Footstep Planning Based on Univector Field Method for Humanoid Robot

Youngdae Hong and Jong-Hwan Kim

Department of Electrical Engineering and Computer Science, KAIST
Daejeon, Korea
{ydhong,johkim}@rit.kaist.ac.kr
http://rit.kaist.ac.kr

Abstract. This paper proposes a footstep planning algorithm based on univector field method optimized by evolutionary programming for humanoid robot to arrive at a target point in a dynamic environment. The univector field method is employed to determine the moving direction of the humanoid robot at every footstep. Modifiable walking pattern generator, extending the conventional 3D-LIPM method by allowing the ZMP variation while in single support phase, is utilized to generate every joint trajectory of a robot satisfying the planned footstep. The proposed algorithm enables the humanoid robot not only to avoid either static or moving obstacles but also step over static obstacles. The performance of the proposed algorithm is demonstrated by computer simulations using a modeled small-sized humanoid robot HanSaRam (HSR)-VIII.

Keywords: Footstep planning, univector field method, evolutionary programming, humanoid robot, modifiable walking pattern generator.

1 Introduction

These days research on a humanoid robot has made rapid progress for dexterous motions with the hardware development. Various humanoid robots have demonstrated stable walking with control schemes [1]-[5]. Considering the future of the humanoid robot as a service robot, research on navigation in indoor environments such as homes and offices with obstacles is now needed.

In indoor environments, most of research on navigation has been carried out for differential drive mobile robots. The navigation method of the mobile robots is categorized into separated navigation and unified navigation. The separated navigation method, such as structural navigation and deliberative navigation, separates path planning and path following as two isolated tasks. In the path planning step, a path generation algorithm is developed which connects the staring point with the end point without crossing the obstacles. To find the shortest path many searching algorithms such as A* algorithm and dynamic programming have been applied [6]. On the other hand, in unified navigation method such as the artificial potential field method [7], [8], the path planning step and the path following step are unified in one task.

In the navigation research, differential drive mobile robots make a detour to avoid obstacles to arrive at a goal position. On the other hand, humanoid robots are able to

J.-H. Kim et al. (Eds.): FIRA 2009, LNCS 5744, pp. 125–134, 2009.

traverse obstacles by their legs. When they move around in an environment, positions of their footprints are important as there are obstacles. Thus, the study of footstep planning for humanoid robots is an important research issue.

As research on footstep planning, algorithm obtaining information of obstacle's shape and location by sensors was presented [9]. Through obtained information, a robot determines its step length which is predefined as three type step lengths and its motion such as circumventing, stepping over or stepping on obstacles. Also, an algorithm finding alternative path employing A* by heuristic cost function was developed [10]. Stable region of robot's footprints is predetermined and then a few of placements of them are selected as a discrete set. This algorithm checks collision between a robot and obstacles by 2D polygon intersection test. Human-like strategy for footstep planning was also presented [11].

In this paper, a footstep planning algorithm based on the univector field method for humanoid robot is proposed. The univector field method is one of the unified navigation methods, which is designed for fast differential drive mobile robots to enhance performances. Using this method, robot can navigate rapidly to the desired position and orientation without oscillations and unwanted inefficient motions [12], [13]. The footstep planning algorithm determines moving direction of a humanoid robot in real time and has low computing cost by employing the univector field method. Besides, it is able to modify foot placement depending on obstacle's position. Inputting the moving direction and step length of a robot at every footstep to modifiable walking pattern generator [14], every joint trajectory is generated. The proposed algorithm generates an evolutionary optimized path by evolutionary programming (EP) considering hardware limit of a robot and makes a robot arrive at a goal with desired direction. Computer simulations are carried out by a model of HanSaRam (HSR)-VIII which is a small-sized humanoid robot developed in Robot Intelligence Technology (RIT) Lab, KAIST.

The rest of the paper is organized as follows: Section 2 describes an overview of univector field method and Section 3 explains MWPG. In Section 4 a footstep planning algorithm is proposed. Computer simulation results are presented in Section 5. Finally concluding remarks follow in Section 6.

