INTECH
open science | open minds

OPEN ACCESS ARTICLE

International Journal of Advanced Robotic Systems

Mobile Robot Collision Avoidance in Human Environments

Regular Paper

Lingqi Zeng[1] and Gary M. Bone[2,*]

1 HATCH Ltd., Sudbury, ON, Canada
2 Department of Mechanical Engineering, McMaster University, W. Hamilton, ON, Canada
* Corresponding author E-mail: gary@mcmaster.ca

Received 16 Jun 2012; Accepted 7 Nov 2012

DOI: 10.5772/54933

Abstract Collision avoidance is a fundamental requirement for mobile robots. Avoiding moving obstacles (also termed dynamic obstacles) with unpredictable direction changes, such as humans, is more challenging than avoiding moving obstacles whose motion can be predicted. Precise information on the future moving directions of humans is unobtainable for use in navigation algorithms. Furthermore, humans should be able to pursue their activities unhindered and without worrying about the robots around them. In this paper, both active and critical regions are used to deal with the uncertainty of human motion. A procedure is introduced to calculate the region sizes based on worst-case avoidance conditions. Next, a novel virtual force field-based mobile robot navigation algorithm (termed QVFF) is presented. This algorithm may be used with both holonomic and nonholonomic robots. It incorporates improved virtual force functions for avoiding moving obstacles and its stability is proven using a piecewise continuous Lyapunov function. Simulation and experimental results are provided for a human walking towards the robot and blocking the path to a goal location. Next, the proposed algorithm is compared with five state-of-the-art navigation algorithms for an environment with one human walking with an unpredictable change in direction. Finally, avoidance results are presented for an environment containing three walking humans. The QVFF algorithm consistently generated collision-free paths to the goal.

Keywords Mobile Robot, Collision Avoidance, Virtual Force Field, Active Region

1. Introduction

1.1 Challenges of avoiding humans

Avoiding collisions with humans and obstacles is a fundamental requirement for mobile robots, particularly those operating in human environments. The objective of navigation is to plan and control the motion of the robot from its initial position to its goal position, while avoiding obstacles. The navigation problem for mobile robots has been the focus of tremendous research effort. Most of the algorithms for mobile robot navigation are only suitable for avoiding stationary obstacles and/or moving obstacles whose future position and moving direction are predictable, such as other mobile robots. However, avoiding humans is different from avoiding other kinds of obstacles and poses the following additional challenges:

1. Human motion is unpredictable, changing its speed or direction arbitrarily. For example, human motion is holonomic; a human can stop and move sideways without turning, or directly move sideways without stopping. Predicted human positions and moving directions based upon previous human motions are not reliable. Therefore, precise predictions are unavailable, making the navigation problem more difficult.
2. For safety, humans must always possess a higher priority in a navigation system than robots or inanimate objects. Humans should be able to pursue their activities unhindered and without having to watch out for the robots working around them.
3. Considering human emotional reactions, when a robot avoids a human it should not come too close to the human or block the human's path. This can be considered rude behaviour and could also frighten the human, which may cause a sudden action (such as jumping away) leading to human injury.

1.2 Related research

Currently, the majority of navigation algorithms for mobile robots can be classified into two categories. The algorithms in the first category directly provide the collision free path of a robot by assuming the shapes of obstacles are known and their motion is predictable. Furthermore, they typically neglect the robot dynamics. The algorithms in the second category indirectly generate collision free paths by using the current kinematic and geometric information of the robot and any obstacles, such as the distances between obstacles and the robot and the current velocities of obstacles. These algorithms consider the robot dynamics. The recent relevant literature on mobile robot navigation algorithms will now be reviewed.

Algorithms in the first category have the advantage that optimal collision free paths can be found with stationary obstacles or with moving obstacles whose motions are predictable. We will only review the algorithms for avoiding moving obstacles [1-7], since that is the focus of this paper. A "dynamic window" algorithm was proposed in [1]. It employs a window around the robot based on the velocities the robot can reach without causing a collision in the next time interval. Other algorithms [2, 5] assume the obstacles will move with zero acceleration and use their current positions and velocities to build forbidden regions for the robot's velocity. Collision avoidance is accomplished by keeping the robot velocity outside of these regions. Randomized path planning algorithms have also been proposed for avoiding moving obstacles [3, 4]. They exploit randomization to explore large state spaces efficiently. Incremental search planning algorithms are another alternative. The D* algorithm [6] performs a grid-based search incrementally, based on the current positions of the robot and obstacles. Similarly, a predicted collision-free region was combined with the A* algorithm in [7]. These algorithms typically do not work well for obstacles like humans, due to a lack of accurate position/moving direction predictions.

The navigation algorithms in the second category are based on the artificial potential field (APF) (introduced in [10]) or the virtual force field (VFF) (introduced in [11]). These algorithms are normally simple to implement and consume less computational loads than those in the first category. Since virtual forces can be obtained by using the gradient descent method on APFs, they are closely related and belong in the same category. These algorithms assume the existence of a repulsive artificial potential field or a repulsive virtual force surrounding each obstacle to push robots away from the obstacle. An attractive artificial potential field or virtual force is also assumed to be acting to pull the robot to the goal. These algorithms do not directly provide a collision-free path for a robot. The desired accelerations of the robot are given through the dynamic interactions of the robot with the virtual forces. The collision-free path can be obtained by integrating the accelerations with Euler's method. VFF or APF based algorithms have been extensively studied [8-19]. With these algorithms, the force functions or the potential fields are related to the distance from the robot to the goal and the distance to the obstacles. Sometimes they incorporate the velocities of the obstacles and robots in the force functions or potential fields to help the robots avoid dynamic obstacles [8, 10-16, 18-19]. An "active region" surrounding each obstacle is often used to define where the repulsive potential field (or repulsive virtual force) from the obstacle is active. If the robot is inside an obstacle's active region, the repulsive potential field/virtual force is applied to repel the robot away from the obstacle. The attractive potential field may also be active and move the robot towards the goal during navigation. If the robot is outside the region, only the attractive potential field is applied to the robot to pull it towards the goal. The APF/VFF is divided into two parts by the boundary of the active region. In [11, 14] a component tangential to the repulsive component of their potential field is added. This tangential component helps the robot to detour around the obstacle. Similarly, a virtual force that acts perpendicularly to the repulsive force was added in [13]. This force helps the robot to detour around the obstacle and reduces the time required for the robot to reach the goal. In this paper we name this force the *detour force*. In [13] another region that is very close to the obstacle is utilized. We will refer to this as the *critical region*. In the critical region a robot may not have sufficient space for complete avoidance, so an alternative to VFF is needed inside this region. It is important to note that methods for systematically designing the size and shape of the active and critical regions are missing from the prior literature.

The APF-based and VFF-based navigation algorithms in the second category are suitable for avoiding moving obstacles whose moving directions are unpredictable, such as humans. This is because the active region provides the space needed to handle the unpredictability. With VFF-based algorithms, if the virtual forces at the boundary of the active region of an obstacle are discontinuous, as in, for example [9, 17], oscillations that waste time and energy may occur as explained in [20]. To both prevent the oscillations and make the motion more efficient, the VFF must be continuous at the edge of the active region.

Another problem with existing APF and VFF-based algorithms, that we will term the *collinear condition*, was described in [13]. In this condition, an obstacle is located between the robot and the goal and the centres of the robot, obstacle and goal are collinear. The velocity directions of the robot and the obstacle are also on this line (see Figure 1). With conventional VFF algorithms, the attractive force and the repulsive force will all be along this line. Hence, the robot either collides with the obstacle, or is pushed away such that it will never reach its goal. Obviously, the solution is to move the robot sideways. Furthermore, severe path oscillations occur with the existing VFF algorithms whenever the robot, obstacle and goal locations are near to the collinear condition. A non-zero detour force can provide collision avoidance without path oscillations, as shown in Figure 1. This force was not provided in the algorithm in [13]. In [11, 14] a non-zero detour force is provided, however its direction may result in a collision when avoiding a moving human.

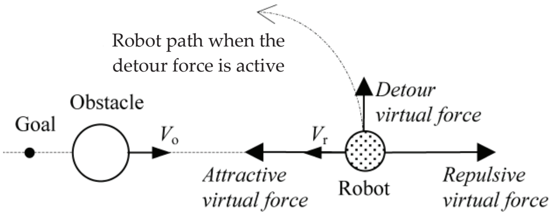

Figure 1. Collinear condition for collision avoidance.

Lastly, proven stability is an unsolved problem for APF-based and VFF-based methods for environments with a variable number of dynamic obstacles. Some researchers have investigated the use of Lyapunov's second method to prove stability. In [11] a pre-known environment with only stationary obstacles was divided into a grid. Using an electrical analogy, each cell of the grid was considered as a resistor, cells occupied by an obstacle were considered as high electric potential, and the goal as ground. Then they established a Lyapunov function, such that the value of the function is the voltage of the cell presently occupied by the robot. The robot is moved to the surrounding cell that has the lowest potential until it reaches the goal. This function is extended for moving obstacles in [14]. However, the number of moving obstacles must be known *a priori*, otherwise their Lyapunov function will be discontinuous and therefore invalid. The APF algorithm in [16] utilized the modified dipolar potential field for multi-robot navigation. By using this field as the Lyapunov function of the navigation system, a backstepping controller is designed to be globally asymptotically stable. The effectiveness of this method is verified through simulations. However, this method requires perfect knowledge of the environment in order to build the APF and therefore cannot be applied to environments with variable numbers of obstacles. In [17] an APF is designed to be proportional to the distance between the robot and its goal and inversely proportional to the distance between the robot and an obstacle. By using their APF function as a Lyapunov function, they verified the asymptotic stability for avoiding stationary obstacles with one important exception. According to their analysis, under the collinear condition the robot fails to reach the goal.

