    Adjust the distance from the nearest node to the step size.
  end If
  If the selected node does not collide with obstacles, then
    Insert the selected node into the tree.
    The distance from the nearest node to the newly inserted node is calculated and stored as the cost value in addition to the distance value of the nearest node.
    If the distance from the newly inserted node to the destination is sufficiently close, then
      compare the cost and store the path with the smallest cumulative distance.
    end If
  end If
end For
Select the path with the smallest cost.
```

First, we consider the hardware limit of the humanoid robot in the path planning stage. The hardware limit refers to the limit of the walking direction based on the previous supporting foot. Restrictions on the walking direction can be divided into an inner angle toward the torso θ_{inner} and an outer angle from the torso θ_{outer} with respect to the supporting foot. The configuration of constraints in the walking direction in the modified RRT algorithm are presented

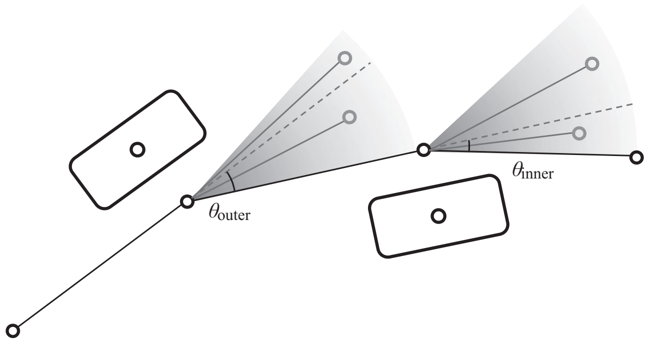


FIGURE 4 Tree structure expansion of the modified rapidly-exploring random tree (RRT). The gray gradation area represents the RRT extension area limited by the robot hardware. θ_{inner} represents the angle limit on the inside of the robot body, and θ_{outer} represents the angle limit to the outside of the robot body

in Figure 4. The convergence time can be reduced by restricting the search space of the RRT algorithm by restricting the hardware to the path planning stage. This constraint does not eliminate the advantage of the continuous optimization approach, which is a more flexible and efficient footstep planning approach because it plays a role in eliminating the walking direction, which is not possible in the case of robot hardware. In addition to reducing the convergence time by considering the limitation of the walking direction in the path planning stage, the path generated by the path planning algorithm can be made smooth.

Second, we generate multiple paths in order to determine the final path. If there is a hardware constraint, the path can be made smooth and suitable for use by a robot, but it may not direct the robot in the direction of the destination. In order to solve this problem, two or more paths are grouped together, the length cost is compared, and the path with the lowest cost is selected as the final path. Generally, the shortcut method can be used to post-process the path to be used by the robot. The direction of the destination while considering the configuration of the obstacles can be secured by applying the shortcut method, but it may give rise to two problems.

First, additional computation time is required. This undermines the fast convergence time, which is one of the advantages of the RRT algorithm.

Second, the constraints of the RRT algorithm become meaningless as shown in Figure 3B. This may result in a widening of the search space in the continuous optimization stage and can ultimately increase the time required for footstep planning. In the proposed footstep planner, the modified RRT algorithm (presented in Algorithm 2) is used instead of the shortcut method as shown in Figure 3C. The steps added to the existing RRT algorithm are steps 6–8 and steps 13–16. Steps 6–8 represent the search space restriction, and steps 13–16 outline the process for comparing the length cost of the final paths.

