



HAL
open science

Reactive navigation of mobile robot using elliptic trajectories and effective on-line obstacle detection

José Miguel Vilca Ventura, Lounis Adouane, Y. Mezouar

► **To cite this version:**

José Miguel Vilca Ventura, Lounis Adouane, Y. Mezouar. Reactive navigation of mobile robot using elliptic trajectories and effective on-line obstacle detection. *Gyroscope and Navigation*, 2013, 4 (1), pp.14 - 25. 10.1134/S2075108713010094 . hal-01714845

HAL Id: hal-01714845

<https://hal.science/hal-01714845>

Submitted on 10 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Reactive Navigation of Mobile Robot using Elliptic Trajectories and Effective On-line Obstacle Detection

José Miguel Vilca Lounis Adouane Youcef Mezouar

*Institut Pascal, UBP – UMR CNRS 6602, Clermont-Ferrand, France
(e-mail: Jose_Miguel.Vilca_Ventura@univ-bpclermont.fr)*

Abstract: This paper deals with the problem of mobile robot navigation in cluttered environment. An adaptive elliptic trajectories are exploited for reactive obstacle avoidance using only position information and uncertain range data. The used obstacle avoidance strategy is based on elliptic limit-cycle principle where each obstacle is surrounded by an ellipse. The ellipse parameters are computed on-line using the sequence of uncertain range data. An on-line heuristic method combined with the Extended Kalman Filter (EKF) is used to compute the ellipse parameters. It is demonstrated that this process insures that all range data are surrounded by the computed ellipse. Moreover, this paper proposes a single control law to the multi-controller architecture where is embedded the reactive obstacle avoidance algorithm. This proposed control law is based on Kanayama control law and it is designed to improve the performance of the controllers. The stability of this control architecture is proved according to Lyapunov synthesis. Simulations and experiments in different environments were performed to demonstrate the efficiency and the reliability of the proposed on-line navigation in cluttered environment.

Keywords: Mobile robots navigation; Multi-controller architecture; Reactive control; Obstacle detection and avoidance; Telemetry; Parameter identification; Extend Kalman Filter.

1. INTRODUCTION

An important issue for successful mobile robot navigation is obstacle avoidance. This function permits to prevent robot collision and insure thus robot safety. One part of the literature considers that the robot control is entirely based on path planning methods while involving the total knowledge of its environment. Voronoi diagrams and visibility graphs [1] or artificial potential fields functions [2] are among these methods. All obstacles configurations are taken thus into account in the planning step. In these methods, it is possible also to deal with dynamic environment while regularly replanning the robot's path [3], [4]. However, planning and replanning require a significant computational time and complexity.

The other community is concentrated on reactive methods to deal with the obstacle avoidance, where only local sensors information is used rather than a prior knowledge of the environment [5], [6], [7]. In [8], the author proposes a real-time obstacle avoidance approach based on the principle of artificial potential fields. In this work, it is assumed that the robot actions are guided by the sum of attractive and repulsive fields. In [9] author extends Khatib's approach while proposing specific schema motors for mobile robots navigation. Another interesting approach, based on a reflex behavior reaction, uses the Deformable Virtual Zone (DVZ) concept, in which a robot movement depends on risk zone surrounding the robot [10]. If an obstacle is detected, it will deform the DVZ and the approach consists of minimizing this deformation by modifying the

control vector. This method deals with any obstacle shape, however, it suffers as schema motors from local minima problem. In general, reactive methods do not require high computational complexities since robot's actions must be given in real-time according to the perception [11].

Many other approaches can be founded in the literature, such as obstacles avoidance using vortex fields [12] and orbital trajectories [13]. The last approach is build on circular limit-cycle differential equations in [13], [14] or [15]. The circular limit-cycles are more stable than the vortex fields and always converge to periodic orbit. This work uses elliptical trajectories that was presented in [16]. Furthermore, this work proposed to use only one control law for the multi-controller architecture [17]. This control law is based on [18] and adapted to this obstacle avoidance algorithm. Therefore, more generic and efficient obstacle avoidance is performed and this even with different obstacle shapes, for instance long walls. In fact, an ellipse fits better this kind of obstacles than a circle (cf. Fig. 2).

Different approaches have been proposed in the literature to enclose the data with an ellipse. In [19], the author proposed a technique to obtain the smallest enclosing ellipse by a set of data using primitive operation with linear increasing time with regards to data dimension. In [20], the author presents a summary of the methods to fit a set of data with an ellipse. The presented methods are the least square fitting based on algebraic and Euclidean distance, Kalman filtering method and robust estimation. In this work, the used method is based on a simple and

efficient heuristic approach based on Euclidean distance estimation [21].

The Extended Kalman Filter has important applications in different fields [22], [23], [24], [25]. In [26], the author presents a method to fit ellipses using bias corrected Kalman filter. This method uses the perpendicular distance to the ellipse to find optimal fits to short sections of ellipse data. In this work, the EKF is used to enhance the ellipse parameters obtained from uncertain data of any obstacle shapes [27].

The rest of the paper is organized as follows: in the next section, the task of navigation using elliptic trajectories is presented. In section 3, the details of the control architecture are introduced. It presents the model of the considered robot and the implemented elementary controllers. Section 4 gives in details the proposed obstacle avoidance algorithm. Section 5 presents the method for enclosing the uncertain range data with an ellipse. Simulation results are given in section 6. Finally, conclusion and some future works are given in section 7.

2. NAVIGATION IN CLUTTERED ENVIRONMENT

Before to give details about the robot navigation, let us assume first that the obstacle O in the environment can be surrounded by elliptical box (cf. Fig. 1). The elliptical shape is represented by its Cartesian form:

$$\frac{(x-h)^2}{a^2} + \frac{(y-k)^2}{b^2} + c(x-h)(y-k) = 1 \quad (1)$$

where $(h, k) \in \mathbb{R}^2$ are the ellipse center coordinates and $a, b \in \mathbb{R}^+$ are the semi-axes ($a \geq b$). $c \in \mathbb{R}$, permits to give the ellipse orientation $\Omega_E = 0.5 \arctan(c/(b^2 - a^2))$ (cf. Fig. 1).

Ellipse boxes rather than circles are exploited to obtain a generic and flexible means to surround and fit accurately different kind of obstacle shapes [16]. Among examples of shapes which can be properly fitted by an ellipse instead of a circle is a wall (or in general, longitudinal shapes). Figure 2 shows this kind of configuration. In fact if we would like to surround this wall by a circle, it will have a large radius which will induce more robot path distance to avoid safely the obstacle [13] (cf. Fig. 2(a)). Figure 2(b) shows that the ellipse fits better the dimension of the obstacle. This figure shows also uncertain perceptions taken by range sensor on the left side of the wall.

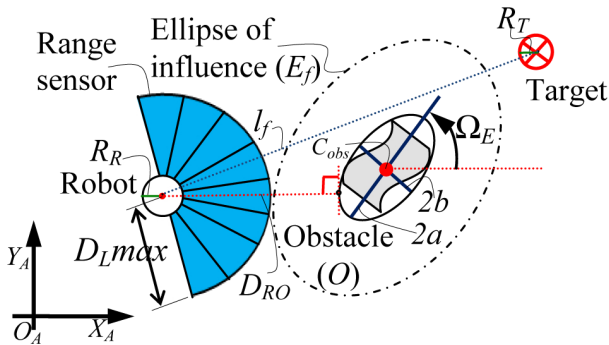


Fig. 1. The obstacle and robot representation.

