Experimental Study on Modifiable Walking Pattern Generation for Handling Infeasible Navigational Commands

Young-Dae Hong* and Bumjoo Lee[†]

Abstract – To accommodate various navigational commands, a humanoid should be able to change its walking motion in real time. Using the modifiable walking pattern generation (MWPG) algorithm, a humanoid can handle dynamic walking commands by changing its walking period, step length, and direction independently. If the humanoid is given a command to perform an infeasible movement, the algorithm substitutes the infeasible command with a feasible one using binary search. The feasible navigational command is subsequently translated into the desired center-of-mass (CM) state. Every sample time CM reference is generated using a zero-moment-point (ZMP) variation scheme. Based on this algorithm, various complex walking patterns can be generated, including backward and sideways walking, without detailed consideration of the feasibility of the navigational commands. In a previous study, the effectiveness of the MWPG algorithm was verified by dynamic simulation. This paper presents experimental results obtained using the small-sized humanoid robot platform DARwIn-OP.

Keywords: Humanoid robot, Bipedal robot, Walking pattern, Gait, Locomotion

1. Introduction

Walking control is a prerequisite for bipedal locomotion. To address this need, real time algorithms for walking patterns have been proposed in many studies. Because of the innate difficulty of solving the differential algebraic equations (DAEs) that represent aspects of dynamic motion, including the contact problem, simplified versions of the equations have been devised for practical use. One such simplification, the linear inverted pendulum model, has attracted the attention of many researchers because of its ease of implementation [1-4].

The modifiable walking pattern generation (MWPG) algorithm also belongs in this class. Modification during walking is regarded as a transition between two periodic walking patterns that each execute a navigational command. In this algorithm, the robot is modeled as a linear inverted pendulum, and the equations of motion between the zero-moment-point (ZMP) and center-of-mass (CM) states are derived in a simple closed form. When a new navigational command is received, it is converted to the desired CM state, which is a representative snapshot of the periodic CM motion. To achieve this desired CM state, a ZMP variation technique is adopted that uses two simple ZMP functions [5, 6]. Sometimes, a navigator may command a robot to take a step that is infeasible given the robot's current state. In this case, the command is replaced with a feasible one identified using a binary search algorithm, and CM motion is successfully generated. The effectiveness of

the proposed algorithm has been verified by computer simulation [7]. In this study, to demonstrate the practicality of the algorithm, several experiments were carried out using an actual humanoid robot, and the results were analyzed.

Because the MWPG algorithm corresponds to the specific way of generating a walking pattern for a feed-forward control reference, it is obviously necessary to implement some type of feedback controller to compensate for the disturbance. Approaches have been proposed for compensating for the disturbance using heuristic damping control [8, 9], the dynamics between a robot and its environment [10, 11], and landing force control [12, 13]. In this study, the torques and ground reaction force were measured using force-sensing resistors (FSRs) on each foot, and the foot of the robot was modeled using three virtual spring-damper (VSD) models for the purpose of disturbance compensation.

This paper is organized as follows. Section 2 describes the key concepts and flow of the MWPG algorithm. Section 3 describes the VSD model for compliance control. The humanoid platform is described in Section 4, and experimental results obtained using the platform are presented and discussed. Concluding remarks are made in Section 5.

2. Modifiable Walking Pattern Generation

2.1 Walking pattern modification

The MWPG algorithm is based on the reduced dynamics model, which is simplified so that the resulting equations

[†] Corresponding Author: Dept. of Electrical Engineering, Myongji University, Korea. (bjlee@mju.ac.kr)

* Dept. of Electrical and Computer Engineering, Ajou University, Korea. (ydhong@ajou.ac.kr)

Received: March 16, 2015; Accepted: June 9, 2015

of motion can be derived in closed form. The solutions of the equations are divided into two parts: the homogeneous part and the particular part. The homogeneous solution part describes the CM motion given by the initial state, and the particular solution part relaxes the motion predetermined by the homogeneous solution part. By allowing variation in the ZMP for the bounded region, the position and velocity of the CM can be changed independently throughout the single-support phase.

In the MWPG algorithm, the position and the weighted velocity of the CM, which are state-space variables of the equation of motion that represents the walking pattern at a given time, are defined as the *walking state (WS)*. In addition, the minimal set of parameters that includes the single- and double-support times, forward and side step lengths, and walking directions is defined as a *command state (CS)*.

