OBSTACLE AVOIDANCE IN LOCAL NAVIGATION

Daniel Castro*, Urbano Nunes†, António Ruano*

*Institute of Systems and Robotics – ISR
Faculty of Science and Technology
University of Algarve
Campus de Gambelas
8000-117 Faro, Portugal
fax: +351 289 819 403
e-mail: dcastro@ualg.pt, aruano@ualg.pt

†Institute of Systems and Robotics – ISR
Electrical and Computer Eng. Department
University of Coimbra, Polo II
3030-290 Coimbra, Portugal
email: urbano@isr.uc.pt

Keywords: Obstacle detection, object tracking, local navigation.

Abstract

A reactive navigation system for an autonomous non-holonomic mobile robot in dynamic environments is presented. A new object detection algorithm and a new reactive collision avoidance method are presented. The sensory perception is based in a laser range finder (LRF) system. Simulation results are presented to verify the effectiveness of the proposed local navigational system in unknown environments with multiple moving objects.

1 Introduction

Indoor mobile robots may need to carry out missions in hazardous and/or populated environments. A typical application is to assist human beings in indoor environments, like offices. These robots should be able to react appropriately to unforeseen changes in the environment or to unpredictable objects blocking their trajectories. Local navigation techniques are the ones responsible to achieve these reactive issues. These techniques are sensory-based approaches, using only local sensory information, or an additional small fraction of the world model.

Some of the most popular reactive collision avoidance methods are based on artificial potential fields [10], where the robots steering direction is determined assuming that obstacles assert repulsive forces on the robot and the goal asserts attractive forces. These methods are extremely fast and they typically consider only a small subset of obstacles near the robot. An extended version of this approach was introduced in [9]. In the vector field histogram approach [1] an occupancy grid representation is used to model the robots environment. The motion direction and velocity of the robot are computed from the transformation of the occupancy information into a histogram description. All these methods calculate the desired motion direction and steering commands in two different steps, which is not acceptable in a dynamic point of view.

The curvature-velocity method [13] and the dynamic window (DW) approach [6] are based on the steer angle field approach [4]. It is assumed that the robot moves in circular paths, and the search of motion commands is performed directly in the space of translational and rotational velocities. In both approaches the robot kinematics and dynamic constraints are considered by constraining the

search space to a set of admissible velocities. They show good results for obstacle avoidance at high velocities (60 cm/s ~ 95 cm/s) although suffer of the local minima problem.

The DW approach was extended to use a map in conjunction with sensory information [7] to generate collision free motion. A Bayesian approach to obstacle avoidance was linked with global path planning [8]. These approaches require a previous knowledge of the environment for the execution of the motion command.

A global dynamic window approach was proposed in [2] to non-holonomic and holonomic robots. This approach allow robust execution of high-velocity, go-to-goal, local minima free, reactive motion for mobile robots in unknown and dynamic environments, combining a wave-propagation technique [11] starting at the goal position with the DW approach.

With obstacle motion prediction, a different framework called the velocity obstacle (VO) approach was proposed in [5], to determine potential collisions and compute the collision-free path of a robot moving in a time varying environment.

This paper is organised as follows. In Section 2, the system architecture is presented. In Section 3, the dynamic window approach is briefly elaborated. Section 4, presents the proposed LRF based obstacle detection and object tracking (perception of object's motion) algorithms. Section 5 introduces the new reactive local navigation and obstacle avoidance methods. In Section 6, simulation results are shown with discussion. Finally, concluding remarks are given in Section 7.

2 System architecture

Our system is based in a hierarchical architecture, with three layers: on the bottom level is the Steering Control system, which is responsible for velocity control and motor driving signals; next is the Reactive Local Navigation system, the one we address in this work, responsible for all local needs, like obstacle detection, obstacle classification, reactive local navigation and obstacle avoidance; on the top is the Strategic Navigation system, which has the ability to coordinate and schedule all robot processes in order to achieve the robot mission. This high level layer is compose by the following subsystems: Global Path Planning, Cartographer (building and reading maps), Interface and Communications, and the Mission Planner. The Mission Planner is where all high level decisions are taken.

