ISSN 2075-1087, Gyroscopy and Navigation, 2013, Vol. 4, No. 1, pp. 14–25. © Pleiades Publishing, Ltd., 2013.

Reactive Navigation of a Mobile Robot Using Elliptic Trajectories and Effective Online Obstacle Detection

J. Vilca, L. Adouane, and Y. Mezouar

Institut Pascal, UBP – UMR CNRS 6602, Clermont-Ferrand, France
E-mail: Jose Miguel.VilcaVentura@univ-bpclermont.fr

Received July 19, 2012

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

DOI: 10.1134/S2075108713010094

1. INTRODUCTION

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

Another community is concentrated on reactive methods dealing with obstacle avoidance, where only information from local sensors is used rather than a priori knowledge of the environment [5–7]. 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 a sum of attractive and repulsive fields. In [9], the author extends Khatib's approach, proposing specific schema motors for mobile robot navigation. Another interesting approach based on a reflex behavior reaction uses the concept of a deformable virtual zone (DVZ) in which a robot movement depends on the risk zone surrounding the robot [10]. If an obstacle is detected, it will deform the DVZ and the approach consists in minimizing this deformation by modifying the control vector. This method deals with any obstacle shape, however, it suffers as motor schema from the local minima problem. In general, reactive methods do not require high computational complexities since a robot's actions must be given in real-time according to the perception [11].

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

Different approaches have been proposed in 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 a 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 algebra and the Euclidean distance, the Kalman filtering method and robust estimation. In this work, the method we use is based on a simple and efficient heuristic approach based on Euclidean distance estimation [21].

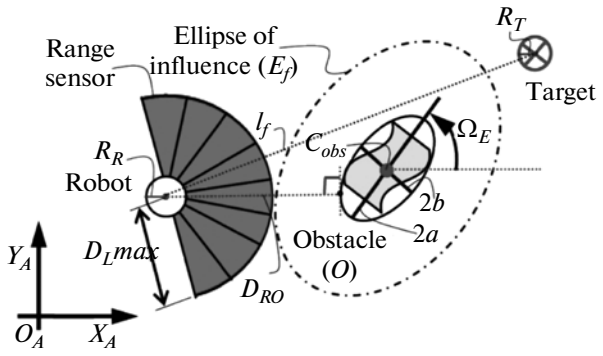

Fig. 1. An obstacle and robot representation.

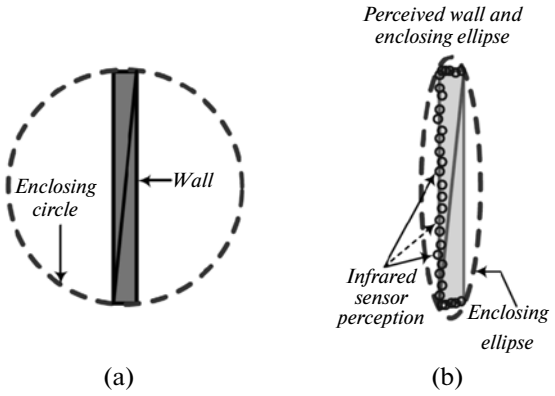

Fig. 2. Interpolated wall using circle and ellipse shapes.

The extended Kalman filter has important applications in different fields [22–25]. In [26], the author presents a method to fit ellipses using the 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 the proposed obstacle avoidance algorithm in detail. Section 5 presents the method for enclosing the uncertain range data with an ellipse. The simulation results are given in Section 6. Finally, conclusions and some prospects for the future work are given in Section 7.

2. NAVIGATION IN CLUTTERED ENVIRONMENT

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

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

where $(h, k) \in \mathbb{R}^2$ are the ellipse center coordinates and $(a, b) \in \mathbb{R}^+$ are the semi-axes ($a \geq b$). $c \in \mathbb{R}$ makes it possible to give the ellipse orientation (see Fig. 1)

$$\Omega_E = 0.5 \arctan\left(c/(b^{-2} - a^{-2})\right).$$

Ellipse boxes rather than circles are exploited to obtain a generic and flexible means to surround and accurately fit different kinds of obstacle shapes [16]. Among the examples of the 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 wanted to surround this wall with a circle, it would have a large radius which will induce a longer robot path distance in order to safely avoid the obstacle [13] (see Fig. 2a). Figure 2b shows that the ellipse fits the dimension of the obstacle better. This figure also shows uncertain perceptions taken by the range sensor on the left side of the wall.

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 radiuses R_R and R_T, respectively (see Fig. 1). We can define:

(1) D_{RO} as the minimal distance between the robot and the obstacle O;

(2) The ellipse of influence (E_f) as an ellipse that has the same center (h, k) and tilt angle Ω_E as the ellipse which surrounds obstacle (1); its major and minor semi-axes, a_{lc} and b_{lc} , are defined as follows:

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

where *Margin* represents safety tolerances encapsuling perception uncertainties, control reliability, and accuracy. This definition is similar to the configuration space for global path planning; 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 below, for our method we only need to know whether there exists intersection points between l_f and E_f (see 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 by avoiding static and dynamic obstacles O which can have different shapes.