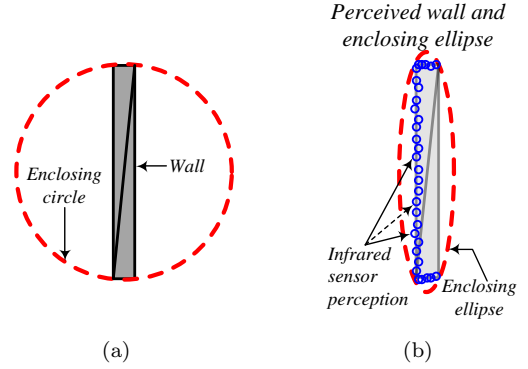


Fig. 2. Interpolated wall using a circle and ellipse shapes.

In section 5, it will be shown how the ellipse parameters can be efficiently computed from range data. Let us also represent the robot and the target by circles C_R and C_T of radius R_R and R_T respectively (cf. Fig. 1). One can define:

- (1) D_{RO} as the minimal distance between the robot and the obstacle “ O ”.
- (2) *Ellipse of influence* (E_f) as an ellipse that has the same center (h, k) and tilt angle Ω_E as the ellipse which surround the obstacle (1) while its major and minor semi-axes, a_{lc} and b_{lc} , are defined as follows

$$\begin{cases} a_{lc} = a + R_R + \text{Margin} \\ b_{lc} = b + R_R + \text{Margin} \end{cases}$$

where *Margin* represents safety tolerances encapsulating: perception uncertainties, control reliability and accuracy. This definition is similar to the configuration space for global path planing, nevertheless, in this paper it is used to focus on the reactive planning approach, i.e, the ellipse of influence is obtained by the robot in real time using the detected obstacle.

- (3) l_f as the line passing through the center of C_R and C_T . As we will see in the sequel, our method only needs to know if it exists intersection points between l_f and E_f (cf. Fig. 1).

The objective of the navigation task in cluttered environment is to lead a mobile robot towards a specific target in an unstructured environment. This task must be achieved while avoiding static and dynamic obstacles O which can have different shapes.

3. CONTROL ARCHITECTURE

The control structure is based on [17] (cf. Fig. 3). It aims to manage the interactions between elementary controllers while guaranteeing the stability of the overall control as proposed in [7]. Its objective is also to insure safe, smooth and fast robot navigation. The specific blocks that compose the global controller are detailed below.

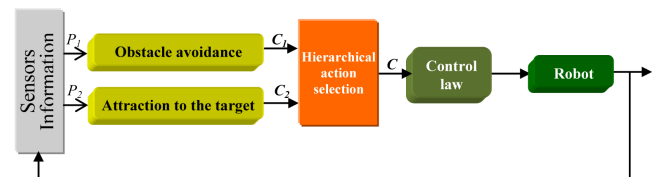


Fig. 3. Control architecture for mobile robot navigation.

This control architecture uses a hierarchical action selection mechanism to manage the switches between the controllers, according to environment perception. The mechanism activates the obstacle avoidance controller as soon as it exists at least one obstacle which can obstruct the future robot movement toward its target (cf. Algorithm 1) [15]. This allows to anticipate the activation of obstacle avoidance controller unlike what is proposed in [28], [29], which wait until the robot is in the immediate vicinity of the obstacle (i.e. $D_{RO} \leq R$ “a certain radius value”). Algorithm 1 allows thus to decrease the time to reach the target, especially in very cluttered environments.

if *It exists at least one constrained obstacle*
{i.e., it exists at least one intersect point between the line “ l_f ” and the ellipse of influence (cf. Fig. 1)} **then**
 | Activate *Obstacle avoidance* controller;
else
 | Activate *Attraction to the target* controller;
end

Algorithm 1: Hierarchical action selection

3.1 Elementary controllers

Each controller composing the control architecture (cf. Fig. 3) is characterized by a stable nominal law. In this work, only one control law synthesized according to Lyapunov theorem is used to improve the performance of the two different controllers (cf. Subsection 3.2). Before describing each elementary controller, let us briefly recall the kinematic model of an unicycle robot (cf. Fig. 4)

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (2)$$

where x , y , θ are configuration state of the unicycle at the point O_m , v and ω are respectively, the linear and the angular velocities of the robot at the point O_m .

Attraction to the target controller This controller guides the robot toward the target which is represented by a circle C_T of center (x_T, y_T) and radius R_T (cf. Fig.

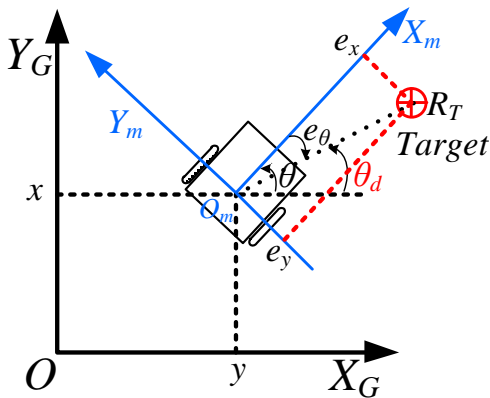


Fig. 4. Robot and target configuration in a Cartesian reference frame.

4). It is based on the configuration of the robot to the target, represented by e_x , e_y and e_θ in Fig. 4. Since we consider a circular target with radius R_T , therefore, to guarantee that the center of robot axis reaches the target with asymptotical convergence, d must be smaller than R_T , where $d = \sqrt{e_x^2 + e_y^2}$ (cf. Fig. 4).

The position and orientation errors w.r.t local reference frame $X_m - Y_m$ are:

$$\begin{cases} e_x = \cos(\theta)(x_T - x) + \sin(\theta)(y_T - y) \\ e_y = -\sin(\theta)(x_T - x) + \cos(\theta)(y_T - y) \\ e_\theta = \theta_d - \theta \end{cases} \quad (3)$$

where θ_d is the orientation of the line passing through the robot and the target. Further, $\dot{\theta}_d = \omega_r$ is given by:

$$\omega_r = v \sin(e_\theta)/d \quad (4)$$

where v is the linear velocity of the robot [17].

Obstacle avoidance controller To perform the obstacle avoidance behavior, the robot needs to follow accurately limit-cycle trajectories as detailed in [13], [15]. In these works, the authors use a circular limit-cycle characterized by a circle of influence of radius R_I . In [16], it is proposed to extend this methodology for more flexible limit-cycle shape (an ellipse). The main ideas of this controller are detailed below (cf. Fig. 6).

The differential equations giving elliptic limit-cycles are:

$$\dot{x}_s = m y_s + x_s(1 - x_s^2/a_{lc}^2 - y_s^2/b_{lc}^2 - c x_s y_s) \quad (5)$$

$$\dot{y}_s = -m x_s + y_s(1 - x_s^2/a_{lc}^2 - y_s^2/b_{lc}^2 - c x_s y_s) \quad (6)$$

with $m = \pm 1$ according to the direction of avoidance (clockwise or counter-clockwise, cf. Fig. 5). (x_s, y_s) corresponds to the position of the robot according to the center of the ellipse; a_{lc} and b_{lc} characterize respectively the major and minor elliptic semi-axes (cf. Fig. 1); c if not equal to 0 gives the Ω_E ellipse angle.

