2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Reactive Biped Robot Walking with On-line Path Generation and Obstacle Avoidance

Ren C. Luo, Jun Sheng, Chin-Cheng Chen, and Peng-Hsi Chang

Abstract— **The objective of this paper is to propose an approach for generating on-line walking patterns for a biped robot to navigate towards destination and reactively avoid obstacles. Kinect-based sensor system detects obstacles with velocity and dimension estimated for obstacle modeling. With identified obstacles, repulsion vectors of potential field given by obstacles modify walking path of a biped robot. On-line path generation (OPG) is proposed to generate walking path towards a target configuration including position and orientation. With reference mapping combined with walking pattern generator based on zero momentum point (ZMP), reactive walking patterns free of obstacle collision are thus created. The effectiveness of this method is confirmed on a biped robot developed in our lab, and the robot is capable of avoiding obstacles while maintaining ZMP stability. The contribution of this paper is to propose OPG for obstacle avoidance of biped robots by extending potential field method. The Demonstration of the proof of concept has been successfully conducted.**

I. INTRODUCTION

A number of dynamic capabilities have been achieved by the biped robot, such as stable walking [1], running [2] and balancing under perturbation [3] [4]. The intelligent capability of global navigation in an environment filled with assorted static and moving obstacles is another significant merit. Extensive works have investigated global path planning and obstacle avoidance for mobile robots and manipulators [5] [6] by using potential field method [7], MDP [8] and A* [9]. For biped robots, collision-free and global navigation focused on foot placement are widely researched. A* is applied to compute optimal sequences of footstep locations [10] and a collection of heuristic metrics for safety, effort and complexity is used [11]. A three-tiered planner for navigation over large distance based on A* is proposed [12]. Instead of A*, an RRT [13] variant is utilized for footstep planning based on swept volume approximation to check collision off-line [14]. Reference review indicates that the biped collision-free navigation system should include the following parts.

To begin with, the system needs to detect obstacles and model their positions and velocities. A lot of sensors, such as camera, laser range finder, ultra-sonic sensor and infrared sensor are used to detect the environment. Laser sensor is adopted to detect and reconstruct the environment [15]. Stereo vision is used to detect obstacles and identify the objective [16]. Kinect [17] with depth measurement offers a powerful and low-cost sensor system for obstacle detection [18]. In the second part, two possible strategies exist. One is to step over the obstacles with limited size and height [19] [20]. The other is to circumvent the obstacles [10]-[12] [14]. Conventional methods manipulate foot steps to avoid obstacles, yet directly changing walking path [21] is a more humanlike and instinctive alternative. Potential field method [7], successful in manipulator reactive motion [18] [22] and wheeled robots navigation [23], is capable of dealing with on-line obstacle avoidance by directly changing motion path. The criticism of local minimum and incompleteness [24] can be overcome by potential field design and on-line path generation (OPG) as the third part, similar to the on-line trajectory generation (OTG) for manipulators [25]. Kajita et al. [26] proposed a method for direction change by modifying foot placement at certain turning step. Real-time walking pattern generation using spline collocation is proposed [27]. We achieve this objective by using a general path comprised of arcs and straight lines and mapping method [21]. Based on zero moment point (ZMP) based walking pattern generator [28] [29], walking patterns are generated on-line.

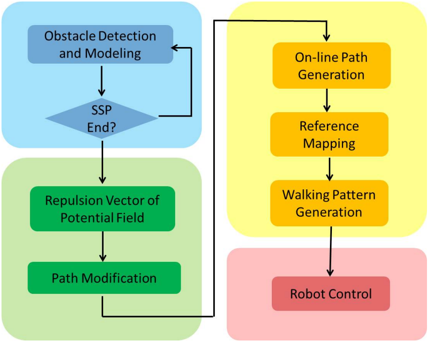
Obstacle Detection and Modeling
SSP End?
Repulsion Vector of Potential Field
Path Modification
On-line Path Generation
Reference Mapping
Walking Pattern Generation
Robot Control

Fig. 1. System architecture of proposed approach

In our scheme, the walking process is a combination of cyclic single support phase (SSP) and double support phase (DSP), with transitions defined by references. In our approach, during DSP in each step, the walking path of the next step (SSP and then DSP) is calculated. As shown in Fig. 1, the architecture contains four sections, and the rest of this paper primarily explains the first three sections. In Section II, the sensor system for obstacle detection and modeling based on Kinect is introduced. As the blue block indicates, the flowchart enters the next section once DSP begins. In Section III, the green block indicates that repulsion vector given by obstacles is applied on the robot to modify its walking path for obstacle avoidance. In Section IV, the yellow section points out that the on-line path generation is proposed. Combined with reference mapping along the path, on-line

The authors are with the Electrical Engineering Department, National Taiwan University, No. 1, Sec. 4, Roosevelt Road, Taipei, Taiwan 106 (corresponding author to provide phone: +886-2-3366-9826; e-mail:renluo@cc.ee.ntu.edu.tw;junsheng,ccchen,pschang@ira.ee.ntu.edu.tw).