1.3 Contributions and paper organization

Although significant progress has been made with robot navigation algorithms, several problems have not been addressed in the prior literature, as discussed above. The main contributions of this paper are:

1. A novel VFF-based robot navigation algorithm with improved force functions. It may be used with holonomic and nonholonomic robots to avoid moving obstacles with unpredictable direction changes, including humans.
2. Methods to calculate the sizes of active and critical regions for different obstacles. These methods consider the worst-case avoidance conditions and utilize the shapes, velocity limits and acceleration limits of the obstacle and robot.
3. Proof that the proposed algorithm is stable for variable numbers of stationary and dynamic obstacles.
4. A comparison of the proposed algorithm with five state-of-the-art collision avoidance algorithms for moving obstacles.

It is important to note that our algorithm assumes that the human(s) wish to be independent of the robot(s), unlike in [21] where the focus is on navigation for human-robot social interaction. The organization of the paper is as follows. In Section 2, the methods for establishing the sizes of the active and critical regions are explained. The virtual force functions are designed in Section 3. The stability analysis is presented in Section 4. In Section 5, simulation and experimental results are presented and discussed. Finally, conclusions are drawn in Section 6.

2. Sizing the active and critical regions

2.1 Active and critical regions for a human

The sizes of the two regions will be determined from the robot and human shapes, and velocity and acceleration limits. The active region, termed $\mathbb{C}_2$, is defined as the region near the human. If any mobile robot is in $\mathbb{C}_2$, the repulsive and detour virtual forces should be activated to make the robot avoid and detour around the human. Therefore, the size of $\mathbb{C}_2$ is a key issue in the design of a navigation system. It determines how much space is available for the robot to avoid the human. If $\mathbb{C}_2$ is large, the space available for navigation is wide and the distance between the human and robot will increase, but the time to reach the goal will also increase. The critical region, termed $\mathbb{C}_3$, is the region in close proximity to the human and it is very dangerous if a robot intrudes into this region. Within this region, the robot cannot complete the avoidance and should stop to prevent, or at least mitigate, the collision. Any other action made by the robot could worsen the situation. Therefore, when the robot just enters $\mathbb{C}_3$, it should be fully decelerated until it is stopped. We define $\mathbb{C}_1$ as the region that excludes $\mathbb{C}_2$ and $\mathbb{C}_3$ for all obstacle(s). $\mathbb{C}_1$ contains the goal. If the robot is in this region, collision avoidance is unnecessary and the robot simply moves towards its goal.

Since a human's motion is arbitrary, it can change its moving velocity and direction at any time, subject to a limit in linear velocity magnitude. For safety, we need to consider the arbitrary motion made by the human, so the active and critical regions for humans are represented by disk-shaped regions to consider all directions (i.e., assuming the angular velocity of the human is infinite). In the following analysis, we will focus on the design of $\mathbb{C}_2$ and $\mathbb{C}_3$ for the human, with consideration of the human's linear velocity limit.

2.2 Size of the critical region for a human

Since a typical mobile robot moves on the floor, the human will be modelled as the projected shape of its body on the floor. The projected shape is highly related to the pose of the human. By modelling a human body as an enclosing cylinder, we can neglect the different poses and simplify the navigation system, while still maintaining safety. The human cylinder shape projected onto the horizontal plane of motion is a disk, as shown in Figure 2. The average step length of a human is 0.8 m [22]. One half of this value will be used as the radius of the human model, *i.e.* $\rho_h = 0.4$ m.

To simplify the design of the critical region, the shape of the robot will be reduced to a point and the region will be appropriately enlarged. We will use 2D Minkowski addition to sum the disk-shaped human model and the shape of the robot. The projection of the robot onto the floor is modelled as a disk with a radius of ρ_r. The resultant Minkowski sum area (shown enclosed by the dashed-line in Figure 2) is a disk with a radius of $\rho_r + \rho_h$. Therefore when the centre of the robot lies on the dashed-circle, the robot will just contact the human model. The shape of the robot can then be regarded as a point in the remaining analysis.

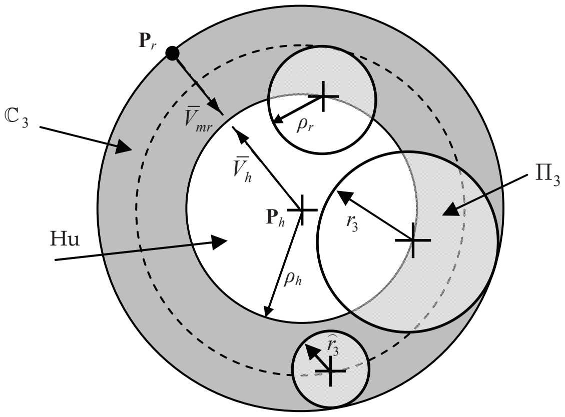

Figure 2. Design of the critical region for a human.

The size of the critical region is related to the stopping distance, $\hat{r}_3$, which ensures that, in the worst-case, robot-human impact will not occur before the robot is stopped, i.e., when the human and robot approach in the collinear condition (defined in Section 1.2) with their maximum velocities $\bar{V}_h$ and $\bar{V}_{mr}$, as shown in Figure 2. When the robot enters $\mathbb{C}_3$ of the human, it should decelerate with its maximum deceleration magnitude until it stops. The stopping distances are given by:

$$\hat{r}_3 = \bar{\delta}_{h1} + \bar{\delta}_{r1} \tag{1}$$

$$\bar{\delta}_{h1} = \bar{V}_h \cdot (\Delta t_1 + T_s) \tag{2}$$

$$\bar{\delta}_{r1} = \bar{V}_r T_s + \tfrac{1}{2}\bar{a}_{mr}\Delta t_1^2 \quad \text{and} \tag{3}$$

$$\Delta t_1 = \bar{V}_{mr} / \bar{a}_{mr} \tag{4}$$

where $\bar{\delta}_{r1}$ is the maximum distance required to stop the robot, $\bar{\delta}_{h1}$ is the maximum distance that the human could travel towards the robot during the interval $\Delta t_1 + T_s$, $\bar{a}_{mrd}$ is the maximum magnitude of the robot deceleration, Δt_1 is the time required for stopping the robot and T_s is the sampling period of the navigation system. T_s must be considered due to the lack of position information of the human and robot between sampling instants. $\hat{r}_3$ also defines a disk, as shown in Figure 2. Minkowski addition is used to add this disk to the sum of the human and robot disks. Equivalently, the human model can be summed with disk Π_3, as shown. The radius of Π_3 must satisfy:

$$r_3 > \hat{r}_3 + \rho_r \tag{5}$$

The combined area defines the critical region for the human. In other words, $\mathbb{C}_3$ incorporates the stopping distance, the shape of the human and the shape of the robot.

2.3 Size of the active region for a human

The active region of a human should provide the robot(s) with enough room to avoid the human. As previously mentioned, the worst-case is the collinear condition, with the human and robot approaching at their maximum speeds, as shown in greater detail in Figure 3. The initial conditions are as follows: the position and velocity of the robot centre are $\mathbf{P}_{r0} = \begin{bmatrix} P_{rx0} & P_{ry0} \end{bmatrix}^{\mathrm{T}}$ and $\mathbf{V}_{mr} = \begin{bmatrix} \bar{V}_{mr} & 0 \end{bmatrix}^{\mathrm{T}}$, respectively and the position and velocity of the human centre are $\mathbf{P}_{h0} = \begin{bmatrix} P_{hx0} & P_{hy0} \end{bmatrix}^{\mathrm{T}}$ and $\mathbf{V}_{h} = \begin{bmatrix} -\bar{V}_{h} & 0 \end{bmatrix}^{\mathrm{T}}$, respectively. The human's velocity here is assumed to be constant. Due to the detour force the robot's velocity will change direction and the robot will move to the point $\mathbf{P}_{r1}$. The displacement of the robot in the Y-direction must be larger than the radius of $\mathbb{C}_3$ (*i.e.* $r_3 + \rho_h$); otherwise the robot will enter $\mathbb{C}_3$.

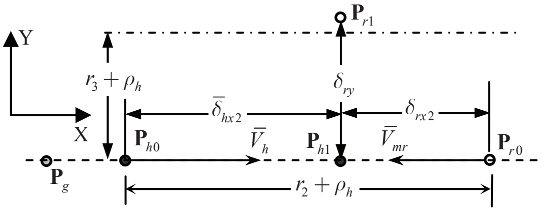

Figure 3. Kinematics of the collinear condition.

