International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Jul-2022)

https://dx.doi.org/10.12785/ijcds/120118

EKF-DWA for Car-Like Robot Navigation in Presence of Moving Obstacles

REBAI Karima[1] and BOUCHAMA Samira[1]

[1]*Laboratory of Innovative Technologies, National Higher School of Technology, Algiers, Algeria*

Received 18 Aug. 2021, Revised 8 May 2022, Accepted 15 Jun. 2022, Published 1 Jul. 2022

Abstract: In the research field of autonomous robots or vehicle navigation, several works have been carried out in order to allow the avoidance of fixed obstacles. However, the presence of a moving obstacles presents a challenge, particularly when the vehicle moves at high-speed. Indeed, in a robot's environment, it is not enough to use only the obstacles positions for avoiding them but it is more valuable and necessary to consider their future predicted trajectories. In this paper, we present a method based on both the principle of the Dynamic Window Approach (DWA) which is extended for car-like robot navigation, and Extended Kalman Filter (EKF) which is based on moving obstacle detection and the tracking module. The former are detected and tracked using laser rangefinder and individual EKF for each obstacle. The proposed method is tested in simulation for different scenarios that are close to real environments and has shown satisfactory results.

Keywords: Car-like robot, Dynamic Window Approach (DWA), Extended Kalman Filter (EKF), Moving Obstacles, Time to Collision, Direction to Collision

1. Introduction

The development of autonomous mobile robots has received considerable attention from many researchers and professionals in the field of robotics. The major research areas in this field are: The trajectory planning, localization, terrain related challenges and obstacle avoidance. The reader can find examples in [1], [2], [3], [4]. The main and critical functionality of robots is to be able to move from a starting position to a given target in an unknown environment in a safe way. Within this framework, different approaches are proposed in the literatures that can be divided into two categories: planning and reactive navigation, in addition to a combination between them. In the first category, the use of path planning algorithms can provide to the robot a sequence of routes and intermediate target points. These algorithms can work well using the environment map and in the presence of only static obstacles. However, the real world environments are partially or completely unknown. The latter are dynamic, change over time and contain moving objects that may easily block the robot's path. To overcome this hindrance, the autonomous vehicle should have the capability to detect, recognize and avoid both static and moving obstacles. Over the last decades, various reactive navigation methods have been developed in the field of mobile robotics [4], [5], [6], [7], [8], [9] but have several hitches when navigating at high speed in dense and crowded environments, which is usually the case in most real robot environments. In parallel with this, the detection and the tracking of moving objects from laser range finder measurement is also an important topic in mobile robotics. The proposed solutions in this context can be exploited for avoiding collision with moving obstacles. In the approaches proposed by Rebai et al.[10] and later Dekan et al. [11], objects are described by basic geometric features (segments) and their tracking was done by Extended Kalman Filters (EKF). In the well-known methods (VFE)[2] and Potential Field (PF)[4], [5], [6], [7], [8] and their variants [9], the obstacles generate repulsive forces that conduct the robot away the obstacles but these methods have several limits depending on the obstacle configuration and the vehicle dynamics. Other methods known as Velocity methods determine the next robot's command in the velocity space configuration such as Dynamic Window Approach (DWA) [5][7] and Curvature Velocity Method (CVM [12][13]. These approaches are more suitable for differential and holonomic robots because each point in this space corresponds to a translation and rotation velocities that can be directly executed by the robot. In the DWA, the robot's dynamics (acceleration and deceleration capabilities) are considered. This particularity made from it one of the attractive approaches for real world applications [14]. The adaptation of this approach for car-like robots [15][16] shows also the possibility of using it principally for self-driving cars. The application of all these methods in dynamic environments seems not to be secure especially when the robot moves at median and high speeds

E-mail address: karima.rebai@enst.dz, samira.bouchama@enst.dz

and in the presence of fast-moving objects. However, not dealing correctly with the moving obstacles in a navigation approach can steer the robot and the other moving objects to dangerous situations of collision. Therefore, the aim of this work is to improve DWA for car-like robot navigation in the presence of moving obstacles. Seder et al. [17] and Molinos et al.[18] schemed new versions of the DWA that consider also moving obstacles. In [17], the moving obstacles are represented by moving cells in the occupancy grid map and their motion is predicted using a procedure similar to the DWA. The idea of grid map has also been exploited in DW4DO method proposed in [18]. In this approach, instead of considering a global map, only a local occupancy map is created at the current robot position and completed by information about the moving obstacles. The command selection is established by the evaluation of a new objective function. The DW4DO showed attractive performances but the size of the local grid map produces a second-rate compromise between the maximal speeds of the considered obstacles and the dimension of the local map. For example, the local grid used in [18] is of 1mx1mx1s. It is a small representation that fails in the detection and the identification of moving obstacle that has a relatively high or medium velocity. These obstacles are detected close to the robot which leads to high collision risk especially for non-holonomic robots. Missura et al.[19] consider also a local grid map but the moving polygonal obstacles are detected and their velocity are computed by the comparison between successive grid maps. The obstacle velocity is supposed constant. The moving polygonal are removed from this map. To avoid these obstacles, the possible commands are then evaluated according to a modified objective function that captures obstacle clearance and progress towards the goal. Among the previously cited works that use grid map for environment representation (both static and dynamic obstacles) [17][18], only Missura et al. [19] proposed to extract moving obstacles from the grid and suppose that their form is polygonal. The consideration of all the grid map during the robot motion for the evaluation of the possible commands using the cost function can lead to unnecessary computation complexity that can growth the risk of fast moving obstacles. In order to overcome this limitation, local grid map was preferred in [18][19] however this also can lead to be surprised by the appearance of a rapid obstacle in this local region and the robot don't have the necessary time to avoid it. In these methods, the prediction of moving obstacle motion was simplified without considering the real possible motions that are nonlinear and random. It is also important to discern that these methods were developed for differential nonholonomic robot and their functionality for car-like robot has not been investigated. It is clear that the building of grid cell map during the robot motion and its exploitation for the evaluation of all possible commands can lead to an important computation cost. It also needs an important compromise between cell size, for reducing the complexity and the map size, and the quality of the available information (pose and velocity) about the moving obstacles. Our proposed solution for car-like robot navigation in the presence of moving obstacles, without using grid map, consists of the exploitation of object tracking by the Extended Kalman Filters (EKF). To this end, an EKF is created for the tracking of each detected moving obstacle and provides all the needed information about its motion. These data are used to estimate the following information: time-to-collision and the direction with higher risk of collision. This information is then introduced in the objective function of the DWA. As a reminder, a car like robot has non-holonomic constraints which reduce the mobility of the mechanical system. To our knowledge, it's the first approach that combines EKF and DWA for robot navigation in dynamic environments. After this introduction and the presentation of the most related works, the reminder of this paper is organized as follows: Section 2 describes the preliminaries of the developed approach where we present the adaptation of the DWA for car-like robot and the developed procedure for moving obstacles detection and tracking. In Section 3, the proposed combination of DWA with EKF, form moving obstacle detection and tracking, is presented. The obtained simulation results for different scenarios that are close to real environments are presented in section 4. Finally, section 5 shows conclusions and proposes some derived future works.

