Navigation Framework for Humanoid Robots Integrating Gaze Control and Modified-univector Field Method to Avoid Dynamic Obstacles

Jeong-Ki Yoo and Jong-Hwan Kim

Abstract—This paper proposes a navigation framework for humanoid robots, which integrates gaze control and modified univector field-based path planning to cope with moving obstacles. To make navigation robust, obstacles are modeled according to their relative velocities and positions. Moreover, partial evaluation values for gaze control architecture are also considered for modifying their virtual size and moving trajectory. In addition, gaze control architecture is proposed, which estimates the size of local map confidence area, self-localization error, surrounding obstacles and obstacle-free distance against those obstacles in the local map. The proposed framework is verified through computer simulations by using a developed simulator for HanSaRam-VIII.

I. INTRODUCTION

Most of humanoid robot researches have focussed on walking issues [1], [2]. From the biologically inspired approach to the dynamic model-based approaches [3], [4], they have focussed on generating the dexterous walking pattern. Due to the remarkable improvements in hardware and walking pattern generation for humanoid robots, researches are now expanding to various other fields of robotics, such as navigation, vision perception, task processing in complex environments, etc [5]–[7].

In the navigation aspect, various navigational concepts for wheeled robots, such as heuristic search algorithm, dynamics-based random state search approach and force-based algorithm, have been applied to humanoid robots. A* algorithm was applied to footstep planning [8], and vision-guided footstep planning in dynamic environment was performed [9]. Among various search algorithms, rapidly-exploring random trees approach was also applied for motion planning of humanoid robots [10]. Since it estimates and explores through sampled states maintaining its dynamic constraints, it can rapidly derive quasi-optimal path. Univector field method using virtual obstacle concept for footstep planning was proposed, which is expanded in this paper [20]. Integration with vision processing algorithms was performed in many researches of humanoids, which focussed on increasing autonomy and obstacle avoidance [11].

Along with these approaches, researches related to gaze control are also important not only for stabilizing vision images, but also for efficient navigation [12]. It plays a major role in gathering surrounding information [13]. Information theory-based approach was proposed in [14]. Even though these are closely dependent on path planning issues, integration aspect has not been considered as a broadly concerned research topic so far.

This paper proposes a navigation framework, which integrates gaze control and modified univector field method [20]. Since the univector field method is robust for real-time applications, it is modified to deal with moving obstacles efficiently by adopting the proposed dynamic virtual obstacle and velocity modification scheme. Instead of using a simple duplicated virtual obstacle, dynamically resizing and moving obstacle integrated with partial evaluation functions for gaze control is proposed. In addition, its move-to-goal univector field function is modified to pass by via points and arrive at a goal position with arbitrary arriving posture angle. Gaze control system integrates four partial evaluation functions for local map confidence area size, self-localization error from covariance matrix, environmental status of obstacles, and obstacle-free distance of local map. These are also used for deriving the size and length of the path of virtual obstacle. Through this interlinked structure, the performance of the whole framework is expected to have synergy effects for avoiding collision with moving obstacles. Verification of the proposed scheme is performed through computer simulations with a model of a small-sized humanoid robot, HanSaRam-VIII (HSR-VIII), developed at the Robot Intelligence Technology (RIT) Laboratory, KAIST since 2000 [15].

This paper is organized as follows. In Section II, gaze control architecture focusing on partial evaluation functions is described. Section III explains the modification scheme for univector field method using dynamic virtual obstacle and velocity modification approach. Section IV describes an integration of the proposed framework. Section V presents simulation environment and results, and finally conclusions follow in Section VI.

II. GAZE CONTROL ARCHITECTURE

This section describes the gaze control architecture based on four types of partial evaluations and their integration scheme.

A. Partial Evaluations for Gaze Control

In case of navigation in a complex environment, various types of information, such as relative distances and speed
of obstacles relative to the robot, error covariance matrix of all objects in the local map, the state of the robot, etc., have to be combined efficiently. In order to implement efficient gaze control architecture using these informations, four types of partial evaluation functions are defined focusing on self-localization uncertainty factor, relative position and velocity of obstacles, obstacle-free distance of surrounding local map and confident area ratio against the unconfident one in its local map. In addition, these are closely related to the path generation scheme which is described in the next section. The following is the detailed description of the four partial evaluation functions.