3 Dynamic window approach

The DW approach [6] is a sensor-based collision avoidance technique that takes into account the kinematics and dynamic constraints of differential and synchro-drive robots. Kinematics constraints are taken into account by directly searching the velocity space *Vs* of the robot. This space is defined by the set of tuples (v, ω) of longitudinal velocities v and rotational velocities ω that are achievable by the robot.

Among all velocity tuples the ones that are selected are those that allow the robot to stop before it reaches a obstacle, given the current position, current velocity and the acceleration capabilities of the robot. These velocities are called admissible

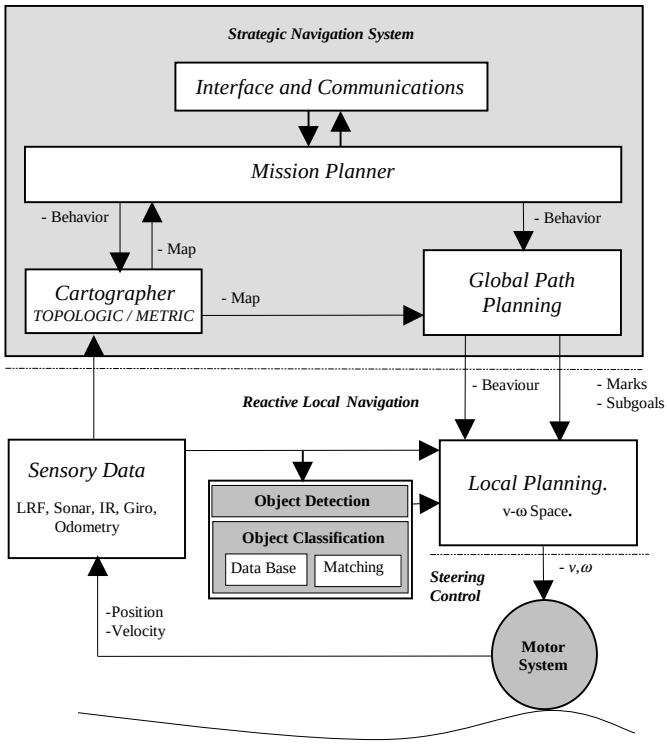


Figure 1: Navigation system architecture.

velocities. The set *Va* of admissible velocities is defined as

$$Va = \left\{(v,\omega),\quad v \le \sqrt{2.dist(v,\omega).\dot{v}_b} \wedge \omega \le \sqrt{2.dist(v,\omega).\dot{\omega}_b}\right\} \quad (1)$$

The function $dist(v,\omega)$ represents the distance to the closest obstacle on the curvature defined by the velocity (v,ω), measured by $r.\gamma$, where $r=v/\omega$ is the radius of the circular trajectory and γ the angle defined in Figure 2. The accelerations for breakage are $\dot{v}_b$ and $\dot{\omega}_b$.

Introducing a rectangular DW, we could reduce the search space to all velocities that can be reached within the next time interval, according to the dynamic limitations of the robot, given its current velocity and its acceleration capabilities. The dynamic window *Vd* is defined as

$$Vd = \{(v,\omega),\quad v \in [v_c - \dot{v}.h, v_c + \dot{v}.h] \quad \wedge \quad \omega \in [\omega_c - \dot{\omega}.h, \omega_c + \dot{\omega}.h]\} \quad (2)$$

where *h* is the time interval during which accelerations $\dot{v}$ and $\dot{\omega}$ will be applied and (v_c,ω_c) is the current velocity.

To determine the next motion command all admissible velocities within the dynamic window are considered, forming the resulting search space *Vr* defined as $Vr=Vs \cap Va \cap Vd$. Among those a velocity is chosen that maximises a certain objective function (linear combination of three functions) where the alignment of the robot to a goal position, the distance to the closest obstacle and its velocity could be considered, as in the following expression:

$$G(v,\omega) = Max(\alpha.head(v,\omega) + \beta.dist(v,\omega) + \delta.vel(v,\omega)) \quad (3)$$

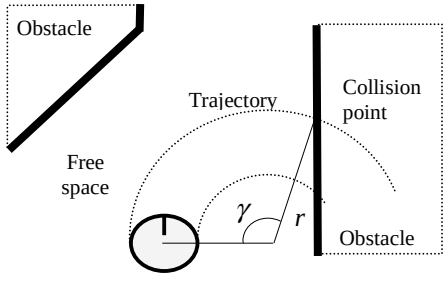


Figure 2: Function $dist(v,\omega)$ computation. The calculation is done for a predefined array of radius.