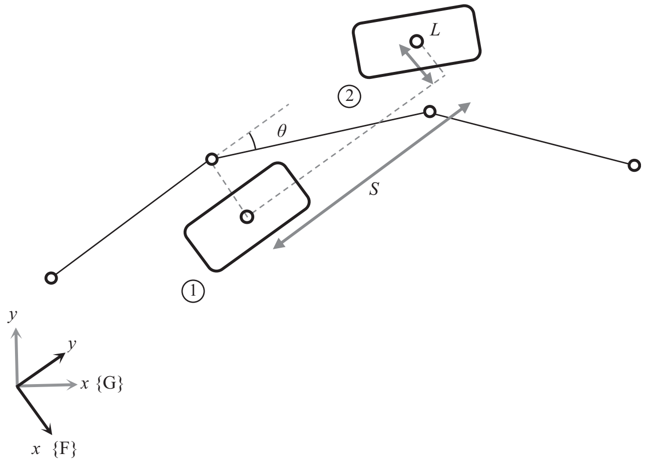


FIGURE 5 Footstep convention based on the modified rapidly-exploring random tree algorithm

Figure 5 represents the footstep convention based on the modified RRT algorithm. In Figure 5, let ① be the supporting foot and ② be the new foot. The supporting foot coordinate system {F} for deriving the footstep command is shown based on the global coordinate system {G} in the lower-left corner of Figure 5. The y axis of the supporting foot coordinate system is set parallel to the connecting line between the node generated by the modified RRT algorithm and the previous node, and the x axis is set to a vector rotated by $-\frac{\pi}{2}$ with respect to the y axis. The center of the foot is set to a position that is at a vertical distance from the corresponding node by a predetermined constant value. The front stride S and side stride L are derived from (2):

$$[L\,S]^T = {}^F_G T(\mathbf{p}_n - \mathbf{p}_s), \tag{2}$$

here, $\mathbf{p}_n$ is the center point of the supporting foot in {G}, $\mathbf{p}_s$ is the center point of the new foot in {G}, and ${}^F_G T$ is the homogeneous transformation matrix. The walking direction θ is derived from (3):

$$\theta = a\tan 2(y_2 - y_1, x_2 - x_1) \tag{3}$$

with

$$\mathbf{d}_1 = [x_1\, y_1]^T, \mathbf{d}_2 = [x_2\, y_2]^T,$$

here, $\mathbf{d}_1$ is a direction vector corresponding to the supporting foot, and $\mathbf{d}_2$ is a direction vector corresponding to the new foot. In this study, we define the result of the footstep planner as a footstep command, and it is defined as follows (4):

$$FC \equiv [S\ L\ \theta\ T]^T. \tag{4}$$

2.4 | Footstep command optimization

The third stage is the continuous optimization stage. At this stage, the previously determined footstep command (FC) is optimized to ensure feasibility. The feasibility is determined by the states of the *FC*. Based on the states of the *FC*, it is determined whether the zero moment point (ZMP) is inside the predetermined stable region inside the foot of the robot, and it is determined that the *FC* is feasible if the ZMP is in the stable region [20]. The role of the two preceding stages is to speed up the third stage. Therefore, in the proposed footstep planning framework, the continuous optimization stage should be executed at the end.

Footstep planning can be categorized into the discrete search and continuous optimization approaches [12]. The discrete search method uses a predetermined discrete set with feasible footsteps based on the supporting foot. It is difficult to determine the proper discrete set if we use the discrete search method considering the walking period as well as the front stride, side stride, and walking direction. This is because it is almost impossible to determine the discrete set that would be applicable to the majority of situations as the number of existing cases is too high. In order to solve this problem, we used the continuous optimization approach, which enables better footstep planning than the discrete search method. This means that it is possible to generate a *FC* that is more suitable to static and dynamic obstacle environments. However, a disadvantage of the continuous optimization approach is that it may be difficult to plan footsteps sufficiently fast to cope with dynamic obstacles. We can achieve better footstep planning while ensuring fast planning time through the methods defined here.

As mentioned previously, we first present a method for limiting the optimization search space of the walking period by using the binary search in the first stage. The maximum walking period is determined from the binary search.

Second, the footstep command resulting from the modified RRT algorithm and the footstep convention limits the range of S, L, and θ. These two stages reduce the optimization range of the PSO and can result in a shortening in the optimization time.

The third method comprises the use of the PSO as an optimization algorithm. The PSO is an evolutionary optimization technique for evaluating the suitability of the solution through the objective function and to find the solution to the problem by repeatedly improving the candidates of the solution. The PSO models the candidates of the solution as a particle swarm. These particles have their own states (positions) and velocities and move to the next state based on them. The velocity of each particle is updated to the direction of the better solution found for the other particle. This is expected to move the swarm toward the best solutions [21]. Representative methods used in the current continuous optimization approach can be classified into evolutionary optimization and convex optimization. A footstep planning study using evolutionary optimization has progressed, and footstep planning on uneven terrain using convex optimization was also performed [12]. However, the majority of footstep planning problems have nonconvex constraints. In order to use convex optimization, the nonconvex constraints must be modified into convex constraints. In contrast, the evolutionary optimization can be used without considering nonconvex constraints. Therefore, evolutionary optimization is selected as an optimization technique. Among the evolutionary optimization techniques, PSO is a sufficiently fast algorithm that can be used in the proposed footstep planning framework. In addition, PSO is known to be a faster algorithm than the genetic algorithm (GA), which is another typical evolutionary algorithm, because PSO does not have complicated evolutionary operations in contrast to the GA.

