ETA-12

AUTONOMOUS NAVIGATION AND OBSTACLE AVOIDANCE CAPABILITY IN OUTDOOR ENVIRONMENTS FOR CAR ROBOT

*Sittichai BOONYARAK, Pradya PREMPRANEERACH
Department of Mechanical Engineering, Rajamangala University of Technology Thanyaburi
39 Moo 1 Rangsit-Nakhonnayok Road, Klong6, Thanyaburi, Pathumthani 12110, Thailand

Corresponding author: Sittichai BOONYARAK. E-mail: south_phuket@hotmail.com

ABSTRACT

In this research, we propose and implement obstacle detection and avoidance methods combining with a local incremental planning technique for navigating a car-like robot, equipped with GPS, IMU, and laser range finder sensors. To detect an obstacle, radial distance data from laser range finder is transformed to cartesian coordinate and mapped to a global occupancy-grid map. Thus, obstacles in real dynamic environment can be detected from a probability of occurrence in the global occupancy-grid map according to a specified threshold. Then, an integration of the local incremental planning and potential field techniques employ obstacles' location from the global map to plan a trajectory from the robot current position to a desired target position. As a result, the vehicle can seek the target and avoid obstacles at the same time. Results of a vehicle simulation, developed in Matlab, reveal that the car-like robot can approach the target point and steer away from static obstacles according to an instantaneous virtual force acting on the vehicle center. Experimental results show that the proposed navigation method, developed in Matlab, can detect obstacle and avoid static obstacles and navigate to the desire target waypoints in the global occupancy map.

Keywords: Obstacle Avoidance; Occupancy-Grid Map; Potential Field; Local Incremental Planning; Car Robot

INTRODUCTION

In present, autonomous and semi-autonomous technologies have played an important role in our daily life in various fields, such as communi-cation, harvesting, manufacturing, and transpor-tation. Particularly, modern transportations on land, in water or in air, are somehow contained an autonomous or semi-autonomous function, like a cruise control for car and an auto-pilot for airplanes and boats, to assist human driving and to reduce human error. To avoid frequent car accidents in Thailand and around the world, most active controls with network communication have been developed for automatic parking, preventive crash stop as well as autonomous driving. For a few decades, researchers in both university and automotive industry have been interested in developing a completely autonomous car, like in Drappra grand challenge or Google car, such a smaller number of accidents will occur on the road.

The autonomous system architecture for car is comprised for two main systems: control and motion-planning systems. Analyzing information from dynamic environments in real time is performed by the motion-planning system, such as Probabilistic roadmaps [1], Cell decomposition technique [1], Potential field technique [1] or Vector field Histogram method [2], which the algorithm of interest. The objective of the motion planning is 1) to evaluate optimal path from current position to the desired location with respect to specified quantity, 2) to generate commands to several control systems, including low-level controls, and 3) to receive feedback signals from all sensors. Most of low-level control systems employ a PID control or a feedback linearization control, which can be applied to our research.

To achieve the autonomous navigation with obstacle avoidance capability, two main systems: perception and motion planning are required. First, one of the popular perception techniques is to measure radial distances using a laser range finder because of its accuracy. S. Geyer and E. Johnson [3] demonstrated three techniques for obstacle mapping with data from the laser range finder 1) grid-based map, 2) map scrolling, and 3) confidence algorithm. According to J. Borenstein and Y. Koren [4],

their perception method used a certainty grid map or a fixed-size grid map, which is similar to that in [3], and contained simple map analysis as well. Moreover, S. Solanki [6] introduced a traversability-grid approach, which is better than the certainty grid map since this map always aligns along the North-East direction. Also, the robot is located at the center of the traversability-grid for safety in obstacle avoidance behavior. J. Roberts and P. Corke [5] descried a threshold of gradient method for building the terrain map. Likewise, R. Zou [7] basically detected obstacles in general environment using a threshold method which in this research, it can detect obstacles using a stereo camera and a radar sensor instead. Second, the motion planning performs the obstacle avoidance and the goal-seeking behavior at the same time, which employ feedback signal from all sensors and control low-level systems. A. Bemporad, etc. [8] proposed a local incremental planning method using potential function. S.S.GE and Y.J.CUI [9] modified the potential function technique, that is applicable for dynamic environment. C. Wuthishuwong, C. Silawatchananai and M. Parnichkun [10] applied the potential field method to an intelligent vehicle for the autonomous navigation with obstacle-avoidance ability. However, the virtual force is directly computed from laser range finder without building the obstacle map.

Main objective of our research is to implement the obstacle avoidance method and the automatic target-seeking navigation for the car-like robot by combining the obstacle detection method with the potential field technique and the local incremental planning. This paper is organized as the following: Section 1 describes hardware of the car-like robot, Section 2 introduces the obstacle detection method and the occup-ancygrid map, Section 3 explains the goal-seeking navigation and obstacle avoidance method using potential field and local incremental planning, and Section 4 presents simulation and experimental results of automatic navigation with obstacle avoidance ability of the car-like robot in real environ-ment.