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

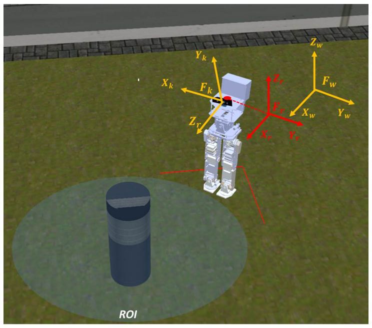

Fig. 2. Construction and frame definition

walking patterns are generated. In Section V, experiments are conducted on an experimental robot developed in our lab, in which the walking robot avoids static and moving obstacles steadily. Finally, the conclusion and future work are presented in Section VI.

II. Obstacle Identification

This section explains how the depth information of Kinect is utilized to detect the nearby obstacles in the walking environment. For each updated frame, the point clusters of obstacles are detected. By calculating geometric centroids and their dimensions, obstacles are modelled with estimated velocity.

A. 3D Position

For the Kinect, eye-in-hand form is adopted with Kinect placed on the top of robot as Fig. 2 shows. Compared with eye-to-hand form detecting spatial information in the world frame F_w by a static Kinect, eye-in-hand form measuring obstacles in the Kinect frame F_k is more instinctive. In F_k, three-dimension (3D) position of a pixel point in the image plane is

$$x_k = \frac{(p_x - c_x)d_p}{f_{sx}} \quad y_k = \frac{(p_y - c_y)d_p}{f_{sy}} \quad z_k = d_p$$

Here, (x_k, y_k, z_k) is 3D position of a pixel point in F_k, (p_x, p_y) is the pixel coordinate in the image plane, and d_p is corresponding depth. (c_x, c_y) is the pixel coordinate of the image centroid, and (f_{sx}, f_{sy}) are camera's focal lengths. F_w is the world frame, and a robot frame F_r is located at the top of the robot with its X axis pointing forward and Z axis pointing upward as Fig. 2 shows. From F_k to F_r, the transformation matrix is

$$M = \begin{bmatrix} 0 & -\sin(\gamma) & -\cos(\gamma) & x_t \\ -1 & 0 & 0 & y_t \\ 0 & -\cos(\gamma) & \sin(\gamma) & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Here, γ is the tilt angle of Kinect, and (x_t, y_t, z_t) is translation part. So, 3D position in F_r is obtained as

$$\begin{bmatrix} x_r \\ y_r \\ z_r \\ 1 \end{bmatrix} = M \begin{bmatrix} x_k \\ y_k \\ z_k \\ 1 \end{bmatrix} \tag{2}$$

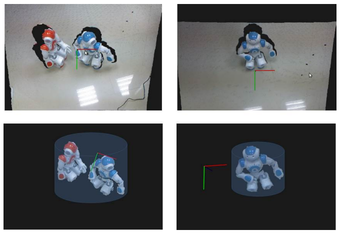

Fig. 3. Results of static obstacle detection and modelling

In this way, 3D positions of all pixel points in the image plane are expressed in F_r.

B. Detection and Modeling

The priority of obstacle modeling is to segment disconnected obstacles and background. All points between the lowest height and a designed upper bound are filtered out as background points and the foreground cluster $C_f(p_1, p_2, \ldots, p_N)$ is formed of remaining points. The second step is to segment obstacles. A minimum interval d_{min} exists for a biped robot to move through, so obstacles with intervals smaller than this minima are merged as a whole. An obstacle segmentation algorithm based on segmentation threshold d_{min} is proposed.