In the PSO stage, the footstep command defined earlier is extended as shown here (5):

$$FC \equiv [S\ L\ \theta\ T_{\mathrm{SS}}\ T_{\mathrm{DS}}]^T. \tag{5}$$

In addition to using the optimization algorithm with a fast convergence rate, the optimization area is reduced to speed up the optimization. We set an incremental value as an optimization parameter instead of the *FC* itself. This serves to reduce the optimization area. In the previous two stages, we did not specify the range of the variables that comprise the *FC* but decided it to be a specific value. Therefore, the footstep configuration derived using the footstep convention can simply be accepted as a pose calculation. However, the specific values in the preceding two stages are reference values, which play a role in reducing the search space. Therefore, the optimized *FC* can be obtained by adding the optimized increment value to the reference value. The parameters to be optimized through the PSO are shown in (6). The variation ranges of the incremental values are represented in Table 1.

$$\Delta FC \equiv [\Delta S\ \Delta L\ \Delta\theta\ \Delta T_{\mathrm{SS}}\ \Delta T_{\mathrm{DS}}]^T. \tag{6}$$

The objective function of the PSO, f_{ob} is set such that a feasible footstep is generated without significantly differing from the *FC* generated by the binary search and the modified RRT algorithm. The objective function of the PSO is shown in (7).

TABLE 1 Variation ranges used in particle swarm optimization

Incremental variable	Ranges
dT_{SS} (s)	[−0.4 to 0]
dT_{DS} (s)	[−0.1 to 0.1]
dS (m)	[−0.01 to 0.01]
dL (m)	[−0.01 to 0.01]
$d\theta$ (m)	[−5 to 5]

$$f_{\mathrm{ob}} = 100 * ||FC - FC_{\mathrm{op}}|| + P \tag{7}$$

with

$$P = \begin{cases} \infty, & \text{if feasibility} = \text{false} \\ 0, & \text{if feasibility} = \text{true}. \end{cases}$$

The optimized footstep command, FC_{op}, is shown as follows:

$$FC_{\mathrm{op}} = FC + \Delta FC. \tag{8}$$

2.5 | Overview of the proposed footstep planner

The footstep planner proposed in this study takes the walking period into consideration for a more flexible and efficient response in dynamic environments without collision. The proposed footstep planner consists of three stages. The starting position P_{start}, the destination point P_{goal}, and the initial parameters *Max_iteration* and *Max_step_size* of the modified RRT algorithm are determined before applying the three stages.

In the first stage, the position of the expected obstacle is determined before applying the path planning algorithm. The proposed footstep planner accounts for the position of the obstacle at the current time as well as the expected position of the obstacle after a certain time in order to cope with dynamic environments. Depending on the position of the expected obstacle, the path planning algorithm may not search a path in the current position. Therefore, the position of the expected obstacle to which the path planning algorithm can be applied is determined by the walking period. If the velocity of the obstacles is known or can be estimated, the position of the obstacles can then be determined after a certain time. In order to reduce the computation time in the continuous optimization stage, we limit the optimization range of the walking period by using a binary search. As the walking period determined using the binary search is the longest walking period for coping with dynamic obstacles, it does not interfere with a more flexible footstep planning, which is one of the advantages of the continuous optimization approach.

In the second stage, the stride and walking direction are determined. The modified RRT algorithm is used to find the path from the current position to the destination. The node generated by the modified RRT algorithm is determined as the center point of the robot. The modified RRT algorithm and footstep convention are used to determine the stride and walking direction. In order to cope with a dynamic environment, it is important to have a direction for a destination at every step. The RRT algorithm has a small convergence time in the majority of environments, but it may not have the direction for the destination, which is why the modified RRT algorithm is used. In contrast to the discrete search method, a transition model is not used to determine the footstep each time the modified RRT algorithm is extended. The footstep is decided preferentially by the footstep convention. The modified RRT algorithm and footstep convention can reduce the computational burden in the continuous optimization stage by preferentially determining the footstep in this stage.