HARDWARE

In this paper, the car-like robot is the modified electrical ATV. A vehicle steering for turning two front wheels is equipped with a 24V DC brush motor, controlled its angular rotation in the position mode. The vehicle is real-wheel drive using a 36V DC brushless motor, controlled its angular velocity in the velocity mode. Both front- and rear-wheels motor drives, using Accelus Panels from Copley Motion Control, are connected to a NI PCI-6221 Data Acquisition (DAQ) card from National Instrument to perform the low-level control of these two DC motors. The low-level control commands are calculated from the robot position and obstacles' position at each instance. After motion-planning command computation using MATLAB, the DAQ card generates the analog control signal to motor drives and receives encoder pulses from two motors.

For the high-level control, an embedded computer receives the car position and the heading angle from the Global Positioning System (GPS) of and Inertia Measurement Unit (IMU), respectively. Microstrain 3DM GX3-45 IMU combined with GPS can provide the robot orientation from 3-axis Euler angles with a sensor x-axis pointing forward and the robot position from Latitude/Longitude. Furthermore, obstacles' location in perception module can be detected with an outdoor LMS511-Pro Sick laser scanner for real-time virtual force calculation. This Sick laser scanner has a maximum detection range of 80 m with a maximum field of view of 180 degrees.

Fig. 1 illustrates the hardware architecture with both low-level and high-level controls of the car-like robot. Fig. 2 shows successfully installation of all sensors, actuators, and controllers in the car-like robot.

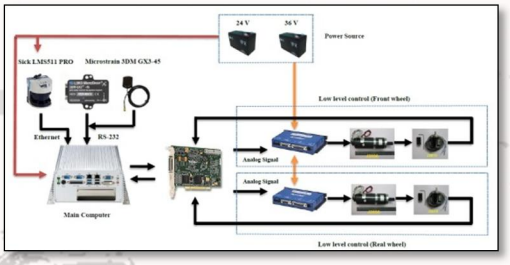

Fig. 1 Hardware architecture of the car-like robot

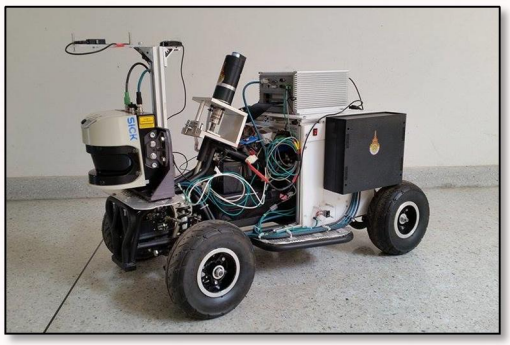

Fig. 2 The car-like robot

OBSTACLE DETECTION METHOD

Main objective of the perception module is to detect all obstacles in the vehicle path, to identify obstacle location in the global coordinates, and to pass through the correct obstacle information to the motion planning algorithm. Our car-like robot is operated in outdoor environment. In this section, the global coordinate's system and ECEF transformation equation are presented for evaluating obstacles' location relative to the global fixed environment. Furthermore, a fixed grid map is constructed as a place holder for obstacle locations. In the last part, a specify threshold is applied to a probability of obstacle occurrence in the global occu-pancy-grid map to provide more accurate obstacle locations.

Coordinate Transformation

In this paper, Earth Centered Earth Fixed (ECEF) frame and tangent plane [11] are employed as the main reference frames. The coor-dinate transformation from Lati-tude/Longitude to the ECEF frame and that from the ECEF frame to the tangent plane are described below.

ECEF Coordinate Transformation:

Latitude/Longitude and altitude measured form GPS at each instance are transformed to the x, y, z position in the ECEF frame as in Eqn. (1):

$$x = (R_N + h)\cos(\phi)\cos(\lambda)$$
$$y = (R_N + h)\cos(\phi)\sin(\lambda)$$
$$z = [R_N(1-e^2) + h]\sin(\phi) \quad \ldots(1)$$

where

$R_N = \frac{a}{\sqrt{(1-e^2\sin^2(\phi))}}$ is the normal radius of earth.

a is equatorial radius = 6378137 [m].

h is altitude or geodetic height [m].

e is eccentricity = 0.08181919.

ϕ is latitude [degree].

λ is longitude [degree].