3. CONTROL ARCHITECTURE

The control structure is based on [17] (see Fig. 3). It aims to manage interactions between elementary controllers, guaranteeing the stability of the overall control

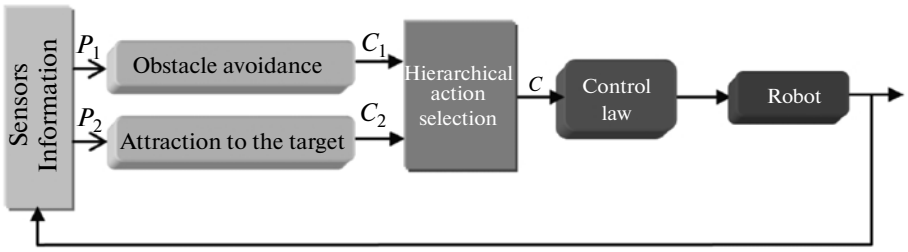

Fig. 3. Control architecture for mobile robot navigation.

as proposed in [7]. Its objective is also to ensure safe, smooth and fast robot navigation. The specific blocks that compose the global controller are detailed below.

This control architecture uses a procedure for selecting a hierarchical action to manage the switches between the controllers according to environment perception. The mechanism activates the obstacle avoidance controller as soon as there exists at least one obstacle which can obstruct the future robot movement toward its target (see Algorithm 1) [15]. This allows us to anticipate the activation of the obstacle avoidance controller unlike what is proposed in [28, 29], when the controller waits until the robot is in the immediate vicinity of the obstacle (i.e., $D_{RO} \le R$, where R is a certain radius value). Thus, Algorithm 1 allows decreasing the time needed to reach the target, especially in very cluttered environment.

If *there exists at least one constrained obstacle* {*i.e., there exists at least one intersection point between the line "l_f" and the ellipse of influence* (*see Fig. 1*)}, **then**

Activate *Obstacle avoidance* controller;

else

Activate *Attraction to the target* controller;

end

Algorithm 1: Hierarchical action selection.

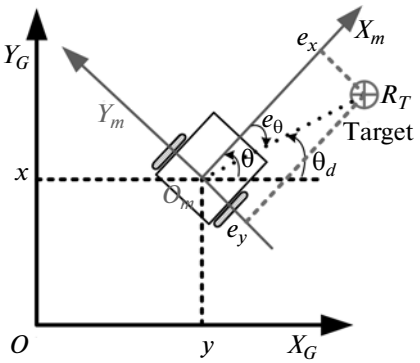

Fig. 4. Robot and target configuration in the Cartesian reference frame.

3.1. Elementary controllers

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

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

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

Attraction to the target controller This controller guides the robot toward the target which is represented by circle C_T of center (x_T, y_T) and radius R_T (see Fig. 4). It is based on the configuration of the robot relative 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 the robot axis reaches the target with asymptotical convergence, d must be smaller than R_T, where $d = \sqrt{e_x^2 + e_y^2}$ (see Fig. 4).

The position and orientation errors with respect to the local reference frame $X_m - Y_m$ are as follows:

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

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:

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

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

Obstacle avoidance controller To perform the obstacle avoidance behavior, the robot needs to accurately follow 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 a

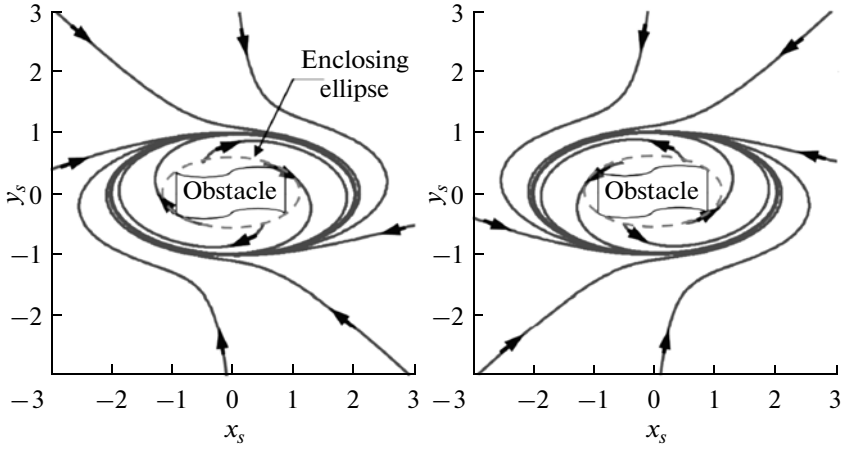

Fig. 5. Clockwise ($m = 1$) and counterclockwise ($m = -1$) shapes for the elliptic limit cycles used.

more flexible limit-cycle shape (an ellipse). The main ideas of this controller are detailed below (see Fig. 6).