Firstly, a random point of the foreground cluster is chosen as the nucleus N_o of an obstacle cluster C_o. Any point of the foreground cluster is categorized into C_o if the distance between this point and N_o is smaller than d_{min}, and otherwise it is regarded as the nucleus of a new obstacle cluster. The distance of each point in the foreground cluster to all nuclei is recursively computed and compared with d_{min}, so that it is categorized into a existing obstacle cluster or regarded as the nucleus of a new obstacle cluster. In this way, segmented obstacle clusters $C_{o1}, C_{o2}, \ldots, C_{oK}$ are formed. The horizontal centroid $p_o(x_o, y_o)$ of obstacle cluster in F_r is calculated by averaging horizontal positions of all its points and the farthest distance r_o in the horizontal plane to the centroid is found. The obstacle cluster is modelled as a cylinder, and in the horizontal plane it is modelled as a circle with a radius r_o at p_o. Two obstacle models overlap if their nuclei are close or the size of a real obstacle is much larger than d_{min}. A refined obstacle segmentation algorithm is proposed. Two obstacle cluster models with centroid p_{oi} and p_{oj} and radius r_{oi} and r_{oj} are merged if the following condition is satisfied.

$$|p_{ci} - p_{cj}| \leq r_{oi} + r_{oj} + d_{min} \tag{3}$$

The centroid and radius of the new merged cluster is computed again. By checking the above condition for all obstacle clusters, this algorithm refines the segmentation process and merges overlapped obstacle clusters. After this, centroid of each obstacle cluster is tracked for deriving velocity if two or more obstacles exist. The centroids in two consecutive frames are viewed as the same one if their displacement is smaller than a designed threshold. The detection and segmentation results are shown in Fig. 3.

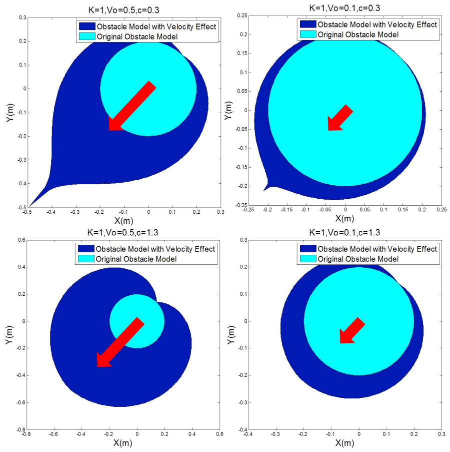

Fig. 4. Horizontal obstacle model considering velocity (red arrow indicates magnitude and direction of velocity)

Obstacle velocity v_o, solved by differentiating the position of centroid in consecutive frames, is a crucial factor in obstacle modeling. Predictive effect of velocity is seen in our life. For example, you dodge more swiftly when confronting a fast car than a slow or standstill one. In our approach, the predictive effect is embodied in the modification of obstacle modeling. The point of boundary of modelled obstacle is given as

$$\begin{bmatrix}x_b\\ y_b\end{bmatrix} = \begin{bmatrix}x_o\\ y_o\end{bmatrix} + (r_o + Kv_o(1-(\frac{\parallel \theta_d \parallel}{\pi})^c))I\begin{bmatrix}\cos(\theta)\\ \sin(\theta)\end{bmatrix} \quad (4)$$

where

$$\theta_d = \begin{cases} \theta-\alpha+2\pi & \theta-\alpha<-\pi \\ \theta-\alpha-2\pi & \theta-\alpha>\pi \\ \theta-\alpha & -\pi\le\theta-\alpha\le\pi \end{cases}$$

Here, (x_b, y_b) is the point of the boundary, θ changes from 0 to 2π in F_r, α indicates the direction of obstacle velocity in F_r, I is identity of dimension 2 by 2, c is the shaping index, and K is a factor to adjust the magnitude of velocity influence. The simulated results of obstacle modeling are shown in Fig. 4.

III. Reactive Walking Path Modification

This section tackles how reactive walking path modification is achieved to deal with nearby obstacles. Configuration of walking path includes position and orientation in the horizontal plane. Potential field method is applied to decide the path configuration at the end of next step and curve path is adopted to smooth the walking path.

A. Potential Field Design

Potential field method indicates that when a robot passes by an obstacle, a repulsion vector pushes the robot away from the obstacle. The walking robot is modelled as a cylinder with physical boundary of radius R_r surrounded by a cylindrical repulsion field of radius $R_r+\varepsilon$. As the walking path is a two-dimension (2D) horizontal trajectory, the 2D relationship between the modelled robot and nearby obstacles indicates that when the boundary of a modelled obstacle enters the repulsion field of the walking robot, a repulsion vector will be applied on the walking robot. The magnitude of repulsion vector is exclusively decided by the closest repulsion distance between the obstacle boundary and the robot physical boundary. Analytical solution for this distance is intractable, so a shortest repulsion distance searching algorithm is proposed. The line connecting the centroids of robot and obstacle intersects the boundary of obstacle model at an intersection angle as