1) Local Map Confidence Area-based Partial Evaluation ($\psi_{lm}$): Fig. 1 shows a sample point diagram in robot coordinate $O_R$. Apart from the probability of occupancy grid map in [16], confidence factor $\mathcal{C} (\cdot)$ is additionally defined for each grid. It is set to one when a grid is included in gaze region $R_g$, and starts to be attenuated by $\xi_m$ after the grid is excluded from the gaze region. In the extended map, local map confidence area-based partial evaluation is defined as

$$\mathcal{M}_{lm}(R_g) = N_{lp}/N_{cp}, \quad (1)$$

with

$$N_{lp} = n \{ (\{ \mathbf{p}_i^s | \mathcal{C}(\mathbf{p}_i^s) < \lambda_{cf}, \mathbf{p}_i^s \in R_{s-g} \} \},$$

$$N_{cp} = n \{ (\{ \mathbf{p}_i^s | \mathbf{p}_i^s \in R_{s-g} \} \} - N_{lp}, \quad R_{s-g} = R_s \cap R_g^c,$$

where $R_g$ is the gaze region, $R_s$ is the sampling region defined by $(d_{am}, d_{fm}, \theta_{tlt}^{MAX})$, and $\mathbf{p}_i^s$ denotes the $i_{th}$ sample point in the grid map. In this paper, $(d_{am}, d_{fm})$ are set to $(30 \, \text{cm}, 130 \, \text{cm})$, respectively. $R_g$ is derived by the maximum pan and tilt angles $(\theta_{pan}^{MAX}, \theta_{tlt}^{MAX})$. For $\lambda_{cf}$ and $\xi_m$, 0.85 and -0.001 are respectively used in this paper. This partial evaluation is defined by the ratio between confident and under-confident sample points as in (1). The candidate pan/tilt angles for $\psi_{lm}$ are derived according to arithmetic mean of representative angles for under-confident sample points.

2) Self-localization-based Partial Evaluation ($\psi_{sf}$):

$$\mathcal{M}_{sf}(\sigma) = \left\{ \begin{array}{ll} \sigma/T_{sf} & \text{if } \sigma \leq T_{sf} \\
1 & \text{otherwise} \end{array} \right \} \quad (2)$$

where $T_{sf}$ denotes the user-defined threshold. For self-localization, a model-based sample point approach for extended kalman filter, i.e. distribution approximation filter, is adopted [17]. As a result, error covariance matrix $C_{pp}$ and its magnitude $\sigma$ of robot posture are obtained. Using $\sigma$, partial evaluation for self-localization uncertainty $\psi_{sf}$ is calculated by (2). Apart from the other already explained two evaluation functions, $\psi_{lm}$ and $\psi_{sf}$, candidate angles for $\psi_{sb}$ and $\psi_{sf}$ are defined for all the detected obstacles in the confident area of the local map, and added up in the process of integration in Section II-B. This is because they vary depending on the number of obstacles located within the gaze area.

3) Obstacle-based Partial Evaluation ($\psi_{ob}$): Surrounding obstacles also needs to be considered in navigation. In particular, the movement of each detected obstacle in the local map has to be considered [13]. During the navigation process, each detected obstacle is updated in the local map. Since humans mainly pay attention to frontal and approaching obstacles, only frontal and approaching-from-behind obstacles in the robot-centered confidence map, $R_{O\theta^{cm}\text{dted/appr}}$, are estimated through their relative positions and velocities in $O_R$ as follows:

$$\mathcal{M}_{ob}(\alpha_i) = \left\{ \begin{array}{ll} \alpha_i/T_{ob} & \text{if } \alpha_i \leq T_{ob} \\
1 & \text{otherwise} \end{array} \right \} \quad (3)$$

with

$$\alpha_i = \omega_o \sigma_o \| R_{V_i} \|_2/\| R_{P_i} \|_2,$$

where $R_{P_i}$ and $R_{V_i}$ denote the position and velocity of $i_{th}$ obstacle, respectively, and $\omega_o$ is a normalization factor. For this evaluation, localization uncertainty factor of $i_{th}$ obstacle is reflected by $\sigma_o$, obtained in the process of localization for $\psi_{sf}$.