In this controller, the desired position is considered as the same robot position, i.e. $e_x = 0$ and $e_y = 0$ in (3) (cf. Fig. 6). The desired robot orientation is given by the differential equation of the limit-cycle (5) and (6) as (cf. Fig. 6):

$$\theta_d = \arctg\left(\frac{\dot{y}_s}{\dot{x}_s}\right) \quad (7)$$

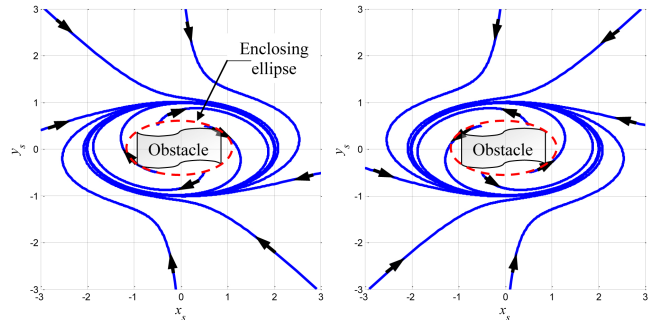


Fig. 5. Clockwise ($m = 1$) and counter-clockwise ($m = -1$) shape for the used elliptic limit-cycles.

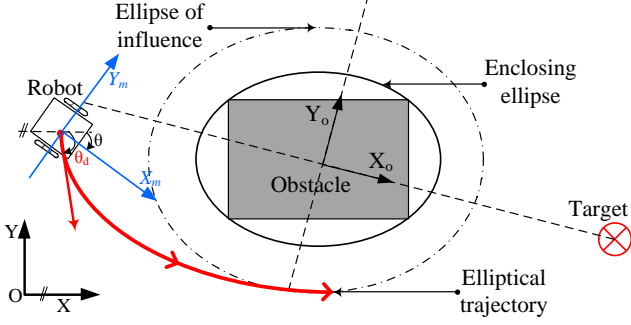


Fig. 6. Obstacle avoidance controller.

3.2 Control law

The proposed control law is based on [18] and adapted to the multi-controller architecture. In [30], the Kanayama control law is used to track a reference trajectory. In this work, the proposed control law is adapted to track a reference trajectory (*Obstacle avoidance* controller) and to reach the target (*Attraction to the target* controller). It provides the desired velocities for these controllers (cf. Sub-subsections 3.1.1 and 3.1.2). Further, this control law allows to improve the performance of these controllers (cf. Section 6).

Consider that the error w.r.t local frame of the robot (e_x, e_y, e_θ) between the desired position (x_d, y_d, θ_d) and the actual position of the robot (x, y, θ) has the following form (cf. Fig. 4 and 6):

$$\begin{cases} e_x = \cos(\theta)(x_d - x) + \sin(\theta)(y_d - y) \\ e_y = -\sin(\theta)(x_d - x) + \cos(\theta)(y_d - y) \\ e_\theta = \theta_d - \theta \end{cases} \quad (8)$$

The derivatives of the errors can be computed using (2) and (8) (cf. [18]):

$$\begin{cases} \dot{e}_x = -v + e_y \omega + v_r \cos(e_\theta) \\ \dot{e}_y = -e_x \omega + v_r \sin(e_\theta) \\ \dot{e}_\theta = -\omega + \omega_r \end{cases} \quad (9)$$

Hence, the desired linear (v) and angular (ω) velocities of the robot that make the errors converge to zero are given by:

$$v = v_r \cos(e_\theta) + K_x e_x \quad (10)$$

$$\omega = \omega_r + K_y v_r e_y + K_\theta e^{(e_y/R_R)^2} \sin(e_\theta) \quad (11)$$

where R_R is the radius of the robot. This work proposes a modification in the Kanayama control law, the term $e^{(e_y/R_R)^2}$ is added in the angular velocity (11). The objective of this term is to improve the convergence of the error for the case where $v_r = 0$ (*Attraction to the target* controller) (cf. Section 6). The form of the term $e^{(e_y/R_R)^2}$ is chosen to improve the convergence when $e_y \geq R_R$, and when $e_y \approx 0$, then the angular velocity depends only on e_θ . Let us consider the following Lyapunov function V_0 [18]:

$$V_0 = \frac{1}{2} (e_x^2 + e_y^2) + \frac{1 - \cos(e_\theta)}{K_y} \quad (12)$$

Therefore, to guarantee the stability of the controller, \dot{V}_0 must be negative definite. By computing \dot{V}_0 using (9), (10) and (11), we obtain:

$$\begin{aligned} \dot{V}_0 &= e_x \dot{e}_x + e_y \dot{e}_y + \frac{\sin(e_\theta) \dot{e}_\theta}{K_y} \\ &= e_x (-K_x e_x + e_y \omega) + e_y (-e_x \omega + v_r \sin(e_\theta)) \\ &\quad + \frac{\sin(e_\theta)}{K_y} (-K_y v_r e_y - K_\theta e^{(e_y/R_R)^2} \sin(e_\theta)) \\ &= -K_x e_x^2 - \frac{K_\theta e^{(e_y/R_R)^2} \sin^2(e_\theta)}{K_y} \leq 0 \end{aligned} \quad (13)$$

where K_x, K_y and K_θ are positive constants to be defined by the designer. $v_r = 0$ when the *attraction to the target* controller is activated, and if the *obstacle avoidance* is activated, v_r and ω_r are computed as:

$$v_r = \sqrt{(\dot{x}_d)^2 + (\dot{y}_d)^2} \quad \text{and} \quad \omega_r = \dot{\theta}_d. \quad (14)$$

It is interesting to notice that only one control law is applied to the robot even if the control architecture contains two different controllers. Only the set points change according to the applied controller such as in [17].

4. REACTIVE OBSTACLE AVOIDANCE ALGORITHM

In what follows, the overall methodology to achieve the proposed obstacle avoidance algorithm will be given [16]. The algorithm is developed according to stimuli-response principle. To implement this kind of behavior it is important to:

- detect the obstacle to avoid (cf. Section 2),
- give the direction of the avoidance (clockwise or counter-clockwise),
- define an escape criterion which defines if the obstacle is completely avoided or not yet.

All these different steps must be followed and applied while guaranteeing that: the robot trajectory is safe, smooth and avoid undesirable situations as deadlocks or local minima; and that the stability of the applied control law is guaranteed (cf. Subsection 3.1). In this paper the obstacle observation is obtained in real time, the robot does not have global information about the hinder obstacle nor the ellipse that encloses it. The robot discovers thus at each sample time the shape of the obstacle and increase progressively the knowledge of the enclosing ellipse to obtain a smooth elliptic trajectory. The global information of the obstacle is not related to the used reactive navigation. The necessary steps to carry out the obstacle avoidance algorithm (2) are given below:

- (1) For each sample time, obtain the distance D_{ROi} for each potentially disturbing obstacle “ i ” (cf. Fig. 1).
- (2) Among the set of disturbing obstacles (which can constrain the robot to reach the target), choose the closer to the robot (the smallest D_{ROi} (cf. Fig. 1)). This specific obstacle has the following features: (x_{obst}, y_{obst}) center position and $2a$ as major axis and $2b$ as minor axis.
- (3) After the determination of the closest constrained obstacle, we need to obtain four specific areas (cf. Fig. 7) which give the robot behavior: clockwise or counter-clockwise obstacle avoidance ; repulsive or attractive phase (cf. Algorithm 2). To distinguish between these 4 areas we need to:

- define a specific reference frame which has the following features (cf. Fig. 7):
 - the X_O axis connects the center of the obstacle (x_{obst}, y_{obst}) to the center of the target. This axis is oriented towards the target,
 - the Y_O axis is perpendicular to the X_O axis and it is oriented while following trigonometric convention.
- apply the reference frame change of the position robot coordinate $(x, y)_A$ (given in absolute reference frame) towards the reference frame linked to the obstacle $(x, y)_O$. The transformation is achieved while using the following homogeneous transformation:

$$\begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_O = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 & x_{obst} \\ \sin \alpha & \cos \alpha & 0 & y_{obst} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{pmatrix} x \\ y \\ 0 \\ 1 \end{pmatrix}_A \quad (15)$$

Once all necessary perceptions are obtained, one can apply the reactive obstacle avoidance strategy given by Algorithm 2. To obtain the robot set points, it is necessary to obtain the value of a_{lc} and b_{lc} (cf. Section 2) of the orbital ellipse and the direction “clockwise or counter-clockwise” of the limit-cycle to follow. The position (x_O, y_O) gives the configuration (x, y) of the robot according to obstacle reference frame. The definition of this specific reference frame provides an accurate means to the robot to know what it must do. In fact, the sign of x_O gives the kind of behavior which must be taken by the robot (attraction or repulsion).