2. Preliminaries

A brief description of the necessary background for DWA and EKF is presented below.

A. Dynamic Window Approach (DWA) for car-like robot

The core idea of the DWA is to select the robot commands (translation and rotation velocities) with the consideration of the following constraints:

- The maximum of velocities that the robot can reach;
- The robot's dynamics: capability of acceleration and deceleration;
- The needed time for executing one cycle of navigation process (perception, execution of the navigation algorithm, robot control);
- The robot short time safety: elimination of the commands that conduct to high risk of collision with obstacles.

The initial DWA has been developed for synchro-drive robots that can move in any direction [5]. It is not possible to apply this method directly to a non-holonomic robot [15][16]. In the case of robot controlled with longitudinal velocity and steering angle (figure 1). The robot configuration is given by $q = [x, y, \theta]^T$ where (x, y) are the coordinates of the robot pose. θ is its orientation and ϕ is the steering angle. L is the length between front and rear wheels of the robot. ICR is the instantaneous center of rotation.The kenematic model of this robot is given by (1). We recall that the robot instantaneous trajectory is a curvature of a center

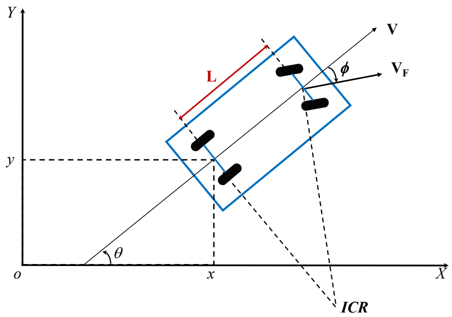

Figure 1. Car-like robot parameters, ICR is the instantaneous center of rotation

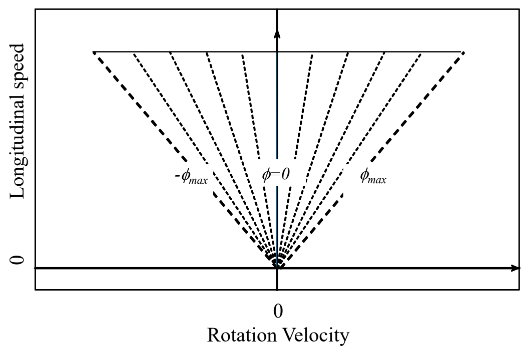

Figure 2. Car-like robot Velocity space

ICR and a rayon ρ. The robot commands are longitudinal velocity (v) and steering angle (ϕ).

$$\begin{cases} \dot{x} = v\cos(\theta) \\ \dot{y} = v\sin(\theta) \\ \dot{\theta} = \omega = v\frac{\tan(\phi)}{L} \end{cases} \tag{1}$$

It is important to remember that for mechanical constraints, the robot steering is limited by ϕ_{max}:

$$|\phi| \leq \phi_{max} \tag{2}$$

This constraint conducts to minimal instantaneous radius of rotation:

$$\rho_{min} = \frac{L}{\tan(\phi_{max})} \tag{3}$$

and leads to the equation (4) that indicates the limits of rotational velocities.

$$|\omega| \leq v\frac{\tan(\phi_{max})}{L} \tag{4}$$

The maximal and minimal rotation velocities are not defined values but linearly dependent of the actual longitudinal speed as shown in figure 2. Therefore, the velocity search space in the case of car-like robot has a triangular shape.

In DWA, for each cycle, the next command is selected by the evaluation of all the velocities (v, ω) in the velocities space determined by the intersection between the velocity search space, admissible velocities and dynamic window (5).

$$(v, \omega) \in V_s \cap V_{adm} \cap V_d \tag{5}$$

Where V_s: Velocity search space limited by minimal and maximal translation and rotation velocities:

$$(v, \omega) \in V_s \left\{v \in [v_{min}, v_{max}], \omega \in [\omega_{min}, \omega_{max}]\right\} \tag{6}$$

V_{adm}: Admissible velocities that allow the robot to stop before hitting an obstacle, It is given by:

$$\begin{aligned} (v, \omega) \in V_{adm} \Big\{ & |v| \leq \sqrt{2 \cdot Dist(v, \omega) \cdot \dot{v}_{dec}}, \\ & |\omega| \leq \sqrt{2 \cdot Dist(v, \omega) \dot{\omega}_{dec}} \Big\} \end{aligned} \tag{7}$$