ECEF to tangent plane Transformation:

the position in ECEF frame can be transformed to the tangent plane, which an origin is located at the vehicle center. The x- and z-axes of the tangent plane point toward true North and perpendicular to the Earth surface. This transformation could be written as in Eqn. (2):

$$\begin{bmatrix} x^t \\ y^t \\ z^t \end{bmatrix} = R_t^e \left(\begin{bmatrix} x_2^e \\ y_2^e \\ z_2^e \end{bmatrix} - \begin{bmatrix} x_0^e \\ y_0^e \\ z_0^e \end{bmatrix} \right) \quad \ldots(2)$$

where

$$R_e^t = \begin{bmatrix} -\sin(\phi)\cos(\lambda) & -\sin(\phi)\sin(\lambda) & \cos(\phi) \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ -\cos(\phi)\cos(\lambda) & -\cos(\phi)\sin(\lambda) & -\sin(\phi) \end{bmatrix}$$

$\begin{bmatrix} x_0^e & y_0^e & z_0^e \end{bmatrix}^T$ is an initial position in the ECEF frame.

$\begin{bmatrix} x_2^e & y_2^e & z_2^e \end{bmatrix}^T$ is a current position in the ECEF frame

$\begin{bmatrix} x^t & y^t & z^t \end{bmatrix}^T$ is a distance from initial position to current position along the tangent plane

Transformation from radial distance of laser data to global coordinate:

all radial distance data from laser range finder sensor, which generally expressed in a polar coordinate, must be transformed to the global vehicle coordinate system that is based on a Cartesian coordinate system. Thus, the transformation from the polar to Cartesian coordinates can be determined as the following Eqn. (3).

$$x_l = r_l\cos(\theta_l)$$
$$y_l = r_l\sin(\theta_l) \quad \ldots(3)$$

where

r_l isradial distance data from laser in the polar coordinate.

θ_l is angle of each radial distance data

x_l is distance along the x-axis of the Cartesian coordinate

y_l is distance along the y-axis of the Cartesian coordinate

With the assumption that the car-like robot motion is restricted only in the horizontal plane and a z-axis of IMU coordinate system is opposite to the z-axis of the tangent plane. As a result, the transformation of the laser sensor coordinate to the global coordinate is expressed as in Eqn. (4).

$$P_t = \begin{bmatrix} & & & x_t \\ & [R_{(3\times3)}] & & y_t \\ & & & z_t \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} y_l \\ x_l \\ -1 \\ 0 \end{bmatrix} \quad \ldots(4)$$

where

P_t is laser data in global coordinates
x_t, y_t, z_t is translation matrix in tangent plane
x_l, y_l is point laser in local coordinate

And a rotation matrix, R(3x3), used in Eqn. (4), can be expressed as the following from

$$R = \begin{bmatrix} \cos(\beta)\cos(\alpha) & \sin(\theta)\sin(\beta)\cos(\alpha) - \cos(\theta)\sin(\alpha) & \cos(\alpha)\cos(\theta)\sin(\beta) + \sin(\theta)\sin(\alpha) \\ \cos(\beta)\sin(\alpha) & \sin(\alpha)\sin(\beta)\sin(\theta) + \cos(\alpha)\cos(\theta) & \cos(\theta)\sin(\beta)\sin(\alpha) - \sin(\theta)\cos(\alpha) \\ -\sin(\beta) & \sin(\theta)\cos(\beta) & \cos(\theta)\cos(\beta) \end{bmatrix}$$

where

θ is a roll angle [radian]
β is a pitch angle [radian]
α is a yaw angle [radian]

All coordinate transformation in Eqn. (1)-(4) mentioned above will be applied to the obstacle detection and obstacle avoidance method, described in later section of this paper.

Occupancy-grid map

One of the existing map construction techniques is the occupancy-grid map representation. The occupancy-grid map, used in this research, consists of two types of grid maps: local- and global-grid maps, representing in the Cartesian coordinate. For the local-grid map, we divide a whole workspace of 4 m^2 into square elements (or called "cell") where each cell dimension is 2 m by 2 m. A center of the local-grid map is coincided with a center of the car-like robot and the local-grid map is translates and rotated within the global-grid map along with the robot motion. The fixed global-grid map contains an entire workspace of 90 m^2, which its origin is located at the lower left corner. A diagram in Fig. 4 illustrates the occupancy-grid map concept. At each instance of time following the robot motion, radial-distance measurements, returned from the laser range finder, are transformed into the global coordinate. Then, an occurrence value of each cell is incremented according to those transformed distances that fall within corresponding cell area. After the occurrence value in the global-grid map is archived, obstacle location could be detected in next step.