$$\theta_i = atan2(x_o, y_o) \quad (5)$$

Here, θ_i also indicates the direction of obstacle in F_r. With a search step $\Delta\theta$, clockwise or counter-clockwise searching direction is decided by comparing the repulsion distances a step away in two directions. Position of boundary point is computed by using (4) and repulsion distance is obtained by subtracting R_r from the distance between the boundary point and robot centroid. Direction in which the repulsion distance is smaller is selected as the search direction. A direction index η is defined to be 1 or -1 for counter-clockwise or clockwise direction. If these two repulsion distances are both larger, the repulsion distance according to the intersection angle is the closest point, and η is set 0. If η is not zero, searching algorithm continues for more steps along the searching direction until the repulsion distance does not decrease any more. This boundary point is denoted as $B(x_{bc}, y_{bc})$ and it also denotes the position vector to the robot centroid. The shortest repulsion distance is

$$d_{min} = \sqrt{x_{bc}^2 + y_{bc}^2} - R_r \quad (6)$$

The repulsion vector is given as

$$V_r = \begin{cases} 0 & d_{min} > \varepsilon \\ -D(d_{min})\frac{B(x_{bc},y_{bc})}{||B(x_{bc},y_{bc})||} & d_{min} \le \varepsilon \end{cases} \quad (7)$$

Here, $D(d_{min}) = M^{\frac{\cos(\frac{d_{min}}{\varepsilon}\pi)+1}{2}}$, and M is the maximal repulsion magnitude, indicating the strength of the repulsion vector.

B. Path Modification

The desired walking vector V_d is tangential to the walking path and has a nominal magnitude of stride length, also indicating the desired foot placement at the end of the next step. The repulsion vector at the beginning of DSP is calculated, and the actual walking vector is a fusion of desired walking vector and the repulsion vector as

$$V_{a(i)} = V_{d(i)} + V_{r(i)}T(\theta_p) \quad (8)$$

where

$$T(\theta_p) = \begin{bmatrix}\cos(\theta_p) & -\sin(\theta_p)\\ \sin(\theta_p) & \cos(\theta_p)\end{bmatrix}$$

Here, T is the rotation from F_r to F_w decided by the walking direction θ_p along a walking path as shown in Fig. 5. This actual walking vector represents the actual path configuration as well as the actual foot placement at the end of the next step. Because a biped robot cannot make an abrupt change

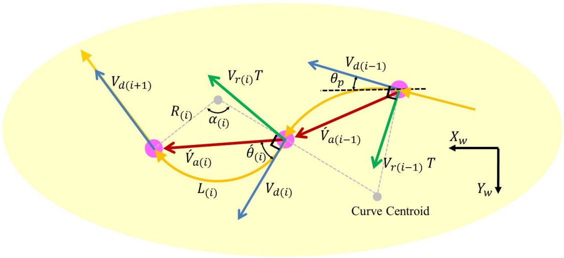

Fig. 5. Reactive walking path modification for consecutive steps (yellow line is generated walking path and pink dot indicates path position at the end of each step.)

in velocity or acceleration, a smooth curve connects the path configurations at the end of two consecutive steps. The direction change from the desired walking vector to the actual walking vector is derived as

$$\theta_{(i)} = \arccos\left(\frac{V_{a(i)}^{\top} V_{d(i)}}{\| V_{a(i)} \| \| V_{d(i)} \|}\right) \tag{9}$$

As indicated by the geometric relationship in Fig. 5, the radius of the curve path is related to the actual walking vector as

$$R_{(i)} = \frac{\| V_{a(i)} \|}{2\sin(\theta_{(i)})} \tag{10}$$

The length of the curved path for the next step is

$$L_{(i)} = \alpha_{(i)} R_{(i)} \tag{11}$$

Here, $\alpha_{(i)} = 2\theta_{(i)}$, and $\alpha_{(i)}$ indicates the rotation angle of the swing leg. In order to avoid physical damage, the biped robot is constrained by hardware limitations about maximal stride length L_{max} and maximal leg rotation angle α_{max} as

$$\begin{cases} \alpha_{(i)} \leq \alpha_{max} \\ L_{(i)} \leq L_{max} \end{cases} \tag{12}$$

By substituting (9)-(11) into (12), the above constraints are rewritten as

$$\begin{cases} \theta_{(i)} \leq \dfrac{\alpha_{max}}{2} \\ \dfrac{\theta_{(i)}}{\sin(\theta_{(i)})} \leq \dfrac{L_{max}}{\| V_{a(i)} \|} \end{cases} \tag{13}$$

Any direction change $(\theta_{(i)})$ or magnitude $(\| V_{a(i)} \|)$ violating above constraints must be adjusted by the following process. Firstly, any $\theta_{(i)}$ is adjusted as

$$\acute{\theta}_{(i)} = \begin{cases} \theta_{(i)} & \theta_{(i)} \leq \frac{\alpha_{max}}{2} \\ \frac{\alpha_{max}}{2} & \theta_{(i)} > \frac{\alpha_{max}}{2} \end{cases} \tag{14}$$