Where $\dot{v}_{dec}$ and $\dot{\omega}_{dec}$ are respectively the maximal translation and rotation deceleration. $Dist(\text{v}, \omega)$ is the distance to the closest obstacle on the corresponding curvature defined by the couple (v, ω). And V_d : dynamic window centred on the actual velocities (v_a, ω_a) and its size is defined by the robot acceleration and deceleration capabilities for both translation and rotation $(\dot{v}_{acc}, \dot{v}_{dec}, \dot{\omega}_{acc}, \dot{\omega}_{dec})$ and the time cycle (T) of the navigation:

$$\begin{aligned} (v, \omega) \in V_d \Big\{ & v \in [v_a - T \cdot \dot{v}_{dec}, v_a + T \cdot \dot{v}_{acc}], \\ & \omega \in [\omega_a - T \cdot \dot{\omega}_{dec}, \omega_a + T \cdot \dot{\omega}_{acc}] \Big\} \end{aligned} \tag{8}$$

To select the adequate command, these velocities are evaluated using an objective function. Then, the steering angle can be deduced from the selected v, ω such as:

$$\phi = \arctan\left(\frac{L \cdot \omega}{v}\right) \tag{9}$$

The used objective function in [15] considers the robot heading toward its goal, the distance to obstacles and when higher velocities are favourite. It cannot be used directly for moving obstacles avoidance where the consideration of only the actual obstacle position conducts to higher collision risk. To overcome this issue, we propose in this paper to use Extended Kalman Filters for the tracking of moving obstacles and we propose other terms in the objective function for the consideration of the predicted moving obstacle motion.

B. Extended Kalman Filter for moving obstacles tracking

In the presence of moving obstacles in the robot's environment, it is not possible to use only their positions for avoiding them but it is more useful to consider their future predicted trajectories. The eventually future position of a moving obstacle can aid the selection of efficient robot controls to avoid collision with this obstacle. For this purpose, moving obstacles detection and tracking module is proposed and described in this section. The robot is equipped with a laser rangefinder that returns a local information about the environment around the robot. Therefore, the robot configuration is used for the computation of the global positions of obstacles. The proposed method starts from raw laser measurements to the tracking of moving obstacles using the following Algorithm 1.

In the following, we describe briefly each step cited in this Algorithm.

Segmentation: a Point-Distance-Based Segmentation (PDBS) method is adopted in this work where the Euclidian distance between two consecutive scan points is compared to a threshold distance in order to decide if the two points are from the same or different objects [10].

Feature extraction: Each segment, representing the perceived part of an object by the laser rangefinder, can be decomposed to group of lines. So, the split algorithm is exploited for line extraction and followed by a line fitting step. In this last, the Total Least Squares is applied to determine the straight-line equation that minimizes the quadratic error. Some simple and efficient solutions was used to eliminate the outliers [10]. Each line can be described by their extremities and its center.

Data association: Once the objects are detected and approximated by a set of lines, the matching between lines using their parameters serves initially in the elimination of the potentially static objects (SO). All the newly perceived objects are saved in a list of appeared objects (List-AO) while the disappeared ones are saved in the list of a disappeared objects (List-DO). It is possible for a moving obstacle to be considered as an object that disappears from a place and appears in a new pose. The next Algorithm describes the proposed operations for data association.

EKF-based tracking algorithm: we consider that mobile objects are generally nonlinear systems and their tracking required the use of the Extended Kalman Filter (EKF) [16]. It is also assumed that both the motion model of moving objects and the measurement present zero-mean Gaussian white noises. Therefore, an EKF is created for

---

Algorithm 1: Moving obstacle detection and tracking

```
while the robot does not reach its goal do
    Read Laser measurments(Data);
    Global coordinate system ⟵ Data;
    Measurement segmentation (PDBS method)
      [10], [20];
    Create a list of obstacles (Segment): S^t;
    foreach Segment do
        Feature extraction (lines): Split Algorithm
          [10];
        foreach Subset ∈ Segment do
            Line fitting;
        end
        Add this object to the list S^t;
    end
    Data association: Static Obstacle (SO) and
      possible Moving Obstacle (MO) identification
      (Algorithm 2);
    foreach appeared obstacle AO do
        if AO ∈ List-MO (comparison with the
          predicted pose provided by an EKF) then
            Tracking it using the associated EKF;
        end
        else if AO is a New-MO then
            List-MO ⟵ New-MO;
            Create EKF for its tracking;
        end
        if a MO disappeared for some number of
          iterations then
            Remove it from the List-MO;
            Delete the corresponding EKF;
        end
    end
end
```

---

tracking each mobile object independently. The moving obstacle state is $X = [Xo, Yo, \Theta o]^T$. Vo is its velocity and $\Delta\theta$ is the orientation change as shown in figure 3.

The discrete-time equation (10) describes the motion of a moving obstacle:

$$\begin{cases} Xo_{k+1} = Xo_k + Vo_k \cdot \Delta T \cdot \cos(\Theta o_k) \\ Yo_{k+1} = Yo_k + Vo_k \cdot \Delta T \cdot \sin(\Theta o_k) \\ \Theta o_{k+1} = \Theta o_k + \Delta\theta \end{cases} \tag{10}$$

The measurement vector $Xm = [Xm, Ym, \Theta m]^T$ is obtained from the moving obstacle detection module that provides directly this vector. We assume that both the motion model and the measurement present zero-mean Gaussian white noises with covariance matrices P and Q respectively.