Definition 1 The CS is a minimal set that represents navigational commands as follows:

$$\mathbf{c} \triangleq [T_{sl} \ T_{sr} \ T_{dl} \ T_{dr} \ F_l \ F_r \ S_l \ S_r \ \theta_l \ \theta_r]^T$$

where T_{sl} , T_{dl} , F_l , S_l , and θ_l represent the single-support time, double-support time, forward step length, side step length, and walking direction, respectively, for the left side. Similarly, T_{sr} , T_{dr} , F_r , S_r , and θ_r represent these parameters for the right side.

A walking pattern is modified in two ways: generation of another periodic pattern that satisfies a given new CS and transitional motion of the WS from the current state to the desired state along the new trajectory. The *desired WS* is uniquely determined from a given CS and the *current WS*, which is a representative snapshot for a periodic walking pattern based on the CS. After the *desired WS* is computed, the ZMP trajectory is adjusted to transfer the *current WS* to the desired one. By means of the ZMP variation with closed-form functions, walking patterns are modified as intended by the CS. Then the CS is infeasible, the CS and subsequent *desired WS* are simultaneously modified using a binary search algorithm.

After initialization, the CS is updated as the mean of the previous CS and the desired CS. Because the current WS is on the walking pattern generated by the previous CS, the *modified CS* is always feasible. Therefore, when the modified *desired WS* is feasible, the modified *desired WS* becomes the first intended infeasible one. Otherwise, it becomes the previous one. This continues until the maximum number of iterations is reached. Because of the simplicity of this *binary search algorithm*, it is sufficiently fast that it can be executed in real time.

2.2 Algorithm flow chart

Walking is a periodic motion involving the alternation between single-support (SS) and double-support (DS) motions.

Fig. 1. Flow chart of the MWPG algorithm: (a) Overall sequence; (b) SS motion; (c) DS motion.

phases. In the SS phase, ZMP variation is executed to modify the walking pattern, while in the DS phase, the CM follows the predetermined trajectory. When a command to walk is given in the standing-still state, the MWPG algorithm executes the initial DS motion. The subsequent SS and DS motions alternate in the finite-state machine (FSM) until a stop instruction is given, as shown in Fig. 1(a). Note that the CS can be updated at any time. However, for the sake of simplicity in implementation, this is done only in the last DS phase.

As shown in Fig. 1(b), the feasibility of the given CS is checked during the SS by examining the parameters of the ZMP functions, which are calculated analytically. When the CS is infeasible, simple binary searching is initiated and continues iteratively until a feasible CS is obtained or the maximum number of iterations is reached. If all of the parameters for the ZMP trajectories are between the minimum and the maximum values, ZMP variations are executed to transfer the current WS to the desired one. Or not, the given CS is regarded as unachievable. When a feasible CS is not found before the maximum number of iterations is reached, the previous CS is utilized instead. The desired WS is then computed to transfer the CM to the desired state. Every sample time reference WS is calculated for the control reference, and the subsequent joint reference is computed by inverse kinematics.

In the DS motion, the walking pattern is not modified but rather follows the previous trajectory. As with the SS motion, every sample time reference WS and the subsequent joint reference is calculated as shown in Fig. 1(c).

3. Compliance Control

When a humanoid robot is walking, the sole of the swing leg contacts the terrain, and as a result, a disturbance between the robot and the terrain occurs. VSD models are used to compensate for such disturbances, as shown in Fig. 2.

Two rotational VSD models are utilized for the x -axis and y -axis motions of the foot, and one linear VSD model is utilized for the z -axis motion of the foot. The rotational foot displacements/velocities, $\Delta\theta_x / \Delta\dot{\theta}_x$ and $\Delta\theta_y / \Delta\dot{\theta}_y$,

Fig. 2. VSD models for the foot.

and the linear foot displacement/velocity, $\Delta d_z / \Delta \dot{d}_z$, are given by the following equations:

$$\begin{aligned} T_x &= k_x \Delta\theta_x + b_x \Delta\dot{\theta}_x \\ T_y &= k_y \Delta\theta_y + b_y \Delta\dot{\theta}_y \\ F_z &= k_z \Delta d_z + b_z \Delta \dot{d}_z \end{aligned} \quad (1)$$

where T_x , T_y , and F_z are the x - and y -axis torques and ground reaction force (GRF) on the foot, respectively, and k_x/b_x , k_y/b_y , and k_z/b_z are the spring/damper coefficients. From (1), the state-space equation of the x -axis VSD model can be written as follows:

$$\Delta\theta_x(k+1) = \left\{ \frac{b_x}{\Delta t} / \left(k_x + \frac{b_x}{\Delta t} \right) \right\} \Delta\theta_x(k) + \left\{ 1 / \left(k_x + \frac{b_x}{\Delta t} \right) \right\} T_x(k) \quad (2)$$

where Δt is a sample time. Note that the state-space equations of the y -axis and z -axis VSD models are obtained in a similar fashion. From (2), the transfer function of the x -axis VSD model is obtained as follows:

$$\frac{\Delta\theta_x}{T_x} = \frac{1/b_x}{s + k_x/b_x} \quad (3)$$

Fig. 3. Step responses of x -axis VSD model. (a) $\Delta\theta_x$. (b) $\Delta\dot{\theta}_x$.

Fig. 3 shows the step responses of $\Delta\theta_x / \Delta\dot{\theta}_x$, from (3). As shown in the figure, the responses rapidly reach a steady state, and they are robust to disturbance. The spring and damper coefficients are determined experimentally, based on the displacements that occur when a unit torque and force are applied and on the time constant b_x/k_x in (3).

4. Experimental Results

The effectiveness of the proposed algorithm was verified using the small-sized humanoid platform DARwIn-OP, shown in Fig. 4(a). The height and weight of this platform

Fig. 4. DARwIn-OP: (a) Small-sized humanoid robot; (b) Overall structure.

Table 1. Experiment parameters.

Z_c	F_x	F_y	d_y	P_{max}	P_{min}	Q_{max}	Q_{min}
20.35	10.4	6.6	7.4	1.25	-1.25	0.75	-0.75

are 45.5 cm and 2.8 kg, respectively. Fig. 5(b) presents the overall structure of the DARwIn-OP system. DARwIn-OP has twenty degrees of freedom (DOFs) and MX-28 actuators. An Intel Atom Z530 central processing unit (CPU) is used as the main controller, which communicates with the sub controller through universal serial bus (USB) connections. Four FSRs are located on the sole of each foot to measure ground reaction force measurements.

Table 1 shows the CM height (Z_c), foot size (F_x , F_y), distance between foot centers (d_y), and ZMP allowable region ($P_{max/min}$, $Q_{max/min}$). Note that all lengths are given in centimeters. Table 2 shows the actual VSD coefficients implemented in DARwIn-OP. The VSD coefficients were determined on the assumption that the robot walks on the rigid surface.

Table 2. VSD coefficients.

x-axis VSD model		y-axis VSD model		z-axis VSD model	
k_x	100	k_y	100	k_z	1000
b_x	20	b_y	20	b_z	10

4.1 Backward walking

Backward walking patterns were produced using the following CS list:

(1) Initial CS

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 6.0 \ 6.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(2) After the third step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 - 5.0 - 5.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(3) After the seventh step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

In the CS list, times and lengths are given in seconds and centimeters, respectively, and angles are given in degrees. Fig. 5 shows the backward walking pattern generated using the proposed method. The thick and thin curves represent

Fig. 5. Walking pattern (backward walking).

Table 3. Sagittal step lengths during backward walking.

Step	Current x-axis position	Previous x-axis position	Sagittal step length
1	6.0	0.0	6.0
2	12.0	6.0	6.0
3	18.0	12.0	6.0
4	14.4	18.0	-3.6
5	9.4	14.4	-5.0
6	4.4	9.4	-5.0
7	-0.6	4.4	-5.0
8	-5.6	-0.6	-5.0
9	-5.6	-5.6	0.0

the CM trajectories in the SS and DS phases, respectively. The thick and thin rectangles represent the foot polygon and ZMP region, respectively. The circles and numbers indicate the center of the foot and the step number, respectively. Table 3 lists the current and previous x-axis positions of the foot centers and the sagittal step lengths at each footstep during the backward walking. According to the list of CS commands, the sagittal step length was changed to -5.0 cm after the third step of the backward walking. However, the sagittal step length was -3.6 (=14.4-18.0) cm at the fourth step, and the commanded step length of -5.0 (=9.4-14.4) cm was achieved at the fifth step. The reason for this is that the infeasible CS commanded after the third step was replaced with a feasible CS by the proposed binary search algorithm.

Fig. 6 presents snapshots of the backward walking experiment result. The humanoid robot successfully walked according to the feasible CSs identified using the proposed method. The ZMP trajectories during the walking experiment were measured and are shown in Fig. 7. The

Fig. 6. Snapshots of backward walking experiment results (left to right, top to bottom).