The adjusted $\theta_{(i)}$ is then substituted into the second constraint, and magnitude is adjusted as

$$\| \acute{V}_{a(i)} \| = \begin{cases} \| V_{a(i)} \| & \| V_{a(i)} \| \leq \frac{L_{max}\sin(\acute{\theta}_{(i)})}{\acute{\theta}_{(i)}} \\ \frac{L_{max}\sin(\acute{\theta}_{(i)})}{\acute{\theta}_{(i)}} & \| V_{a(i)} \| > \frac{L_{max}\sin(\acute{\theta}_{(i)})}{\acute{\theta}_{(i)}} \end{cases} \tag{15}$$

The result of adjusted actual walking vector is

$$\acute{V}_{a(i)} = \frac{\| \acute{V}_{a(i)} \|}{\| V_{d(i)} \|} V_{d(i)} T(\acute{\theta}_{(i)}) \tag{16}$$

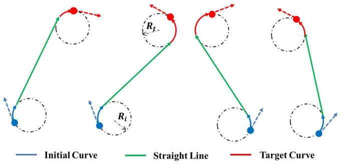

Fig. 6. Four types of generated walking path (type 1 to type 4 from left to right. Blue dash dashed is the initial path orientation and red dashed is the target path orientation.)

Here, transformation matrix T takes the same form in (8). The desired walking vector at the end of the next step is

$$V_{d(i+1)} = V_{d(i)} T(2\acute{\theta}_{(i)}) \tag{17}$$

Here, the transformation matrix also has the same definition in (8). Because it always takes several consecutive steps to avoid an obstacle, Fig. 5 shows the result of continuous steps under the influence of repulsion vector and how smooth path is formed by several curves. The approach of velocity of repulsion vector introduced in [16] is utilized as well, so that local minimum and other problems described in [16] are avoided.

IV. On-line Walking Pattern Generation

With certain path configuration and destination of position and orientation, the on-line path generation (OPG) produces a smooth path to guide the biped robot to the destination. Reference mapping method [21] effectively generates ZMP and foot trajectory references along a smooth path expressed by explicit functions. Based on the mapping method and OPG, walking patters are generated for a biped robot.

A. On-line Path Generation

This approach utilizes curves and straight line to smoothly connect the initial path configuration (x_i, y_i, θ_i) and the target path configuration (x_t, y_t, θ_t). As shown in Fig. 6, the robot firstly turns along an initial curve, moves straight forward, and then turns along a target curve to reach the target path configuration. Both initial curve and target curve have two possible directions: clockwise and counter-clockwise. If the target path position is on the left (right) side of the initial path orientation, counter-clockwise (clockwise) curve is selected. If the initial path orientation points to the target path position, target path orientation decides the initial curve. The direction index ϵ_i of the initial curve is defined as

$$\epsilon_i = \begin{cases} 1 & \theta_{ti} < \theta_i \quad or \quad \theta_{ti} = \theta_i, \theta_t > \theta_i \\ -1 & \theta_{ti} > \theta_i \quad or \quad \theta_{ti} = \theta_i, \theta_t < \theta_i \end{cases} \tag{18}$$

Here, $\theta_{ti} = atan2(x_t - x_i, y_t - y_i)$, and ϵ_i is 1 or -1 for clockwise or counter-clockwise direction. So, the centroid of the initial curve is expressed as

$$\begin{bmatrix} x_{ic} \\ y_{ic} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \end{bmatrix} + R_i \begin{bmatrix} \cos(\theta_i - \epsilon_i 0.5\pi) \\ \sin(\theta_i - \epsilon_i 0.5\pi) \end{bmatrix} \tag{19}$$

Here, R_i is the radius of the initial curve. Then the direction of the target curve is decided by the relationship between the target path orientation and a vector tangential to the initial curve and pointing to the target path position. The tangential point on the initial curve is

$$\begin{bmatrix}x_{tan}\\ y_{tan}\end{bmatrix} = \begin{bmatrix}x_{ic}\\ y_{ic}\end{bmatrix} + R_i \begin{bmatrix}\cos(\phi+\epsilon_i\theta_{tic})\\ \sin(\phi+\epsilon_i\theta_{tic})\end{bmatrix} \tag{20}$$