In repulsive phase, the limit-cycle takes an increase value of a'_{lc} and b'_{lc} values to guarantee the trajectory smoothness. The sign of y_O gives the right direction to avoid the obstacle. In fact, if $y_O \geq 0$ then apply clockwise limit-cycle direction else apply counter-clockwise direction. This choice permits to optimize the length of robot trajectory to avoid obstacles. Nevertheless, this direction is forced to the direction taken just before if the obstacle avoidance controller was already active at $(t - \delta T)$ instant and this to avoid local minima and dead-end [15].

In algorithm 2, some conflicting situations which are due to local minima or dead ends have to be managed to improve the performance. These specific local and reactive rules are detailed in [15].

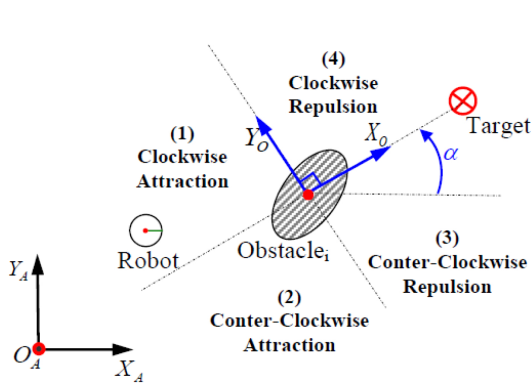


Fig. 7. The 4 specific areas surrounding the obstacle to avoid [16].

Input: All the features of the closest constrained obstacle.

Output: Features of the limit-cycle trajectory to follow.

```
// I) Obtaining the values of  $a'_{lc}$  and  $b'_{lc}$  of the
limit-cycle to follow
1 if  $x_O \leq 0$  then
2    $\begin{cases} a'_{lc} = a_{lc} - \xi \\ b'_{lc} = b_{lc} - \xi \end{cases}$  (Attractive phase)
3   {with  $\xi$  a small constant value as  $\xi \ll \text{Margin}$  which
   guarantees that the robot do not navigate very
   closely to the obstacle (cf. Section 2).}
4 else
5   {Escape criterion: go out of the obstacle ellipse of
   influence with smooth way}
6    $\begin{cases} a'_{lc} = a_{lc} + \xi \\ b'_{lc} = b_{lc} + \xi \end{cases}$  (Repulsive phase)
7 end

// II) Obtaining the limit-cycle direction
8 if obstacle avoidance controller was active at  $(t - \delta T)$ 
instant then
9   Apply the same direction already used, equation (5)
or (6) is thus applied.
10  {This will permit to avoid oscillations and several
conflicting situations [15]}
11 else
12  {The limit-cycle set-point is given by:}
 $\dot{x} = \text{sign}(y_O)y + x(1 - x^2/a_{lc}'^2 - y^2/b_{lc}'^2 - cxy)$ 
 $\dot{y} = -\text{sign}(y_O)x + y(1 - x^2/a_{lc}'^2 - y^2/b_{lc}'^2 - cxy)$ 
13 end
```

Algorithm 2: Obstacle avoidance algorithm [16].

5. ENCLOSING UNCERTAIN RANGE DATA WITH AN ELLIPSE

During the robot movement, it is important to detect on-line and to avoid the hinder obstacle. At this aim, the observed noisy range data are surrounded with the closest ellipse to apply elliptic limit-cycle approach.

For this purpose, let us consider a set of n points in \mathbb{R}^2 with coordinates $\mathbf{P}_i(x_i, y_i)$ (cf. Fig. 8). These points are computed from the data range of the robot, and the outliers are erased while using the Mahalanobis distance [31]. In this section, it will be shown how to compute the

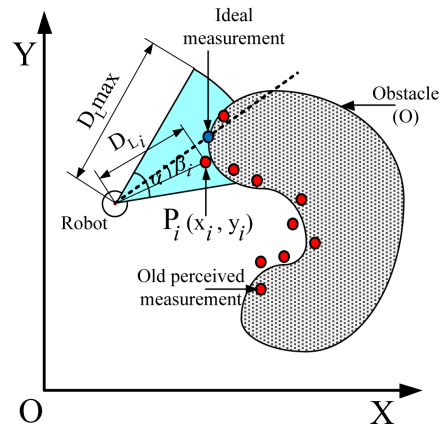


Fig. 8. Range sensor model and data set of n points.

ellipse that encloses all points. An important condition in this work is that the method need to start at least with three different points.

This paper considers that the robot detects one obstacle at a time. The segmentation method of the set of points will be used in future works to the detection of more than one obstacle at a time and for big obstacles.

Before describing the proposed method to obtain the enclosing ellipse, let us present the model of the perceived data from range sensor in the following subsection.

5.1 Range sensor model

The position of the obstacle with respect to the range sensor in \mathbb{R}^2 can be denoted by the polar coordinates (D_{Li}, β_i) , where D_{Li} is the distance between the center of the robot and the impact point of the sensor and β_i is the orientation with respect to the mobile reference frame (cf. Fig 8).

The sensor specifications and the real behavior of the sensor have significant differences [32], [33]. In this work, we focus on the accuracy of the range sensor, because we have observed that short range readings are more accurate than the long range ones. In [33] it is observed how the mean and the standard deviation of the errors between the real and the measured range tend to increase with distance.

The reading range data provided by the range sensor at each time step is modeled by the Normal distribution $D_{Li}^t = N(\hat{D}_{Li}^t, P_{Li}^t)$, where $\hat{D}_{Li}^t = [D_{Li}, 0]^T$ is the mean vector, D_L is the range measure of the sensor and $P_{Li}^t = \text{diag}([\sigma_{D_{Li}}^2, \sigma_{\beta_i}^2])$ is the covariance that is defined as the model of the range and angular uncertainties. The angular uncertainties are related to the sonar opening β_i and the range uncertainty is given according to the accuracy of the range sensor (cf. Fig. 8) [33]. The representation of the range data in Cartesian frame is given by:

$$\mathbf{z}_i = \mathbf{z}'_i + \mathbf{v}_i \quad (16)$$

where $\mathbf{z}_i = [x_i, y_i]^T$ is the point computed using the noisy range data, \mathbf{z}'_i is the point computed using the range data without noise, the Gaussian noise v_i has $E[\mathbf{v}_i] = 0$ and $E[\mathbf{v}_i \mathbf{v}_i^T] = \mathbf{R}_{v_i}$. The covariance \mathbf{R}_{v_i} is given by:

$$\mathbf{R}_{v_i} = \begin{bmatrix} \cos(\beta_i) & -D_{Li} \sin(\beta_i) \\ \sin(\beta_i) & D_{Li} \cos(\beta_i) \end{bmatrix} \begin{bmatrix} \sigma_{D_{Li}}^2 & 0 \\ 0 & \sigma_{\beta_i}^2 \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \end{bmatrix}^T \quad (17)$$

The following sub-sections will present the proposed methods to enclose the data with an ellipse. Subsection 5.2 permits to address the problem of enclosing ellipse and subsection 5.3 is the extension of the proposed method to deal with uncertainty data and to enhance the identification of the ellipse parameters to get round the obstacle.