4) Way point-based Partial Evaluation ($\psi_{way}$): The partial evaluation function for the closest way point is defined using an obstacle-free distance, $d_{of}$, as follows:

$$\mathcal{M}_{way}(d_{of}) = \left\{ \begin{array}{ll} d_{of}/d_{nThres} & \text{if } d_{of} \leq d_{nThres} \\
1 & \text{otherwise} \end{array} \right \} \quad (4)$$

where $d_{of}$ is the minimum distance without obstacles from robot bounded by a user-defined value $d_{nThres}$. It is obtained by the closest point $R_{p_of}$ of extended shapes of obstacles in $R_{O\theta^{cm}\text{dted/appr}}$. The extended shape of moving obstacles is defined and used for the navigation process in Section III.
according to $R_{\psi_\alpha} R_{\psi_\alpha}$, and gaze control parameters. The candidate pan/tilt angle for $\psi_{way}$ is the direction of way point in $O_R$.

B. Integration of Partial Evaluations ($\psi_{int}$)

One candidate gaze direction can be converted to two dimensional pan/tilt gaze angle region. If the pan/tilt angles are set to any of the gaze angles in the derived region, the original candidate gaze direction is shown on the captured image. The conversion for partial evaluations, $\alpha \in \{lm, way, sf, ob\}$, are defined as

$$R_\alpha(\theta_1) = \begin{cases} \psi_\alpha & \text{if } \theta_1 \in R_{\psi_\alpha}^* (\hat{\theta}_\alpha) \\ 0 & \text{otherwise} \end{cases}$$

(5)

where $\hat{\psi}_\alpha$ and $\hat{\theta}_\alpha = [\hat{\psi}_\alpha^T, \hat{\theta}_\alpha^T]^T$ mean the derived partial evaluation value and its candidate pan/tilt angles. For target gaze angle $\theta_1$, $R_{\psi_\alpha}^*$ represents the gaze region according to $\hat{\theta}_\alpha$, as depicted in Fig. 1. The obtained gaze angle regions are combined by weighted arithmetic mean over a gaze angle integration map $G\mathcal{M}$, defined as

$$G\mathcal{M}(\theta_1) = \omega_{lm} R_{\omega_{lm}}(\theta_1) + \omega_{way} R_{\omega_{way}}(\theta_1) + \frac{1}{N_o} \sum_{i=1}^{N_o} \{ \omega_{sf} R_{\omega_{sf}}(\theta_1) + \omega_{ob} R_{\omega_{ob}}(\theta_1) \},$$

where $N_o$ is the number of $R_{\psi_\alpha}$, for $\psi_{lm}$, $\psi_{way}$, $\psi_{sf}$, and $\psi_{ob}$ are normalization weights for partial evaluation values. In case of $\psi_{sf}$ and $\psi_{ob}$, they are added up for $R_{\psi_\alpha}$ and $\theta_1$. Due to the discrete trait of $R_\alpha$, $G\mathcal{M}$ is formed in step shape. The highest center point of obtained $G\mathcal{M}$ is selected as a final representative gaze angle, $\theta^*$, (see Fig. 7(b)).

III. MODIFIED UNIVECTOR FIELD METHOD FOR DYNAMIC OBSTACLE AVOIDANCE

This section describes a modification scheme of univector field method considering moving obstacles by integrating partial evaluations, virtual obstacle concept and velocity modification scheme.