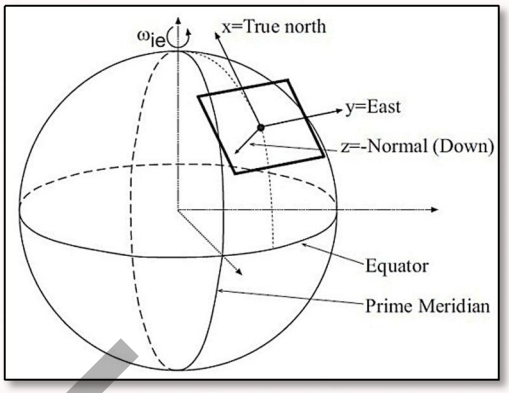

Fig. 3 ECEF coordinate and tangent plane [11]

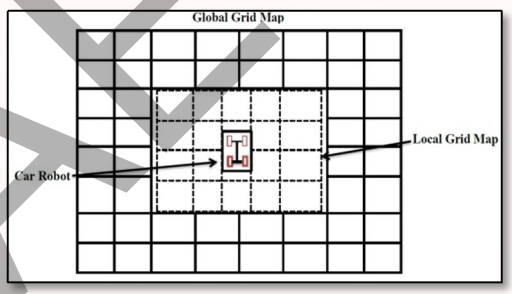

Fig. 4 Diagram of the occupancy-grid map, consisted of local- and global-grid maps

Obstacle detection

The obstacle detection technique is one of the important part of perception module so that the obstacle avoidance capability could be achieved. In this research, the occurrence probability of obstacles in the global occupancy-grid map is proposed to help us identify the obstacles' location. After the global-grid map is updated at each instance following the robot motion, the occurrence probability in each cell increases. Then, the obstacle location could be detected when the occurrence probability increases above a specified threshold. If the occurrence probability of each cell in the global-grid map exceeds a pre-defined threshold, the algorithm will label that individual cell as then obstacle instead of free space. In next iteration, the algorithm contin-uously updates until the robot reaches the target point. The obstacle position from this detection algorithm then passes to the potential filed method to generate the virtual force for the motion-planning system.

OBSTACLE AVOIDANCE

The proposed autonomous obstacle avoidance method for outdoor environment combines the potential field method with the local incremental planning technique, which is a feedback control algorithm using the virtual force from the potential field. In this section, both the potential field method along with its attractive and repulsive functions and the local incremental planning are introduced.

Potential field method

The potential field method is used for creating a virtual force, acting on the robot, to attract the robot to a desired target point and to push the robot away from nearby obstacles at the same time. Two relative distances between obstacles and current robot position as well as between current robot locations and the target point help to compute the repulsive force and the attractive force, respectively. The advantage of the potential field are 1) the virtual force can be updated in real time such that the robot path in the global occupancy grid does not need to be pre-planned in advance and 2) the planning method can be coupled directly to the potential field. Its main disadvantage is that the robot might be trapped at the local minim position. The potential field is comprised of two potential functions as the following:

Attractive Potential Function:

In literature, the existing attractive potential function [12] is defined as a Euclidean norm or a relative distance between car robot and target. A weighted sum of the relative distance and relative speed between robot and target is used in a new attractive potential function [9], in Eqn. (5), so that target movement can be accounted for.

$$U_{att}(\mathbf{p},\mathbf{v}) = \alpha_p \left\| \mathbf{p}_{tar}(t) - \mathbf{p}(t) \right\|^m + \alpha_v \left\| \mathbf{v}_{tar}(t) - \mathbf{v}(t) \right\|^n \quad \ldots(5)$$

where $p(t)$ and $p_{tar}(t)$ are positions of robot and target, respectively. Similarly, $v(t)$ and $v_{tar}(t)$ correspondingly denote the robot and target velocities. The symbol $\|\bullet\|_m$ is represented the m-norm of vector, α_p and α_v are positive scalar weights for relative position and velocity norms, and m and n are positive constants. Note that all positions are expressed in a 3-dimensional space as $p=[x,y,z]^T$ or in a 2-dimensional space as $p=[x,y]^T$.

The virtual attractive force is defined as a negative gradient of the attractive potential function with respect to the robot position and velocity as in Eqn. (6).

$$\begin{aligned} \mathbf{F}_{att}(\mathbf{p},\mathbf{v}) &= -\nabla U_{att}(\mathbf{p},\mathbf{v}) \\ &= -\nabla_p U_{att}(\mathbf{p},\mathbf{v}) - \nabla_v U_{att}(\mathbf{p},\mathbf{v}) \end{aligned} \quad \ldots(6)$$

where, the gradient of the attractive potential with respect to position is

$$\nabla_p U_{att}(\mathbf{p},\mathbf{v}) = \partial U_{att}(\mathbf{p},\mathbf{v}) / \partial \mathbf{p} \quad \ldots(7)$$

And the gradient of the attractive potential with respect to velocity is

$$\nabla_v U_{att}(\mathbf{p},\mathbf{v}) = \partial U_{att}(\mathbf{p},\mathbf{v}) / \partial \mathbf{v} . \quad \ldots(8)$$

Repulsive Potential Function:

To avoid collision with obstacles, the hyperbolic repulsive potential function, calculated as inverse of the relative position between robot and obstacle, is generally used to generate the repulsive force. With an assumption of convex-shape obstacles, a new repulsive potential function, defined in [9], included both of the inverse of relative position and relative velocity, is expressed as the following.