Therefore, from Figure 3 the radius of $\mathbb{C}_2$ (i.e., $r_2 + \rho_h$) must be larger than the summation of the distances moved by the human and robot in the X-direction before the robot reaches $\mathbf{P}_{r1}$. If only motion in the X-direction is considered, the radius of Π_2 must satisfy:

$$r_2 + \rho_h > \bar{\delta}_{hx2} + \delta_{rx2} \tag{6}$$

$$\bar{\delta}_{hx2} = P_{hx} - P_{hx0} = \bar{V}_h \cdot (\Delta t_2 + T_s) \quad \text{and} \tag{7}$$

$$\delta_{rx2} = P_{rx} - P_{rx0} = \bar{V}_{mr} \cdot (\Delta t_2 + T_s) + \iint_{\Delta t_2 + T_s} a_{mrx} d^2 t \tag{8}$$

where a_{mrx} is the acceleration component of the robot in the X-direction, $\bar{\delta}_{hx2}$ and δ_{rx2} are as defined in Figure 3 and Δt_2 is the time required for the robot to move by r_3 in the Y direction. However, to avoid a collision the robot must both decelerate in the X-direction (i.e., a_{mrx}<0) and accelerate in the Y-direction to reach the dashed-dot line shown in Figure 3. Noting that the velocity of the robot is also limited, we have:

$$\delta_{ry2} = \bar{V}_{mr} \cdot (\Delta t_2 - t_a) + \iint_{t_a} a_{mry} dt > r_3 + \rho_h \tag{9}$$

where a_{mry} is the acceleration of the robot in the Y-direction and t_a is the time spent by the robot accelerating from zero velocity to $\bar{V}_{mr}$ in the Y-direction. It is impossible to obtain the acceleration in the Y-direction analytically, since it will be different for different algorithms, so Δt_2 cannot be obtained directly. We will estimate Δt_2 by first assuming the robot is driven with maximum acceleration magnitude[1], $a_{mry} = \bar{a}_{mr}$ in the Y-direction. Therefore, (9) is rewritten as:

$$\bar{V}_{mr} \cdot (\Delta t_2 - t_a) + \tfrac{1}{2} \bar{a}_{mr} t_a^{\ 2} = \delta_{ry2} > r_3 + \rho_h \tag{10}$$

So Δt_2 can be estimated with:

$$\Delta t_2 > \frac{r_3 + \rho_h - \frac{1}{2} \bar{a}_{mr} t_a^2}{\bar{V}_{mr}} + t_a \quad \text{and} \tag{11}$$

$$t_a = \frac{\bar{V}_{mr}}{\bar{a}_{mr}} \tag{12}$$

where t_a is the acceleration time. The robot acceleration magnitude in the Y-direction will actually be less than $\bar{a}_{mr}$, since the robot must also decelerate in the X-direction, so the actual Δt_2 will be longer than the computed values and (6) will underestimate the value of r_2 . To compensate for this error, we can assume the worst case of $a_{mrx} \approx 0$ and then the conservative estimate for r_2 is given by (7) and:

$$r_2 > \bar{\delta}_{hx2} + \bar{\delta}_{rx2} - \rho_h \quad \text{and} \tag{13}$$

$$\bar{\delta}_{rx2} = \bar{V}_{mr} \cdot \left(\Delta t_2 + T_s \right). \tag{14}$$

In the analysis presented above, it was assumed that the robot could be directly accelerated sideways (Y-direction). This is true if the robot is holonomic. A nonholonomic robot must be turned $\pi / 2$ first and then it can be accelerated in the Y-direction. Therefore, the turning time t_{turn} also needs to be considered. By assuming the robot turns with its maximum angular acceleration $\bar{\dot{\omega}}$, we have:

$$\frac{\pi}{2} = \frac{1}{2} \bar{\dot{\omega}} t_{turn}^2 \tag{15}$$

$$\therefore \quad t_{turn} = \sqrt{2\left(\tfrac{\pi}{2}\right) / \bar{\dot{\omega}}} = \sqrt{\pi / \bar{\dot{\omega}}} \ . \tag{16}$$

Note, when calculating the size of $\mathbb{C}_3$, t_{turn} was not considered, since the robot will be linearly decelerated to a stop. For the human and a nonholomic robot, (13) is changed to:

$$r_{2,nh} > \bar{\delta}_{hx2} + \bar{\delta}_{rx2} = \bar{V}_{mr} \cdot \left(\Delta t_2 + t_{turn} + T_s \right) + \bar{V}_h \cdot \left(\Delta t_2 + t_{turn} + T_s \right) - \rho_h .\text{a} \tag{17}$$

[1] It is assumed that the maximum acceleration magnitude of the mobile robot equals the maximum deceleration magnitude.

2.4 Sizes of active and critical regions for stationary circular obstacles

The procedure for determining r_2 and r_3 for humans can be used for other obstacles, such as stationary disk-shaped obstacles. Since the velocity of a stationary obstacle is zero, avoiding the obstacle requires:

$$r_{3,c} > \bar{\delta}_{r1} + \rho_r \tag{18}$$

$$r_{2,cir} > \bar{\delta}_{rx2,c} \tag{19}$$

$$\bar{\delta}_{rx2,c} = \bar{V}_{mr} \cdot \left(\Delta t_{2,c} + T_s\right) \text{ and} \tag{20}$$

$$\Delta t_{2,c} > \frac{r_{3,c} + \rho_c - \frac{1}{2}\bar{a}_{mr} t_a{}^2}{\bar{V}_{mr}} + t_a \tag{21}$$

where $r_{2,c}$ and $r_{3,c}$ are the radii of Π_2 and Π_3, respectively, ρ_c is the radius of the obstacle model and t_a is given by (12).

2.5 Sizes of the active and critical regions for stationary rectangular and polygonal obstacles

For a stationary rectangular obstacle, modelling it as a circle will waste too much floor area. The size of its active and critical regions should be determined by modelling it as a rectangle, as shown in Figure 4. As in Section 2.2, we use Minkowski summation to add the rectangle and the shape of the robot, to allow us to reduce the shape of the robot to a point. The resultant area is enclosed by the dashed line in Figure 4.

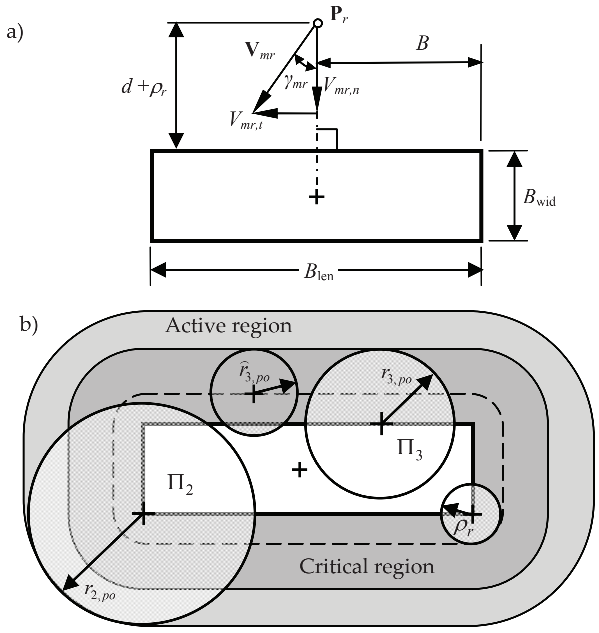

Figure 4. (a) Worst-case avoidance condition for a rectangular obstacle. (b) Design of the active and critical regions.

To obtain the region sizes, we need to know the worst-case for a robot avoiding this obstacle. As shown in Figure 4a, the robot approaches the obstacle with a velocity $\mathbf{V}_{mr}$. It is clear that its velocity component $V_{mr,t}$ (i.e., tangential to the boundary of the Minkowski sum area) will not cause a collision and it will help the robot to detour around the obstacle. The other component $V_{mr,n}$ may cause a collision. The worst-case occurs when $\gamma_{mr} = 0$ and $V_{mr,n} = \bar{V}_{mr}$. For this case, stopping the robot before contact occurs requires:

$$\hat{r}_{3,po} = \bar{\delta}_{r1} \tag{22}$$

$$r_{3,po} > \hat{r}_{3,po} + \rho_r \tag{23}$$

where $\hat{r}_{3,po}$ is the stopping distance and $\bar{\delta}_{r1}$ is given by (3). So $\mathbb{C}_{3,po}$ may be obtained by Minkowski summation of the obstacle model and disk Π_3 with a radius of $r_{3,po}$, as shown in Figure 4b.

For the active region and a holonomic robot similar to (19), we require:

$$\hat{r}_{2,po} > \bar{\delta}_{rx2,po} \tag{24}$$

$$\bar{\delta}_{rx2,po} = \bar{V}_{mr} \cdot \left(\Delta t_{2,po} + T_s\right) \tag{25}$$

$\Delta t_{2,rect}$ must satisfy:

$$\bar{V}_{mr} \cdot \left(\Delta t_{2,po} - t_a\right) + \tfrac{1}{2}\bar{a}_{mr} t_a^2 > r_{3,po} + B \tag{26}$$

where B is the distance between the robot's position and the boundary of the obstacle in the $V_{mr,t}$ direction. For the worst-case of $B = \frac{1}{2}B_{len}$, similar to (21), $\Delta t_{2,po}$ must satisfy:

$$\Delta t_{2,po} > \frac{r_{3,po} + \frac{1}{2}B_{len} - \frac{1}{2}\bar{a}_{mr} t_a^2}{\bar{V}_{mr}} + t_a \tag{27}$$

The corresponding worst-case value of $\hat{r}_{2,po}$ is termed $r_{2,po}$. If the robot is nonholonomic then $r_{2,po}$ should be increased by $\bar{V}_{mr} t_{turn}$. $\mathbb{C}_{2,po}$ may then be obtained by Minkowski summation of the obstacle model and disk Π_2 with a circle of radius $r_{2,po}$, as shown in Figure 4b.

This can be easily extended to convex and non-convex polygonal obstacles. In the convex case:

1. Enclose the polygon in a minimal area bounding rectangle.
2. Calculate $r_{2,po}$ and $r_{3,po}$ for the bounding rectangle.
3. $\mathbb{C}_{2,po}$ and $\mathbb{C}_{3,po}$ equal the Minkowski summations of the polygon with disks Π_2 and Π_3, respectively.