The function $head(v,\omega) = 1 - |\theta|/\pi$, where θ is the angle between the direction of motion and the goal heading, is maximal to trajectories that are directed towards the goal position. For a realistic measure of the target heading we have to consider the dynamics of the rotation, therefore θ is computed at the predicted position that the robot will reach when exerting maximal deceleration after the next interval. The function $vel(v,\omega) = v/v_{\max}$ is used to evaluate the progress of the robot on their trajectory.

The three components of the objective function are normalised to the interval [0,1]. Parameters α, β and δ are used to weight the different components. Their values are crucial to the performance of the method and robot behaviour.

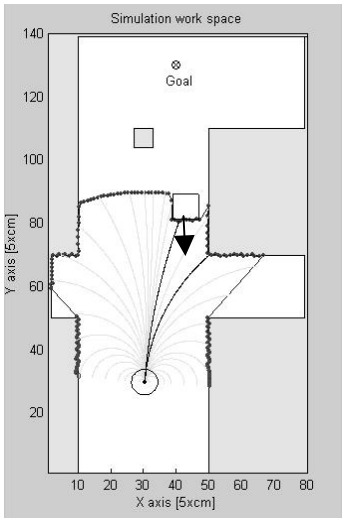


Figure 3: Robot workspace simulation.

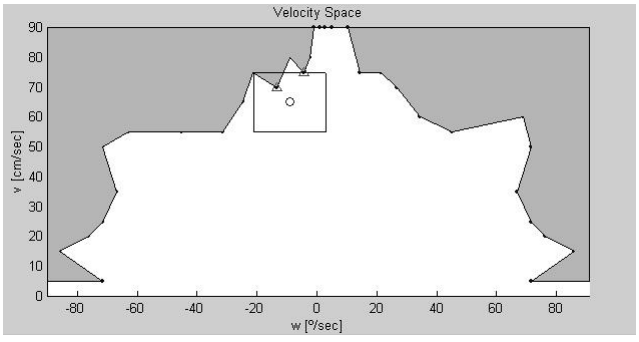


Figure 4: Velocity space. The gray area represents the non-admissible velocity tuples. The rectangle represents the dynamic window. The two triangles are the building left corner and the object, represented by dark gray radius in Figure 3.

Using this approach, robust obstacle avoidance behaviors have been demonstrated at high velocities [6]. However, since the dynamic window approach only considers goal heading and no connectivity information about free space, it is still susceptible to local minima. Figures 3 and 4 show for the same time instante, a robot scan with a moving object and the respective velocity space, with a DW for $\dot{v}$ = 50 cm/s^2 and $\dot{\omega}$ = 60 º/s^2.

4 LRF based obstacle detection

Our object detection system is laser range finder based, which detects dynamic activity from two consecutive laser scans. Object detection is achieved by the segmentation of one or two consecutive scan sets, and vectorization of the segments found. Object tracking is also possible from the object classification procedure presented in this paper. It can be combined with the DW reactive navigational approach (in section 5) as shown in Figure 5.

A typical laser range finder sensor has a view angle of about 180°, scanned with an angular resolution of 0.25°-1°, a distance range from 10 m (indoor devices) to more than 100 m (outdoor devices), and an accuracy of the distance measurement from