$$U_{rep}(\mathbf{p},\mathbf{v}) = \begin{cases} 0, \text{ if } \rho_S(\mathbf{p},\mathbf{p}_{obs}) - \rho_m(\upsilon_{RO}) \geq \rho_0 \text{ or } \upsilon_{RO} \leq 0 \\ \eta\left(\dfrac{1}{\rho_S(\mathbf{p},\mathbf{p}_{obs}) - \rho_m(\upsilon_{RO})} - \dfrac{1}{\rho_0}\right), \text{ if } 0 < \rho_S(\mathbf{p},\mathbf{p}_{obs}) - \rho_m(\upsilon_{RO}) < \rho_0 \\ \text{not defined, if } \upsilon_{RO} > 0 \text{ and } \rho_S(\mathbf{p},\mathbf{p}_{obs}) < \rho_m(\upsilon_{RO}) \end{cases} \quad \ldots(9)$$

where $\rho_S(\mathbf{p}_{obs}(t),\mathbf{p}(t))$ is defined as a shortest Euclidean distance between robot and obstacle, $v_{RO}(t)$ is a relative velocity between robot to obstacle, $\rho_m(v_{RO})$ is defined as a distance traveled before robot's v_{RO} reduce to zero, ρ_o denotes a maximum obstacle size, and η is positive constants. Thus, the virtual repulsive force can be computed as a negative gradient of the repulsive potential function, as in eqn. (10).

$$\begin{aligned} \mathbf{F}_{rep}(\mathbf{p},\mathbf{v}) &= -\nabla U_{rep}(\mathbf{p},\mathbf{v}) \\ &= -\nabla_p U_{rep}(\mathbf{p},\mathbf{v}) - \nabla_\upsilon U_{rep}(\mathbf{p},\mathbf{v}) \end{aligned} \quad \ldots(10)$$

where, $\nabla_p, \nabla_\upsilon$ are the gradient with respect to position and velocity, as well. Or, Eqn. (10) can be rewritten as

$$\mathbf{F}_{rep}(\mathbf{p},\mathbf{v}) = \begin{cases} 0, \text{ if } \rho_S(\mathbf{p},\mathbf{p}_{obs}) - \rho_m(\upsilon_{RO}) \geq \rho_0 \text{ or } \upsilon_{RO} \leq 0 \\ \mathbf{F}_{rep1} + \mathbf{F}_{rep2}, \text{ if } 0 < \rho_S(\mathbf{p},\mathbf{p}_{obs}) - \rho_m(\upsilon_{RO}) < \rho_0 \text{ and } \upsilon_{RO} > 0 \\ \text{not defined, if } \upsilon_{RO} > 0 \text{ and } \rho_S(\mathbf{p},\mathbf{p}_{obs}) < \rho_m(\upsilon_{RO}) \end{cases}$$

where the first part of the repulsive force, $\mathbf{F}_{rep1}$, points in a direction away from the obstacle and the second part of repulsive force, $\mathbf{F}_{rep2}$, helps pushing the robot to go around the obstacle.

Local Incremental Planning

The local incremental planning is one of the motion-planning techniques that can generate a real-time robot trajectory. The local incremental planning consists of a high-level nonlinear control algorithm that incorporates both real-time velocity and position of the car-like robot, target, as well as all obstacles. The generated trajectory from this planning method includes the resultant virtual force acting on the robot, computed from the potential functions. As a result, the car-like robot can avoid obstacles and approach the desired target at the same time.

Nonlinear Control Law:

According to a kinematic model of the carlike robot in [8], a state-space representation can be described in a nonlinear form as in Eqn. (11).

$$\dot{X} = G(X)u \qquad \ldots(11)$$

where $X = [x \quad y \quad \alpha l\theta \quad \beta]^T$ and α is a positive weighting constant. Given a specify desired feasible trajectory: $X_d = [x_d \quad y_d \quad \alpha l\theta_d \quad \beta_d]^T$, a control law or command input (u) can be computed from a pseudo-inverse of the nonlinear function G(X) as

$$u = G^{\#}(X)\dot{X}_d = [G^T(X)G(X)]^{-1}G^T(X)\dot{X}_d \qquad \ldots(12)$$

A feedback linearization technique, minimizing the trajectory tracking error in a least-square sense, is applied to the above control law. Therefore, the feedback law for the robot steering and for rear-wheel drive can be reformulated as the following

$$u_1 = \frac{\dot{x}_d \cos\beta + \dot{y}_d \sin\beta + \alpha^2 l\dot{\theta}_d \sin\phi}{1 + \alpha^2 \sin\phi}, \text{ and}$$

$$u_2 = \dot{\beta}_d . \qquad \ldots(13)$$

Local Trajectory Generation:

To create the desired trajectory, the virtual forces could be obtained from the potential function. The virtual attractive force in Eqn. (6) must act only on front wheels to steer the robot toward the target, while the virtual repulsive force in Eqn. (10) must be imposed on both front and rear wheels as shown in Fig. 5. $\mathbf{F}_f = (F_{f,x}, F_{f,y})$ and $\mathbf{F}_r = (F_{r,x}, F_{r,y})$ res-pectively represent forces acting on front wheels and rear wheels. These forces relate to each other and to a moment on a front-wheel axe (M_θ) according to the virtual work principle [8], as shown in Eqn. (14).

$$\begin{bmatrix} F_{f,x} \\ F_{f,y} \\ M_\theta \\ M_\beta \end{bmatrix} = \begin{bmatrix} F_{r,x} \\ F_{r,y} \\ l(F_{r,x}\sin\theta - F_{r,y}\cos\theta) \\ 0 \end{bmatrix} \qquad \ldots(14)$$

As a result, the total virtual force and moment acting on the robot's front wheel are expressed as

$$\mathbf{F} = \mathbf{F}_f + \mathbf{F}_r$$

$$\mathbf{M} = M_\theta \qquad \ldots(15)$$

Derivatives of the desired trajectory are required in the feedback linearization command (u_1,u_2) in Eqn. (13), thus these derivatives could be computed from the total virtual force and moment acting on front wheels, as in Eqn. (16), with a simple quasi-static assumption.

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \end{bmatrix} = k_{f1}\mathbf{F} \quad \text{and} \quad \dot{\theta}_d = k_{f2}\mathbf{M} \qquad \ldots(16)$$

where, positive constants, (k_{f1}, k_{f2}), are equivalent to inverse linear and angular damping coefficients. Thus, the calculation of the control input u_1 in Eqn. (13) should obtain sufficient damping from these derivatives in Eqn. (16).

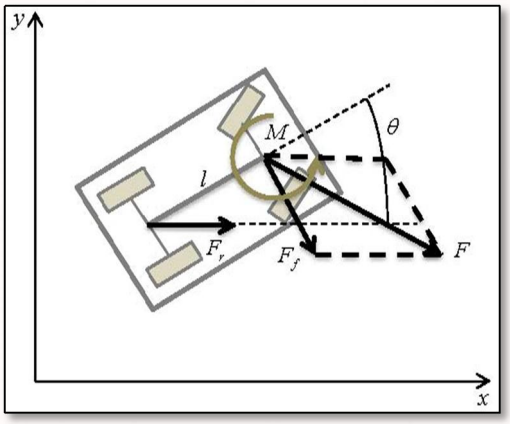

Fig. 5 Forces and torque acting on the car-like robot.

Furthermore, the steering control input u_2 could be expressed as in Eqn. (17),

$$u_2 = -\frac{k_\beta(\beta-\beta_d)}{sign(\cos(\beta-\beta_a))} + \frac{u_1}{\|F\|^2}\begin{bmatrix}-F_y & F_x\end{bmatrix}\nabla F\begin{bmatrix}\cos\beta\\ \sin\beta\end{bmatrix} \quad \ldots(17)$$

where $\beta_a = ATAN2(F_y, F_x)$ is a directional angle of the front-wheel force, $\beta_d = \beta - \arcsin(\sin(\beta-\beta_a))$ is a directional angle of the desired front-wheel force, k_β is a positive constant gain of a steering response time, and a partial derivative of the total force with respect to the horizontal plane axes is given below

$$\nabla F = \begin{bmatrix}\frac{\partial F_x}{\partial x} & \frac{\partial F_x}{\partial y}\\ \frac{\partial F_y}{\partial x} & \frac{\partial F_y}{\partial y}\end{bmatrix} \quad \ldots(18)$$

According to the virtual force from the potential field, these input commands produce desired linear and angular velocities of the car-like robot as well as the steering angle of the front wheel. Next, the motion planning of the robot using this trajectory generation technique is tested on the car-like robot kinematics simulations.

SIMULATION RESULTS

To validate the motion planning algorithm combining with the potential field, kinematic model of the car-like robot is implemented in MATLAB/Simulink, then goal-seeking simulations with obstacles in the global-grid map. This section demonstrates two main simulations: 1) the obstacle detection in the global-grid map and 2) target-seeking with two fixed obstacles in the global map.

In the first simulation, constructions of the global- and local-grid maps are tested for the obstacle detection by threshold the probability of occurrence. There are two obstacles in the global map and the robot moves forward along a straight line parallel to the x-axis from left to right. In the global-grid map, we subdivide workspace into small square grids with dimension of 100 x 100 cells. The local-grid map around the car-like robot has a dimension of 30 x 30 cells. As the robot moves along the x direction with 10 cell increments until its center locates at 80 unit, as in Fig. 6 the global map contain value in each cell equal zero but if it can detect obstacle as show in Fig. 6. The probability of occurrence in the global-grid map is updated as in Fig. 7. A circular obstacle becomes prominent such that the thresholding could help identifying the obstacle location.