An example of this procedure is shown in Figure 5.

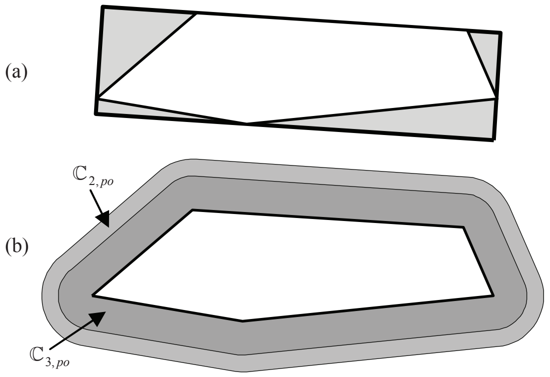

Figure 5. Region design example for a convex polygonal obstacle. (a) Obstacle with its bounding rectangle. (b) Designed regions.

For non-convex polygonal obstacles, the polygon must first be decomposed into convex pieces. The regions are then created for each piece. If the robot enters the active regions of two or more pieces then the virtual forces from those pieces are summed. An example of a U-shaped obstacle will be presented in Section 4.

3. Design of the QVFF virtual force functions

This section begins by discussing the case of a robot avoiding a single human. The geometric configuration of the goal, the robot and the human is presented in Figure 6. We assume that the robot goal is stationary and outside of the active region(s) of the obstacle(s) for safety. In this figure, $\mathbf{P}_g$, $\mathbf{P}_r$ and $\mathbf{P}_h$ are the current position vectors of the goal, the robot and the human, respectively. $\mathbf{V}_{mr}$ and $\mathbf{V}_h$ are the velocity vectors of the robot and the human. We have:

$$\mathbf{E} = \mathbf{P}_g - \mathbf{P}_r, \tag{28}$$

$$\mathbf{D} = \mathbf{P}_r - \mathbf{P}_h \quad \text{and} \tag{29}$$

$$\mathbf{W} = \mathbf{P}_g - \mathbf{P}_h \tag{30}$$

where $\mathbf{E}$ is the vector from the robot to its goal, $\mathbf{D}$ is the vector from the obstacle to the robot and $\mathbf{W}$ is the vector from the obstacle to the goal.

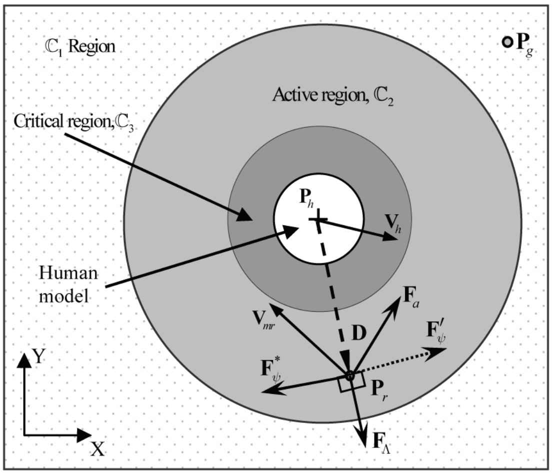

Figure 6. Example of the attractive, repulsive and detour forces when avoiding a human.

An attractive virtual force is used to guide the robot to the goal. It is activated when the robot is within $\mathbb{C}_1$ or $\mathbb{C}_2$. Its force function is defined as:

$$\mathbf{F}_a = \begin{cases} K_1\mathbf{E} + K_2\dot{\mathbf{E}} & \text{if } \mathbf{P}_r \in \mathbb{C}_2 \cup \mathbb{C}_1 \\ \text{undefined} & \text{if } \mathbf{P}_r \in \mathbb{C}_3 \end{cases} \tag{31}$$

where K_1 and K_2 are the positive attractive virtual force gains.

A repulsive virtual force is used to keep the robot away from the human and is activated when the robot is within $\mathbb{C}_2$. The repulsive force is normally a function of the distance between the human and robot. We use the closest distance[2] $d = \|\mathbf{D}\|_2 - \rho_r - \rho_h$ and its rate of change $\dot{d}$. Its direction is along $\mathbf{D}$. Note that when $d \le 0$ a collision has occurred. If we define $d_2 = d - r_2 + \rho_r$ and $d_3 = d - r_3 + \rho_r$ then $d_2 \le 0 \wedge d_3 \ge 0 \Leftrightarrow \mathbf{P}_r \in \mathbb{C}_2$ and $d_3 < 0 \Leftrightarrow \mathbf{P}_r \in \mathbb{C}_3$. We propose the new repulsive force function:

$$\mathbf{F}_\Lambda = \begin{cases} \left(K_3\Lambda + K_4\dot{\Lambda}^*\right)\mathbf{u}_\Lambda & \text{if } \mathbf{P}_r \in \mathbb{C}_2 \\ \text{undefined} & \text{if } \mathbf{P}_r \in \mathbb{C}_1 \cup \mathbb{C}_3 \end{cases} \tag{32}$$

where K_3 and K_4 are the positive repulsive force gains, $\mathbf{u}_\Lambda$ is a unit vector along $\mathbf{D}$, pointing away from the human, $\Lambda = \frac{{d_2}^2}{d_3}$ and $\dot{\Lambda}^* = -\dot{d}\frac{{d_2}^2}{{d_3}^2} = -\kappa_\Lambda \dot{d}$. Note that $\mathbf{P}_r \in \mathbb{C}_2 \Leftrightarrow r_3 < d < r_2$.

The numerator term of Λ, ${d_2}^2$ causes the repulsive force to gradually increase from zero when the robot enters the active region. The denominators, d_3 and ${d_3}^2$, cause the force to increase greatly when the robot is near the boundary of $\mathbb{C}_3$. This is useful for avoiding multiple obstacles. When the robot is pushed close to $\mathbb{C}_3$ by another obstacle, the repulsive force from this obstacle is significantly increased. This tends to push the robot away from $\mathbb{C}_3$ of the obstacle. Note that the velocities of the human and robot are included in $\dot{\Lambda}^*$ of (32). When the human and robot approach each other $\dot{d} < 0$, so $\dot{\Lambda}^* > 0$ and the repulsive force will increase to push the robot away.

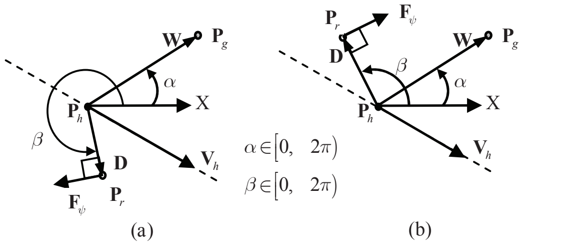

Figure 7. Determination of the correct sense of the detour force.

[2] Please note that the definitions of d, r_2 and r_3 used in this paper are different than in [20] and [24].

The detour force, $\mathbf{F}_\psi$, is a virtual force perpendicular to $\mathbf{D}$. Its main purpose is to push the robot to detour around the human while still moving to the goal. Its other purpose is dealing with the collinear condition. The first issue when designing this force is selecting its sense. This issue has not been adequately addressed in the existing literature. In Figure 6, the two possible senses for $\mathbf{F}_\psi$ are labelled $\mathbf{F}_\psi^*$ and $\mathbf{F}_\psi^{'}$. If the sense of the force was $\mathbf{F}_\psi^{'}$ the robot would try to pass in front of the obstacle. This is acceptable for stationary obstacles, but is not desirable with humans for physical and emotional reasons, as was discussed in Section 1.1. So the sense of $\mathbf{F}_\psi^*$ is the correct choice. Figure 7 shows the correct sense of $\mathbf{F}_\psi$ in relation to the kinematics of the robot and human. In Figure 7a, the goal and robot are on different sides of the human's velocity line. In this case, the detour force should cause the robot to detour around behind the human to prevent blocking the human. In Figure 7b, since the goal and robot are on the same side of the human's velocity line, the robot will not block the human. Then, the sense of $\mathbf{F}_\psi$ should point it towards the goal to help the robot reach its goal faster. For the collinear condition, since the robot, its goal, and the obstacle are collinear and their velocity lines are also along this line, the condition for avoidance is symmetrical to this line. When there are other obstacles, we should choose the sense that satisfies $\mathbf{u}_\psi \cdot \sum \mathbf{F}_{\psi,i} > 0$, where $\sum \mathbf{F}_{\psi,i}$ is the sum of the detour forces from the other obstacles and $\mathbf{u}_\psi$ is the unit vector in the direction of the detour force. This force should be zero when the robot is between the goal and the human (i.e., when $\alpha = \beta$). Furthermore, since they produce no velocity line, the sense of $\mathbf{u}_\psi$ should be towards the goal with stationary obstacles to reach the goal earlier. So, we propose the new detour force function:

$$\mathbf{F}_\psi = \begin{cases} \left(K_5 \psi + K_6 \dot{\psi}^*\right)\mathbf{u}_\psi & \text{if } \mathbf{P}_r \in \mathbb{C}_2 \\ \text{undefined} & \text{if } \mathbf{P}_r \in \mathbb{C}_1 \cup \mathbb{C}_3 \end{cases} \tag{33}$$

where K_5 and K_6 are the positive detour force gains, $\psi = d_2{}^2\Phi$, $\Phi = \alpha - \beta$ and $\dot{\psi}^* = d_2{}^2\dot{\Phi}$. The sense of $\mathbf{u}_\psi$ should be chosen to satisfy:

$$\begin{cases} \mathbf{u}_\psi \bullet \left(\mathbf{V}_h - \mathbf{D}\right) \le 0 & \text{if } \left(\mathbf{V}_h \times \mathbf{D}\right) \bullet \left(\mathbf{V}_h \times \mathbf{W}\right) < 0 \\ \mathbf{u}_\psi \bullet \left(\mathbf{W} - \mathbf{D}\right) \ge 0 & \text{otherwise} \end{cases} \tag{34}$$