1) Modification of Univector Field Method: Due to the simplicity of univector field navigation method, it can be used for real-time control of mobile robots [18]. Compared to conventional potential field methods, it is robust to oscillations. It employs move-to-goal univector field (MUF) and avoid-obstacle univector field (AUF), respectively, for leading robot to the goal and for avoiding collision against obstacles. A robot can be modeled as a point if its maximum radius is added to those of obstacles [19]. Then, path planning can be simplified as a line searching problem. Moreover, it can reflect the result of partial evaluations, which are closely related to path planning and obstacle avoidance. The extended radius of obstacle $S_{oe}$ is defined as

$$S_{oe} = S_r + S_o + \omega_{oe}(\eta_{oe} || R_{\psi_\alpha} \|_2 + \eta_{imp}(\psi_{sf} + \psi_{obs})), \quad (7)$$

where $S_r$ and $S_o$ are the outer shapes of robot and obstacle simplified to cylinders. $\eta_{oe}$ and $\eta_{imp}$ are the normalization factors, respectively, and $\omega_{oe}$ is the predefined control coefficient.

MUF is modified to control its approaching direction to the goal by adopting a controller concept within a user-defined distance $d_{con}$ as follows:

$$u_{muf}(p_r) = \begin{cases} \{ -c(\theta_{rd}) - s(\theta_{rd}) \theta_{rd}^T & \text{if } d_{con} < \rho_{rd} \\ -k_p s(\theta_{rd}) - k_r s(\theta_{rd}) & \text{if } \rho_{rd} \leq d_{con} \end{cases}$$

(6)

with

$$\theta_{rd} = \alpha(p_d - p_r), \quad \alpha = \omega_{rd} - \beta, \quad \beta = \theta_{tg} - \theta_{rd},$$

$$\theta_{rd} = \beta \alpha + k_3 \beta,$$

where $\cos(\alpha)$ and $\sin(\alpha)$ are abbreviated to $c(\alpha)$ and $s(\alpha)$, $p_r$ and $p_d$ are the positions of robot and destination, respectively, and $\rho_{rd}$ is their Euclidian distance. $\theta_{tg}$ is the current posture angle of robot and $\theta_{tg}$ is the target posture angle of robot at $p_d$. $k_3$ is the user-defined parameter to control the magnitude of MUF. If robot just passes through a via point, $k_p$ is set to one. By adjusting arriving control gains, $k_p, k_3, k_3$, the curvature of arriving to the goal is determined.

In case of AUF, the hyperbolic spiral univector field $\varphi_h(p_r)$ is defined as

$$\varphi_h(p_r) = \begin{cases} \{ (\varphi_{l} p_r), s(\varphi_h(p_r)) \} & \text{if } S_{oe} > \rho_{ov} \leq S_{oe} + d_e + d_k \\ \rho_{ov} + S_{oe} \geq \rho_{ov}, \quad \text{and } \rho_{o} > \rho_{ard} > 0 \end{cases}$$

with

$$\varphi_{l} p_r = \begin{cases} \frac{\alpha_{or} + \pi}{2} \sqrt{S_{oe} - d_e} & \text{if } S_{oe} \leq \rho_{or} < S_{oe} + d_e \\ \alpha_{or} + \pi \sqrt{2 - \frac{S_{oe} + d_e + K_3}{\rho_{or} + R_r}} & \text{if } S_{oe} + d_e \leq \rho_{or} \end{cases},$$

(9)

where $p_o$ and $\theta_{or}$ denotes the position of obstacle and angle between the robot and the $i_{th}$ obstacle, respectively. $d_k$ is the predefined radius offset used for defining spiral, $d_k$ is the size of the boundary offset of AUF, and $K_3$ adjusts the contour curvature of spiral as depicted in Fig. 3. $\pm$ sign denotes the direction of avoiding obstacle, where $\pm$ means counter clockwise and vice versa. When $(\hat{p}_{od} \times \hat{p}_{od} - \hat{p}_{od})[0 0 1]^T < 0$, $+$ sign is selected in $\varphi_h(\cdot)$, where $\hat{p}_{od} = \hat{p}_{od} - \hat{p}_{od}$. $\hat{p}$ = [0 0 1].

The total univector field at $p_r$ is calculated by accumulating $u_{muf}$ and $u_{muf}$ as follows:

$$u_T(p_r) = \omega_{muf} u_{muf}(p_r) + \omega_{muf} \hat{N} \sum_{i=1}^{n} u_{muf}(p_r), \quad (10)$$

where $n$ is the total number of obstacles, $\omega_{muf}$ and $\omega_{muf}$ are the normalizing weighting factors, and $\hat{N}(\cdot)$ is the normalization function.