The differential equations giving elliptic limit-cycles are the following:

$$\dot{x}_s = my_s + x_s\left(1 - x_s^2/a_{lc}^2 - y_s^2/b_{lc}^2 - cx_sy_s\right), \quad (5)$$

$$\dot{y}_s = -mx_s + y_s\left(1 - x_s^2/a_{lc}^2 - y_s^2/b_{lc}^2 - cx_sy_s\right), \quad (6)$$

with $m = \pm 1$ according to the direction of avoidance (clockwise or counterclockwise, see Fig. 5). (x_s, y_s) correspond to the position of the robot according to the center of the ellipse; a_{lc} and b_{lc} characterize the major and minor elliptic semi-axes, respectively (see 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) (see Fig. 6). The desired robot orientation is given by the differential equation of the limit cycle (5) and (6) as (see Fig. 6):

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

3.2. Control law

The proposed control law is based on [18] and adapted to the multicontroller 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 (see 3.1.1 and 3.1.2). Further, this control law makes it possible to improve the performance of these controllers (see Section 6).

Consider that the error with respect to the 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 (see 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) (see [18]):

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

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

$$\upsilon = \upsilon_r\cos(e_\theta) + K_xe_x, \quad (10)$$

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

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

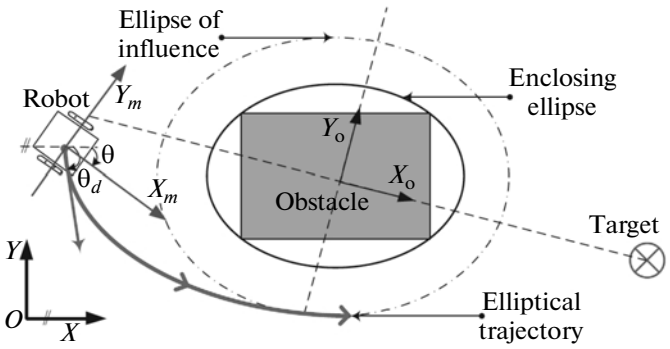

Fig. 6. Obstacle avoidance controller.

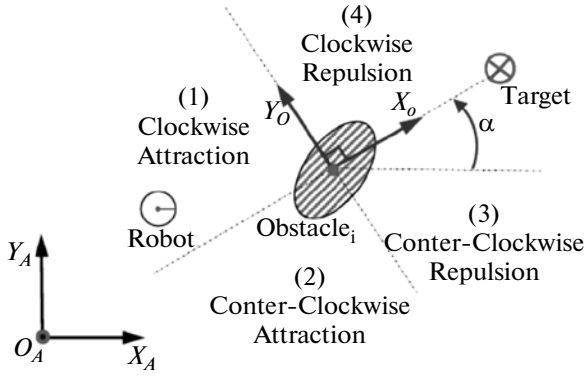

Fig. 7. The four specific areas to avoid an obstacle [16].

the error for the case where $\upsilon_r = 0$ (*Attraction to the target* controller) (see 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 only depends 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:

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

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, υ_r and ω_r are computed as

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

It is interesting to notice that a single 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 [16]

In what follows, the overall methodology for achieving the proposed obstacle avoidance algorithm will be given [16].

The algorithm is developed according to the stimulus-response principle. To implement this kind of behavior, it is important:

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

All these different steps must be followed and applied, guaranteeing that the robot trajectory is safe, smooth and avoids undesirable situations such as deadlocks or local minima, and that the stability of the applied control law is guaranteed (see Subsection 3.1). In this paper, the obstacle observation is obtained in real time, the robot does not have global information either about the hinder obstacle or the ellipse that encloses it. Thus, at each sample time, the robot discovers the shape of the obstacle and increases progressively the knowledge of the enclosing ellipse to obtain a smooth elliptic trajectory. The global information of the obstacle is not related to the reactive navigation used. 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*' (see Fig. 1).

(2) Among the set of disturbing obstacles (which can constrain the robot to reach the target), choose the one closer to the robot (the smallest D_{ROi} (see Fig. 1)). This specific obstacle has the following features: (x_{obst}, y_{obst}) center position, and 2*a* as the major axis and 2*b* as the minor axis.

(3) After the determination of the closest constrained obstacle, we need to obtain four specific areas (see Fig. 7) which give the robot behavior – clockwise or counterclockwise obstacle avoidance; repulsive or attractive phase (see Algorithm 2). To distinguish between these 4 areas we need:

- To define a specific reference frame which has the following features (see 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 the 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 by using the following homogeneous transformation:

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

Once all the necessary perceptions are obtained, we can apply the reactive obstacle avoidance strategy given by Algorithm 2. To obtain the robot set points, it is necessary to obtain the values of a_{lc} and b_{lc} (see Sec-

tion 2) of the orbital ellipse and the clockwise or counterclockwise direction of the limit cycle to follow. The position (x_O, y_O) gives the configuration (x, y) of the robot according to the 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).