To ensure stability, we add the following stabilizing virtual force into the VFF:

$$\mathbf{F}_L = \begin{cases} \dfrac{K_L \mathbf{u}_L}{\left\|\mathbf{v}_L\right\|_2} & \text{if } \mathbf{P}_r \in \mathbb{C}_2 \\ \text{undefined} & \text{if } \mathbf{P}_r \in \mathbb{C}_1 \cup \mathbb{C}_3 \end{cases} \tag{35}$$

where $\mathbf{v}_L = K_1 \dot{\mathbf{E}} + K_3 \dot{\Lambda}^* \mathbf{u}_\Lambda + K_5 \dot{\psi}^* \mathbf{u}_\psi$, $\mathbf{u}_L$ is a unit vector in the direction of $\mathbf{v}_L$,

$$K_L = \begin{cases} \Omega & \Omega > 0 \\ 0 & \Omega \le 0 \end{cases},$$

$$\Omega = -K_1 \dot{\mathbf{E}}^{\mathrm{T}} \left(K_3 \Lambda \mathbf{u}_\Lambda + K_5 \psi \mathbf{u}_\psi\right) - K_3^2 \frac{2\dot{d} d_2}{d_3} \Lambda + \tfrac{1}{2} M_r K_3 \dot{\kappa}_\Lambda \dot{d}^2 + K_5 \dot{\psi}^* \left(\mathbf{a}_o^{\mathrm{T}} \mathbf{u}_\psi - K_1 \mathbf{E}^{\mathrm{T}} \mathbf{u}_\psi\right) - K_3 \dot{\Lambda} \left(K_1 \mathbf{E}^{\mathrm{T}} \mathbf{u}_\Lambda - \mathbf{a}_o^{\mathrm{T}} \mathbf{u}_\Lambda\right) + \tfrac{1}{2} M_r K_5 d_2{}^2 \dot{d} \dot{\Phi}^2 - M_r K_5 d_2 d \dot{d} \dot{\Phi}^2 - K_5^2 d_2 \dot{d} \Phi^2$$

and M_r is the virtual mass of the robot.

Our VFF is a combination of the four forces and is termed quad-VFF (QVFF). Thus, the QVFF for the robot with a single human/obstacle is:

$$\mathbf{F}_V = \begin{cases} \mathbf{F}_a & \text{if } \mathbf{P}_r \in \mathbb{C}_1 \\ \mathbf{F}_a + \mathbf{F}_\Lambda + \mathbf{F}_\psi + \mathbf{F}_L & \text{if } \mathbf{P}_r \in \mathbb{C}_2 \\ \text{undefined} & \text{if } \mathbf{P}_r \in \mathbb{C}_3 \end{cases} \tag{36}$$

If the robot intrudes into $\mathbb{C}_3$, since it may not have enough space for collision avoidance, the VFF is not used and the robot is decelerated to a stop. This stopping action should prevent or at least mitigate the collision. Any other action made by the robot could worsen the situation.

A VFF should be continuous in order to reduce path oscillations. For the QVFF to be continuous the virtual forces at the boundary between $\mathbb{C}_1$ and $\mathbb{C}_2$ must be equal. From (31) and (36), at the boundary of $\mathbb{C}_1$ we have:

$$\mathbf{F}_{V,\mathbf{P}_r \in \mathbb{C}_1} = K_1 \mathbf{E} + K_2 \dot{\mathbf{E}} \tag{37}$$

From (32)-(36), at the outer boundary of $\mathbb{C}_2$ we have:

$$\lim_{d_2 \to 0} \Lambda = \lim_{d_2 \to 0} d_2{}^2 / d_3 = 0, \tag{38}$$

$$\lim_{d_2 \to 0} \dot{\Lambda}^* = \lim_{d_2 \to 0} -\dot{d} d_2{}^2 / d_3{}^2 = 0, \tag{39}$$

$$\lim_{d_2 \to 0} \psi = \lim_{d_2 \to 0} d_2{}^2 \Phi = 0 \text{ and} \tag{40}$$

$$\lim_{d_2 \to 0} \dot{\psi}^* = \lim_{d_2 \to 0} d_2{}^2 \dot{\Phi} = 0 \tag{41}$$

From (38)-(41) it follows that:

$$\lim_{d_2 \to 0} \mathbf{F}_\Lambda = \lim_{d_2 \to 0} \mathbf{F}_\psi = \lim_{d_2 \to 0} \mathbf{F}_L = 0 \tag{42}$$

Therefore:

$$\lim_{d_2 \to 0} \mathbf{F}_{V,P_r \in \mathbb{C}_2} = K_1 \mathbf{E} + b K_1 \dot{\mathbf{E}} = \mathbf{F}_{V,P_r \in \mathbb{C}_1} \tag{43}$$

From (43) the piecewise QVFF is continuous.

Extending the above analysis, if the robot is sharing its environment with N static or dynamic obstacles (which could include other robots), the force field will be:

$$\mathbf{F}_V = \begin{cases} \mathbf{F}_a & \text{if } \mathbf{P}_r \in \mathbb{C}_1 \\ \text{undefined} & \text{if } \mathbf{P}_r \in \mathbb{C}_{3,i} \; i \in [1,...,N] \\ \mathbf{F}_a + \sum_{i=1}^{N} c_i \left(\mathbf{F}_{\Lambda,i} + \mathbf{F}_{\psi,i} \right) + \mathbf{F}_L & \text{otherwise} \end{cases} \quad (44)$$

where $c_i = \begin{cases} 1 & \text{if } \mathbf{P}_r \in \mathbb{C}_{2,i} \\ 0 & \text{otherwise} \end{cases}$, $\mathbf{F}_{\Lambda,i}$ and $\mathbf{F}_{\psi,i}$ are the repulsive and detour forces for the i^{th} obstacle, respectively, $\mathbf{v}_L = K_1 \dot{\mathbf{E}} + \sum_{i=1}^{N} c_i \left(K_{3,i} \dot{\Lambda}_i^* \mathbf{u}_{\Lambda,i} + K_{5,i} \dot{\psi}_i^* \mathbf{u}_{\psi,i} \right)$, $\mathbf{F}_L = \frac{K_L \mathbf{u}_L}{\left\| \mathbf{v}_L \right\|_2}$ and $\mathbf{u}_L$ is the unit vector along the direction of $\mathbf{v}_L$. With (44), the QVFF is again piecewise. It is also continuous. Proof is given in [24] and is omitted here for brevity.

4. Stability analysis of the QVFF

As mentioned in Section 1.2, with VFF-based algorithms the desired acceleration of the robot is obtained by dividing $\mathbf{F}_v$ by M_r and the desired velocity may be obtained by Euler integration. Assuming the robot can track the desired motion trajectory, stability of the QVFF implies stability of the robot motion. Our stability analysis is based on Lyapunov's second method. We will start the stability analysis for the navigation condition with only one static or dynamic obstacle. From (36), when the robot is inside $\mathbb{C}_1$, $\mathbf{F}_v$ only depends on $\mathbf{E}$ and $\dot{\mathbf{E}}$, so the state vector is $\mathbf{X}_1 = \begin{bmatrix} \mathbf{E}^{\mathrm{T}} & \dot{\mathbf{E}}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$. When $\mathbf{X}_1 = \mathbf{0}$ the robot is stopped at its goal. When the robot is inside $\mathbb{C}_2$, the QVFF state vector is $\mathbf{X}_2 = \begin{bmatrix} \mathbf{E}^{\mathrm{T}} & \dot{\mathbf{E}}^{\mathrm{T}} & \Lambda & \dot{\Lambda}^* & \psi & \dot{\psi}^* \end{bmatrix}^{\mathrm{T}}$. The piecewise Lyapunov function candidate for the QVFF is:

$$V = \begin{cases} V_1 = \tfrac{1}{2} \mathbf{X}_1^{\mathrm{T}} \mathbf{Q}_1 \mathbf{X}_1, & \text{if } P_r \in \mathbb{C}_1 \\ V_2 = \tfrac{1}{2} \mathbf{X}_2^{\mathrm{T}} \mathbf{Q}_2(d) \mathbf{X}_2, & \text{if } P_r \in \mathbb{C}_2 \end{cases} \quad (45)$$

where $\mathbf{Q}_1 = \begin{bmatrix} K_1^2 & K_1^2 & M_r K_1 & M_r K_1 \end{bmatrix} \cdot \mathbf{I}_4$, $\mathbf{Q}_2(d) = \begin{bmatrix} K_1^2 & K_1^2 & M_r K_1 & M_r K_1 & K_3^2 & h_\Lambda & K_5^2 & h_\psi \end{bmatrix} \cdot \mathbf{I}_8$, $h_\Lambda = M_r K_3 \frac{d_3^2}{d_2^2}$ and $h_\psi = M_r K_5 \frac{d}{d_2^2}$. Note that when the robot is inside $\mathbb{C}_3$ the robot is commanded to stop. V_1 and V_2 are clearly positive definites. A piecewise Lyapunov function candidate must be continuous to be valid. To check the continuity of V at the boundary between $\mathbb{C}_1$ and $\mathbb{C}_2$:

$$\lim_{d \to r_2} V_2 = \tfrac{1}{2} K_1^2 \mathbf{E}^{\mathrm{T}} \mathbf{E} + \tfrac{1}{2} M_r K_1 \dot{\mathbf{E}}^{\mathrm{T}} \dot{\mathbf{E}} \quad (46)$$

Therefore:

$$\lim_{d \to r_2} V_{2, P_r \in \mathbb{C}_2} = V_{1, P_r \in \mathbb{C}_1}, \quad (47)$$

and the continuity is proven.