The EKF algorithm consists on doing the next two steps:

Algorithm 2: Data association

S^t: extracted segments from current measurement;
S^{t-1}: extracted segments from the previous measurement;
foreach *Seg1* $\in S^t$ **do**
  foreach *Seg2* $\in S^{t-1}$ **do**
    if *parameters(Seg1)≈ parameters(Seg2)* **then**
      SO ⟵ Seg1;
    end
  end
  if *Seg1 is not static* **then**
    List-AO ⟵ Seg1;
  end
end
Generate List-AO and List-DO;
Affine the data association:
forall *AO* $\in$ *List-AO* **do**
  forall *DO* $\in$ *List-DO* **do**
    Distance=distance(AO, DO);
    if *Distance* $\leq$ *Threshold* **then**
      AO is associated to DO;
      Remove AO from List-AO;
      Remove DO from List-DO;
    end
  end
end

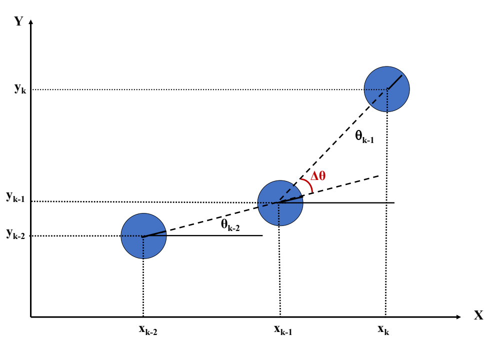

Figure 3. Motion parameters of a moving obstacle

1) *Prediction step*
State prediction using the motion model:

$$X^*_{k+1} = f(\hat{X}_k, U_k) \tag{11}$$

Prediction of the state covariance matrix:

$$P^*_{k+1} = F\hat{X}_k F^T + Q \tag{12}$$

Where F is the *Jacobian* matrix of $f()$.

2) *Correction (Filter) step*
The predicted measurement:

$$XM^*_{k+1} = X^*_{k+1} \tag{13}$$

The KF gain

$$K = P^*_{k+1}(P^*_{k+1} + P_{k+1})^{-1} \tag{14}$$

Take the new observation: XM_{k+1}
The correction of the predicted state:

$$\hat{X}_{k+1} = X^*_{k+1} + K(XM_{k+1} - XM^*_{k+1}) \tag{15}$$

Correction of the Predicted covariance matrix:

$$\hat{P}_{k+1} = P^*_{k+1} + KP^*_{k+1} \tag{16}$$

Each created EKF for motion tracking of a moving obstacle provides an information about its motion. This information can be exploited for the prediction of collision risk between the mobile robot and the moving obstacle as explained in the next section.

3. EKF-DWA For Mobing Obstacle Avoidance

One should put in mind that the possible commands are the set of velocities pairs within the intersection between the search space, the admissible velocities and the dynamic window. To simplify the DWA algorithm and avoid the building of grid map, we've reformulated the following algorithm. We suggested in this paper a new objective function (17) for velocities evaluation that consider both the collision risk with static or moving obstacles. Especially, two sub-functions are introduced: **Time_C** and **Direct_C**. The first one favors commands that lead to a large difference between the moments of traversing this region by the robot and the moving obstacle. The second term avoids the commands that make the robot and the moving obstacle get aligned in order to conduct the robot away from the region of collision risk.

$$\begin{aligned} G(v,\omega) =& \alpha \cdot Heading(v,\omega) + \beta \cdot dist(v,\omega) + \\ & \gamma \cdot Velocity(v,\omega) + \delta \cdot Time\_C(v,\omega) + \\ & \sigma \cdot Direct\_C(v,\omega) \end{aligned} \tag{17}$$

The terms *Heading*, *dist* and *Velocity* are the same as the ones used in the original DWA. The remainder of this section describes briefly these terms and with more detail the new terms.

Heading term: this parameter awards the curvature arcs that leads the robot towards the goal. To calculate this term, we compute the predicted robot orientation θ', for the couple (v, ω), according to the goal position. Then this term is calculated using the equation (18).

$$Heading(v,\omega) = 1 - \frac{|\theta'|}{\pi} \tag{18}$$

It has a maximum for the commands that lead to a good alignment of the robot with the goal (θ' close to $0rd$). It decreases symmetrically for coming null if the goal is

Algorithm 3: DWA algorithm without grid map (One cycle)

Initialize: $G_{max} = 0$;
for $v \in [v_a - T\dot{v}_{dec}, v_a + T.\dot{v}_{acc}]$ **do**
  Calculate $\omega_{max} = \frac{v \cdot \tan\phi_{max}}{L}$;
  for $\omega \in [\omega_a - T \cdot \dot{\omega}_{dec}, \omega_a + T \cdot \dot{\omega}_{acc}]$ **do**
    if $(v,\omega) \in V_s \left\{ v \in [0, v_{max}], \omega \in [-\omega_{max}, \omega_{max}] \right\}$ **then**
      Calculate $dist(v,\omega)$;
      if $(v,\omega) \in V_{adm}$ **then**
        Calculate $G(v,\omega)$;
        if $G(v,\omega) > G_{max}$ **then**
          $G_{max} = G(v,\omega$;
          $(v_c, \omega_c) = (v,\omega)$;
        end
      end
    end
  end
end
Applied the command $(v_a, \omega_a) = (v_c, \omega_c)$;

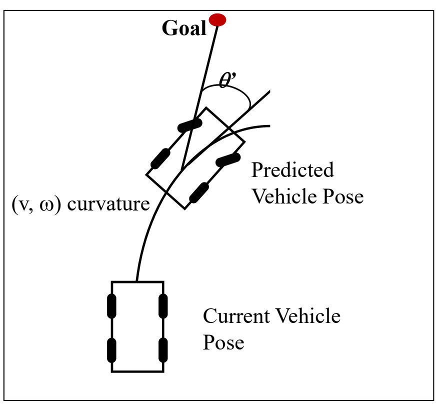

Figure 4. Illustration of the heading term

behind the robot as shown in figure 4. This term is very important and has the highest weight; it is like a force that conduct the robot toward the goal.

Distance to obstacles term: this parameter favors the commands that guide the robot far from the static obstacles. It is given, for each pair of commands, by (19) where d_{col} is the minimal distance between the robot and the obstacle if it takes the corresponding curvature. In this approach, the rectangular shape of the robot is considered and an analytic method is used [15][16][21] to determine this distance. d_{max} is the maximal distance between the robot and obstacles to be considered. For distances upper than dmax, the robot is considered in secured situation and it shall not trying to avoid a very far obstacle.

$$dist(v,\omega) = \frac{d_{col}}{d_{max}} \tag{19}$$

We defined dmax as the twice of the needed distance to stop the robot when it has a maximal velocity. This distance is considered as sufficient for the robot for avoiding an obstacle or to stop before hitting it.