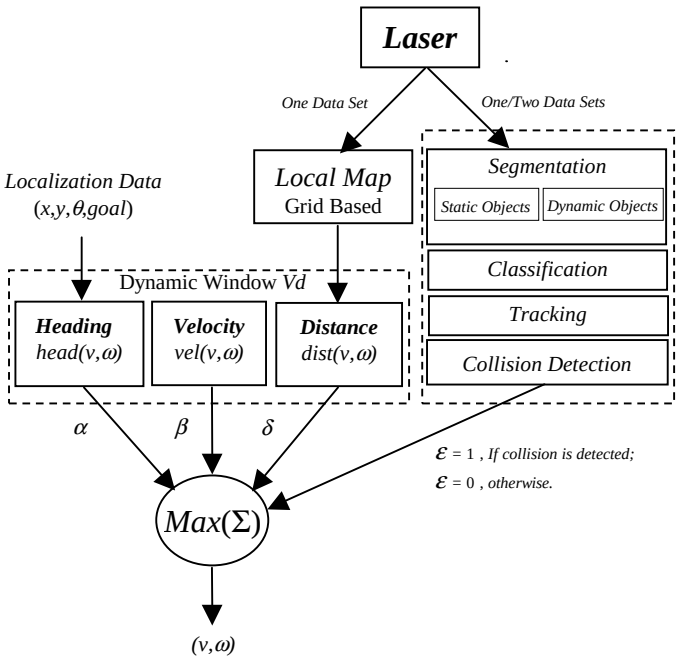


Figure 5: LRF based obstacle detection system.

±1mm to ±5cm, depending all these specifications of the sensor operational mode and model. As an example, the SICK LMS 200-30106 indoor model [12] has a scanning frequency of 75Hz, ±15 mm / ±4 cm accuracy values for distance ranges of [1 m to 8 m] / [8 m to 20 m], 10 mm resolution, and an angular resolution of 0.5° / 1° for 180° scan.

Our object detection method is quite similar to the Dietmayer et al. [3] method. Some modifications were introduced in the data preparation, before the segmentation, and in the segments filtration process.

In this work the LRF has a scan angle of 180°, with an angular resolution $\Delta\alpha$. The laser range data measured in the instant k is represented by the set LRF_K, of N data points P_i, with angle α_i and distance value r_i.

$$LRF_k = \left\{ P_i = \begin{pmatrix} \alpha_i \\ r_i \end{pmatrix}, \quad i \in [0, N] \right\} \tag{4}$$

4.1 Data analysis

The set of scan points that have some dynamic activity between two consecutive measurements is represented by $SEGM_k$. The point selection is done by the calculation of an error value for the measurements of the same scan point in two different scans. A constant ΔP is used as an error threshold. Figure 6, shows two consecutive laser scans, for two different robot positions, (LRF_{k-1}) and (LRF_k). The set of scan points can be determined by,

$$SEGM_k = \left\{ P_i \in LRF_k \wedge P_j \in LRF_{k-1} : \left| P_i - P_j \right| > \Delta P,\ k > 1 \right\} \tag{5}$$

with

$$i = \arctan\left(r_j . \sin\left(\alpha_j\right) - \left(y_k - y_{k-1}\right), \; r_j . \cos\left(\alpha_j\right) - \left(x_k - x_{k-1}\right) \right) - \Delta P_\theta \tag{6}$$

and

$$\begin{cases} \omega > 0, & j \in [0 + \Delta P v_{ini} + \Delta P_\theta, 180] \\ \omega = 0, & j \in [0 + \Delta P v_{ini}, 180 - \Delta P v_{end}] \\ \omega < 0, & j \in [0, 180 - \Delta P v_{end} + \Delta P_\theta] \end{cases} \tag{7}$$

The angular displacements

$$\Delta Pv_{ini} = \arctan(v.h, r_i), \quad i = 0 \qquad (8)$$

$$\Delta Pv_{end} = \arctan(v.h, r_i), \quad i = 180 \qquad (9)$$

and

$$\Delta P_\theta = (\theta_2 - \theta_1) \qquad (10)$$

are introduced to compensate the robot motion trajectory. The sampling rate is given by *h*. The reference of the angular velocity ω is counter clockwise. Figure 7 shows the three possible robot motion movements in each instant. The gray areas are the ones where scan points cannot be compared by successive scan data. Parameter ΔP_θ is associated to circular motion (v=0,ω≠0), see Figure 7a). In Figure 7b), related with the forward motion (v≠0,ω=0), usually ΔPv_{ini} and ΔPv_{end} are different due to the robot position in space. ΔPv_{ini}=ΔPv_{end} only occurs for the robot moving in the middle of two wall, parallel to them. The last situation (see Figure 7c)) is the most common one for a non-holonomic robot. In this last case, both motion parameters are taken into account.