At the 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 the clockwise limit-cycle direction or else apply the counterclockwise direction. This choice permits optimization of the robot trajectory length 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 to avoid the local minima and dead-end [15].

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

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 x a small constant value as* $x \ll$ *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

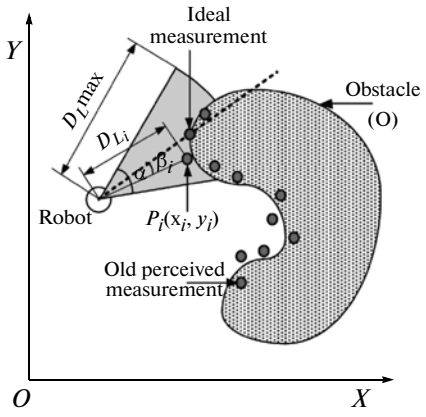

Fig. 8. Range sensor model and data set of n points.

12 {*The limit - cycle set - point is given by* :}

$$\dot{x} = \text{sgn}(y_O)y + x(1 - x^2/a'^2_{lc} - y^2/b'^2_{lc} - cxy)$$

$$\dot{y} = -\text{sgn}(y_O)x + y(1 - x^2/a'^2_{lc} - y^2/b'^2_{lc} - cxy)$$

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 a hinder obstacle online and to avoid it. With this aim in view, the observed noisy range data are surrounded with the closest ellipse to apply the elliptic limit-cycle approach.

For this purpose, let us consider a set of n points in $\mathbb{R}^2$ with coordinates $\mathbf{P}_i(x_i, y_i)$ (see Fig. 8). These points are computed from the data range of the robot, and the outliers are erased by using the Mahalanobis distance [31]. In this section, it will be shown how to compute an ellipse that encloses all points. An important condition in this work is that the method needs to start at least with three different points.

In this paper, the robot detects one obstacle at a time. The segmentation method of a set of points will be used in future works to detect more than one obstacle at a time and big obstacles as well.

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

5.1. Range sensor model

The position of an 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

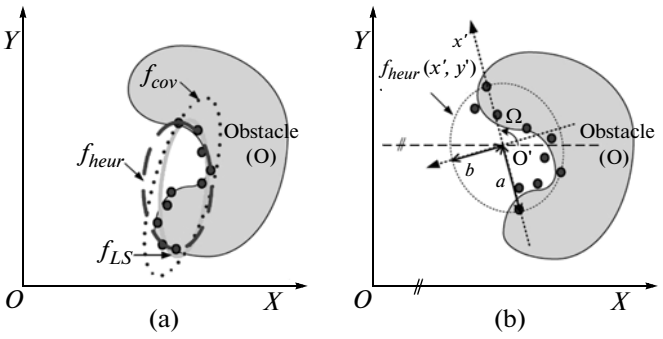

Fig. 9. The ellipses obtained with the use of the least-square (f_{LS}), covariance (f_{cov}) and heuristic (f_{heur}) approaches.

and β_i is the orientation with respect to the mobile reference frame (see 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 $\hat{D}_{Li}^t = [D_{Li}^t, 0]^T$ is the mean vector, D_L is the range measure of the sensor and $P_{Li}^t = \mathrm{diag}\left(\left[\sigma_{D_L}^2, \sigma_{\beta_i}^2\right]\right)$ 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 (see Fig. 8) [33]. The representation of the range data in the Cartesian frame is given by

$$\mathbf{z}_i = \mathbf{z}_i' + \mathbf{v}_i, \tag{16}$$

where $\mathbf{z}_i = [x_i, y_i]^{\mathrm{T}}$ 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 $\mathbf{v}_i$ has $E[\mathbf{v}_i] = 0$ and $E\left[\mathbf{v}_i \mathbf{v}_i^T\right] = \mathbf{R}_{vi}$. The covariance $\mathbf{R}_{vi}$ is given by

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

The following subsections will present the proposed methods to enclose the data with an ellipse. Subsection 5.2 addresses the problem of enclosing an 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 for enclosing an ellipse is given; a heuristic method was proposed and compared with the existing approaches (based on least square or covariance). This method makes it possible to enclose all range data and to obtain smooth changes of the ellipse parameters, different from the least square and the covariance method that could have a point outside of the obtained ellipse and abrupt change of parameters (see Fig. 9a). In this work, the heuristic approach is given in more detail with the mathematical proof that the obtained ellipse parameters enclose 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 $i = 1, ..., n$. The parameters of the ellipse that enclose all points are computed as follows:

(1) Compute the distance between all the points $d_{ij} = \|p_i - p_j\|$ with $i, j = 1, ..., n$; 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 the maximum distances and the first semiaxis is $a_1 = d_{max}/2$ (see Fig. 9b).