The final stage is the continuous optimization stage. The goal of the optimization is to determine the feasible footstep command that is closest to the footstep command determined by the previous two stages. In particular, T is optimized such that it is shorter than or equal to the walking period determined by the binary search. The discrete search method based on the transition model cannot be used when considering T as well as S, L, and θ. The creation of a model that includes a walking period based on a previous support foot results in too many cases. By dividing the transition model into several stages and simplifying the states used in constructing the transition model, this method can respond to dynamic environments. However, we use PSO to respond to dynamic environments in a more flexible manner. Before we apply the PSO, we determine whether the predetermined footstep is a feasible footstep command or whether PSO is applied only if the predetermined footstep is unfeasible. The next robot center position $P_{\mathrm{next\_node}}$ is determined using S_{op}, L_{op}, and θ_{op}, which were determined in the previous stages. If the footstep command determined in the stages prior to the continuous optimization stage is a feasible command, $P_{\mathrm{next\_node}}$ is determined based on S, L, and θ. This position now becomes the new starting point, and the previous stages are repeated to reach the destination. The proposed footstep planner is shown in Figure 6.

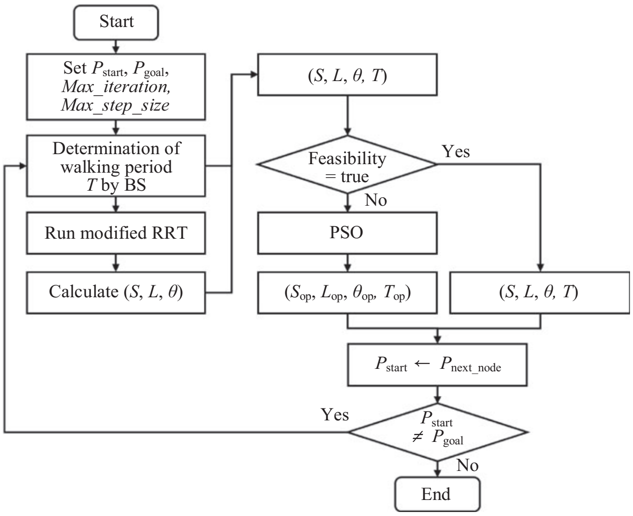


FIGURE 6 Flow chart of the proposed footstep planner

3 | SIMULATIONS

In this section, we explain the reason why the walking period should be taken into considered when determining the footstep sequence through a simulation. We model situations in which the walking period should be considered in three environments and explain the characteristics of each environment. We verify the performance of the proposed footstep planning framework in each environment through a simulation. We use Webots, a dynamic simulator, to verify the feasibility of the footstep command resulting from the proposed footstep planner. Owing to the limitation of the robot hardware, the simulation is performed by setting the ZMP area that is determined to be stable to be smaller than the humanoid robot's foot. The first version of DARwIn-OP is used in the simulation, which is performed in a computing environment equipped with a CPU (Intel (R) CORE (TM) i7 4790, 3.60 GHz).

3.1 | Revolving door

The first obstacle environment to be considered in the walking period is a revolving door. The simulation environment is modeled as an environment in which the range of response for the dynamic obstacles increases by considering the walking period. As the walking speed is increased by further considering the walking period, the range of responses to the obstacle is widened. The "B" area in Figure 1 is modeled as a revolving door. To show the increase in the response range according to changes in the speed of the obstacle more efficiently, we modeled the revolving door as a rod rotating automatically at a constant speed. The radius of the revolving door is 50 cm, and the revolving door rotates clockwise starting from the vertical position. In the revolving door environment, the range that can cope with the dynamic obstacle increases when the walking period is considered. In humanoids, in contrast to mobile robots, there are two factors that determine the movement speed: the stride and walking period. In the existing footstep planning, the walking speed was determined using only the stride. In addition to the stride, it is obvious that the walking speed is more freely controlled, and the range of the walking speed is broadened when the walking period is factored in.

In order to show the wider range corresponding to the dynamic obstacle in the simulation, three case of revolving door speed were considered: 8 °/s, 14 °/s, and 17 °/s. At the revolving door speed of 8 °/s, the humanoid robot could reach the destination without considering the walking period, but in the remaining cases, if the walking period is not considered, it is confirmed that the humanoid robot did not reach the destination but collided with the obstacle. The robot could cope with a revolving door speed of up to 17 °/s by taking into consideration the walking period. The simulation results are presented in Figure 7.