4.2 Segmentation

The segmentation process is based on the computation of the distance between two consecutive scan points, calculated by

$$d(P_i, P_{i+1}) = \left\| P_{i+1} - P_i \right\| = \sqrt{r_{i+1}^2 + r_i^2 + 2.r_{i+1}.r_i.\cos\Delta\alpha} \qquad (11)$$

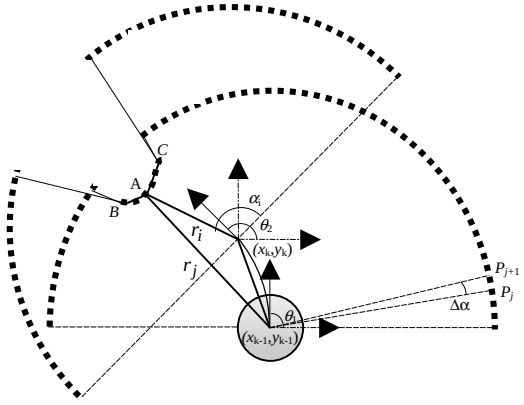


Figure 6: Figure shows two consecutive robot positions for a robot motion with linear and rotational components (v≠0,ω≠0).

If the distance is less than the threshold

$$d(P_i, P_{i+1}) \le C_0 + C_1 \min\{r_i, r_{i+1}\} \qquad (12)$$

with

$$C_1 = \sqrt{2(1 - \cos\Delta\alpha)} \qquad (13)$$

the point P_{i+1} belongs to the same segment as point P_i. The constant C_0 allows an adjustment of the algorithm to noise and strong overlapping of pulses in close range [3]. The constant C_1 is the distance between consecutive points in each scan.

This distance is also used to search the scan data, on the left and on the right, for points not selected in data analysis. This operation allows the detection of points with different dynamic activity from the rest of the points just selected, like edge segment corners. Special tests could be done to combine segments that probably belong to the same object.

4.3 Filtration

Pairs and isolated scan points are rejected. Segments have a minimum size established in the beginning (3 points). It is also a simple way of noise filtering.

4.4 Object classification

After the basic segmentation process, the segments founded need to be described by a set of points {*A*,*B*,*C*}, where {*A*} is the closest point to the sensor

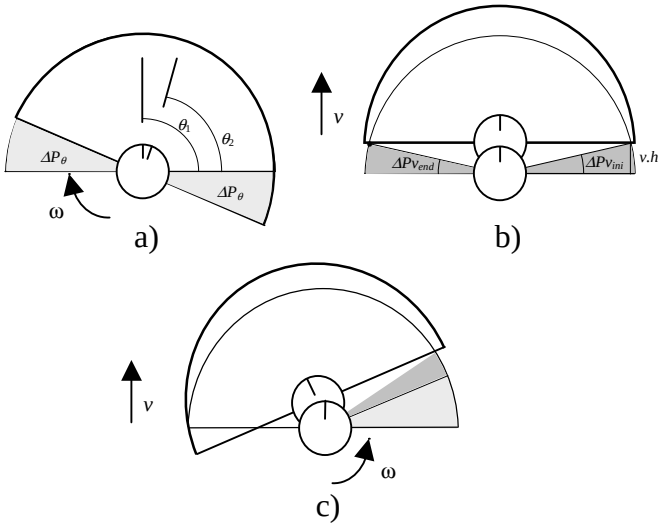


Figure 7: Typical robot motion movements, a)- Circular motion (v=0,ω≠0); b)- Forward motion (v≠0,ω=0); c)- Curve motion (v≠0,ω≠0).

and the other two {*B*,*C*}, the two segment extremes. Object classification could be done with a certain probability, by matching the segment representation data, points {*A*,*B*,*C*}, with models of possible objects that could exist in the robot scene. Generally, most objects of interest in the robot's or vehicle environment are either almost convex or can be decomposed into almost convex objects.

A database of such possible objects is necessary, and *a priori* knowledge of the working environment is crucial to create it.

4.5 Object tracking

In order to perceive the object motion it is necessary to select a reference point of the object. Usually, segment centre point *A* is the one chosen [3,14]. In Figure 8, the object's velocity is calculated by the displacement of point *A* between two successive time instants. Point *A* at the instant *k* is defined in polar coordinates by the distance $r(A_{s,k})$ and angle $\alpha_{s,k}$, referred to LRF space.