Here, $\theta_{tic} = atan2(x_t - x_{ic}, y_t - y_{ic})$, and $\phi = atan2(\sqrt{-R_i^2 + (x_t - x_{ic})^2 + (y_t - y_{ic})^2}, R_i)$. The orientation of the tangential vector is

$$\theta_{tan} = atan2(x_t - x_{tan}, y_t - y_{tan}) \tag{21}$$

The direction index for the target curve is decided as

$$\epsilon_t = \begin{cases} -1 & -\pi < \rho \leq 0 \quad or \quad \pi < \rho \leq 2\pi \\ 1 & 0 < \rho \leq \pi \qquad or - 2\pi < \rho \leq -\pi \end{cases} \tag{22}$$

Here, $\rho = \theta_{tan} - \theta_t$, and the centroid of the target curve is

$$\begin{bmatrix}x_{tc}\\ y_{tc}\end{bmatrix} = \begin{bmatrix}x_t\\ y_t\end{bmatrix} + R_t \begin{bmatrix}\cos(\theta_t - \epsilon_t 0.5\pi)\\ \sin(\theta_t - \epsilon_t 0.5\pi)\end{bmatrix} \tag{23}$$

By developing the tangential line connecting these two circles, the walking path is obtained. External tangent occurs in type 1 and 4, and internal tangent occurs in type 2 and 3. Curve ranges are thus concluded for various types as

1 $\phi_i : \theta_i + 0.5\pi \rightarrow \theta_{ti} + 0.5\pi, \phi_t : \theta_{ti} + 0.5\pi \rightarrow \theta_i + 0.5\pi$

2 $\phi_i : \theta_i + 0.5\pi \rightarrow \theta_{ti} + \eta, \phi_t : \theta_{ti} + \eta - \pi \rightarrow \theta_i - 0.5\pi$

3 $\phi_i : \theta_i - 0.5\pi \rightarrow \theta_{ti} - \eta, \phi_t : \theta_{ti} - \eta + \pi \rightarrow \theta_i + 0.5\pi$

4 $\phi_i : \theta_i - 0.5\pi \rightarrow \theta_{ti} - 0.5\pi, \phi_t : \theta_{ti} - 0.5\pi \rightarrow \theta_i - 0.5\pi$

Here, $\eta = \arccos(\frac{R_t + R_i}{\sqrt{(x_{ic} - x_{tc})^2 + (y_{ic} - y_{tc})^2}})$, ϕ_i and ϕ_t are ranges of initial curve and target curve. Ending point of initial curve (starting point of straight line), and starting point of target curve (ending point of straight line) are solved according to curve ranges, centroids and radius. Therefore, a general walking path trajectory is generated.

B. Reference Mapping Method

Tangential direction at a point of a walking path is regarded as a virtual x axis as X_v, and the direction rightwards perpendicular to X_v is a virtual y axis as Y_v. These two virtual axes form a virtual frame F_v in the ground surface with its origin coinciding with origin of F_w. All original reference trajectories like ZMP and foot trajectories are on-line scheduled in F_v. The essence of reference mapping method is to map walking process from F_v to the world frame F_w. ZMP and foot trajectory references are mapped using this method [21]. With mapped references, conventional walking pattern generation methods, like the preview control [28] [29], can be applied to generate desired motion of center of mass (COM).

For the on-line implementation, this system generates the walking path for the next step cycle during DSP of the previous step. Target path configuration at the end of the next DSP is determined by integrating repulsion vector and OPG and path for the next step is generated with a curve. Combined with reference mapping method and preview control, the whole architecture of on-line walking pattern generation for obstacle avoidance is constructed as Fig. 1.

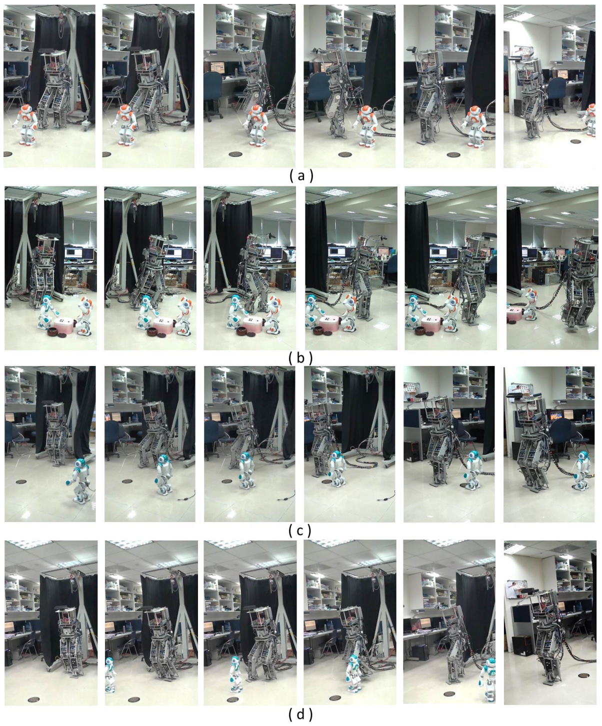

Fig. 7. Snapshots of experiments for (a) small obstacle, (b) clustered obstacles, (c) slowly moving obstacle, and (d) fast moving obstacle