2 Univector Field Method

The univector field method is one of path planning methods developed for a differential drive mobile robot. The univector field consists of *move-to-goal univector field* which leads a robot to move to a destination and *avoid-obstacle univector field* which makes a robot avoid obstacles. Its moving direction is decided by combining move-to-goal univector field and avoid-obstacle univector field. The univector field method requires relatively low computing power because it does not generate a whole path from a start point to a destination before moving, but generates a moving direction decided at every step in real time. In addition, it is easy to plan a path in a dynamic environment with moving obstacles. Thus, this method of path planning is adopted and extended for a humanoid robot.

2.1 Move-to-Goal Univector Field

The move-to-goal univector field is defined as

$$\mathbf{v}_{muf} = [-\cos(\theta_{muf}) \; -\sin(\theta_{muf})]^T, \tag{1}$$

where

$$\theta_{muf} = \cos^{-1}(\frac{p_x - g_x}{d_{goal}}), d_{goal} = \sqrt{(p_x - g_x)^2 + (p_y - g_y)^2},$$

θ_{muf} is the angle from x-axis of the goal at robot's position, d_{goal} is the distance between the center of a goal and robot's position, and (p_x, p_y) and (g_x, g_y) are the robot's position and the goal position, respectively.

2.2 Avoid-Obstacle Univector Field

The avoid-obstacle univector field is defined as

$$\mathbf{v}_{auf} = [\cos(\theta_{auf}) \; \sin(\theta_{auf})]^T, \tag{2}$$

where

$$\theta_{auf} = \cos^{-1}(\frac{p_x - o_x}{d_{ob}}), d_{ob} = \sqrt{(p_x - o_x)^2 + (p_y - o_y)^2},$$

θ_{auf} is the angle from x-axis of an obstacle at robot's position, d_{ob} is the distance between the center of an obstacle and robot's position and (o_x, o_y) is the position of an obstacle.

Total univector field is determined by properly combining the move-to-goal univector field and the avoid-obstacle univector field. Total univector $\mathbf{v}_{tuf}$ is defined as

$$\mathbf{v}_{tuf} = w_{muf}\mathbf{v}_{muf} + w_{auf}\mathbf{v}_{auf}, \tag{3}$$

where w_{muf} and w_{auf} represent the scale factor of the move-to-goal univector field and the avoid-obstacle univector field, respectively.

3 Modifiable Walking Pattern Generator

The modifiable walking pattern generator (MWPG) extended the conventional 3D-LIPM method by allowing the ZMP variation while in single support phase. In the conventional 3D-LIPM without the ZMP variation, only the homogeneous solutions of the 3D-LIPM dynamic equation were considered. However, considering the particular solutions, more extensive and unrestricted walking patterns could be generated by allowing the ZMP variation. The solutions with both homogeneous and particular parts are as follows:
Sagittal motion:

$$\begin{bmatrix} x_f \\ v_f T_c \end{bmatrix} = \begin{bmatrix} C_T & S_T \\ S_T & C_T \end{bmatrix} \begin{bmatrix} x_i \\ v_i T_c \end{bmatrix} - \frac{1}{T_c} \begin{bmatrix} \int_0^T S_t \bar{p}(t)dt \\ \int_0^T C_t \bar{p}(t)dt \end{bmatrix}, \tag{4}$$

Lateral motion:

$$\begin{bmatrix} y_f \\ w_f T_c \end{bmatrix} = \begin{bmatrix} C_T & S_T \\ S_T & C_T \end{bmatrix} \begin{bmatrix} y_i \\ w_i T_c \end{bmatrix} - \frac{1}{T_c} \begin{bmatrix} \int_0^T S_t \bar{p}(t)dt \\ \int_0^T C_t \bar{p}(t)dt \end{bmatrix}, \tag{5}$$

where $(x_i, v_i)/(x_f, v_f)$ and $(y_i, w_i)/(y_f, w_f)$ represent initial/final position and velocity of the CM in the sagittal and lateral plane, respectively. S_t and C_t are defined as $\cosh(t/T_c)$ and $\sinh(t/T_c)$ with time constant $T_c = \sqrt{Z_c/g}$. The functions $p(t)$ and $q(t)$ are ZMP trajectories for the sagittal and lateral planes, respectively. $\bar{p}(t) = p(T-t)$ and $\bar{q}(t) = q(T-t)$. Through the variation of the ZMP, the walking state (WS), which is the state of the point mass in the 3D-LIPM represented in terms of CM position and linear velocity can be moved to the desired WS in the region of possible trajectories expanded by applying the particular solutions. By means of the MWPG, a humanoid robot can change both sagittal and lateral step lengths, rotation angle of ankles and the period of the walking pattern [14].