The Cartesian position of the segment *S* at instant *k* is calculated by the following expressions:

$$\begin{aligned} x_{s,k} &= x_{robo} + r(A_{s,k}).\cos(\theta + \delta(A_{s,k})), \\ y_{s,k} &= y_{robo} + r(A_{s,k}).\sin(\theta + \delta(A_{s,k})) \\ &\left\{(x_{s,k}, y_{s,k}) \wedge A_{s,k} \in \Re^2\right\} \end{aligned} \quad (14)$$

and the velocity, as follows,

$$\begin{aligned} v_{x_{s,k}} &= (x_{s,k} - x_{s,k-1})/h, \\ v_{y_{s,k}} &= (y_{s,k} - y_{s,k-1})/h \end{aligned} \quad (15)$$

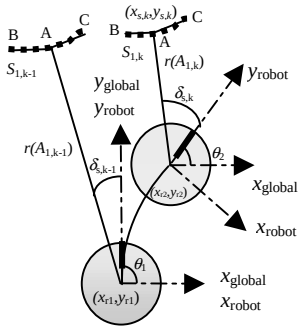


Figure 8: Object position and velocity estimation.

5 Reactive local navigation method

Our method combines a LRF based object detection methodology (allowing velocity and heading estimation) with a velocity based approach, like the DW approach. Similar to Fiorini's VO approach, we have a collision avoidance technique, which takes into account the dynamic constraints of the robot.

This method aims to calculate avoidance trajectories simply by selecting velocities outside a predicted colision cone, penalizing the cone and nearby velocity areas and adding a bonus to escape velocities. The new objective function to be maximized is:

$$\begin{aligned} G(v,w) = Max(&\alpha.head(v,\omega) + \beta.dist(v,\omega) \\ &\delta.vel(v,\omega) + \varepsilon.bonus(v,\omega)) \end{aligned} \quad (16)$$

where the alignment of the robot to a goal position, the distance to the closest obstacles and its velocity can be combined with a bonus/penalty function.

Function $bonus(v,\omega)$ is described as follows,

$$bonus(v,\omega) = \begin{cases} -1, & w \in \left[\frac{v}{\hat{r}} - \Delta\omega, \frac{v}{\hat{r}} + \Delta\omega\right] \\ 0, & otherwise \\ 1, & w \in \left[-\frac{v}{\hat{r}+\Delta R} - \Delta\omega, -\frac{v}{\hat{r}+\Delta R} + \Delta\omega\right] \end{cases} \quad (17)$$

where $\hat{r}$ is the estimated collision radius, $\Delta\omega$ an angular increment necessary to penalize the collision area and nearby areas. This prevents the robot to get undesirable collisions, from unknown changes in the object path.

The collision radius is estimated by the object collision prediction. Knowing the robot's and object's actual position, P_R and P_O, and their actual velocities, it is easy to compute their estimated positions, $\hat{P}_R$ and $\hat{P}_O$. From the estimated robot position we estimate the collision radius by simple trigonometric equalities. Considering the inner triangles of the collision arc (see Figure 9) between the robot and object estimated positions, we have $\Delta ABC \sim \Delta DBC$ and so the following equalities result:

$$\frac{b}{a} = \frac{BD}{AB} = \frac{BC}{BD} = \frac{c}{b} \quad (18)$$

and

$$b^2 = a.c \tag{19}$$

Considering

$$2.\hat{r} = a + c \tag{20}$$

we obtain

$$\hat{r} = \frac{1}{2}\left(\frac{b^2}{a} + a\right) \tag{21}$$

The desired velocity is

$$\tilde{R}(v,\omega) = -\hat{R}(v,w) \tag{22}$$

The parameter ΔR is necessary to compensate the collision radius $\hat{r}$ when the radius value is too big. Equation (22) leads the robot almost to the same trajectory when the radius is too big (R=700cm). Below this value the method reacts in a correct proportion with the objects proximity, above that, a proportional value to the longitudinal distance from the robot to the object, must be set.

All the components of the objective function are normalized to the interval [0,1]. Parameters α, β and δ are used to weight the different components.