As described in the previous section, it is difficult to create a practical transition model using the existing discrete search approach for footstep planning while considering the walking period. To solve this problem, the footstep command was simultaneously optimized using PSO. The results produced by the binary search and modified RRT algorithm are shown in Figure 8D. The optimized results obtained through PSO are also shown in Figure 8D, and we could verify that the results are obtained without deviating greatly from the results produced by the two stages, as intended. The reason for using PSO is feasible footstep planning. We used a dynamics simulator Webots to demonstrate the feasible footstep planning results. In the case of DARwIn-OP, which is used in the simulation, the ZMP area was set such that it was smaller than the foot because the humanoid robot's center of gravity is low, and its foot is wide as compared to the robot's height. Therefore, we programmed the simulation to stop when an unfeasible footstep is generated. Figure 8 shows that a feasible result is generated, and the robot reached the destination from the start at a revolving door speed of 17 °/s. Table 2 shows the footstep commands generated by the proposed footstep

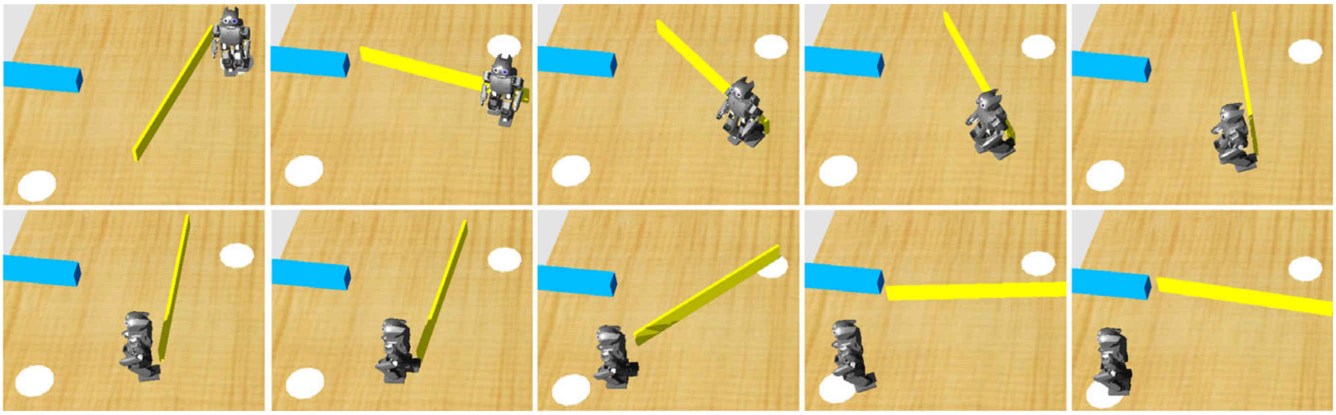
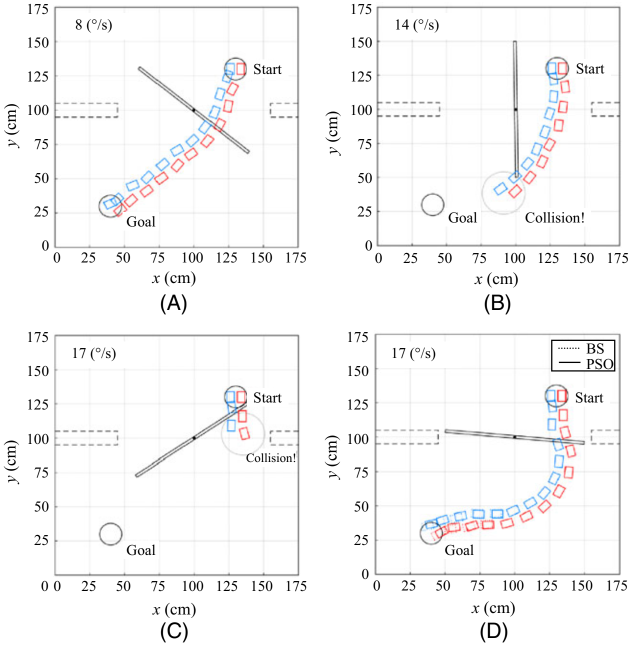


FIGURE 7 Dynamic simulation wherein the angular speed of the revolving door is 17 °/s (left to right and top to bottom)



(A) (B) (C) (D)

FIGURE 8 Footstep planning for a revolving door: (A–C) is the result without considering the walking period. (D) is the result considering the walking period

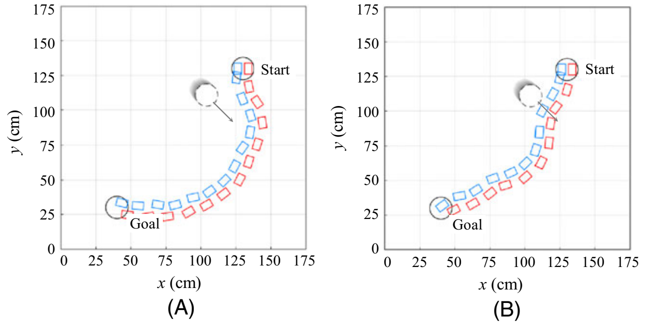
planning framework. Some of the results of changing the walking period are shown in Table 2. Among them, the results optimized using PSO are shown in bold. In the case of the footstep command optimized by the PSO, the footstep commands determined in the previous two stages are presented together, and the footstep command before the optimization is presented in parentheses.