4 Footstep Planning Algorithm

In this section, a footstep planning algorithm for a humanoid robot is described. It decides moving orientation at every footstep by univector field navigation method. Using the determined orientations, it calculates exact foot placement. Subsequently, by inputting the moving direction and step length of a robot at every footstep by proposed footstep planning algorithm to MWPG, every joint trajectory is generated to satisfy the planned footstep.

4.1 Path Planning

To apply univector field method to the path generation of a humanoid robot, the following three issues are considered. To generate a natural and effective path, obstacle's boundary and virtual obstacle [15] are introduced to the avoid-obstacle univector field considering the obstacle's size and movement, respectively. Also, a hyperbolic spiral univector field is developed as a move-to-goal univector field in order to reach a destination with a desired orientation [13].

Boundary of Avoid-Obstacle Univector Field. The repulsive univector field by obstacles is not generated at every position but generated in a restricted range by applying a boundary to the avoid-obstacle univector field. Also, the more the robot's position becomes distant from the center of an obstacle, the more the magnitude of the repulsive univector field decreases linearly. Consequently, a robot is not influenced the repulsive univector field at the region which is away from the boundary of obstacles. Considering this boundary effect, the avoid-obstacle univector $\mathbf{v}_{auf}$ is defined as

$$\mathbf{v}_{auf} = k_b[\cos(\theta_{auf}) \ \sin(\theta_{auf})]^T \tag{6}$$

where

$$k_b = \frac{d_{boun} - (d_{ob} - o_{size})}{d_{boun}},$$

o_{size} is the obstacle's radius, d_{boun} is the size of boundary and k_b is a scale factor. By introducing the boundary into the avoid-obstacle univector field, an effective path is generated.

Virtual Obstacle. The virtual obstacle is defined by introducing a shifting vector to the center position of a real obstacle, where the direction of shifting vector is opposed to the robots moving direction and the magnitude is proportional to the robots moving velocity. Then, the position of the center of the virtual obstacle is obtained as

$$[o_x^{virtual}\ o_y^{virtual}]^T = [o_x^{real}\ o_y^{real}]^T + \mathbf{s}, \tag{7}$$

$$\mathbf{s} = -k_v \mathbf{v}_{robot},$$

where $(o_x^{virtual}, o_y^{virtual})$ is the virtual obstacle's position, (o_x^{real}, o_y^{real}) is the real obstacle's position, $\mathbf{s}$ is the shifting vector, k_v is the scale factor of the virtual obstacle and $\mathbf{v}_{robot}$ is the robot's velocity vector. When calculating the avoid-obstacle univector, the virtual obstacle's positions are used instead of the real obstacles. By introducing the virtual obstacle, a robot can avoid obstacles more safely and smoothly by a generated path at every step.

Hyperbolic Spiral Univector Field. The move-to-goal univector field is designed by the hyperbolic spiral for a robot to get to a target point with a desired orientation. The hyperbolic spiral univector field $\mathbf{v}_{huf}$ is defined as

$$\mathbf{v}_{huf} = [\cos(\phi_h)\ \sin(\phi_h)]^T, \tag{8}$$

where

$$\phi_h = \begin{cases} \theta \pm \frac{\pi}{2}(2 - \frac{d_e + k_r}{\rho + k_r}) & \text{if } \rho > d_e \\ \theta \pm \frac{\pi}{2}\sqrt{\frac{\rho}{d_e}} & \text{if } 0 \le \rho \le d_e \end{cases},$$

θ is the angle from x-axis of the goal at robot's position. The notation $\pm$ represents the direction of movement, where + is when a robot moves clockwise and - counter clockwise. k_r is an adjustable parameter. If k_r becomes larger, the maximal value of curvature derivative decreases and the contour of the spiral becomes smoother. ρ is the distance between the center of the destination and robot's position d_e is predefined radius that decides the size of the spiral.