When $\mathbf{P}_r \in \mathbb{C}_1$, from (31) and (44) we observe that $\mathbf{F}_{V, P_r \in \mathbb{C}_1}$ has the same form as proportional plus derivative control. It is straightforward to prove that the QVFF is locally asymptotically stable at the goal, subject to the condition $\mathbf{P}_r \in \mathbb{C}_1$. The details are omitted here for brevity and are provided in [24].

When $\mathbf{P}_r \in \mathbb{C}_2$, if we set:

$$\frac{K_2}{K_1} = \frac{K_4}{K_3} = \frac{K_6}{K_5} = b \quad (48)$$

where $b > 0$, then the first derivative of V_2 is:

$$\dot{V}_2 = \begin{cases} -b \left\| \mathbf{v}_L \right\|_2^2 & \Omega > 0 \\ -b \left\| \mathbf{v}_L \right\|_2^2 + \Omega & \Omega \le 0 \end{cases} \quad (49)$$

From (49) it is clear that:

$$\dot{V}_2 \le -b \left\| \mathbf{v}_L \right\|_2^2 \le 0 \quad (50)$$

When $\mathbf{P}_r \in \mathbb{C}_2$, according to (50), $\dot{V}_2 \le 0$. As shown in [24], $\dot{V}_2 = 0$ if $K_1 \mathbf{E} + K_3 \Lambda \mathbf{u}_\Lambda + K_5 \psi \mathbf{u}_\psi = \mathbf{0}$, $\dot{\mathbf{P}}_r = \mathbf{0}$ and $\dot{\mathbf{P}}_h = \mathbf{0}$ simultaneously. This is clearly unlikely. When $\dot{V}_2 < 0$ the QVFF will drive the robot out of $\mathbb{C}_2$ and into $\mathbb{C}_1$. When it is inside $\mathbb{C}_1$ the robot will asymptotically reach the goal as described above.

The stability analysis can also be expanded to include multiple static or dynamic obstacles. If the robot is in the active region of N obstacles, the VFF state for V_2 is:

$$\mathbf{X}_2 = \begin{bmatrix} \mathbf{E}^{\mathrm{T}} & \dot{\mathbf{E}}^{\mathrm{T}} & \Lambda_1 & \dot{\Lambda}_1^* & \psi_1 & \dot{\psi}_1^* & \ldots & \Lambda_N & \dot{\Lambda}_N^* & \psi_N & \dot{\psi}_N^* \end{bmatrix}^{\mathrm{T}} \quad (51)$$

and $\mathbf{Q}_2(d)$ becomes:

$$\mathbf{Q}_2(d) = \begin{bmatrix} K_1^2 & K_1^2 & M_r K_1 & M_r K_1 & K_{3,1}^2 & h_{\Lambda,1} & K_{5,1}^2 & h_{\psi,1} & \cdots \\ & K_{3,N}^2 & h_{\Lambda,N} & K_{5,N}^2 & h_{\psi,N} \end{bmatrix}^{\mathrm{T}} \mathbf{I}_{4N+4} \quad (52)$$

Then we can use (45) to build the Lyapunov function for N obstacles. This function for N obstacles is continuous, since Λ_i, $\dot{\Lambda}_i^*$, ψ_i and $\dot{\psi}_i^*$ for $i = 1, \ldots, N$ equal zero at the edge of the active region of all *N* obstacles. We can see that if the robot comes into the active region of another obstacle, the value of *N* in (51) and (52) will be increased to *N*+1. Since Λ_{N+1}, $\dot{\Lambda}^*_{N+1}$, ψ_{N+1} and $\dot{\psi}^*_{N+1}$ equal zero at the edge of the active region of the *N*+1 obstacle, V_2 is still continuous. The Lyapunov function for *N*+1 obstacles is also continuous and the stability analysis can be performed using the same approach as in the single obstacle case. Therefore, arbitrary numbers of obstacles can be included in the stability analysis. Furthermore, it can be shown that the same stability conclusions from the single obstacle case apply to the multiple obstacle case, except that $\dot{V}_2 = 0$ is more likely to occur. Detailed proof is given in [24] and is omitted here for brevity. The $\dot{V}_2 = 0$ case is more likely to occur than with a single obstacle due to the interaction of the virtual forces with the obstacles. The corresponding equilibrium point is termed a "local minima" point in the robot collision avoidance literature (e.g., [14, 19]). This is particularly a problem with stationary obstacles since the robot will be prevented from reaching its goal indefinitely. However, local minima can often be prevented by using large values of K_5 and K_6 to create a large detour force. An example is shown in Figure 8. The robot starts inside the U-shaped obstacle. The obstacle has been decomposed into three rectangular obstacles whose regions were calculated as described in Section 2.5. Initially, $\mathbf{P}_r \in \mathbb{C}_1$, $\mathbf{F}_V = \mathbf{F}_a$ and the robot is driven straight towards its goal. It next enters $\mathbb{C}_2$. While $\mathbf{P}_r \in \mathbb{C}_2$, $\mathbf{F}_V$ is dominated by the repulsive and detour forces. $\mathbf{F}_\psi$ keeps it outside of $\mathbb{C}_3$, while $\mathbf{F}_\Lambda$ drags it around the obstacle. Finally it is pulled out of $\mathbb{C}_2$ and to the goal by $\mathbf{F}_a$.

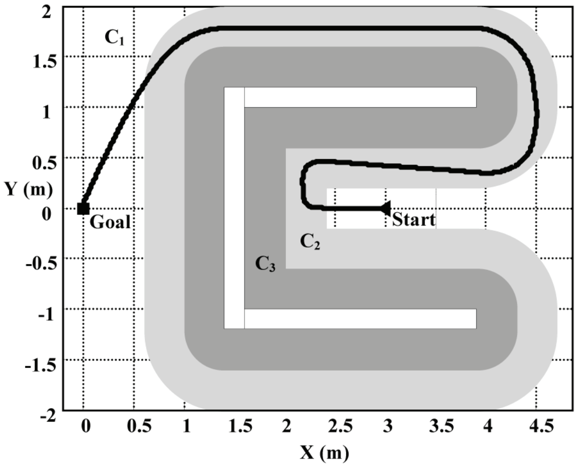

Figure 8. Simulation of the QVFF algorithm with a holonomic robot and a U-shaped obstacle.

5. Results and discussion

5.1 Experimental Hardware

A differential-drive nonholonomic robot and vision system was built for experimental verification of our algorithm. A bottom view of the robot is shown in Figure 9. Wheels 1 and 2 are driven by DC motors via a 10:1 worm gear. Wheel 3 is a passive omni-directional wheel that prevents the robot from tipping. The wheel positions are measured by optical encoders. The encoder data and motor voltages are transmitted to/from a PC-based controller using an umbilical cable. The experimental setup also includes a calibrated colour video camera for capturing images of the human and the robot. Colour patches are attached to the human's shoulder and the robot at known Z heights, as shown in Figure 10. With a standard PC, the X-Y positions of the robot and the X position of the human are estimated from the image centroids of the colour patches using the camera calibration matrix and the patch heights. Note that the low ceiling height in our laboratory prevented the camera from being placed overhead. Due to the camera's location and the human's shoulder height, the accuracy of the Y position estimates is very poor, so only the human's X position estimates are used. The velocities of the robot and the human are calculated by backward differencing their estimated positions. The experimental hardware has a sampling period of $T_s = 0.06$ s. Unfortunately, it is only possible to reliably track one human with our setup since the colour patch on the 2nd human could be occluded by the body of the 1st human. Further details on this hardware are provided in [24].

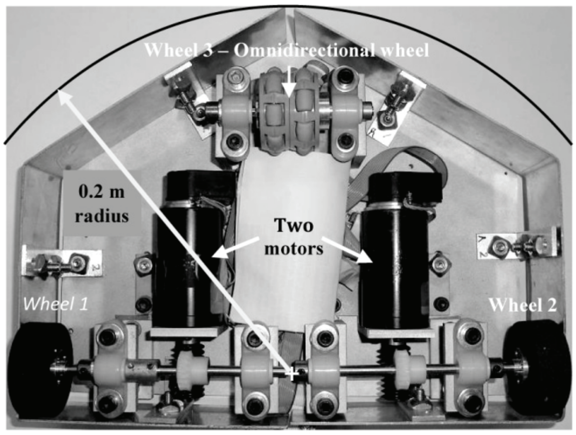

Figure 9. The nonholonomic robot used in the experiments.