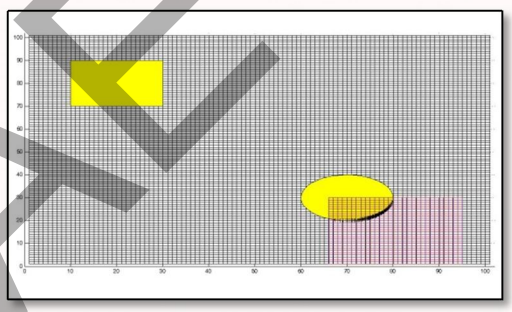

Fig. 6 Simulation of obstacle detection

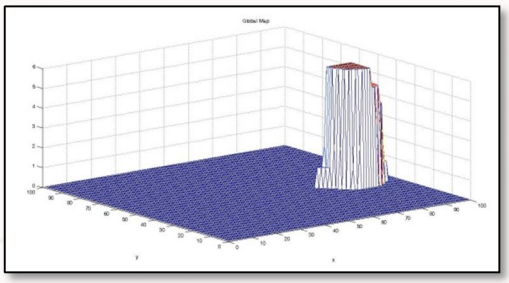

Fig. 7 The probability of occurrence in the global-grid map when the robot moves along x direction.

In the second simulation implemented with MATLAB/Simulink, two fixed obstacles, located at (6m,10m) and (20m,20m), along the straight path from current robot position to the target point in the workspace. In this simulation, a dimension of the car-like robot is 1.2 m long x 0.8 m wide. An initial center position of the car-like robot is located at (0,0) with initial heading angle of 0° and initial front-wheel steering angle of 0° and a target point is located at (30m,30m). All feedback linearization control constants are $k_\beta = 1$, $k_{f1} = 2.5$, k_{f2}

= 1 and α = 0.95. The attractive virtual force is initially large and it gradually decreases to zero as it approaches the target within 30 sec. The repulsive force is almost zero at an initial position and near the target, which is far away from obstacles. But, when the robot is near the obstacle, the repulsive force reaches a maximum value. Therefore, the car-like robot can reaches the target point softly and avoid colliding with obstacles, as show in Fig. 8.

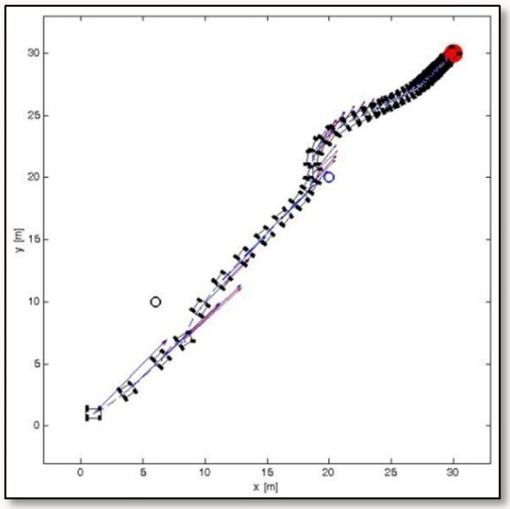

Fig. 8 Motion planning simulation with the virtual force acting on the car-like robot when there exists two fixed obstacles, located at (6m,10m) and (20m,20m).

EXPERIMENTAL RESULTS

Similar to simulation results, we demonstrate two experimental cases: 1) obstacle detection using laser range finder in the occupancy-grid map 2) real-time obstacle avoidance in outdoor environment.

In the first experiment, the global occupancy-grid map with a dimension of 90 x 90 cells is created in a soccer field at mechanical engineering department, Rajamangala University of Technology Thanyaburi, as shown in Fig. 9, so that the obstacle detection algorithm using the local-grid map with a dimension of 5 x 10 cells. Before testing with the car-like robot, we test a car movement simulation using GPS, IMU, and laser range finder, installed on a cart. The cart is moved along a straight path from North to East or along x-axis of the global-grid map when there exists a box obstacle in the front. This experimental result are shown in Fig. 10, it shows a correction identification of the box obstacle location at (x,y) = (8,1) m in the global-grid map. So, in next step, the obstacle location is used as an input for the local incremental planning.

Fig. 9 Obstacle detection test in outdoor environment.

In the second experiment, we use a similar outdoor setup, as in previous test, to perform a real-time computation of the virtual force for the car robot. Then, command inputs for front-wheel steering and rear-wheel drive are generated from motion-planning control law and send to low-level motor controls. Also, this test shows that our obstacle detection technique could be operated with the obstacle avoidance. Both target and initial robot position are located along x axis of the global-grid map. Moreover, a box obstacle is placed in between robot and target point.