By designing a move-to-goal univector field with hyperbolic spiral, a robot can arrive at a destination with any orientation angle. In this paper, in order to obtain the desired posture at a target position, two hyperbolic spiral univector fields are combined. The move-to-goal univector field is defined as

$$\phi_{muf} = \begin{cases} \theta_{up} + \frac{\pi}{2}(2 - \frac{d_e + k_r}{\rho_{up} + k_r}) & \text{if } p_y^h > g_{size} \\ \theta_{down} - \frac{\pi}{2}(2 - \frac{d_e + k_r}{\rho_{down} + k_r}) & \text{if } p_y^h < -g_{size} \\ \theta_{dir} & \text{otherwise} \end{cases}, \tag{9}$$

with

$$\rho_{up} = \sqrt{p_x^{h2} + (p_y^h - d_e - g_{size})^2},\ \rho_{down} = \sqrt{p_x^{h2} + (p_y^h + d_e + g_{size})^2},$$

$$\theta_{up} = \tan^{-1}\left(\frac{p_y^h - d_e - g_{size}}{p_x^h}\right) + \theta_{dir}, \ \theta_{down} = \tan^{-1}\left(\frac{p_y^h + d_e + g_{size}}{p_x^h}\right) + \theta_{dir},$$

$$\mathbf{p}^h = \mathbf{M}_{rot}\mathbf{M}_{trans}\mathbf{p},$$

$$\mathbf{M}_{trans} = \begin{bmatrix} 1 & 0 & -g_x \\ 0 & 1 & -g_y \\ 0 & 0 & 1 \end{bmatrix}, \ \mathbf{M}_{rot} = \begin{bmatrix} \cos(-\theta_{dir}) & -\sin(-\theta_{dir}) & 0 \\ \sin(-\theta_{dir}) & \cos(-\theta_{dir}) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{p} = \begin{bmatrix} p_x & p_y & 1 \end{bmatrix}^T, \ \mathbf{p}^h = \begin{bmatrix} p_x^h & p_y^h & 1 \end{bmatrix}^T,$$

where g_{size} is the radius of the goal region and θ_{dir} is the desired arrival angle at a target. By using the move-to-goal univector field which is composed of the hyperbolic spiral univector fields, a robot can arrives at a goal with any arrival angles.

4.2 Footstep Planning

While a humanoid robot moves towards a destination, there is a situation when it has to step over an obstacle if it is not too high. This is the main difference from the path planning for a differential drive mobile robot, as it tries to find a detour route to circumvent obstacles instead of stepping over them. In this section, a footstep planning algorithm is proposed, which enables a robot to traverse over the obstacles effectively.

It is very natural and efficient way that a robot steps over them instead of detouring, if its moving direction is maintained. The proposed algorithm enables a robot step over the obstacles with minimal step length while maintaining its moving direction. It is

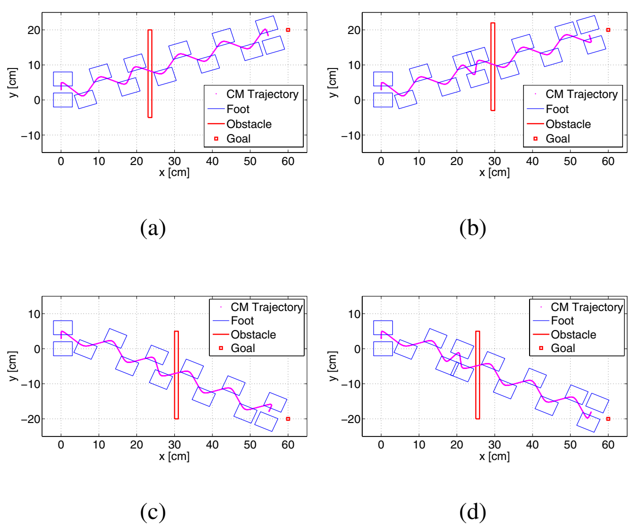


Fig. 1. Stepping over an obstacle. (a) Left leg is supporting leg without additional step (b) Left leg is supporting leg with additional step (c) Right leg is supporting leg without additional step (d) Right leg is supporting leg with additional step.