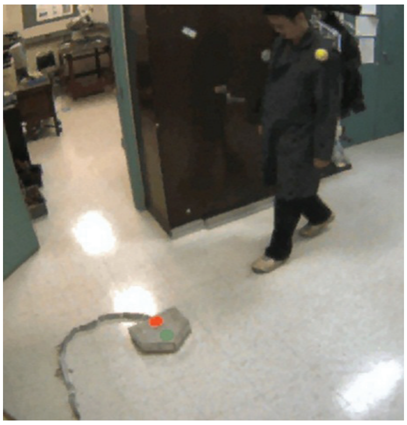

Figure 10. Captured image of the robot avoiding a walking human. The red, green and yellow colour patches are used for position tracking.

5.2 *Avoiding a walking human in the collinear condition*

As explained in Sections 1.2 and 2.3, the collinear condition is the worst-case for a robot and a single moving obstacle. Simulation and experimental results for a walking human and nonholonomic and holonomic mobile robots in the collinear condition are shown in Figure 11. The robots both have a radius of $\rho_r = 0.2$ m, a maximum velocity of $\bar{V}_{mr} = 0.7$ m/s and maximum acceleration of $\bar{a}_{mr} =$ 10 m/s^2. The maximum angular acceleration of the nonholonomic robot is $\bar{\dot{\omega}} = 70\ \text{rad/s}^2$. The maximum human velocity was assumed to be $\bar{V}_h =$ 1 m/s [23]. Substituting these values and the ρ_h value from Section 2.2 into our region sizing equations (Sections 2.2 and 2.3) we obtained $r_{2,nh} =$ 2.1m and $r_{3,nh} =$ 0.4m for the nonholonomic robot and $r_2 =$ 1.7m and $r_3 =$ 0.4m for the holonomic robot. The human starts from position (0.1, 0.6)m, moves from left to right and stops at (4, 0.6)m. At the same time, the robot starts from (4, 0.6)m and moves from right to left towards its goal at (0, 0.6)m. From Figure 11 we can see that the simulation and experimental results for the nonholonomic robot are similar. The discrepancy is mainly due to wheel slip. In comparison to the simulation with the holonomic robot, we can see that the nonholonomic robot moves away from the goal (i.e., in the positive X-direction) more than the holonomic robot. The time required for the nonholonomic robot to turn to avoid the human allowed *d* to decrease further as the human approached. This caused a larger repulsive force that pushed the robot farther from the goal. This is apparent in the plots of *e* and the robot path. The *d* values are always positive, demonstrating that no collision occurred. Furthermore, the *d* and *e* plots show that the holonomic robot was safer (since its minimum *d* was larger) and arrived at the goal 24% sooner than the nonholonomic robot. Note that the measured human position and velocity data were used in the computer simulations. As explained in section 5.1, our hardware can only track one human's motion in the X direction. Since we wanted to evaluate the performance of the QVFF for more challenging environments we simulated more complex human motions to obtain the results presented in the following sections.

5.3 *Comparison with five state-of-the-art navigation algorithms*

In this section simulations with five state-of-the-art navigation algorithms for moving obstacles are compared with the QVFF algorithm. The human and robot paths, the closest distance from the robot to the human and the distance from the robot centre to the goal are plotted for the six algorithms in Figure 12. A human path with an unpredictable direction change is used (plotted in light green). The robot is holonomic with the same specifications as used in the previous section.

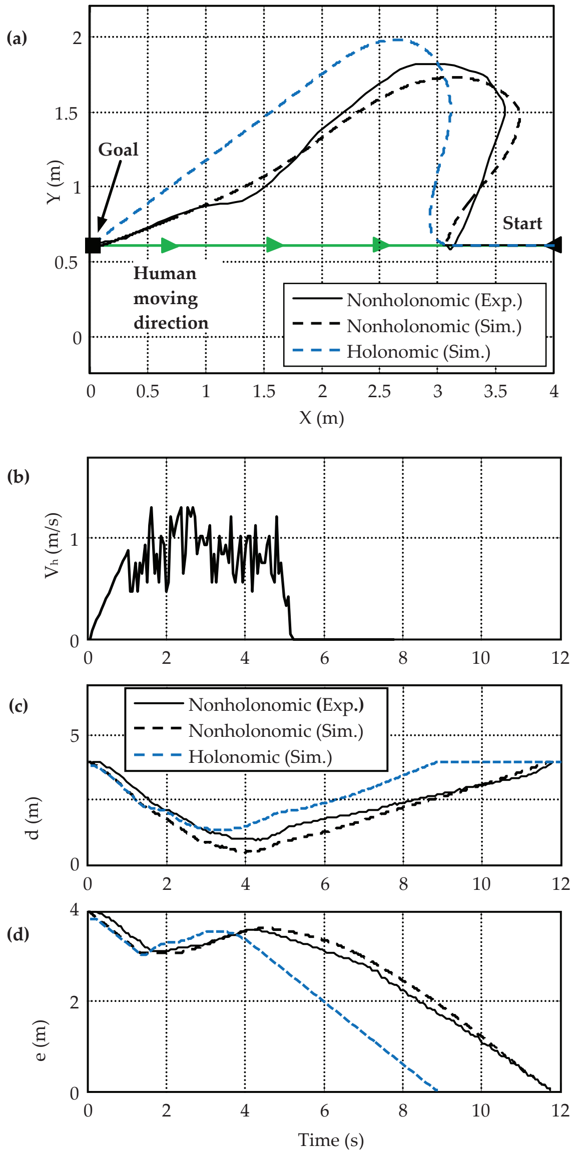

Figure 11. Experimental and simulation results for nonholonomic and holonomic robots navigated by the QVFF algorithm, with a human walking in the collinear condition. (a) Paths. (b) Human velocity. (c) Closest distances to the human. (d) Distances to the goal.

The robot starts from (4, 0)m and must avoid the walking human to reach its goal at (0, 0)m. The walking human initially starts from (0.6, 0)m with a (1, 0)m/s velocity. After 1s, the human slows down with a (-1, 0)m/s^2 deceleration until it stops at (2, 0)m and then moves sideways with a (0, 1)m/s^2 acceleration. After the human's velocity reaches (0, 1)m/s, the human maintains a constant velocity. In the figure, VO denotes the velocity obstacle algorithm [2], RRT denotes the rapidly exploring random trees algorithm [15], D* denotes the D* algorithm [6], Ge & Cui denotes their algorithm [13] and Masoud denotes his algorithm [14]. The gains for the Masoud, Ge & Cui and QVFF algorithms were optimally tuned. The detailed tuning process is omitted here for brevity and can be found in [24]. The first three algorithms (i.e., VO, D* and RRT) belong to the first category described in Section 1.2, while the remainder belong to the second category. During the simulations, the future motion of the human, including the walking direction change, is unknown for all six algorithms. From Figure 12b we can see that the first three algorithms caused collisions (indicated by *d* reaching zero) after the human changed its motion direction. These collisions occurred at the locations shown by the asterisks in Figure 12a. The algorithms in the second category all successfully completed the navigation. The robot path with the Ge & Cui algorithm ran straight from the robot start position to its goal. It was not plotted to avoid obscuring the human's path. It did not detour around the human since their detour force is zero under the collinear condition. As shown by the *e* plot, their repulsive force caused the robot to stop momentarily at 1.9s and then move backwards until the time reached 4.1s.

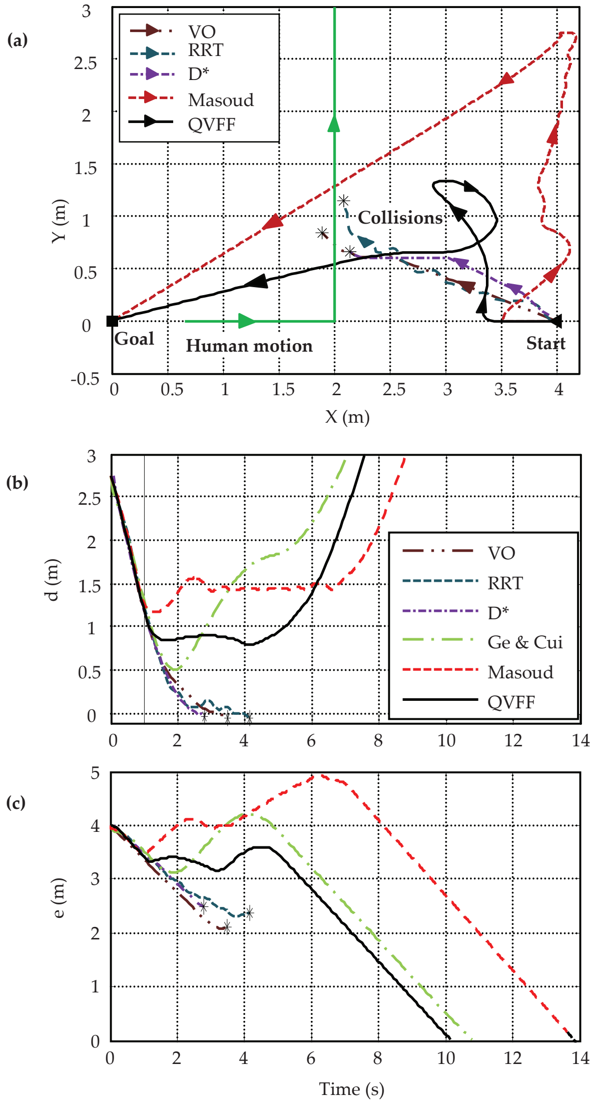

Figure 12. Comparison of simulation results for six navigation algorithms avoiding a human with unpredictable motion. (a) Paths. (b) Closest distances between the robot and humans. (c) Distance from the robot centre to the goal.