2) Dynamic Virtual Obstacle and Velocity Modification Scheme: Virtual obstacle concept is already used for various navigation methods, such as a potential field method, limit-cycle navigation, univector field method, etc [20]–[22]. However, these are not appropriate to deal with various traits of dynamic obstacles because they used simple duplicated
virtual obstacle model placed on the path of moving obstacles. Therefore, a dynamic virtual obstacle concept having extended shape and path according to relative velocity and position to robot is proposed. In addition, for efficient collision avoidance, a velocity modification scheme according to collision checking result is proposed.

**Dynamic virtual obstacle:** By modeling a robot as a point, the collision between a robot and surrounding obstacles can be checked by solving a second order polynomial as suggested in [19] under the assumption of constant velocities of robot and obstacles. Fig. 4 shows their relative situations and a collision band, a zone swept by the obstacle moving along its direction of velocity. As shown in Fig.4(c), if a robot is located in the collision band $R_{\text{coll}}$ of moving obstacles, and the extended shape of approaching obstacle overlaps x-axis of $O_R$, the robot cannot avoid the collision simply by adjusting its velocity maintaining the direction of movement. Therefore, it has to modify its path according to the velocity and direction of approaching obstacles. If a simply duplicated virtual obstacle scheme is used for this case, robot has to steer sharply when it encounters the virtual obstacles instantly generated by moving obstacles. In order to cope with this case, the virtual obstacle concept needs to be modified to adjust its size and position dynamically according to $(R_{p_o}, R_{v_o})$ as in Fig.4(c). The radius of dynamic virtual obstacle, $R_{v_o}$, and its velocity, $R_{v_{\perp,op}}$ are defined as

$$R_{v_o} = \min \left( \frac{\rho (1-\xi)}{\rho - 1} S_{\text{sc}}, \| R_{p_{vo}} \|_2 \right),$$

$$R_{v_{\perp,op}} = \left( R_{v_o} \cdot R_{p_{vo}} \right) \frac{R_{p_{vo}}}{\| R_{p_{vo}} \|_2},$$

with

$$R_{p_{vo}} = (R_{v_{vo}} - R_{p_{oi}}), \quad h_{vo} = \| R_{p_{vo}} \|_2, \quad \rho = \frac{\xi^2}{(1-\xi)^2},$$

$$\delta_{\text{dist}} = \omega_{\text{dist}} (\| R_{v_{vo}} \|_2 + \eta_{\text{imp}} (\psi_{sf} + \psi_{obs})).$$

The position of dynamic virtual obstacle, $R_{p_{vo}}$, is the projected point of origin on the path of obstacle in $O_R$. $R_{v_{vo}}$, is decided according to the control variable of oval curvature $\xi$ and the relative position and velocity of an obstacle. It takes a smaller value between the calculated radius and the distance between the robot and its corresponding center of a virtual obstacle to maintain the effect of a virtual obstacle. $\xi$ decides the convexity of virtual obstacle outer shape. The bigger $\xi$ generates the more convex one. If it is set to 0.5, the oval shape changes to a triangular one. As it is limited to $(0, 1)$, $\omega_{\text{dist}}$ has to be assigned to maintain this constraint. In deciding $\delta_{\text{dist}}$ and $R_{v_{vo}}$, extended size of obstacle $S_{\text{sc}}$ and partial evaluations, $\psi_{sf}$ and $\psi_{obs}$, are also considered. By virtue of this, the robot can avoid collisions against moving obstacles.
**Velocity modification:** When \( R_{\text{coll}} \) sweeps the path of a robot, it may collide with the obstacle according to its velocity. Though it may avoid collision by virtue of AUF around the obstacle, moving obstacles have to be considered in a different way from static obstacles. Except for the definite collision case, waiting until the obstacle passes by would be one good alternative to avoid collisions with surrounding moving obstacles. When a robot is out of \( R_{\text{coll}} \) as shown in Fig. 4(a), it collides with the obstacle if \( ||R^V r||_2 \in (v_{c_1}, v_{c_2}) \), where \( v_{c_1}, v_{c_2} \) are speed values of the robot reaching to the corresponding collision points in Fig. 4(a) on \( t_{c_1}, t_{c_2} \), respectively. In this case, two methods are considerable, accelerating or decelerating by \( a_i = 2 \left( ||R^V p_{oc_i}||_2 - ||R^V v_r||_2 t_{c_i} \right) / t_{c_i}^2 \). If it is not possible for the robot to accelerate its speed to \( v_{c_2} \), deceleration is automatically selected.