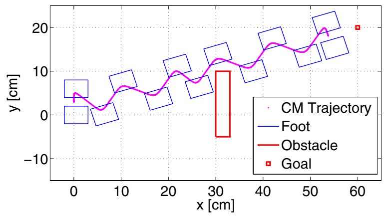


Fig. 2. Stepping over an obstacle when an obstacle is in front of one leg

assumed that the shape of obstacles is a rectangle with narrow width and long length as shown in Fig. 1.

The forward and backward step length from a supporting leg of a humanoid robot are restricted because of hardware limitation. If an obstacle is wider in width than the maximum step length of a humanoid robot, it is not able to step over an obstacle. Thus, a humanoid robot has to step over an obstacle with the shortest possible step length in order to step over the widest possible obstacle. The step length of a humanoid robot is determined by which leg is a supporting leg when it steps over an obstacle. As the proposed algorithm considers these facts, it enables a robot to step over obstacles with the shortest step length. Fig. 1 shows the footprints to step over an obstacle using this algorithm. Fig. 1(a) and Fig. 1(d) are situations when a left foot comes close to the obstacle earlier than a right foot and Fig. 1(b) and Fig. 1(c) are situations when a right foot approaches the obstacle closely than the other one. In case of Fig. 1(a) and 1(b), the left leg is appropriate as a supporting leg for the minimum step length. On the other hand, the right leg is appropriate as a supporting leg in Fig. 1(c) and 1(d). Therefore, in order to make a left leg as a supporting leg in Fig. 1(b) and a right leg as a supporting leg in Fig. 1(d), one more step is needed before stepping over the obstacle, while such an additional step is not needed in Fig. 1(a) and 1(c).

There is a situation when an obstacle is only in front of one leg such that the other leg can be placed without considering the obstacle. The proposed algorithm deals with this situation such that it can step over the obstacle effectively like a human being. Fig. 2 shows the footprints of a robot in this case.

4.3 Parameter Optimization by Evolution Programming

A humanoid robot has the constraint of change in rotation of legs on account of the hardware limitation. Hence, when planning footsteps for a biped robot by the proposed algorithm, the maximum change in rotation of legs has to be assigned. In this algorithm, there are seven parameters to be assigned such as k_v in the virtual obstacle, d_{boun} in the avoid-obstacle univector field, d_e, k_r, g_{size} in the move-to-goal univector field and w_{muf}, w_{auf} in composition of the move-to-goal univector field and the avoid-obstacle univector field, respectively. A robot can arrive at a goal with the change in rotation of legs within constraints by selecting appropriate values of parameters mentioned above. Also to generate the most effective path, EP is employed to choose the values of parameters. The fitness function in EP is designed considering the followings:

- A robot should arrive at a destination with a minimum position error.
- The facing direction of a robot at a destination should be the desired one.
- A robot should not collide with obstacles.
- The change in rotation of legs should not exceed the constraint value.

Consequently, the fitness function is defined as

$$f = -(k_p P_{err} + k_q \mid \theta_{err} \mid + k_{col} N_{col} + k_{const} N_{const}) \quad (10)$$

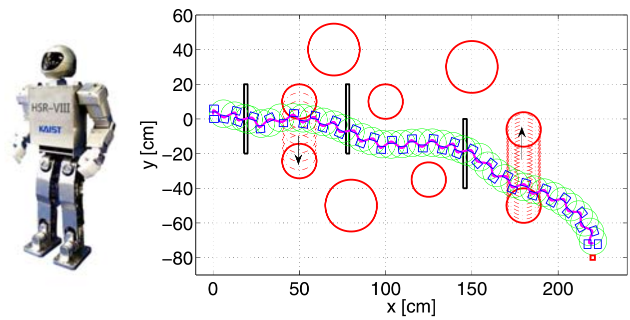
where N_{const} is the number of constraint violations of change in rotation of legs, N_{col} is the number of obstacle collisions of the robot, θ_{err} is the difference between the desired orientation and the orientation of a robot at a goal, P_{err} is the position error at a goal and $k_{const}, k_{col}, k_q, k_p$ are constants.