(3) Transform the n points to a new coordinate system $X' - Y'$ using (18) to obtain the second ellipse semi-axis a_2.

$$\mathbf{P}_i' = \begin{bmatrix} \cos(\Omega) & \sin(\Omega) \\ -\sin(\Omega) & \cos(\Omega) \end{bmatrix} (\mathbf{P}_i - C_O), \tag{18}$$

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 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 $\left|y_i'\right|$ of the points is greater than a threshold $\varepsilon > 0$. This threshold is used to eliminate the points that are collinear with the two points that have the maximum distance (first axis) and the points in the perpendicular line to the 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 ellipse (1) are obtained:

$$a = max\{a_1, a_2\},$$
$$b = min\{a_1, a_2\} \tag{19}$$

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

$$b_i^2 \leq b^2,$$

$$\frac{y_i'^2}{1 - x_i'^2/a^2} \leq b^2,$$

$$x_i'^2/a^2 + y_i^2/b^2 \leq 1. \quad (20)$$

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

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 either uncertain data or the sequence of the obtained the data which characterize the real experiments. The following subsection deals with this issue.

5.3. Identification of Optimal Parameters Using EKF

The 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 an analytic equation of a 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 $\{p_i\} = \{(x_i, y_i)\} | i = 1,..., n$. This set of points is 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, the 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 $i = 0, 1, ..., n$. 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 a 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 the function of the ideal measurement $\mathbf{z}_i'$ (i.e., 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 5.1. We expand $f_i(\mathbf{z}_i', \mathbf{x}_i)$ into the Taylor series about $(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})$

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

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

$$\mathbf{M}_i = \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{x}_i},$$

$$\mathbf{y}_i = -f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1}) + \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{x}_i}\hat{\mathbf{x}}_{i|i-1},$$

$$\xi_i = \frac{\partial f_i(\mathbf{z}_i, \hat{\mathbf{x}}_{i|i-1})}{\partial \mathbf{z}_i'}(\mathbf{z}_i' - \mathbf{z}_i).$$

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

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

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

The extended Kalman filter (EKF) is then used, knowing that the output equation (23) is obtained

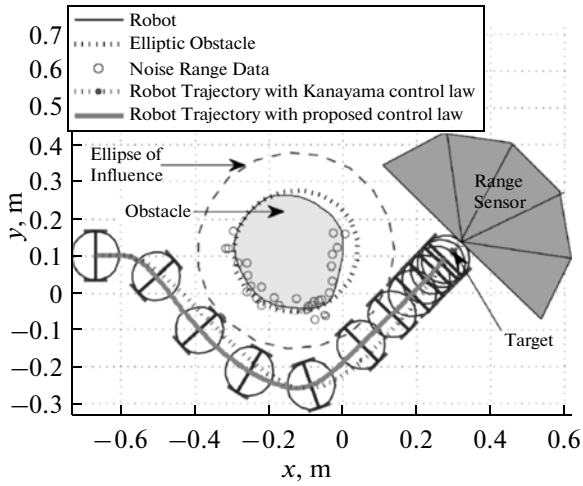

Fig. 10. Robot trajectory using the two different control laws for the multicontroller architecture.

from the nonlinear measurement equation (24). The 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:

$$\Pi_{i|i-1} = \mathbf{F}_{i-1}\Pi_0\mathbf{F}_{i-1}^T + \mathbf{Q}_{i-1}.$$

- Kalman gain matrix:

$$K_i = \Pi_{i|i-1}\mathbf{H}_{i-1}^T(\mathbf{H}_{i-1}\Pi_{i|i-1}\mathbf{H}_{i-1}^T + \mathbf{R}_v)^{-1}.$$

- 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 another and may run the risk of being trapped into a local minimum [20].

6. SIMULATION RESULTS

To demonstrate the efficiency of the proposed control law in the multicontroller architecture and the online obstacle detection method to enclose the obstacle with an ellipse and avoid it, two statistical surveys were made. In this paper, a mobile robot with a radius of $R_R = 0.065\ m$ and six infrared range sensors with the maximum detected range equal to $D_L max = 0.30\ m$ is considered (see Fig. 8). These sensors are in front of the robot, with 30° between each pairs of sensors (see Fig. 10). The accuracy of the sensors based on the datasheet is around 10% of $D_L max$. In the simulation, we consider an uncertainty range with the maximum value of 20% of $D_L max$, thus ensuring 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 $v_{max} = 0.4$ m/s and $\omega_{max} = 3$ rad/s. The sample time is 0.01 s. For each simulation, the robot starts at the same configuration and reaches the same final configuration. The heuristic method does not start until there are enough range data ($n_{data} \geq 3$).