5.2 Heuristic approach

In [21], a review of different methods to enclosing an ellipse is given, an heuristic method was proposed and compared to the existing approaches (based on least square or covariance). This method permits to enclose all range

data and to obtain a smooth changes of the ellipse parameters, different from least square and covariance method that could have point outside of the obtained ellipse and abrupt change of parameters (cf. Fig. 9(a)). In this work, the heuristic approach is given in more details with the mathematical proof that the obtained ellipse parameters encloses all points without regard of the obstacle shape. This approach uses the distance between the points to obtain one of the axes.

Lemma 1. Consider a set of n points (\mathbf{P}_N) in \mathbb{R}^2 with coordinates $\mathbf{P}_i(x_i, y_i)$ with $i = 1, \dots, n$. The parameters of the ellipse that enclose all points are computed as follow:

- (1) Compute the distance between all the points $d_{ij} = \|p_i - p_j\|$ with $i, j = 1, \dots, n$; and select the maximum distance d_{max} . This d_{max} is not decreasing if more data points are added.
- (2) The ellipse center C_O is the middle point between the points with maximum distances and the first semi-axis is $a_1 = d_{max}/2$ (cf. Fig 9(b)).
- (3) Transform the n points to new coordinates system $X' - Y'$ using (18) to obtain the second ellipse semi-axis a_2 .

$$\mathbf{P}'_i = \begin{bmatrix} \cos(\Omega) & \sin(\Omega) \\ -\sin(\Omega) & \cos(\Omega) \end{bmatrix} (\mathbf{P}_i - \mathbf{C}_O) \quad (18)$$

where Ω is the orientation of the line between the two points that have the maximum distance. $\mathbf{P}'_i(x'_i, y'_i)$ are the coordinates in the new system, $\mathbf{P}_i(x_i, y_i)$ are the coordinates in the initial system and \mathbf{C}_O are the coordinates of the ellipse center in the initial system.

- (4) Compute the distance of \mathbf{P}'_i to the origin O' , if the value of $|y'_i|$ of the points is greater than a threshold $\epsilon > 0$. This threshold is used to eliminate the points that are colinear with the two points that have the maximum distance (first axis) and the points in the perpendicular line to first axis (which could produce a large axis).
- (5) Choose $a_2 = \max\{b_i\}$, where b_i is the computed semi-axis using \mathbf{P}'_i in (1).
- (6) Finally, the semi-axes of the ellipse (1), are obtained:

$$\begin{aligned} a &= \max\{a_1, a_2\} \\ b &= \min\{a_1, a_2\} \end{aligned} \quad (19)$$

and, the orientation of the ellipse is $\Omega_E = \Omega + \Pi/2$ if a_2 is the major axis, otherwise $\Omega_E = \Omega$.

Proof 1. To proof that this heuristic method enclose all the points, we assume that a data point $p_i \in \mathbf{P}_N$, \mathbf{P}_N is the set of n points used to compute the parameters of the ellipse. First, if $p_i \notin \text{Ellipse}$ then $d_{ij} = \|p_i - p_j\|$ with $p_j \in \mathbf{P}_N - \{p_i\}$ could be $d_{ij} \geq d_{max}$, however, the first axis of ellipse is the maximum distance between all points, then, d_{ij} is new first axis and $p_i \in \text{Ellipse}$. Therefore, if $p_i \notin \text{Ellipse}$ then $d_{ij} \leq d_{max}$, now, we know that the second axis satisfy $b_i^2 \leq b^2$, using (18) and (1), we obtain:

$$\begin{aligned} b_i^2 &\leq b^2 \\ \frac{y_i'^2}{1 - x_i'^2/a^2} &\leq b^2 \\ x_i'^2/a^2 + y_i'^2/b^2 &\leq 1 \end{aligned} \quad (20)$$

Therefore, p_i satisfy the ellipse equation ($p_i \in \text{Ellipse}$) with $i, j = 1, \dots, n$.

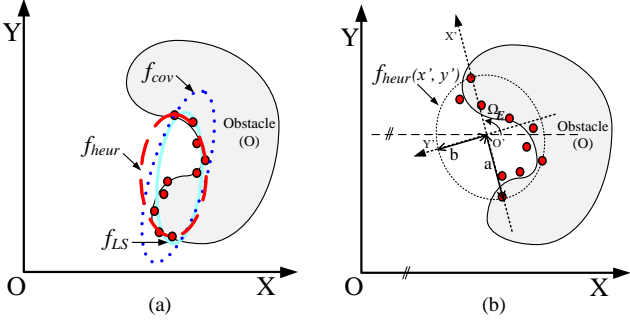


Fig. 9. Obtained ellipses using Least Square (f_{LS}), Covariance (f_{cov}) and Heuristic (f_{heur}) approaches.

The heuristic approach is an efficient method to enclose data with an ellipse. It produces a smooth change of the ellipse parameters and the average time of computation is $O(n \log(n))$, where n is the size of the data, however, this method does not consider neither uncertain data nor the sequentiality of obtained data which characterize the real experiments. The following subsection deal with this issue.

5.3 Optimal parameters identification using EKF

Kalman Filter is used in many fields as a general method for integrating noisy measurements [24] and [25]. In this paper, the Kalman Filter is used to improve the obtained ellipse parameters. The general conic equation is given by:

$$f(x, y) = Ax^2 + 2Bxy + Cy^2 + 2Dx + 2Ey + F = 0 \quad (21)$$

According to the real constants A, B, C, D, E and F , we obtain the analytic equation of the different kind of conics (parabola, ellipse and hyperbole). An ellipse is defined if the conic parameters (21) satisfy the following condition $B^2 - AC < 0$.

The problem is to fit a conic section (21) with a set of n points $\{p_i\} = \{(x_i, y_i)\} \mid i = 1, \dots, n$. This set of points are selected using (21) with the ellipse parameters from the heuristic method. These points satisfy the following condition $|f(x_i, y_i)| < \delta$, and $\delta \in \mathbb{R}^+$ is close to zero, i.e., these points are close to the boundary of the obtained ellipse, other points are not considered for this method. As the data are noisy, it is unlikely to find a set of parameters (A, B, C, D, E, F) (except for the trivial solution $A = B = C = D = E = F = 0$) such that $f(x_i, y_i) = 0$. This method is applied to the conic fitting.