After that, the motion of the human in the positive Y direction allowed their algorithm to drive the robot to the goal. It arrived at 10.9s. Of course this "backup and wait" strategy would prevent the robot from reaching the goal if the obstacle was static. The Masoud and QVFF algorithms made detours to avoid the human. With the QVFF algorithm the minimum *d* was 0.8m and the robot reached the goal at 10.2s. The larger detour made by the Masoud algorithm increased its minimum *d* to 1.2m, but also increased its arrival time to 14.0s.

5.4 Collision avoidance with three walking humans

The QVFF algorithm was also simulated with an environment containing three walking humans (denoted Human 1, Human 2 and Human 3). The paths of the humans and robot are plotted in Figure 13a.

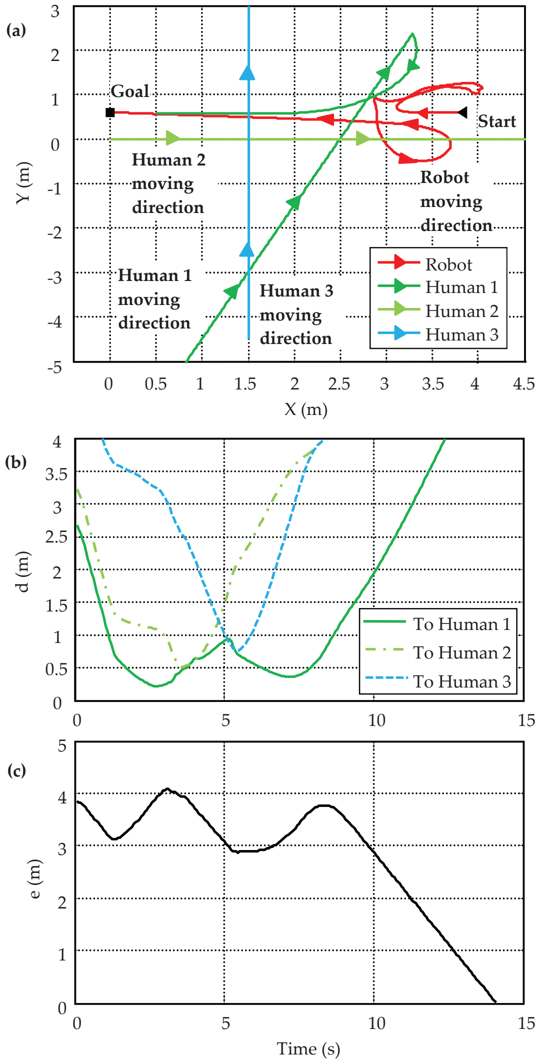

Figure 13. Simulation of the QVFF algorithm for an environment with three walking humans. (a) Paths. (b) Closest distances between the robot and humans. (c) Distance from the robot centre to the goal.

The closest distances from the robot to each human are plotted in Figure 13b and the distance from the robot centre to the goal is plotted in Figure 13c. The robot was identical to the previous section. It started from (3.85, 0.6)m and its goal was at (0, 0.6)m. The motion of Human 1 was unpredictable. Starting from (0.6, 0.6)m, he moved to the right at 0.7m/s velocity. After 1.4s, he smoothly turned counter-clockwise until the time equalled 4.4s. Next, he stopped for 1.0s. Finally, he moved left and downwards at 1m/s. Simultaneously, Human 2 started from (0, 0)m and moved left at 1m/s and Human 3 started from (1.5, -4.5)m and moved upwards at 1m/s velocity. From Figure 13b we can see that the robot successfully avoided all three humans. The minimum *d* was 0.22m to Human 1. This is not surprising since his motion was the most unpredictable. The robot arrived at its goal in 14.2s.

6. Conclusions

A novel VFF-based navigation algorithm, termed QVFF, was presented. It is primarily intended for mobile robots sharing working environments with people. It features improved virtual force functions for avoiding moving obstacles, it may be used with both holonomic and nonholonomic robots and its stability was proven using Lyapunov's second method. Active and critical regions were used to deal with the uncertainty of human motion. A procedure was introduced to calculate the region sizes for disk-shaped and polygonal obstacles based on the worst-case avoidance conditions. Simulation and experimental results were presented. The QVFF algorithm outperformed five state-of-the-art navigation algorithms for an environment with one human walking with an unpredictable change in direction. For a human walking in the collinear condition and a challenging environment containing three walking humans, the QVFF algorithm successfully generated collision-free paths to the goal.

7. Acknowledgments

The funding provided by the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged. The authors also wish to thank Graham Ashby and Justin Flett for their assistance with the final formatting of this paper.

8. References

[1] Fox D., Burgard W., Thrun S. (1997) The dynamic window approach to collision avoidance, *IEEE Robotics and Automation Magazine,* Vol. 4, No. 1, 1997, pp. 23–33.

[2] Fiorini P., Shiller Z. (1998) Motion planning in dynamic environments using velocity obstacles, *The International Journal of Robotics Research*, Vol. 17, No. 7, 1998, pp. 760–772.

[3] LaValle S. M., Kuffner J. J. (2001) Randomized kinodynamic planning, *The International Journal of Robotics Research,* Vol. 20, No. 5, 2001, pp. 378–400.

[4] Hsu D., Kindel R., Latombe J., Rock, S. (2002) Randomized kinodynamic motion planning with moving obstacles, *The International Journal of Robotics Research*, Vol. 21, No. 3, 2002, pp. 233–255.

[5] Park S. H., Lee B. H. (2006) Analysis of robot collision characteristics using the concept of the collision map, *Robotica*, Vol. 24, No. 3, 2006, pp. 295–303.

[6] Ferguson D., Stentz, A. (2006) Using interpolation to improve path planning: the field D* algorithm, *Journal of Field Robotics*, Vol. 23, No. 2, 2006, pp. 79–101.

[7] Jacobs S., Ferrein, A., Schiffer, S., Beck D. Lakemeyer, G. (2010) Robust collision avoidance in unknown domestic environments, *RoboCup 2009: Robot Soccer World Cup XIII*, pp. 116-127, 2010, Springer, Berlin.

[8] Khatib O. (1985) Real-time obstacle avoidance for manipulators and mobile robots, *Proceedings of 1985 IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 500–505, 1985.

[9] Borenstein J., Koren Y. (1989) Real-time obstacle avoidance for fast mobile robots, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 19, No. 5, 1989, pp. 1179–1187.

[10] Khatib O., Yokoi K., Brock O., Chang K., Casal, A. (1999) Robots in human environments: basic autonomous capabilities, *The International Journal of Robotics Research*, Vol. 18, No. 7, 1999, pp. 684–696.

[11] Masoud S. A., Masoud A. A. (2000) Constrained motion control using vector potential fields, *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 30, No. 3, 2000, pp. 251–272

[12] Yang S. X., Meng M. (2001) Neural network approaches to dynamic collision-free trajectory generation, *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Vol. 31, No. 3, 2001, pp. 302–318.

[13] Ge S. S., Cui Y. J. (2002) Dynamic motion planning for mobile robots using potential field method, *Autonomous Robots*, Vol. 13, No. 3, 2002, pp. 207–222.

[14] Masoud A. A. (2007) Decentralized self-organizing potential field-based control for individually motivated mobile agents in a clustered environment: a vector-harmonic potential field approach," *IEEE Transaction on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 37, No. 3, 2007, pp. 372–390

[15] Mastellone S., Stipanović D., Graunke C., Intlekofer K., Spong M. (2008) Formation control and collision avoidance for multi-agent non-holonomic systems: theory and experiments, *The International Journal of Robotics Research*, Vol. 27, No. 1, 2008, pp. 107–126.

[16]Loizou S., Kyriakopoulos K. (2008) Navigation of multiple kinematically constrained robots, *IEEE Transactions on Robotics*, Vol. 24, No. 1, 2008, pp. 221–231.

[17] Vanualailai J., Sharma B., Nakagiri S. (2008) An asymptotically stable collision-avoidance system, *International Journal of Non-Linear Mechanics*, Vol. 43, No. 9, 2008, pp. 925–932.

[18] Fahimi F., Nataraj C., Ashrafiuon H. (2009) Real-time obstacle avoidance for multiple mobile robots, *Robotica*, Vol. 27, No. 2, 2009, pp. 189–198.

[19] Deng M., Inoue A., Sekiguchi K., Jiang L. (2010) Two-wheeled mobile robot motion control in dynamic environments, *Robotics and Computer-Integrated Manufacturing*, Vol. 26, No. 3, 2010, pp. 268–272.

[20] Zeng L., Bone, G. M. (2010) Collision avoidance for nonholonomic mobile robots among unpredictable dynamic obstacles including humans, *Proceedings of 2010 IEEE Conference on Automation Science and Engineering*, pp. 940-947, 2010.

[21] Svenstrup, M., Hansen, S. T., Andersen, H. J., Bak, T. (2011) Adaptive human-aware robot navigation in close proximity to humans, *International Journal of Advanced Robotic Systems*, Vol. 8, No. 2, 2011, pp. 7-21.

[22] Martin P., Marsh A. (1992) Step length and frequency effects on ground reaction forces during walking, *Journal of Biomechanics*, Vol. 25, No. 10, 1992, pp. 1237–1239

[23] van Emmerik R., Wagenaar R. (1996) Effects of walking velocity on relative phase dynamics in the trunk in human walking, *Journal of Biomechanics*, Vol. 29, No. 9, 1996, pp. 1175–1184

[24] Zeng L. (2011) Design and control of human-friendly robots, *Ph.D. Thesis*, McMaster University, 2011, Hamilton, Canada.