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

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 that 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 improves 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 improves the convergence 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 the Lyapunov function V_0 (12) attributed to the control law (see Fig. 3) when the navigation is performed. This function decreases asymptotically to the equilibrium point. It is observed that the Lyapunov function of the proposed control law has a lower value in the switch

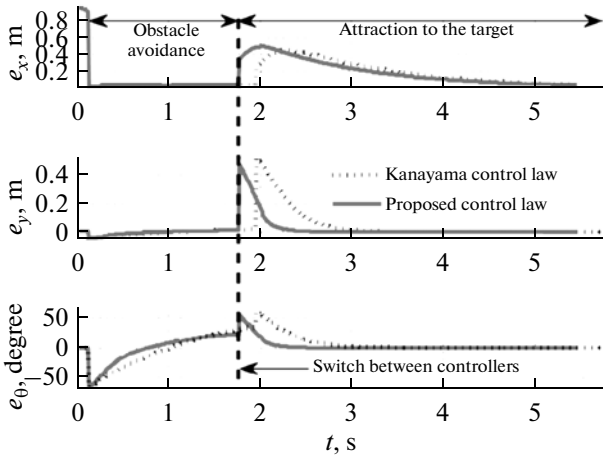

Fig. 11. Position and orientation errors.

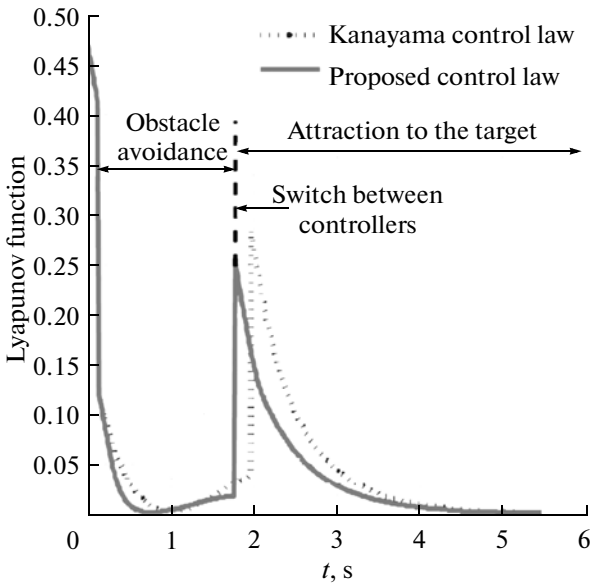

Fig. 12. Evolution of the Lyapunov functions for the two different control laws during the robot navigation.

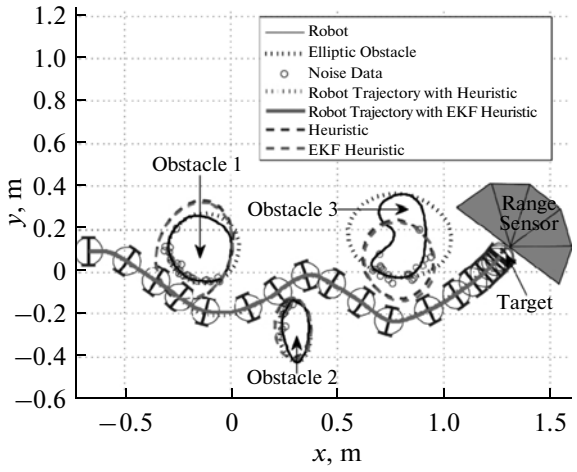

Fig. 13. Robot trajectory using the heuristic and EKF approaches to enclose an obstacle.

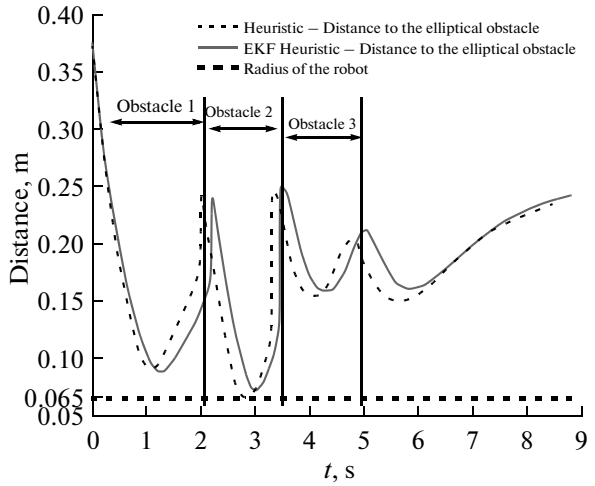

Fig. 14. Distance from the robot to the elliptical obstacles.

between the controller and faster convergence than in the case of the Kanayama control law.

The second survey makes a focus around the proposed heuristic method and the Kalman filter which give satisfactory results when they are used for online navigation in cluttered environment (see 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 the entire trajectory. The range data buffer used to compute the ellipse parameters is deleted for each new discovered obstacle. This figure shows that the ellipse obtained by using the heuristic method encloses all the points as it was demonstrated above. Further, it is observed that the robot avoids the obstacles with a smooth trajectory. This trajectory was obtained by using the online obstacle avoidance algorithm [16] with the proposed control law which takes its parameters (elliptical limit cycle to follow) from the combination of the proposed heuristic approach and EKF.