The state vector is defined by the conic parameters as $\mathbf{x} = [A, B, C, D, E, F]^T$ and the measurement vector by the point as $\mathbf{z}_i = [x_i, y_i]^T$, a linear dynamic system (in discrete-time form) can be described by

$$\mathbf{x}_{i+1} = \mathbf{F}_i \mathbf{x}_i + \mathbf{w}_i \quad (22)$$

$$\mathbf{z}_i = \mathbf{H}_i \mathbf{x}_i + \mathbf{v}_i \quad (23)$$

Where $i = 0, 1, \dots, n$. The matrix state \mathbf{F}_i is the identity matrix of order 6 (\mathbf{I}_6), \mathbf{w}_i is the vector of random disturbance of the state and is usually modeled as white noise:

$$E[\mathbf{w}_i] = 0, \quad E[\mathbf{w}_i \mathbf{w}_i^T] = \mathbf{Q}_i$$

The measurement equation (23) is nonlinear in function of the ideal measurement \mathbf{z}'_i (is a polynomial equation that

does not satisfy the superposition principle [34]) and it is described by the observation function:

$$f_i(\mathbf{z}'_i, \mathbf{x}_i) = x_i'^2 A + 2x_i' y_i' B + y_i'^2 C + 2x_i' D + 2y_i' E + F \quad (24)$$

The real measurement \mathbf{z}_i is assumed to be corrupted by additive noise \mathbf{v}_i . The model of noise \mathbf{v}_i is described in subsection 5.1. We expand $f_i(\mathbf{z}'_i, \mathbf{x}_i)$ into a Taylor series about $(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})$:

$$\begin{aligned} f_i(\mathbf{z}'_i, \mathbf{x}_i) &= f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1}) + \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{z}'_i} (\mathbf{z}'_i - \mathbf{z}_i) \\ &+ \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{x}_i} (\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1}) + O((\mathbf{z}'_i - \mathbf{z}_i)^2) \\ &+ O((\mathbf{x}_i - \hat{\mathbf{x}}_{i|i-1})^2) \end{aligned} \quad (25)$$

By ignoring the second order terms, we get a linearized measurement equation:

$$\mathbf{y}_i = \mathbf{M}_i \mathbf{x}_i + \xi_i \quad (26)$$

where \mathbf{y}_i is the new measurement vector, ξ_i is the noise vector of the new measurement, and \mathbf{M}_i is the linearized transformation matrix. They are given by

$$\begin{aligned} \mathbf{M}_i &= \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{x}_i} \\ \mathbf{y}_i &= -f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1}) + \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{x}_i} \hat{\mathbf{x}}_{i|i-1} \\ \xi_i &= \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{z}'_i} (\mathbf{z}'_i - \mathbf{z}_i) \end{aligned}$$

Clearly, we have $E[\xi_i] = 0$, and $E[\xi_i \xi_i^T] = R_{\xi_i}$. We consider then there is no correlation between the noise process of the system and the observation. The derivative of $f_i(\mathbf{z}_i, \mathbf{x})$ with respect to \mathbf{x} and with respect to \mathbf{z}_i , are given by

$$\frac{\partial f_i(\mathbf{z}_i, \mathbf{x})}{\partial \mathbf{x}} = [x_i^2, 2x_i y_i, y_i^2, 2x_i, 2y_i, 1] \quad (27)$$

$$\frac{\partial f_i(\mathbf{z}_i, \mathbf{x})}{\partial \mathbf{z}_i} = 2[x_i A + y_i B + D, y_i C + x_i B + E] \quad (28)$$

The Extended Kalman Filter (EKF) is then used while knowing that output equation (23) is obtained from the nonlinear measurement equation (24). EKF is described by the well known following steps:

- Initialization: $\Pi_{0|0} = \Pi_0, \hat{\mathbf{x}}_{0|0} = E[\mathbf{x}_0]$
- Prediction of states: $\hat{\mathbf{x}}_{i|i-1} = \mathbf{F}_{i-1} \hat{\mathbf{x}}_{i-1}$
- Prediction of the state covariance matrix:

$$\Pi_{i|i-1} = \mathbf{F}_{i-1} \Pi_{i-1} \mathbf{F}_{i-1}^T + \mathbf{Q}_{i-1}$$

- Kalman gain matrix:

$$\mathbf{K}_i = \Pi_{i|i-1} \mathbf{H}_{i-1}^T \left(\mathbf{H}_{i-1} \Pi_{i|i-1} \mathbf{H}_{i-1}^T + \mathbf{R}_v \right)^{-1}$$

- Update of the state estimation:

$$\hat{\mathbf{x}}_i = \hat{\mathbf{x}}_{i|i-1} + \mathbf{K}_i (\mathbf{z}_i - \mathbf{H}_i \hat{\mathbf{x}}_{i|i-1})$$

- Update of the covariance matrix of states:

$$\Pi_i = (\mathbf{I} - \mathbf{K}_i \mathbf{H}_i) \Pi_{i|i-1}$$

Note that the Kalman filtering technique is usually applied to a temporal sequence. Here, it is applied to a spatio-temporal sequence. This spatio-temporal sequence is composed of data from each sensor at each time. Due

to its recursive nature, it is more suitable to problems where the measurements are available in a serial manner. Otherwise, if all measurements are available or could be made available (with no serious overhead) at the same time, it is advantageous to apply the Kalman Filter in a single joint evaluation (all the spatial sequence at the same time). Indeed, the Kalman filtering technique is equivalent to the least-squares technique only if the system is linear. For nonlinear problems, the EKF will yield different results depending on the order of processing the measurements one after the other, and may run the risk of being trapped into a local minimum [20].

6. SIMULATIONS RESULTS

To demonstrate the efficiency of the proposed control law in the multi-controller architecture and the on-line obstacle detection method to enclose the obstacle with an ellipse and avoid it, two statistical survey was made. In this paper, a mobile robot with a radius of $R_R = 0.065\text{ m}$ and six infrared range sensors with the maximum detected range equal to $D_{Lmax} = 0.30\text{ m}$ is considered (cf. Fig. 8). These sensors are in the front of the robot, with 30° between each pairs of sensor (cf. Fig. 10). The accuracy of the used sensors based on the datasheet is around 10% of D_{Lmax} . In the simulation, we consider uncertainty range with maximum value of 20% of D_{Lmax} ensuring thus to take the worst range value.

For the control law (10) and (11), the gains are defined as $K_x = 0.8$, $K_y = 5$ and $K_\theta = 3$. The values of these gains were chosen heuristically to obtain a smooth trajectory, fast response and velocity but within limit velocities of the mobile robot, which are $v_{max} = 0.4\text{ m/s}$ and $w_{max} = 3\text{ rad/s}$. The sample time is 0.01 s . For each simulation, the robot starts at the same configuration and reach the same final configuration. The heuristic method does not start until to have enough range data ($n_{data} \geq 3$).

The first survey is used to compare the performance of the reactive obstacle avoidance between proposed control law and Kanayama control law in the multi-controller architecture presented above to obtain safe and smooth navigation (cf. Subsection 3.1).

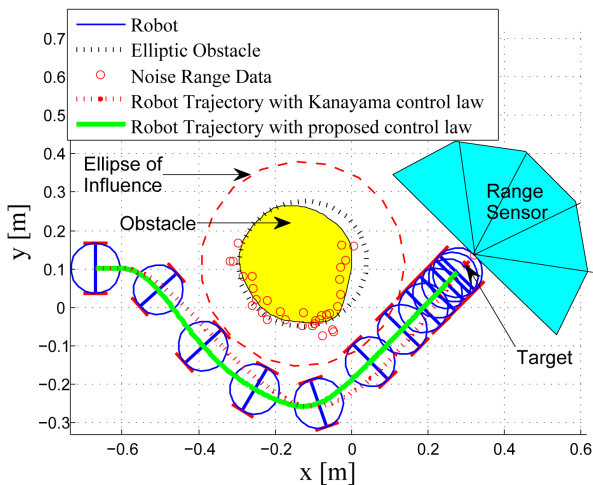


Fig. 10. Robot trajectory using the two different control laws for the multi-controller architecture.