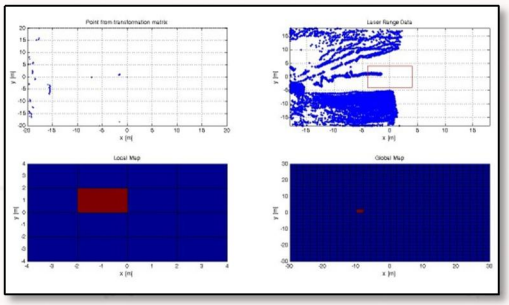

Fig. 10 Experimental results of the obstacle detection technique. Top Left: Initial point cloud from laser scanner at initial instance, Top Right: Overlay point cloud from laser scanner as the robot moves forward from right to left, Bottom Left: Local-grid map updated at the final time, Bottom Right: Global-grid map updated at the final time.

According to Fig. 11, initially the car-like robot moves directly toward the obstacle because the repulsive force is still smaller than the attractive force. Once the robot moves closer to the obstacle, its front wheels start

steering to the right to avoid the obstacle in front or the robot moves around the obstacle. Then, the robot is steered back toward the target. Thus, the result shows that the car-like robot can avoid the obstacle, move to the safe zone, and gradually approach the target at the same time.

Fig. 11 Experimental results of the obstacle avoidance for the car-like robot.

CONCLUSION

In this research, we proposed and developed the obstacle detection technique and a combination of the potential field with the local incremental planning methods for autonomous navigation with obstacle avoidance ability for the car-like robot. This car robot is equipped with GPS, IMU, and laser range finder sensors. First, the proposed method is tested by simulating the robot in MATLAB/Simulink: 1) the obstacle detection using the occupancy-grid map, consisting of the local- and global-grid map, which can identify the obstacle location according to the probability of occurrence and 2) auton-omous navigation for the car-like robot, which pursues the target and avoids fixed obstacles along its path.

For experimental results in outdoor environment, we demonstrated two tests in the soccer field parallel with those from simulation: 1) the obstacle detection algorithm using laser range data and GPS position can identify the obstacle location in the global occupancy-grid map, and 2) obstacle avoidance can be achieved when the car-like robot moves toward the target along the straight path.

ACKNOWLEDGMENT

Authors would like to thank Rajamangala University of Technology Thanyaburi and the National Research Council of Thailand (NRCT) for 2557 annual research grant to financially support this research.

REFERENCES

[1] Coenen, S.A.M. Motion planning for mobile robots-a guide. Master Thesis, Mechanical Engineering, Eindhoven University of Technology, Eindhoven (2012)

[2] Borenstein, J. and Koren, Y. Real-time obstacle avoidance for fast mobile robot in cluttered environments. Proc. of the IEEE International Conference on Robotics and Automation, Cincinnati, Ohio, pp. 572-577 (1990)

[3] Geyer, Mark J. and Johnson, Eric N. 3D obstacle avoidance in adversarial environments for unmanned aerial vehicles. Proc. of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado (2006)

[4] Borenstein, J. and Koren, Y. Real-time obstacle avoidance for fast mobile robot. IEEE Transactions on systems, man, and cybernetics. 19: 1179-1187 (1989)

[5] Roberts, J. and Corke, P. Obstacle detection for a mining vehicle using a 2-D laser. Proc. of the Australian Conference on Robotics and Automation. (2000)

[6] Sanjay Champalal Solanki. Development of sensor component for terrain evaluation and obstacle detection for an unmanned autonomous vehicle. PhD. Thesis, Computer science and Electrical Engineering, University of Florida, California (2007)

[7] Ruimin Zou. Free space detection based on occupancy gridmaps. Master Thesis, Darmstadt University of Technology, Darmstadt (2012)

[8] Bemporad, A. Luca, A.D. and Oriolo, G. Local incremental planning for a car-like robot navigation among obstacles. Proc. of the 1996 IEEE International Conference on Robotics and Automation, pp. 1205-1211 (1996)

[9] Ge, S.S. and Cui, Y.J. Dynamic motion planning for mobile robots using potential field method. Autonomous Robots. 13: 207-222 (2002)

[10] Wuthishuwon, C. Silawatchananai, C. and Parnichkun, M. Navigation of intelligent vehicle by using stand-alone GPS, compass and laser range finder. Proc. of the 2008 IEEE International Conference on Robotics and Biomimetics, Bangkok, Thailand, pp. 2121-2127 (2009)

[11] Farrell, Jay A. Aided navigation GPS with high rate sensors, McGrawHill (2008)

Koren, Y. and Borenstein, J. Potential field methods and their inherent limitations for mobile robot navigation. Proc. of the 1991 IEEE Internat-ional Conference on Robotics and Automation, pp. 1398-1404 (1991)