Figure 14 shows the minimum distance between the effective elliptical obstacles (obtained from the knowledge of 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 the heuristic method and the EKF (green continuous line). This figure shows that the robot never collides

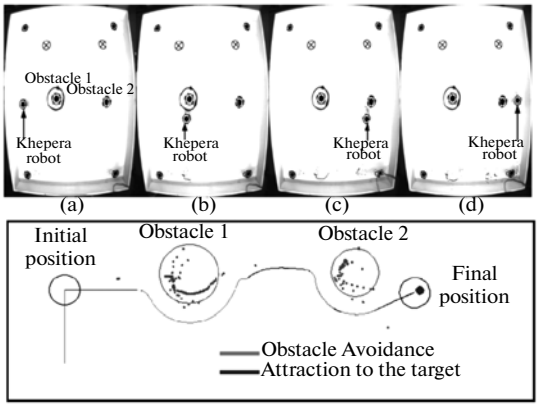

Fig. 15. Top view of the robot trajectory in the platform and uncertain range data observed from the robot.

with any obstacles when the proposed control law is used; therefore, the proposed online approach is efficient to deal with cluttered and unstructured environment.

Experimentations are implemented using a Khepera® III robot (see Fig. 15). Its kinematic model is given by (2). As very first tests, navigation is achieved on a platform equipped with a camera on the top, which gives the 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 demonstrates the efficiency of the proposed robust approach with the use of enclosing ellipses.

The real trajectory of the robot avoiding two obstacles is given in Fig. 15. It can be seen that the robot successfully converges to its target after avoiding two obstacles (surrounded with two ellipses of influence).

7. CONCLUSIONS

This paper proposes an online and adaptive elliptic trajectory to perform smooth and safe mobile robot navigation in reactive way. These trajectories use a limit-cycle principle to obtain generic and flexible navigation in very unstructured environments. This elliptic limit-cycle trajectory is obtained by using the proposed heuristic method combined with extended Kalman filter which deals with uncertain range data to obtain ellipse parameters. This method was demonstrated and implemented online to enclose all range data. The proposed reactive navigation was embedded in multicontroller architecture. Furthermore, the proposed control law to this multicontroller architecture improves the performance of the controllers. Otherwise, the stability proof of the overall control architecture using a single control law for two different controllers has been shown. Simulations and experiments in different environments have been presented. The efficiency and the flexibility of the proposed control architecture and the obstacle detection have been proved.

In the future works, the problem of detecting outliers of the range data will be accurately considered and the proposed control structure will be extended for a multirobot system. Furthermore, the presence of dynamic obstacles will be developed.

REFERENCES

1. Latombe, J.C., *Robot Motion Planning*, Kluwer Academic Publishers, Boston, MA, 1991.
2. Rimon, E. and Koditschek, D., Exact Robot Navigation Using Artificial Potential Fields, *IEEE Transactions on Robotics and Automation*, 1992, vol. 8, no. 5, pp. 501–518.
3. Fraichard, T., Trajectory Planning in a Dynamic Workspace: a "State-Time Space" Approach, *Advanced Robotics*, 1999, vol. 13, no. 1, pp. 75–94.
4. Jur-Van-Den, B., and Overmars, M., Roadmap-Based Motion Planning in Dynamic Environments, *IEEE Transactions on Robotics*, 2005, vol. 21(5), pp. 885–897.
5. Egerstedt, M. and Hu, X., A Hybrid Control Approach to Action Coordination for Mobile Robots, *Automatica*, 2002, vol. 38(1), pp. 125–130.
6. Toibero, J., Carelli, R., and Kuchen, B., Switching Control of Mobile Robots for Autonomous Navigation in Unknown Environments, *IEEE International Conference on Robotics and Automation*, 2007, pp. 1974–1979.
7. Adouane, L., Hybrid and Safe Control Architecture for Mobile Robot Navigation, *9th Conference on Autonomous Robot Systems and Competitions*, Portugal, May 2009.
8. Khatib, O., Real-Time Obstacle Avoidance for Manipulators and Mobile Robots, *International Journal of Robotics Research*, 1986, vol. 5, pp. 90–99.
9. Arkin, R. C., Motor Schema-based Mobile Robot Navigation, *International Journal of Robotics Research*, 1989, vol. 8, no. 4, pp. 92–112.
10. Zapata, R., Cacitti, A., and Lepinay, P., DVZ-Based Collision Avoidance Control of Non-holonomic Mobile Manipulators, *JESA, European Journal of Automated Systems*, 2004, vol. 38(5), pp. 559–588.
11. Arkin, R.C., *Behavior-Based Robotics*, MIT Press, 1998.
12. De Luca, A. and Oriolo, G., Local Incremental Planning for Nonholonomic Mobile Robots, *IEEE International Conference on Robotics and Automation*, May 1994, vol. 1, pp. 104–110.
13. Kim D.-H. and Kim, J.-H., A Real-Time Limit-Cycle Navigation Method for Fast Mobile Robots and its Application to Robot Soccer, *Robotics and Autonomous Systems*, 2003, vol. 42(1), pp. 17–30.
14. Jie, M.S., Baek, J.H., Hong, Y.S., and Lee, K.W., Real Time Obstacle Avoidance for Mobile Robot Using Limit-Cycle and Vector Field Method, *Knowledge-Based Intelligent Information and Engineering Systems*, October 2006.
15. Adouane, L., Orbital Obstacle Avoidance Algorithm for Reliable and On-Line Mobile Robot Navigation, *9th Conference on Autonomous Robot Systems and Competitions*, May 2009, Portugal.
16. Adouane, L., Benzerrouk, A., and Martinet, P., Mobile Robot Navigation in Cluttered Environment Using