Figure 10 shows the trajectory of the robot for two different control laws in a specific environment where the parameters of the ellipse that enclose the obstacle are known. It is observed that the trajectory with the Kanayama control (red dotted line) is closer to the obstacle than the trajectory with the proposed control law (green continuous line) in the *obstacle avoidance* controller. Furthermore, in the *attraction to the target* controller, the trajectory with the proposed control law is shorter than the other trajectory. This figure shows that the proposed control law improve the trajectory safety to reach the target.

Figure 11 shows the position and orientation errors for the two different control laws. It is observed that the proposed control law improve the converge in the *attraction to the target* controller. Further, the convergence of e_x is close between the two different control laws, that is because the modification to obtain the proposed control law is only in the angular velocity (more related to e_y and e_θ).

Otherwise, Fig. 12 shows the progress value of Lyapunov function V_0 (12) attributed to the control law (cf. Fig. 3) when the navigation is performed. This function decreases asymptotically to the equilibrium point. It is observed that Lyapunov function of the proposed control law has less value in the switch between controller and faster convergence then that the Kanayama control law.

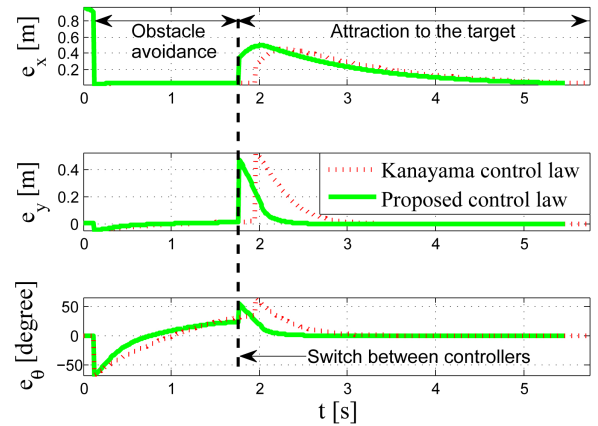


Fig. 11. Position and orientation errors.

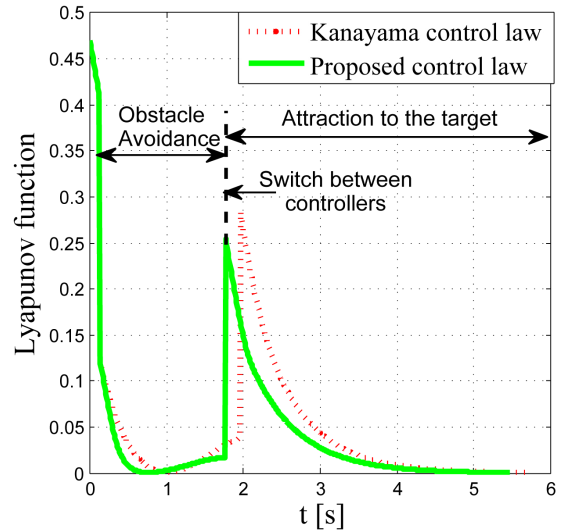


Fig. 12. Evolution of Lyapunov functions for the two different control laws during the robot navigation.

The second survey is used to make a focus around the proposed heuristic method and Kalman filter which gives satisfactory results and this while making an on-line navigation in cluttered environment (cf. Fig. 13).

Figure 13 shows the trajectory of the robot in the environment with three obstacles. Moreover, the red points represent the range data from the sensor along all the trajectory. The range data buffer used to compute the ellipse parameters is deleted for each new discovered obstacle. This figure shows that the obtained ellipse using the heuristic method enclose all points such as was demonstrated above. Further, it is observed that the robot avoid the obstacles with a smooth trajectory. This trajectory was obtained while using the on-line obstacle avoidance algorithm [16] with the proposed control law which takes its parameters (elliptical limit-cycle to follows) from the combination of the proposed heuristic approach and EKF.

Figure 14 shows the minimum distance between the effective elliptical obstacles (obtained while knowing all the range data, without noise, which surround the obstacle) and the position of the robot along of the elliptical trajectory using only the heuristic method (red dotted line), and the combination of heuristic method and EKF (green continuous line). This figure shows that the robot does never

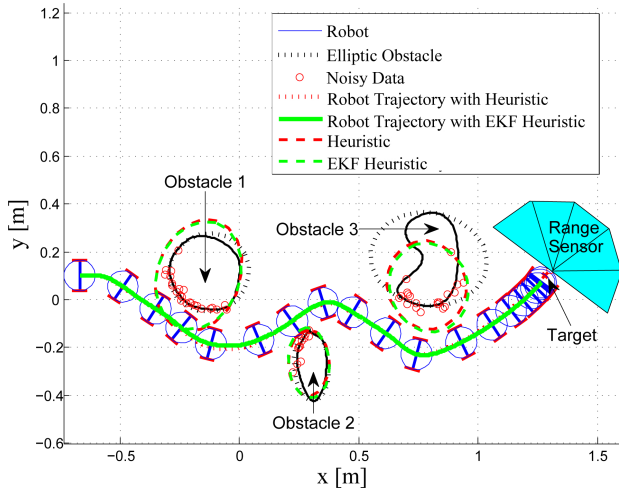


Fig. 13. Robot trajectory using the heuristic and EKF approach to enclosing the obstacle.

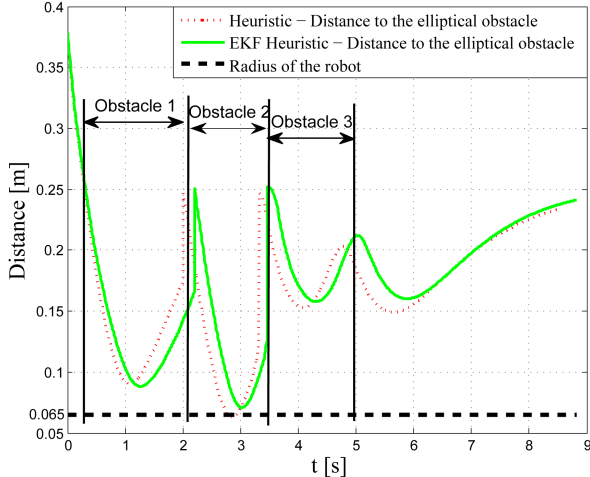


Fig. 14. Distance from the robot to the elliptical obstacles.

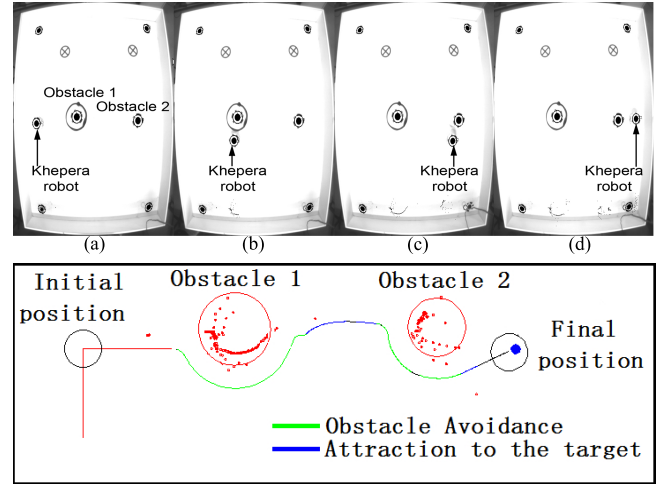


Fig. 15. Top view of the robot trajectory in the platform and observed uncertain range data from the robot.

collide with any obstacles when the proposed control law is used, therefore, the proposed on-line approach is efficient to deal with cluttered and unstructured environment.