V. Experiment Results

In order to test our proposed algorithm and system, a series of experiments are conducted on an experimental robot. This test robot consists of two legs, each with 6 degrees of freedom (DoF), and a trunk with pitch and roll. The robot is 140 cm high and weighs approximately 50 kg. The joint actuation consists of ”Maxon” DC motor, harmonic drive, and pulley and belt to achieve high torque output. Industrial PC sends signal via an I/O board and motor driver to control joint torque. With COM and foot trajectory references, inverse kinematics (IK) solves joint references and torque commands are computed using a PD controller combined with gravity compensation. The experiments include two parts, towards static and moving obstacles respectively.

Static obstacles, like furnitures and standstill people, are common in the walking environment. In the first test of static obstacles, we design a scenario that a standing NAO robot has its back towards the test robot and blocks the originally straight walking path of the test robot, so NAO doesn’t notice the approaching test robot, and the test robot must detect NAO to avoid it actively. Fig. 7(a) shows the walking process of this scenario. Because NAO stands right in the way, the local minimum case possibly occurs. In our scheme, the test robot turns right and avoids the obstacle by considering of velocity of repulsion vector. Fig. 7(b) shows another scenario that two NAO robots are playing chess with their full attention and do not notice the test robot. Because the intervals of obstacles are very small, the Kinect-based system models the two NAO robots and the small desk as a whole obstacle cluster. Because this obstacle cluster locates on the right side of the orientation of the test robot, the repulsion vector pushes the robot to the left side to avoid obstacles.

If moving obstacles occur, their estimated velocities are taken into consideration. The developed system models the moving obstacles and determines the magnitude and orientation of repulsion vector applied on the test robot. With different obstacle velocities and various obstacle moving paths, two strategies of avoiding moving obstacles occur for the biped robot. As Fig. 7(c) shows, when NAO walks at a relative slow speed from left to right, the test robot chooses to circumvent the NAO by walking before it. If NAO walks with a relative fast speed, as Fig. 7(d) shows, the test robot decides to circumvent the obstacle from its back. This difference is generated because of the influence of obstacle velocity. If the velocity is small, the repulsion vector directly pushes away the test robot. If the velocity is larger, the boundary of obstacle model extends to the vicinity of the test robot, and the repulsion vector decelerates the test robot and pushes it towards the back of the moving obstacle.

VI. CONCLUSIONS

The contribution of this paper is that we apply the potential field method in a biped robot and propose on-line path generation for obstacle avoidance. Depth information helps us model obstacles with velocity estimation to deal with static and moving obstacles. Reference mapping method is adopted for efficiently on-line walking pattern generation. In addition, the proposed mechanism considers hardware constraints. Our proposed algorithms are proved by experiments in which the biped robot avoid obstacles and maintain walking stability. The future work is to replace robot location method, which is accumulation and forward kinematics, by advanced techniques for a better performance.

REFERENCES

[1] T. Takenaka, T. Matsumoto, and T. Yoshiike, "Real time motion generation and control for biped robot-1st report: Walking gait pattern generation," in Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., 2009, pp. 1084-1091.

[2] T. Takenaka, T. Matsumoto, T. Yoshiike, and S. Shirokura, "Real time motion generation and control for biped robot-2nd report: Running gait pattern generation," in Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst., 2009, pp. 1092-1099.

[3] B. J. Stephens and C. G. Atkeson, "Push recovery by stepping for humanoid robots with force controlled joints," in Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on, 2010, pp. 52-59.

[4] B. Stephens, "Humanoid push recovery," in Humanoid Robots, 2007 7th IEEE-RAS International Conference on, 2007, pp. 589-595.

[5] J. C. Latombe, Robot Motion Planning. Boston, MA: Kluwer Academic Publishers, 1991.

[6] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," The international journal of robotics research, 1986, 5(1): 90-98.

[7] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," IEEE Trans. Robot. and Autom., vol. 8, no. 1, pp. 23-32, Feb. 1992.

[8] R. Bellman. A Markovian Decision Process. Journal of Mathematics and Mechanics 6, 1957.

[9] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," Systems Science and Cybernetics, IEEE Transactions on , vol.4, no.2, pp.100-107, July 1968.