Velocity term: the DWA was developed for robot navigation at high speed in a way that the robot moves at high speed if it is far from the goal. This term allows also a graduated deceleration for preparing the robot to stop when approaching the goal. It is given by (20). The distance between the predicted robot position and the goal ($Dist\_r\_g$) is compared to a threshold distance D_{th}. If this distance is less than D_{th}, the robot shall prepare to stop and shall select small velocities. In the situation where the robot is far from object, it can move at higher speed.

$$Velocity(v,\omega) = \begin{cases} \frac{|v|}{v_{max}}, & \text{if } Dist\_r\_g > D_{th} \\ 1 - \frac{|v|}{v_{max}}, & \text{otherwise.} \end{cases} \tag{20}$$

Time to collision: as the aim of this work is to modify and improve DWA to avoid moving obstacles, we proposed to consider the time to collision risk as a term in the objective function. When robot moves at high speed and moving objects can move also at high speed with non-linear equation, the time to collision has more signification and importance rather than distance to collision. Therefore, we proposed this term that favors the choice of the commands that conduct the robot away from the collision region (figure 5). Using the information delivered by the Extended Kalman Filter about the object motion, the region of eventual collision is the intersection between the object trajectory and the curvature arc of the robot defined by (v,ω). This term is defined by the equation (21).

$$Time\_C(v,\omega) = \begin{cases} \frac{\max(t_{mo},t_{mr})}{\min(t_{mo},t_{mr})}, & \text{if } |t_{mo} - t_{mr}| > t_{min} \\ 1 - \frac{\max(t_{mo},t_{mr})}{\min(t_{mo},t_{mr})}, & \text{otherwise.} \end{cases} \tag{21}$$

Where: t_{mo} and t_{mr} are the times taken by the moving obstacle and the mobile robot respectively to reach the collision region. t_{min} is the minimal secure difference between t_{mo} and t_{mr} to avoid the collision. This time can be influenced by the robot and moving obstacle size but in this work, it was chosen empirically.

If the difference between these times is greater than t_{min}, the collision risk is weak and the corresponding curvature is preferred. To be in higher security, the couple (v,ω) that mains to a higher ratio between t_{mo} and t_{mr}. The commands that lead to a small difference between t_{mo} and t_{mr} are

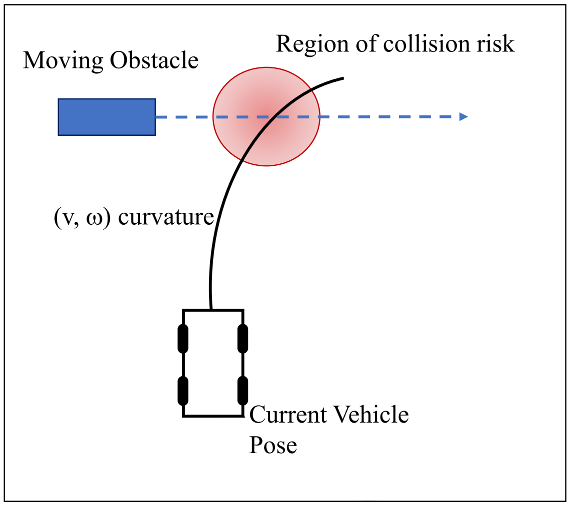

Figure 5. Predicted Region of collision

penalized.

Both t_{mo} and t_{mr} can be determined as a ratio between the distance and the velocity of the moving obstacle and the robot respectively.

Direction of collision: the risk of collision face to face or face to back between the robot and the moving obstacles cannot be avoided only by the consideration of time to collision. This risk is particularly dangerous for car-like robot with limited steering angle rather that other types of robots (holonomic robot, Differential and synchrodrive robots) that can change easily their orientation. Hence, we introduced a new term **Direct_C** to favors the velocities that lead to a considerable gap between the directions of motion of the mobile robot and moving obstacle and given by the equation (22). Fig. 6 shows a face-to-face collision risk between the robot and a moving obstacle. In the region of collision risk, if the predicted direction of the mobile robot and mobile object are very close, the term $Direct\_C(v,\omega)$ shall give a very lower evaluation (close to zero). In the other hand, the velocities that lead to a large difference between these angles are favored.

$$Direct\_C(v,\omega) = \begin{cases} \left|\frac{\max(\theta_{mo},\theta_{mr})}{\min(\theta_{mo},\theta_{mr})}\right|, & \text{if } ||\theta_{mo}| - |\theta_{mo}|| > \theta_{min} \\ 1 - \left|\frac{\max(\theta_{mo},\theta_{mr})}{\min(\theta_{mo},\theta_{mr})}\right|, & \text{otherwise.} \end{cases} \tag{22}$$

Where θ_{mo} is the orientation of the moving obstacle and θ_{mr} is the predicted robot orientation for the couple of velocities (v,ω) and θ_{min} is the considered threshold to distinguish between aligned and no aligned robot-obstacle. This term penalizes the velocities that conduct to face-to-face or face to back risk of collision. During the robot motion, this term help the robot to change gradually its direction and avoid collision with obstacles.