Experimentations are implemented using Khepera[®] III robot (cf. Fig. 15), its kinematic model is given by (2). As a very first tests, navigation is achieved on a platform equipped with a camera in the top which gives positions and orientations of the robots and the obstacles to avoid. The navigation is achieved on a platform using the local infrared sensors of the robot. This test demonstrate the efficiency of the proposed robust enclosing ellipse approach. The real trajectory of the robot avoiding two obstacles is given in figure 15. It can be seen that the robot successfully converges to its target after avoiding two obstacles (surrounded with two ellipses of influence).

7. CONCLUSION

This paper proposes an on-line and adaptive elliptic trajectory to perform smooth and safe mobile robot navigation with reactive way. These trajectories use limit-cycle principle to obtain generic and flexible navigation in very unstructured environments. This elliptic limit-cycle trajectory is obtained while using the proposed heuristic method combined with Extended Kalman Filter which deals with the uncertain range data to obtain the ellipse parameters. This method was demonstrated and implemented on-line to enclose all range data. The proposed reactive navigation was embedded in multi-controller architecture. Furthermore, the proposed control law to this multi-controller architecture improve the performance of the controllers. Otherwise, the stability proof of the overall control architecture using only one control law for the two different controllers was shown. Simulations and experiments in different environments was presented. The efficiency and the flexibility of the proposed control architecture and the obstacle detection was proved.

In future works, the problem of outliers detection of the range data will be accurately considered and the proposed control structure will be extended for multi-robot system. Furthermore, the presence of dynamic obstacles will be developed.

REFERENCES

- [1] J.-C. Latombe, *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.
- [2] E. Rimon and Daniel Koditschek, "Exact robot navigation using artificial potential fields," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [3] T. Fraichard, "Trajectory planning in a dynamic workspace: a "state time" approach," *Advanced Robotics*, vol. 13, no. 1, pp. 75–94, 1999.
- [4] B. Jur-Van-Den and M. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Transactions on Robotics*, vol. 21(5), pp. 885–897, 2005.
- [5] M. Egerstedt and X. Hu, "A hybrid control approach to action coordination for mobile robots," *Automatica*, vol. 38(1), pp. 125–130, 2002.
- [6] J. Toibero, R. Carelli, and B. Kuchen, "Switching control of mobile robots for autonomous navigation in unknown environments," in *IEEE International Conference on Robotics and Automation*, 2007, pp. 1974–1979.
- [7] L. Adouane, "Hybrid and safe control architecture for mobile robot navigation," in *9th Conference on Autonomous Robot Systems and Competitions*, Portugal, May 2009.
- [8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, pp. 90–99, 1986.
- [9] R. C. Arkin, "Motor schema-based mobile robot navigation," *International Journal of Robotics Research*, vol. 8, no. 4, pp. 92–112, 1989.
- [10] R. Zapata, A. Cacitti, and P. Lepinay, "Dvz-based collision avoidance control of non-holonomic mobile manipulators," *JESA, European Journal of Automated Systems*, vol. 38(5), pp. 559–588, 2004.
- [11] R. C. Arkin, *Behavior-Based Robotics*. The MIT Press, 1998.
- [12] A. De Luca and G. Oriolo, "Local incremental planning for nonholonomic mobile robots," in *IEEE International Conference on Robotics and Automation*, vol. 1, May 1994, pp. 104–110.
- [13] D.-H. Kim and J.-H. Kim, "A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer," *Robotics and Autonomous Systems*, vol. 42(1), pp. 17–30, 2003.
- [14] M. S. Jie, J. H. Baek, Y. S. Hong, and K. W. Lee, "Real time obstacle avoidance for mobile robot using limit-cycle and vector field method," *Knowledge-Based Intelligent Information and Engineering Systems*, October 2006.
- [15] L. Adouane, "Orbital obstacle avoidance algorithm for reliable and on-line mobile robot navigation," in *9th Conference on Autonomous Robot Systems and Competitions*, Portugal, May 2009.
- [16] L. Adouane, A. Benzerrouk, and P. Martinet, "Mobile robot navigation in cluttered environment using reactive elliptic trajectories," in *18th IFAC World Congress*, August 2011.
- [17] A. Benzerrouk, L. Adouane, and P. Martinet, "Lyapunov global stability for a reactive mobile robot navigation in presence of obstacles," in *ICRA'10 International Workshop on Robotics and Intelligent Transportation System*, 2010.
- [18] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1990, pp. 384–389.
- [19] E. Welzl, "Smallest enclosing disks (balls and ellipsoids)," in *Results and New Trends in Computer Science*. Springer-Verlag, 1991, pp. 359–370.
- [20] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, pp. 59–76, 1997.
- [21] J. Vilca, L. Adouane, and Y. Mezouar, "On-line obstacle detection using data range for reactive obstacle avoidance," in *12th International Conference on Intelligent Autonomous Systems*. Korea, June 2012.
- [22] K. Xiong, C. Wei, and L. Liu, "Robust kalman filtering for discrete-time nonlinear systems with parameter uncertainties," *Aerospace Science and Technology*, 2011.
- [23] C. Fouque, P. Bonnifait, and D. Bétaille, "Enhancement of global vehicle localization using navigable road maps and dead-reckoning," in *IEEE Position Location and Navigation Symposium*, 2008.
- [24] G. G. Rigatos, "Extended kalman and particle filtering for sensor fusion in motion control of mobile robots," *Mathematics and Computers in Simulation*, vol. 81, no. 3, pp. 590–607, November 2010.
- [25] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *IEEE International Conference on Robotics and Automation*. Alaska, USA, May 2010.
- [26] J. Porrill, "Fitting ellipses and predicting confidence envelopes using a bias corrected kalman filter," *Image and Vision Computing*, vol. 8, no. 1, pp. 37–41, February 1990.
- [27] J. Vilca, L. Adouane, and Y. Mezouar, "Robust on-line obstacle detection using range data for reactive navigation," in *10th International IFAC Symposium on Robot Control*. Croatia, September 2012.
- [28] R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. RA-2, pp. 14–23, March 1986.
- [29] L. Adouane and N. Le Fort-Piat, "Behavioral and distributed control architecture of control for minimalist mobile robots," *Journal European des Systèmes Automatisés*, vol. 40, no. 2, pp. 177–196, 2006.
- [30] E. Maalouf, M. Saad, and H. Saliyah, "A higher level path tracking controller for a four-wheel differentially steered mobile robot," *Robotics and Autonomous Systems*, vol. 54, pp. 23–33, 2006.
- [31] R. De Maesschalck, D. Jouan-Rimbaud, and D. Massart, "The mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, 2000.
- [32] B. Barshan and R. Kuc, "Active sonar for obstacle localization using envelope shape information," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, April 1991, pp. 1273–1276.
- [33] A. Burguera, Y. Gonzalez, and G. Oliver, "Sonar sensor models and their application to mobile robot localization," *Sensors*, vol. 9, no. 12, pp. 10 217–10 243, December 2009.
- [34] H. K. Khalil, *Nonlinear Systems*, 3rd ed., P. Hall, Ed., 2002.