3.2 | General environment

The second environment in which the walking period should be considered is the general environment, which is an environment in which footstep planning results in changes according to the walking priority owing to dynamic obstacles. In this simulation, the walking priority refers to the shorter path length. In real environments, we can observe changes in the walking period in view of the speed of a moving car in the case of crossing a road. We model the general environment as an environment with a dynamic obstacle. We construct the simulation environment such that the dynamic obstacle would travel cross the path of the humanoid robot and disturb its motion. This simulation environment represents the "C" area in Figure 1. We compare cases wherein the walking period is considered to cases wherein the walking period is not considered. The results are presented in Figure 9 and Table 3, and it could be confirmed that more efficient results are obtained with respect to the walking length and number of steps. Although it may be possible to cope with a dynamic obstacle environment without changing the walking period, it could be confirmed that the inefficient result shown in this simulation could be generated.

TABLE 2 Footstep command list in revolving door

Steps	S (cm)	L (cm)	θ (rad)	T_{DS} (s)	T_{SS} (s)
13	–8.87	6.07	0.21	0.20	0.28
14	6.88	7.30	0.09	0.20	0.26
15	–7.62	6.93	0.02	0.20	0.25
16	6.49	7.45	0.14	0.20	0.23
17	**(–9.28)** **–8.28**	**(5.70)** **4.70**	**0.28**	**0.20**	**0.21**
18	7.86	6.79	–0.05	0.20	0.20
19	–8.10	6.65	0.09	0.20	0.18
20	**(9.67)** **8.67**	**(5.30)** **4.30**	**(–0.35)** **–0.36**	**(0.20)** **0.23**	**0.17**
21	**(–8.33)** **–9.33**	**(6.49)** **5.49**	**(0.12)** **0.13**	**(0.20)** **0.10**	**(0.60)** **0.58**
22	7.09	6.05	0.07	0.20	0.60



(A) (B)

FIGURE 9 General environment: (A) is an inefficient path without considering the walking period and (B) is an efficient path considering the walking period

TABLE 3 Simulation results with specific features

	Number of steps	Length of path (cm)
Figure 7A	26	260
Figure 7B	20	190

3.3 | Closed environment

The third environment in which the walking period is considered is the closed environment, which is an environment wherein a robot is surrounded by obstacles, and a collision occurs only when we modify the stride. We model a crowd that is a typical closed environment as a cylinder-shaped dynamic obstacle moving around the robot, which creates a closed environment with a different velocity. This modeled environment represents the "D" area in Figure 1. Figure 10 illustrates the modeled environment. As mentioned earlier, the robot is surrounded by obstacles, and the obstacles

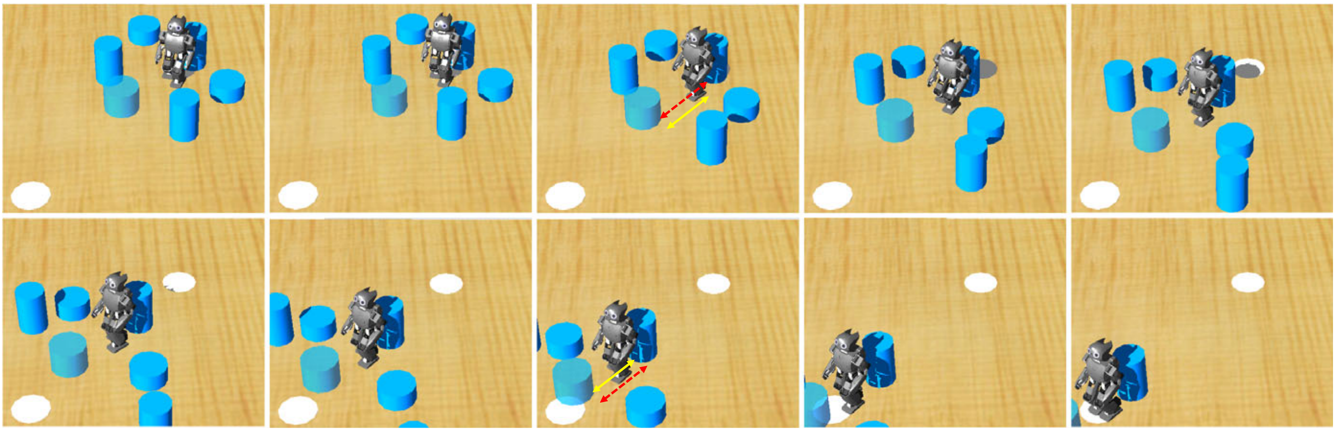


FIGURE 10 Closed environment (left to right and top to bottom). The dotted line in the center figure shows the length of the closed space of the third snapshot in the direction of motion of the robot, and the solid line shows the length of the closed space of the eight snapshots in the direction of motion of the robot

around the robot are configured to move at a different velocity. This led to variations in the size of the closed environment. It could be confirmed that the robot moves from the start point to the destination and changes its walking period as well as stride in order to avoid a collision with the obstacle. Table 4 shows that the walking period and the front stride are reduced in order to deal with the speed of the dynamic obstacle. We could confirm that the stride alone cannot cope with changes in the closed environment.