5 Simulation Results

HSR-VIII (Fig. 3(a)) is a small-sized humanoid robot that has been continuously undergoing redesign and development in RIT Lab, KAIST since 2,000. Its height and weight are 52.8 cm and 5.5 kg, respectively. It has 26 DOFs which consists of 12 DC motors with harmonic drives for reduction gears in the lower body and 14 RC servo motors in the upper body. HSR-VIII was modeled by Webot which is the 3D mobile robotics simulation software [16]. Simulations were carried out with Webot of the HSR-VIII model by applying the proposed footstep planning algorithm.

Through the simulation, seven parameters in the algorithm were optimized by EP. Maximum rotating angle of the robot's ankles was selected heuristically as 40°. After 100 generations, the parameters were optimized as k_v=1.94, d_{boun}=20.09, d_e=30.04, k_r=0.99, g_{size}=0.94, w_{muf}=1.96, w_{auf}=1.46.

Fig. 3(b) shows the sequence of robot's footsteps as a 2D simulation result, where there were ten obstacles of three different kinds such as five static circular obstacles



(a) (b)

Fig. 3. (a) HSR-VIII. (b) Sequence of footsteps in the environment with ten obstacles of three different kinds.

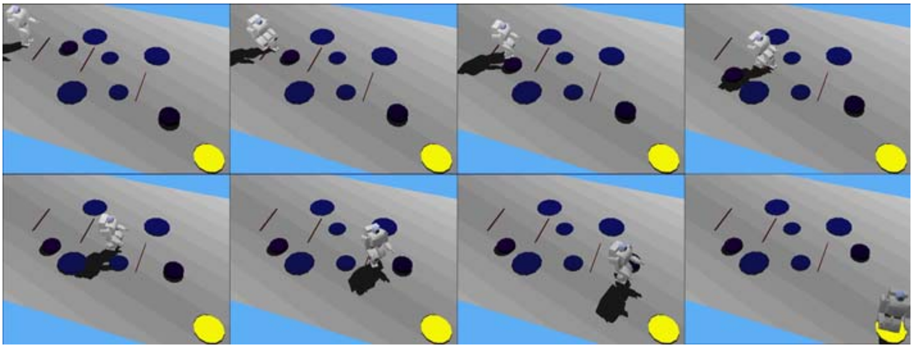


Fig. 4. Snap shots of 3D simulation result by Webot in the environment with ten obstacles of three different kinds. (A goal is a circle in the right bottom corner.)

and two moving circular obstacles and three static rectangular obstacles with a height of 1.0 cm. The desired angle at a destination was fixed at 90° from x-axis. As shown in the figure, by the proposed algorithm the robot moves from a start point to a target goal in the right bottom corner, while avoiding static and moving circular obstacles and stepping over static rectangular ones by adjusting its step length. In addition, the robot faces the desired orientation at the goal. Fig. 5 shows the 3D simulation result by Webot, where the environment is the same as that used in the 2D simulation. Similar result was obtained as in Fig. 3(b). In particular, in third and sixth snapshots of the Fig. 10, it can be seen that the robot makes a turn before colliding with the moving circular obstacles predicting their movement.

6 Conclusion

The real-time footstep planning algorithm was proposed for a humanoid robot to travel to a destination avoiding and stepping over obstacles. The univector field method was adopted to determine the heading direction and using the determined orientations, exact foot placement was calculated. The proposed algorithm generated the efficient path by applying a boundary to the avoid-obstacle univector field and introducing the virtual obstacle concept. Furthermore, it enables a robot to get to a destination with a desired orientation by employing the hyperbolic spiral univector field. The proposed algorithm made a robot possible to step over an obstacle with minimal step length maintaining its heading orientation. It also considered the situation when an obstacle is in front of only one leg. In this case, it steps over the obstacle while placing the other leg properly as a supporting one. The effectiveness of the algorithm was demonstrated by computer simulations in dynamic environment. As a further work, experiments with a real small-sized humanoid robot HSR-VIII will be carried out using a global camera to demonstrate the applicability of the proposed algorithm.