[10] J. Chestnutt, M. Lau, G. Cheung, J. Kuffner, J. Hodgins, and T. Kanade, "Footstep planning for the honda ASIMO humanoid," in Proc. IEEE Int. Conf. Robot. Autom., 2005, pp. 631-636.

[11] J. Chestnutt, J. Kuffner, K. Nishiwaki, and S. Kagami, "Planning biped navigation strategies in complex environments," in Proc. IEEE-RAS/RSJ Int. Conf. on Humanoid Robots (Humanoids'03), Munich, Germany, Oct. 2003.

[12] J. Chestnutt, J. Kuffner, "A tiered planning strategy for biped navigation," in Proc. IEEE Int. Conf. Humanoid Robots, 2004. p. 422-436.

[13] Z. Xia, J. Xiong and K. Chen, "Global Navigation for Humanoid Robots Using Samling-Based Footstep Planners," IEEE Transactions on mechatronics, Vol 16, No 4, pp. 716-723, August 2011.

[14] N. Perrin, O. Stasse, L. Baudouin, F. Lamiraux, and E. Yoshida, "Fast humanoid robot collision-free footstep planning using swept volume approximations," Robotics, IEEE Transactions on, 2012, 28.2: 427-439.

[15] L. Bascetta, G. Magnani, P. Rocco, R. Migliorini, and M. Pelagatti, "Anti-collision systems for robotic applications based on laser Timeof-Flight sensors," in IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, July 2010, pp. 278-284.

[16] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, M. Inaba, and H. Inoue, "Online footstep planning for humanoid robots," in Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA'03), 2003.

[17] Microsoft Corporation, 1 Microsoft Way, Redmond, WA 98052-7329, USA, "Microsoft kinect homepage. http://xbox.com/Kinect (accessed: Mar. 28, 2011)," Internet, 2011.

[18] F. Flacco, T. Kroger, A. De Luca, and O. Khatib, "A depth space approach to human-robot collision avoidance," Robotics and Automation (ICRA), 2012 IEEE International Conference on. IEEE, 2012: 338-345.

[19] B. Verrelst, O. Stasse, K. Yokoi, and B. Vanderborght, "Dynamically stepping over obstacles by the humanoid robot HRP-2," in Proc. IEEE/RAS Int. Conf. Humanoid Robots, 2006, pp. 117-123.

[20] O. Stasse, B. Verrelst, B. Vanderborght, and K. Yokoi, "Strategies for humanoid robots to dynamically walk over large obstacles," IEEE Transactions on Robotics, vol. 99, pp. 1-8, 2009.

[21] Ren Luo, Peng Hsi Chang, Jun Sheng, Shao Cheng Gu, and Chun-Hung Chen, "Arbitrary Biped Robot Foot Gaiting Based on Non-Constant COM Height," in Proc. of IEEE-RAS International Conference on Humanoid Robots, Oct. 2013.

[22] S. Haddadin, H. Urbanek, S. Parusel, D. Burschka, J. Rossmann, A. Albu-Schaffer, and Hirzinger, "Real-time reactive motion generation based on variable attractor dynamics and shaped velocities," Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, 2010: 3109-3116.

[23] C. W. Warren, "Global path planning using artificial potential fields," in Proc. of the IEEE International Conference on Robotics and Automation, vol. 1, Scottsdale, AZ, USA, May 1989, pp. 316-321.

[24] T. Kroger and F. M. Wahl, "Online Trajectory Generation: Basic Concepts for Instantaneous Reactions to Unforeseen Events," Robotics, IEEE Transactions on, On page(s): 94-111, Volume: 26, Issue: 1, Feb. 2010

[25] M. Elmogy, C. Habel, and J. Zhang, "Online motion planning for HOAP-2 humanoid robot navigation," Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pp.3531-3536, 2009.

[26] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Yokoi, and H. Hirukawa, "A realtime pattern generator for biped walking," Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on. IEEE, 2002. p. 31-37.

[27] T. Buschmann, S. Lohmeier, M. Bachmayer, H. Ulbrich, and F. Pfeiffer, "A collocation method for real-time walking pattern generation," in Proc. IEEE Int. Conf. Humanoid Robots (Humanoids), 2007, pp. 1-6.

[28] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, et al., "Biped walking pattern generation by using preview control of zero-moment point," in Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on, 2003, pp. 1620-1626.

[29] H. Diedam, D. Dimitrov, P. B. Wieber, K. Mombaur, and M. Diehl, "Online walking gait generation with adaptive foot positioning through linear model predictive control," in Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, 2008, pp. 1121-1126.