4 | CONCLUSION AND FUTURE WORKS

The main contribution of this study is the consideration of walking periods in terms of navigation for humanoid robots. We explained why the walking period should be included in the footstep planning stage and proposed a framework for planning the footstep sequence in a short time. The footsteps were planned using the continuous optimization approach and the feasibility was secured through the use of PSO using the p, q parameters of the developed walking pattern generator. To secure a small amount of computation, the walking period was first determined using a binary search. The modified RRT algorithm was introduced to reduce the amount of computation in the path planning stage, and S, L, and θ were set based on the modified RRT algorithm. Through these stages, the search space of the PSO could be reduced, and the computation time was greatly reduced as well.

TABLE 4 Footstep command list in closed environment

Steps	S (cm)	L (cm)	θ (rad)	T_{DS} (s)	T_{SS} (s)
18	**6.24**	–8.66	–0.17	0.20	**0.60**
19	**7.00**	7.50	0.00	0.20	**0.60**
20	**6.24**	–8.66	–0.17	0.20	**0.60**
⋮	⋮	⋮	⋮	⋮	⋮
38	**5.60**	9.38	0.30	0.20	**0.54**
39	**5.70**	–9.28	–0.28	0.20	**0.54**
40	**5.70**	9.28	0.28	0.20	**0.54**

We outlined circumstances in which the walking period should be considered in the footstep planning stage based on the simulation results. We also confirmed that a feasible footstep sequence can be planned at a high speed using the simulation.

In this study, a footstep was determined using the greedy method, which always ensures optimal results each time a footstep is determined but does not guarantee that the overall solution will be optimal. Therefore, our future works will include the optimization of the footstep sequence while taking multi-steps into consideration. In addition, the PSO can be used to study the footstep planner based on the number of steps while considering the walking period, hardware constraints, travel time, energy consumption, and feasibility. Real experiments with DARwIn-OP require the development of a more complicated controller and a vision system for measuring the slip of the robot caused by variations in its walking period; hence, they remain to be explored in future studies.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. 2016R1C1B1006691).

ORCID

Young-Dae Hong http://orcid.org/0000-0002-6174-6442

REFERENCES

1. J. J. Kuffner et al., Footstep planning among obstacles for biped robots, *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Maui, HI, USA, Oct. 29–Nov. 3, 2001, pp. 500–505.
2. J. Chestnutt et al., Planning biped navigation strategies in complex environments, *IEEE/RAS Int. Conf. Human. Robots*, Karlsruhe, Germany, Sept. 29–30, 2003, pp. 13–18.
3. J. J. Kuffner et al., Motion planning for humanoid robots, *The Eleventh International Symposium Robotics Research*, Springer, Heidelberg, Berlin, 2005, pp. 365–374.
4. J. Chestnutt et al., Footstep planning for the Honda ASIMO humanoid, *IEEE Int. Conf. Robot. Autom.*, Barocelona, Spain, Apr. 18–22, 2005, pp. 631–636.
5. J. Chestnutt et al., An adaptive action model for legged navigation planning, *IEEE/RAS Int. Conf. Human. Robots*, Pittsburgh, PA, USA, Nov. 29–Dec. 1, 2007, pp. 196–202.
6. Z. Xia et al., Parameter optimization in non-uniform randomized footstep planning for biped navigation, *IEEE Int. Conf. Mechatronics Autom.*, Changchun, China, Aug. 9–12, 2009, pp. 3745–3750.