Fig. 7. Measured ZMP trajectories during backward walking experiment.

ZMP trajectories along the x- and y-axes followed the foot trajectories with little variation.

4.2 Sideways walking

To demonstrate the effectiveness of the binary search algorithm, sideways walking patterns generated from the following CS list were also examined:

(1) Initial CS

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 2.0 \ 2.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(2) After the second step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 2.0 \ 2.0 \ 7.4 - 12.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(3) After the seventh step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 2.0 \ 2.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(4) After the eighth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

Fig. 8 shows the footprints and CM trajectories of the sideways walking pattern generated from the CS list. Table 4 lists the current and previous y-axis positions of the foot centers and the lateral step lengths at each footstep during the sideways walking. The lateral step length of the right leg was changed to -12.4 cm after the second step. However, the commanded step length was not achieved at the third step ($-11.6 (= -4.2 - 7.4)$ cm). The commanded step length of $-12.4 (= -9.2 - 3.2)$ cm was achieved at the fifth step. Figs. 9 and 10 present snapshots of the sideways walking experiment results and the measured ZMP trajectories. As in the backward walking experiment, the

Fig. 8. Walking pattern (sideways walking).

Fig. 9. Snapshots of sideways walking experiment results (left to right, top to bottom).

Table 4. Lateral step lengths during sideways walking.

Step	Current y-axis position	Previous y-axis position	Lateral step length
1	0.0	7.4	-7.4
2	7.4	0.0	7.4
3	-4.2	7.4	-11.6
4	3.2	-4.2	7.4
5	-9.2	3.2	-12.4
6	-1.8	-9.2	7.4
7	-14.2	-1.8	-12.4
8	-6.8	-14.2	7.4
9	-14.2	-6.8	-7.4

humanoid robot walked according to the feasible CSs, and the ZMP trajectories followed the foot trajectories successfully.

4.3 Complex walking

As previously stated, the MWPG algorithm can generate a walking pattern even if the navigational command is complex. To demonstrate this ability, two complex walking experiments were carried out. In the first walking pattern, the CS included forward, backward, sideways, and turning motions:

(1) Initial CS

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 6.0 \ 6.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

Fig. 10. Measured ZMP trajectories during sideways walking experiment.

(2) After the second step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 3.0 \ 3.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(3) After the fourth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 3.0 \ 3.0 \ 7.4 - 8.4 \ 0.0^\circ - 20.0^\circ]^T$$

(4) After the fifth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 - 3.0 - 3.0 \ 9.4 - 7.4 - 30.0^\circ \ 0.0^\circ]^T$$

(5) After the sixth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 - 3.0 - 3.0 \ 7.4 - 8.4 \ 0.0^\circ - 20.0^\circ]^T$$

(6) After the seventh step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 - 6.0 - 6.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(7) After the ninth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

Fig. 11 shows the first complex walking pattern, including forward, backward, and sideways walking and turning motions. After the second step, the sagittal step length was shortened. The walking direction changed to the right, with a different lateral step length after the fourth step. After the fifth step, backward walking was initiated by changing the sagittal and lateral step lengths and the walking direction.

In the second walking pattern, asymmetric forward walking motions were generated from the following CS list:

(1) Initial CS

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 6.0 \ 6.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(2) After the third step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(3) After the fourth step

Fig. 11. Walking pattern (complex walking: case 1).

Fig. 12. Walking pattern (complex walking: case 2).

Fig. 13. Measured ZMP trajectories during complex walking experiment: case 1.

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 6.0 \ 6.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(4) After the fifth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

(5) After the sixth step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 6.0 \ 6.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

Fig. 14. Measured ZMP trajectories during complex walking experiment: case 2.

(6) After the seventh step

$$\mathbf{c} = [0.25 \ 0.25 \ 0.2 \ 0.2 \ 0.0 \ 0.0 \ 7.4 - 7.4 \ 0.0^\circ \ 0.0^\circ]^T$$

Fig. 12 shows the second complex walking pattern. In this case, after the third step, the sagittal step length of the left leg was changed to 0.0 cm, and the sagittal step length of the right leg was changed to 6.0 cm. Figs. 13 and 14 present the ZMP trajectories measured during the complex walking experiments. The ZMP trajectories followed the foot trajectories.