When it has already entered into \( R_{\text{coll}} \) as shown in Fig. 4(b), escaping speed \( v_{\text{esc}} \) along with its reaching time \( t_{\text{esc}} \) is the only option. If the speed of robot is smaller than \( v_{\text{esc}} \), acceleration is performed as described above. If a target acceleration is beyond robot’s capability, path modification using dynamic virtual obstacle concept is automatically triggered.

**Images from vision sensor**

- Object detection (color)
- Self-localization
- Obstacle localization
- Occupancy grid map (with confidence)
- Path planner: waypoint
- Dynamic virtual obst.
- AUF
- Integration: weighted sum
- Velocity mod.
- Footstep command
- Gaze command

Fig. 5. Integrated architecture of navigation for HSR-VIII.

IV. INTEGRATION OF THE PROPOSED FRAMEWORK

Fig. 5 shows the proposed framework. The locations and velocities of detected obstacles are obtained from color-based object detection process. Self and obstacle localizations are performed using sampling-based kalman-filter approach [17]. Using the results, four types of partial evaluations for gaze control are calculated along with the update process of occupancy grid map and path planner. In this paper, path planner simply updates sequential waypoints for the next destination. For the efficient moving obstacle avoidance process, univector field method is extended to use dynamic virtual obstacle along with velocity modification scheme, which makes robot to overtake a moving obstacle or wait until a moving obstacle passes by (see Section III). In addition, the radius of obstacles is extended to reflect those partial evaluations and their velocities. At last, this navigational information is converted to a control command for HSR-VIII. On every 5ms control period, the walking pattern generation is performed according to transferred footstep and gaze commands by Modifiable Walking Pattern Generator (MWPG) algorithm [15], [23].

V. SIMULATION ENVIRONMENT AND RESULTS

This section explains simulation environments and results using a developed simulator, which simulates the whole navigation framework of small-sized humanoid robot, HSR-VIII [15].

A. Simulation Environments

As in Fig. 6, OpenGL-based simulator was implemented using mechanical frame design files of HSR-VIII by Open-Inventor library. As described in [15], color-based vision processing was implemented using the captured images of virtual vision sensor. Moreover, the same walking pattern generation program with HSR-VIII was implemented for the simulation.

B. Partial evaluation integration

In Fig. 7(a), obstacles depicted with their extended shapes and covariance matrices of localization error are marked as red ovals. Double circle means that the obstacle is shown on the captured screen at least once. Red dot and green line means the sample point and the confident area in the local map. Fig. 7(b) shows the result of integration \( \psi_{\text{int}} \) described in Section II-B.

C. Navigation using the proposed framework

For the verification of the proposed approach, a simulation of navigation was performed with 16 randomly generated obstacles, including four moving obstacles. Fig. 9 shows gaze
VI. CONCLUSIONS

This paper proposed the navigation framework composed of gaze control and modified univector field-based path planning scheme. In addition to path modification by using the dynamic virtual obstacle, velocity modification corresponding to the partial evaluations of gaze control was proposed. Using the proposed dynamic virtual obstacle and velocity modification scheme, humanoid robot could change its walking direction to follow a smooth and effective trajectory in simulations. Moreover, the partial evaluations of gaze control architecture was not only used on its own, but was also closely connected to the path planning scheme through the whole framework. By these, humanoid robot could cope with moving obstacles efficiently for collision avoidance in the performed simulations. However, the proposed modification of univector field scheme assumes the exact detection of obstacle velocity and distance. For this issue, uncertainty terms according to relative distance to obstacles are need to be considered.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0000831).

REFERENCES

Fig. 9. Sequential simulation results of integrated framework.