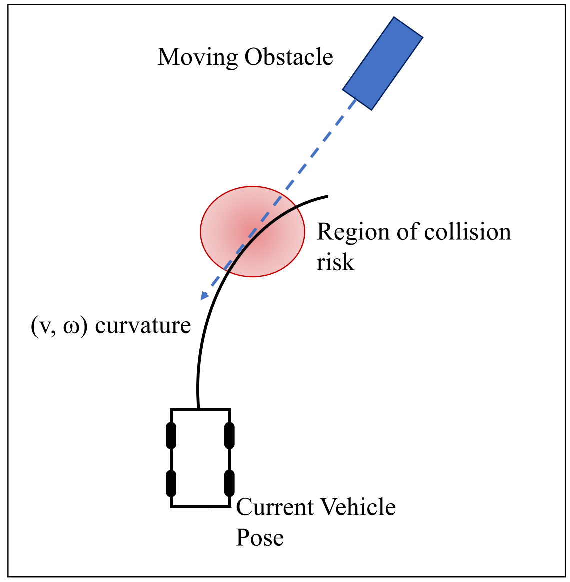

Figure 6. Prediction of face-to-face risk of collision

4. Simulation Results

Several simulation tests are carried out to select the weights parameters of the objective function and to verify the performance of the proposed EKF-DWA approach for robot navigation in dynamic environments. The test procedure of this approach is a two-stage where simple and complex scenarios are considered. The blue and red circles indicate the ground robot's initial and goal positions, respectively. We resume in Table I the simulation settings (used parameters). These given cost function weights are the best ones according to our tests.

TABLE I. Simulation Parameters

Simulation Parameters		
Robot Characteristics	V_{max}	$1.5m/s$
	ϕ_{max}	$18°$
	L	$1.2m$
	$\lvert\dot{v}_{acc}\rvert = \lvert\dot{v}_{dec}\rvert$	$2m/s^2$
	$\lvert\dot{\omega}_{acc}\rvert = \lvert\dot{\omega}_{dec}\rvert$	$0.75rd/s^2$
	T	$250ms$
Cost Function Weights	α	1.2
	β	0.1
	γ	0.1
	σ	2.5
	δ	2

A. Simple scenarios

As explained previously, the aim of this paper is to improve DWA for moving obstacles avoidance by a car-like robot. In this section, two scenarios are presented where the robot moves in an environment with only moving obstacles. In the first one, the robot will avoid three moving obstacles. The simulation illustrated in figure 7 shows how the robot avoid the first green obstacle (Figure 7(a)) and the second one (figure7(b) and 7(c)). The second obstacle is a big brown object that passes between the robot and its goal. In this test, the obstacle has been detected and tracked sufficiently earlier. While the third obstacle (pink) appears to be, for the first time, very close to the robot as shown in the subfigure (7(d)).

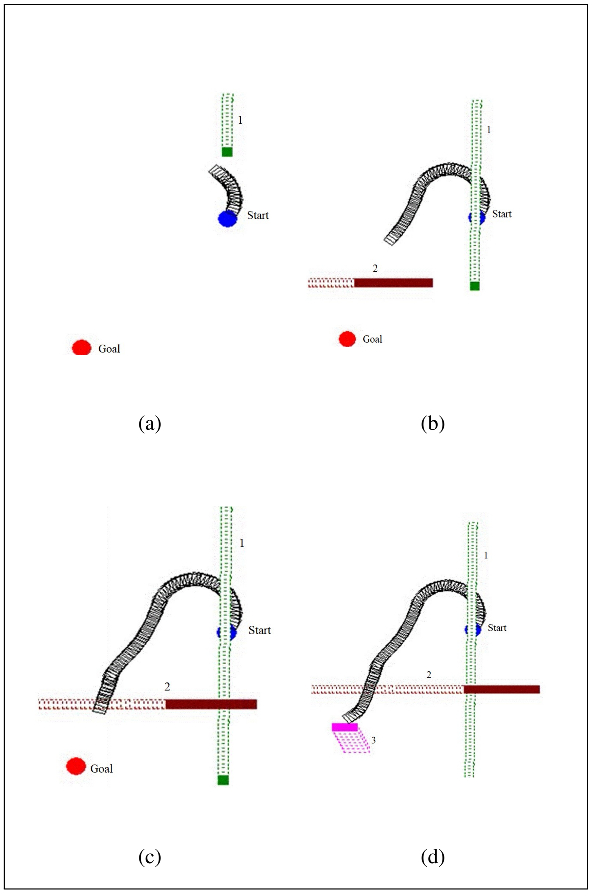

Figure 7. Simulation results: Simple Scenario 1
(a) The appearance of the first moving obstacle (1: green), **(b)**the appearance of the second moving obstacle (2: brown big obstacle), **(c)** how the robot avoids collision with obstacle, **(d)** the appearance of the third obstacle (3: pink) in a close position to the robot. The robot stops because it cannot avoid this obstacle (No sufficient time to react).

This situation cannot be avoided by the robot because of its kinematics constraints as explained in the subsection 2.A and equations (2 and 3). The size of the dynamic window depends also on the robot capabilities of acceleration and deceleration. Therefore, in this situation, the moving obstacle appears very close to the robot and the evaluation of all the possible commands leads to collision risk. The created EKFs to track the detected moving obstacles can provide a proper estimation of their future position and as a consequence a good estimation of the position of the region of collision risk. The improved DWA can then select the appropriate command for avoiding the two moving obstacles thanks to the introduced terms in the DWA cost function ($Time\_C$ and $Direct\_C$). Fig. 8 represents the second simple scenario where the robot moves in a cluttered environment composed of five moving obstacles. In this experiment, all the obstacles are directed toward the robot with different angles. In figure 8, the subfigures 8(a), 8(b), 8(c), 8(d) and 8(e) are selected to show the robot reaction for avoiding each obstacle. The last subfigure 8(f) presents the final robot trajectory.

It is important to notice that the robot's behavior in presence of moving obstacles is the result of the combination of the evaluated commands using the objective function. For this reason, the robot's reaction depends on the robot's actual velocities and the sensor measurement. From these figures (7 and 8), we can consider that the robot behavior in presence of moving obstacle is very acceptable.