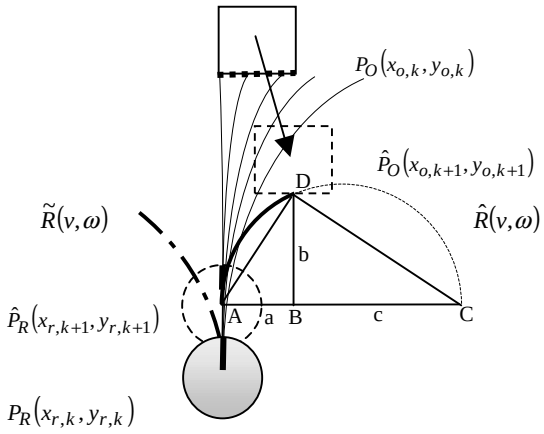


Figure 9: Collision avoidance based on object motion prediction: P_R, P_O – actual robot / object position, $\hat{P}_R$, $\hat{P}_O$ - estimated robot/object position, $\hat{R}$, $\tilde{R}$ - estimated/desired velocity radius.

6 Simulation Results

Extensive simulation tests were done to validate and test the methods exposed above. A sensor model was created for the LRF with the following specifications: each scan has an angle of view of about 180°, with an angular resolution of 1°; the maximum distance measured between the target and the sensor was 300 cm, with ±1 cm of accuracy. The sensor is centred with the robots coordinate axis, aligned with the motion axis.

The workspace is a typical indoor dynamic environment, a corridor with two escape ways for doors and a turn on the top (see Figure 10). The workspace dimensions are similar to narrow indoor real spaces, where the LRF is always detecting static objects, the building walls.

Static and dynamic objects were modelled by rectangular shapes. The user in the beginning of a new simulation can specify their moving direction, velocity and diameter. In all examples the robot started without any knowledge of the environment.

Figure 10 compares the one-scan method [3] and the two-scans method proposed in this paper. There is one robot and two obstacles in the workspace, with different moving directions, as shown in Figure 10a). Constant velocities are considered. In Figure 10, the gray dots represent scanning points and the black points the resulting segment points. As can be seen in Figure 10a), the two-scans method gives only segments corresponding to real moving objects. In Figure 10b) there are always more then two segments.

Figure 11 prints two different go-to-goal tests using the DW approach. The robot starts with (v_0,ω_0)=(50cm/s, 0 rad/s) in both tests. The parameters set {α, β, δ} are equal for both tests, {α=0.8, β=0.3, δ=0.2}. This parameters set enhances the heading function on the navigational scheme, searching the velocity space for the goal position instead of searching, as an example, middle distances from the walls.

Local navigation strategies need, as well as global methodologies, some high level commands (decided in high level structures) which lead the robot to a correct behaviour, needed to accomplish

their mission. This can be done by changing the parameters set values, when it is necessary. Another point of view is that some manoeuvres can be decomposed in several simple procedures. As a simplified example, the avoidance manoeuvre could be decomposed in two procedures:

1)- Avoid_object;

2)- Return_to_path.

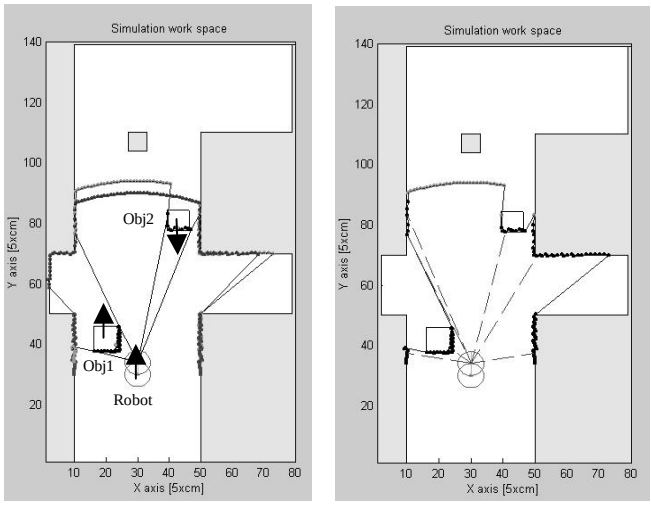
The first procedure enhances the avoidance, which to be secure must increase the robots velocity until the object be avoided, while the second procedure is necessary to lead the robot again to its path. This second procedure is very important when the robot motion is between lanes. These aspects are being studied, and were taken into consideration in the tests, shown in Figure 12. Two different tests are presented, both with the same object avoidance procedure using the parameter set {α=0.3, β=0.5, δ=0.8, ε=1}. One of the return_to_path procedures use the set {α=0.8, β=0.2, δ= -0.1, ε=0} and the other {α=0.8, β=0.2, δ=0.2, ε=0}. The difference in the δ value leads the robot to a longer trajectory to the goal position, due to an increase in the velocity. The procedure that reduces the velocity gets the smallest path until the target, because small velocity values allow the robot to re-orient its heading direction faster.