Reactive Elliptic Trajectories, *18th IFAC World Congress,* August 2011.

17. Benzerrouk, A., Adouane, L., and Martinet, P., Lyapunov Global Stability for a Reactive Mobile Robot Navigation in Presence of Obstacles," *ICRA'10 International Workshop on Robotics and Intelligent Transportation System*, 2010.
18. Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T., A Stable Tracking Control Method for an autonomous mobile robot, *Proceedings of the IEEE International Conference on Robotics and Automation*, May 1990, pp. 384 – 389.
19. Welzl, E., Smallest Enclosing Disks (Balls and Ellipsoids), *Results and New Trends in Computer Science,* Springer-Verlag, 1991, pp. 359–370.
20. Zhang, Z., Parameter Estimation Techniques: A Tutorial with Application to Conic Fitting, *Image and Vision Computing*, 1997, vol. 15, pp. 59–76.
21. Vilca, J., Adouane, L., and Mezouar, Y., On-Line Obstacle Detection Using Data Range for Reactive Obstacle Avoidance, *12th International Conference on Intelligent Autonomous Systems*, Korea, June 2012.
22. Xiong, K., Wei, C., and Liu, L., Robust Kalman Filtering for Discrete-Time Nonlinear Systems with Parameter Uncertainties, Aerospace Science and Technology, 2011.
23. Fouque, C., Bonnifait, P., and Betaille, D., Enhancement of Global Vehicle Localization Using Navigable Road Maps and Dead-Reckoning, *IEEE Position Location and Navigation Symposium*, 2008.
24. Rigatos, G.G., Extended Kalman and Particle Filtering for Sensor Fusion in Motion Control of Mobile Robots, *Mathematics and Computers in Simulation*, November 2010, vol. 81, no. 3, pp. 590–607.
25. Levinson, J. and Thrun, S., Robust Vehicle Localization in Urban Environments Using Probabilistic Maps, *IEEE International Conference on Robotics and Automation,* Alaska, USA, May 2010.
26. Porrill, J., Fitting Ellipses and Predicting Confidence Envelopes Using a Bias Corrected Kalman Filter, *Image and Vision Computing,* February 1990, vol. 8, no. 1, pp. 37–41.
27. Vilca, J., Adouane, L., and Mezouar, Y., Robust Online Obstacle Detection Using Range Data for Reactive Navigation, *10th International IFAC Symposium on Robot Control*, Croatia, September 2012.
28. Brooks, R.A., A Robust Layered Control System for a Mobile Robot, *IEEE Journal of Robotics and Automation*, vol. RA-2, March 1986, pp. 14–23.
29. Adouane, L. and Le Fort-Piat, N., Behavioral and Distributed Control Architecture of Control for Minimalist Mobile Robots, *Journal Europen des Systèmes Automatisés*, 2006, vol. 40, no. 2, pp. 177–196.
30. Maalouf, E., Saad, M., and Saliah, H., A Higher Level Path Tracking Controller for a Four-Wheel Differentially Steered Mobile Robot, *Robotics and Autonomous Systems*, 2006, vol. 54, pp. 23–33.
31. De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D., The Mahalanobis Distance, *Chemometrics and Intelligent Laboratory Systems*, 2000, vol. 50, no. 1, pp. 1–18.
32. Barshan, B. and Kuc, R., Active Sonar for Obstacle Localization Using Envelope Shape Information, *International Conference on Acoustics, Speech, and Signal Processing*, April 1991, vol. 2, pp. 1273–1276.
33. Burguera, A., Gonzlez, Y., and Oliver, G., Sonar Sensor Models and Their Application to Mobile Robot Localization, *Sensors*, December 2009, vol. 9, no. 12, pp. 10217–10243.
34. Khalil, H.K., *Nonlinear Systems*, 3rd ed., P. Hall, Ed., 2002.