B. Complex scenario

To be more close to real situation, we conduct other simulation tests in environment with both static and moving obstacles. In this subsection, an example of these tests is presented (figure 9). In this scenario, the robot moves toward its goal and meets in addition of the static obstacle, seven moving obstacles. It can reach successfully its goal without collision avec obstacles. The extended DWA for moving obstacles avoidance by car-like robot using the motion prediction of dynamic obstacles shows a good quality of robot behavior. The failures cases of the proposed approach are caused by robot perception and dynamics. We remember that the principal source of car-like robot navigation complexity is its mechanical constraint (steering angle limit) that reduces the possible instantaneous commands. This leads also to the risk of the non-visibility of some obstacles when the sensors covered only the robot front. Face to this situation, our approach stops the robot. While this solution is very acceptable in static environments, it can be dangerous where a moving obstacle hit the robot.

5. Conclusion

In this paper, we have presented a new approach for car-like navigation at high speed in dynamic environments. It tackles the avoidance of collision with both static and moving obstacles. This approach is based on the well-known DWA that has been extended for taking into account the car-like robot constraints such as its non-holonomy and the limit of its steering angle. It has been modified for avoiding moving obstacles that are detected and tracked using EKF. Each detected moving obstacle is associated with an appropriate EKF to track its motion and predict its trajectory. The outputs of the created trackers (EKFs) are used for computing two new terms added to the DWA objective function. These terms are proposed to maximally avoid face to face collision with obstacles and to increase the gap between the time taken by vehicle and obstacle for reaching the region of collision risk. Although the available approaches [17], [18], [19] use grid map for

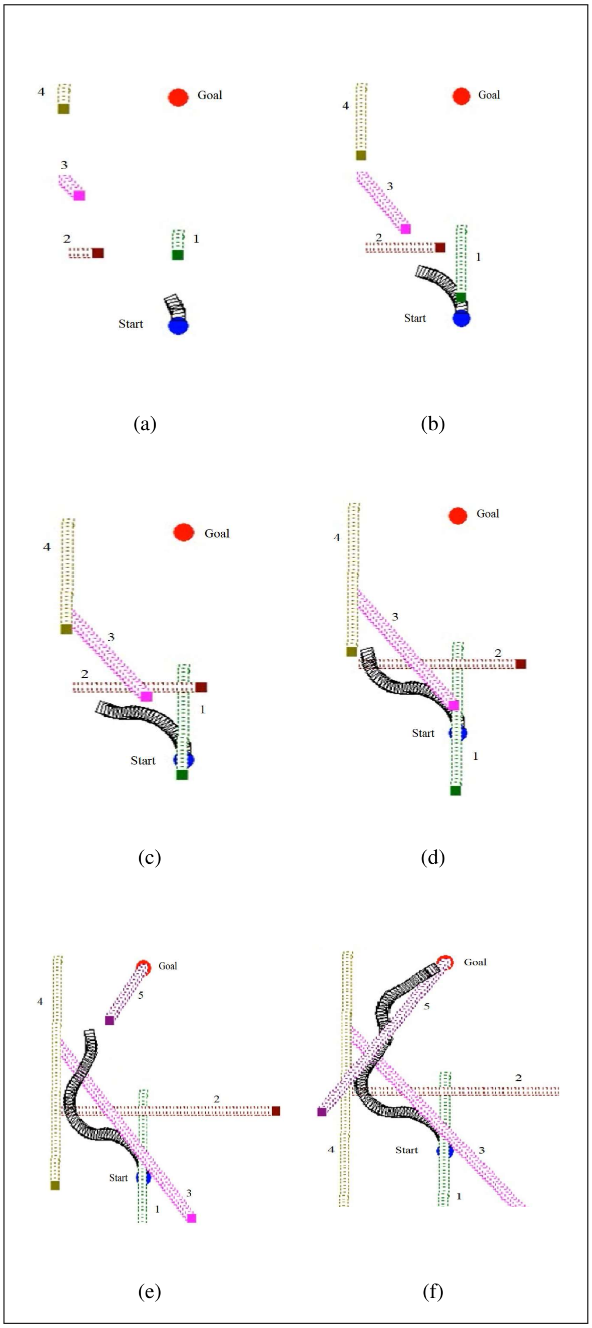

Figure 8. Simulation results: Simple Scenario 2
(a) The appearance of the four moving obstacles, the robot change its direction to avoid the first obstacle (1: green), **(b)** the robot avoids the second obstacle (2: brown), **(c)** the robot avoids the third obstacle (3: pink) and changes its direction to align with the goal, **(d)** the robot avoids the fourth moving obstacle, **(e)** the robot avoids a face-to-face collision risk with the five moving obstacle, **(f)** the robot achieves its goal.

presenting static and moving obstacles and exploit this information for the evaluation of the possible commands

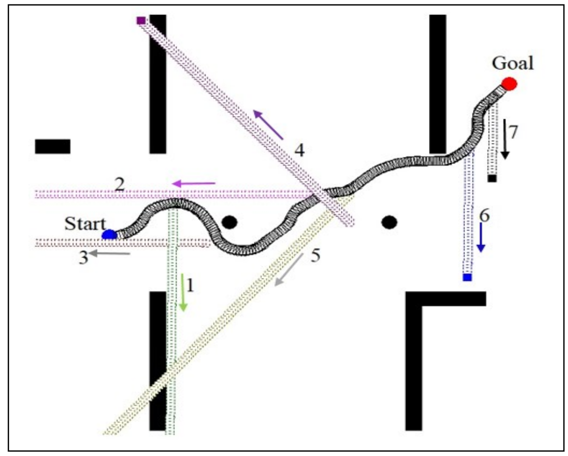

Figure 9. Simulation results: Complex scenario

using an objective function, our approach is completely reactive and create only a list of moving objects for their tracking. Each disappeared object for some number of iterations is removed from this list and the associated EKF is deleted. Our approach differs also from the others by the quality of the used algorithm for the prediction of moving obstacles motion. We used EKF known by its performance for the tracking of random and nonlinear motion. Various simulations have been conducted to analyze the robot's reaction toward moving obstacles and also static ones. This approach solved the problem of moving obstacle avoidance for the more complicated type of wheeled robots. The motions constraints of vehicle represent the principal reason for the difficulty of the problem of vehicle navigation in dynamic environments. Various approaches have been proposed for other types of robot-like differential ones with less constraints and that cannot be directly extended for vehicle navigation. The study of the limits of the presented approach shows the difficulty of avoiding big moving object especially if it appears for the first time close to the vehicle. This situation is caused by the vehicle dynamics and constraints and also by the laser observed region (in the robot front in our case). To overcome this limit, we propose, a future work, the consideration of back motion and its integration as a solution when all forward motions can lead or result in collision with moving obstacles. Accordingly, it is also important to perceive the 360° around the robot. Also, this system can be reinforced by sensors which improve the range and the scope of the robot's vision in order to improve the prediction.