References

1. Nishiwaki, K., Sugihara, T., Kagami, S., Kanehiro, F., Inaba, M., Inoue, H.: Design and Development of Research Platform for Perception- Action Integration in Humanoid Robot: H6. In: Proc. IEEE/RSJ Int. Conference on Intelligent Robots and Systems, pp. 1559–1564 (2000)
2. Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., Akachi, K., Isozumi, T.: Humanoid Robot HRP-2. In: Proc. IEEE Int'l. Conf. on Robotics and Automation, ICRA 2004 (2004)
3. Sakagami, Y., Watanabe, R., Aoyama, C., Matsunaga, S., Higaki, N., Fujimura, K.: The intelligent ASIMO: system overview and integration. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2478–2483 (2002)
4. Ogura, Y., Aikawa, H., Shimomura, K., Kondo, H., Morishima, A., Lim, H., Takanishi, A.: Development of a New Humanoid Robot WABIAN-2. In: Proc. IEEE Int'l. Conf. on Robotics and Automation, ICRA 2006 (2006)
5. Kim, Y.-D., Lee, B.-J., Ryu, J.-H., Kim, J.-H.: Landing Force Control for Humanoid Robot by Time-Domain Passivity Approach. IEEE Trans. on Robotics 23(6), 1294–1301 (2007)
6. Kanal, L., Kumar, V. (eds.): Search in Artificial Intelligence. Springer, New York (1988)
7. Borenstein, J., Koren, Y.: Real-time obstacle avoidance for fast mobile robots. IEEE Trans. Syst., Man, Cybern. 20, 1179–1187 (1989)
8. Borenstein, J., Koren, Y.: The vector field histogram-fast obstacle avoidance for mobile robots. IEEE Trans. Syst., Man, Cybern. 7, 278–288 (1991)
9. Yagi, M., Lumelsky, V.: Biped Robot Locomotion in Scenes with Unknown Obstacles. In: Proc. IEEE Int'l. Conf. on Robotics and Automation (ICRA 1999), Detroit, MI, May 1999, pp. 375–380 (1999)
10. Chestnutt, J., Lau, M., Cheung, G., Kuffner, J., Hodgins, J., Kanade, T.: Footstep planning for the honda asimo humanoid. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 631–636 (2005)
11. Ayaz, Y., Munawar, K., Bilal Malik, M., Konno, A., Uchiyama, M.: Human-Like Approach to Footstep Planning Among Obstacles for Humanoid Robots. In: Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 5490–5495 (2006)
12. Kim, Y.-J., Kim, J.-H., Kwon, D.-S.: Evolutionary Programming-Based Uni-vector Field Navigation Method for Fast Mobile Robots. IEEE Trans. on Systems, Man and Cybernetics - Part B - Cybernetics 31(3), 450–458 (2001)
13. Kim, Y.-J., Kim, J.-H., Kwon, D.-S.: Univector Field Navigation Method for Fast Mobile Robots. Korea Advanced Institute of Science and Technology, Ph.D. Thesis
14. Lee, B.-J., Stonier, D., Kim, Y.-D., Yoo, J.-K., Kim, J.-H.: Modifiable Walking Pattern of a Humanoid Robot by Using Allowable ZMP Variation. IEEE Transaction on Robotics 24(4), 917–925 (2008)
15. Lim, Y.-S., Choi, S.-H., Kim, J.-H., Kim, D.-H.: Evolutionary Univector Field-based Navigation with Collision Avoidance for Mobile Robot. In: Proc. 17th World Congress The International Federation of Automatic Control, Seoul, Korea (July 2008)
16. Michel, O.: Cyberbotics Ltd. WebotsTM: Professional mobile robot simulation. Int. J. of Advanced Robotic Systems 1(1), 39–42 (2004)