5. Conclusions

The MWPG algorithm is an effective tool for controlling the walking motion of a humanoid robot in real time. Use of this algorithm makes it possible to modify a walking pattern even when infeasible commands are issued. A FSM-based flow chart of the algorithm is presented in this paper. By using an actual humanoid platform, the experimental results demonstrated that the algorithm substitutes infeasible navigational commands with feasible ones so that walking motions are successfully generated. A conventional compliance control scheme was also adopted to compensate for disturbances. In future research, the search algorithm for feasible replacements for infeasible commands will be enhanced.

Acknowledgement

This work was supported by the Industrial Strategic

technology development, 10047635, Development of Hydraulic Robot Control Technology based on Accurate and Fast Force Control for Complex Tasks, funded By the Ministry of Trade, Industry & Energy (MI, Korea).

References

- [1] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "A realtime pattern generator for Biped walking," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, vol. 1, pp. 31-37.
- [2] T. Sugihara, Y. Nakamura, and H. Inoue, "Realtime humanoid motion generation through ZMP manipulation based on inverted pendulum control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, vol. 2, pp. 1404-1409.
- [3] Y.-D. Hong and J.-H. Kim, "3-D command state-based modifiable bipedal walking on uneven terrain," *IEEE/ASME Trans. Mechatron.*, vol. 18, no. 2, pp. 657-663, Apr. 2013.
- [4] Y.-D. Hong, B.-J. Lee, and J.-H. Kim, "Command state-based modifiable walking pattern generation on an inclined plane in pitch and roll directions for humanoid robots," *IEEE/ASME Trans. Mechatron.*, vol. 16, no. 4, pp. 783-789, Aug. 2011.
- [5] B.-J. Lee, D. Stonier, Y.-D. Kim, J.-K. Yoo, and J.-H. Kim, "Modifiable walking pattern of a humanoid robot by using allowable ZMP variation," *IEEE Trans. Robot.*, vol. 24, no. 4, pp. 917-925, Apr. 2008.
- [6] B.-J. Lee and K. I. Kim, "Modifiable walking pattern generation handling infeasible navigational commands for humanoid robots," *J. Elect. Eng. Technol.*, vol. 9, no. 1, pp. 344-351, 2014.
- [7] Y.-D. Hong, K.-B. Lee, and B.-J. Lee, "Dynamic simulation of modifiable walking pattern generation to handle infeasible navigational commands for humanoid robots," *J. Elect. Eng. Technol.*, 2015 (submitted).
- [8] Q. Huang and Y. Nakamura, "Sensory reflex control for humanoid walking," *IEEE Trans. Robot. Autom.*, vol. 21, no. 5, pp. 997-984, Oct. 2005.
- [9] S. Kajita, T. Nagasaki, K. Kaneko, K. Yokoi, and K. Tanie, "A running controller of humanoid biped HRP-2 LR," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 618-624.
- [10] J.-H. Park, "Impedance control for biped robot locomotion," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 870-882, Dec. 2001.
- [11] H.-O. Lim, S. Setiawan, and A. Takanishi, "Position-based impedance control of a biped humanoid robot," *Adv. Robot.*, vol. 18, no. 4, pp. 415-435, 2004.
- [12] Y.-D. Kim, B.-J. Lee, J.-H. Ryu, and J.-H. Kim, "Landing force control for humanoid robot by time domain passivity approach," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1294-1301, Dec. 2007.

- [13] Y. -D. Hong, C.-S. Park, and J. -H. Kim, "Stable bipedal walking with a vertical center of mass motion by an evolutionary optimized central pattern generator," *IEEE Trans. Ind. Electron.*, vol. 61, no. 5, pp. 2246-2355, May. 2014.

Young-Dae Hong received the B.S., M.S., and Ph.D. degrees in electrical engineering from KAIST, Daejeon, Korea, in 2007, 2009, and 2013 respectively. Since 2014, he has been with the department of electrical and computer engineering, Ajou University, Suwon, Korea, where he is currently an Assistant Professor. His current research interests include humanoid robotics, especially in bipedal walking pattern generation, footstep planning, multi/single objective evolutionary optimization, and biologically inspired control.

Bumjoo Lee received the B.S. degree in electrical engineering from Yonsei University, Seoul, Korea, in 2002, and the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea, in 2004 and 2008, respectively. Since 2012, he has been with the Department of Electrical Engineering, Myongji University, Korea, where he is currently an Assistant Professor. His research interests include the areas of Humanoid Robotics, especially in motion planning and control algorithm.