References

[1] K. Karpe, A. Chatterjee, P. Srinivas, D. Samiappan, K. Ramamoorthy, and L. Sabattini, "Sprinter: A discrete locomotion robot for precision swarm printing," in *IEEE International Conference on Advanced Robotics (ICAR)*, 2019, pp. 733–738.

[2] M. Bakhale, V. Hemalatha, S. Dhanalakshmi, R. Kumar, and M. Siddharth Jain, "A dynamic inertial weight strategy in micro pso for

swarm robots," *Wireless Personal Communications*, vol. 110, pp. 573–592, 2020.

[3] M. G. H. Nampoothiri, B. Vinayakumar, Y. Sunny, and R. Antony, "Recent developments in terrain identification, classification, parameter estimation for the navigation of autonomous robots," *SN Applied Sciences*, vol. 3, 2021.

[4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, pp. 90–98, 1986.

[5] C. V. B. Murthy, M. L. Shri, S. Kadry, and S. Lim, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4, pp. 23–33, 1997.

[6] I. Ulrich and J. Borenstein, "Vfh+: reliable obstacle avoidance for fast mobile robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 1998, pp. 1572–1577.

[7] P. Ogren and N. Leonard, "A convergent dynamic window approach to obstacle avoidance," *IEEE Transactions on Robotics*, vol. 21, pp. 188–195, 2005.

[8] H.-T. Chiang, N. Malone, K. Lesser, M. Oishi, and L. Tapia, "Path-guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments," in *In IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2347–2354.

[9] C. Kim, Y. Kim, and W.-S. Ra, "Modified 1d virtual force field approach to moving obstacle avoidance for autonomous ground vehicles," *Journal of Electrical Engineering and Technology*, vol. 14, pp. 1367–1374, 2019.

[10] K. Rebai, A. Benabderrahmane, O. Azouaoui, and N. Ouadah, "Moving obstacles detection and tracking with laser range finder," in *In IEEE International Conference on Advanced Robotics (ICAR)*, 2009, pp. 1–6.

[11] M. Dekan, D. František, B. Andrej, R. Jozef, R. Dávid, and M. Josip, "Moving obstacles detection based on laser range finder measurements," *International Journal of Advanced Robotic Systems*, vol. 15, pp. 1–18, 2018.

[12] R. immons, "The curvature-velocity method for local obstacle avoidance," in *In IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4, 1996, pp. 3375–3382.

[13] C. Shi, Y. Wang, and J. Yang, "A local obstacle avoidance method for mobile robots in partially known environment," *Robotics and Autonomous Systems*, vol. 58, pp. 425–434, 2010.

[14] H. Q. T. Ngo, V. N. Le, V. Thien, T. P. Nguyen, and H. Nguyen, "Develop the socially human-aware navigation system using dynamic window approach and optimize cost function for autonomous medical robot," *Advances in Mechanical Engineering*, vol. 12, pp. 1–17, 2020.

[15] K. Rebai, O. Azouaoui, M. Benmami, and A. Larabi, "ar-like robot navigation at high speed," in *In IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 2007, pp. 2053–2057.

[16] D. A. de Lima and G. A. S. Pereira, "Navigation of an autonomous car using vector fields and the dynamic window approach," *Journal of Control, Automation and Electrical Systems*, vol. 24, pp. 106–116, 2013.

[17] M. Seder, K. Macek, and I. Petrovic, "An integrated approach to real-time mobile robot control in partially known indoor environments," in *Annual conference of the IEEE Industrial Electronics Society*, 2005.

[18] E. J. Molinos, Ángel Llamazares, and M. Ocaña, "Dynamic window based approaches for avoiding obstacles in moving," *Robotics and Autonomous Systems*, vol. 8, pp. 112–130, 2019.

[19] M. Missura and M. Bennewitz, "Predictive collision avoidance for the dynamic window approach," in *In IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8620–8626.

[20] C. Premebida and U. Nunes, "Segmentation and geometric primitives extraction from 2d laser range data for mobile robot applications," *Robotica*, pp. 17–25, 2005.

[21] K. Arras, J. Persson, N. Tomatis, and R. Siegwart, "Real-time obstacle avoidance for polygonal robots with a reduced dynamic window," in *In IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, 2002, pp. 3050–3055.

Karima REBAI Karima REBAI received her Doctorate es Sciences from the University of Science and Technology- Houari Boumedien (USTHB) Algiers in 2015, her Magister's degree in Robotics and Process control from the same university and the center of development of innovative technologies (CDTA). Since 2011, she is with the department of Electrical engineering and industrial informatics of the higher national school of technology (ENST). Her main research activities are in the areas of robotics, autonomous robots and artificial intelligence.

Samira BOUCHAMA Samira BOUCHAMA has received the Electrical engineering degree from the National Polytechnic School (ENP) of Algiers. She achieved the Master's and Doctorate degree from the same school (Signal and Communications Laboratory) in 2014. She has worked at the Research Center on Scientific and Technical Information (CERIST) of Algiers and now she is working as Teacher/Researcher at the Higher National School of Technology (ENST) in the field of image and video watermarking and Digital Signal Processing.