7. Z. Xia, J. Xiong, and K. Chen, *Global navigation for humanoid robots using sampling-based footstep planners*, IEEE/ASME Trans. Mechatronics **16** (2011), no. 4, 716–723.
8. N. Perrin et al., *Fast humanoid robot collision-free footstep planning using swept volume approximations*, IEEE Trans. Robot. **28** (2012), no. 2, 427–439.
9. Y.-D. Hong et al., *Evolutionary multiobjective footstep planning for humanoid robots*, IEEE Trans. Syst., Man, Cybern. C, Appl. Rev. **41** (2011), no. 4, 520–532.
10. Y.-D. Hong and J.-H. Kim, *An evolutionary optimized footstep planner for the navigation of humanoid robots*, Int. J. Humanoid Robot. **9** (2012), no. 1, 1250005:1–1250005:18.
11. K.-B. Lee and J.-H. Kim, *Multiobjective particle swarm optimization with preference-based sort and its application to path following footstep optimization for humanoid robots*, IEEE Trans. Evol. Comput. **17** (2013), no. 3, 755–766.
12. R. Deits and R. Tedrake, Footstep planning on uneven terrain with mixed-integer convex optimization, *IEEE/RAS Int. Conf. Human. Robot.*, Madrid, Spain, Nov. 18–20, 2014, pp. 279–286.
13. K.-B. Lee, H. Myung, and J.-H. Kim, *Online multiobjective evolutionary approach for navigation of humanoid robots*, IEEE Trans. Ind. Electron. **62** (2015), no. 9, 5586–5597.
14. B. Stephens and C. Atkeson, Push recovery by stepping for humanoid robots with force controlled joints, *IEEE/RAS Int. Conf. Human. Robot.*, Nashville, TN, USA, Dec. 2010, pp. 52–59.
15. M. Morisawa et al., A biped pattern generation allowing immediate modification of foot placement in real-time, *IEEE /RAS Int. Conf. Human. Robot.*, Genova, Italy, Dec. 4–6, 2006, pp. 581–586.
16. T. Takubo et al., Emergent walking stop using 3-D ZMP modification criteria map for humanoid robot, *IEEE Int. Conf. Robot. Autom.*, Roma, Italy, Apr. 10–14, 2007, pp. 2676–2681.
17. P. Kryczka et al., Online regeneration of bipedal walking gait pattern optimizing footstep placement and timing, *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Hamburg, Germany, Sept. 28–Oct. 2, 2015, pp. 3352–3357.
18. M. Khadiv et al., Step timing adjustment: a step toward generating robust gaits, *IEEE Int. Conf. Human. Robots*, Cancun, Mexico, Nov. 15–17, 2016, pp. 35–42.
19. M. Brandao et al., *Footstep planning for slippery and slanted terrain using human-inspired models*, IEEE Trans. Robot. **32** (2016), no. 4, 868–879.
20. B.-J. Lee et al., *Modifiable walking pattern of a humanoid robot by using allowable ZMP variation*, IEEE Trans. Robot. **24** (2008), no. 4, 917–925.
21. J. Kennedy and R. Eberhart, Particle swarm optimization, *IEEE Int. Conf. Neural Netw.*, Perth, Australia, Nov. 27–Dec. 1, 1995, pp. 1942–1948.

AUTHOR BIOGRAPHIES

Woong-Ki Lee received his BS and MS degrees in electrical and computer engineering from Ajou University, Suwon, Rep. of Korea in 2015 and 2017, respectively, where he is currently working toward the PhD degree. His current research interests include humanoid robotics, especially footstep planning and bipedal walking pattern generation and control.

In-Seok Kim received his BS and MS degrees in electrical and computer engineering from Ajou University, Suwon, Rep. of Korea in 2016 and 2018, respectively. He is currently a research engineer for Hyundai Robotics, Yongin, Rep. of Korea. His current research interests include humanoid robotics, especially bipedal walking pattern generation & control, and path planning for robot navigation.

Young-Dae Hong received his BS, MS, and PhD degrees in electrical engineering from KAIST, Daejeon, Rep. of Korea, in 2007, 2009, and 2013, respectively. Since 2014, he has been with the Department of Electrical and Computer Engineering, Ajou University, Suwon, Rep. of Korea, where he is currently an assistant professor. His current research interests include humanoid robotics, especially bipedal walking pattern generation and control, footstep planning, and optimization-based robot control.