Two more procedures were used, one to regular navigation, the set {α=0.8, β=0.2, δ=0.2, ε=0}, and another to decrease velocity near the target, {α=0.8, β=0.2, δ= -0.05, ε=0}.

In both tests of Figure 12, the robot starts from an inertial position with an object in front, moving with (v,ω)=(30 cm/s, 0 rad/s).

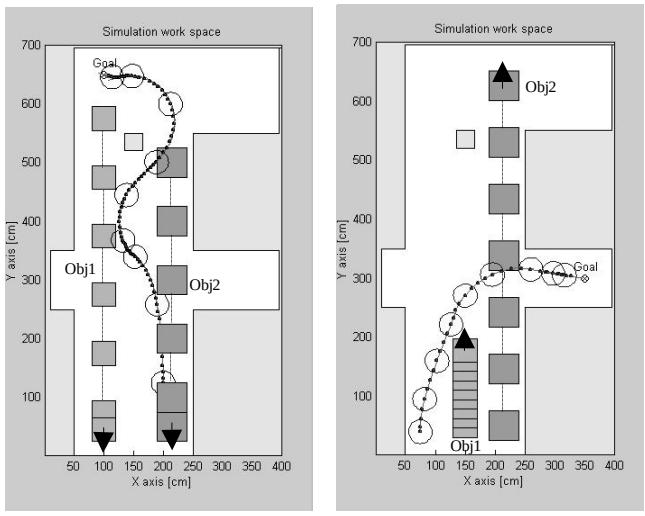
7 Conclusions

A segmentation method, which differentiates between static and dynamic objects, was proposed. This approach restricts all possible data segments to a small set of segments, representing objects with dynamic activity. The method is very effective if the longitudinal velocity of the robot is not too fast. Fast rotational velocities are admissible for small longitudinal velocities.



a) b)

Figure 10: a) Represents the two-scan dynamic obstacle detection algorithm; b) Represents the one-scan object detection approach [3].



a) b)

Figure 11: Two different go-to-goal tasks, performed by the DW approach. In both tests the robot starts with (v_0,ω_0)=(50 cm/s, 0 rad/s). a) Both objects are moving with (v,ω)=(50 cm/s,0 rad/s); b) Object 1 is moving with (v,ω)=(20 cm/s, 0 rad/s) and object 2 with (v,ω)=(120cm/s,0rad/s).

Our local navigation scheme looks promising, and the velocity workbench supplied by the DW approach, is well adapted to local navigation. The manoeuvres decomposition scheme seems to be a good choice to enhance the robot behaviour.

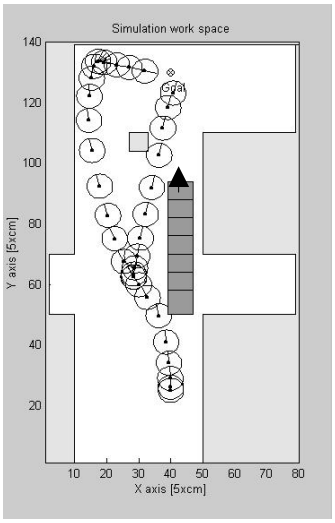


Figure 12: An object avoidance task with two different rescue-target procedures. In both tests the robot starts from an inertial position, with an object in front, moving with (v,ω)=(30 cm/s, 0 rad/s).

Future work will be done in the object classification, using that information to improve the behaviour of the local navigation system with object moving characteristics. The avoidance collision method must be extended to the multiple moving objects situation.

Acknowledgements

This work was partially supported by a PhD student grant (BD/1104/2000) given to the author Daniel Castro by FCT (Fundação para a Ciência e Tecnologia). The author Urbano Nunes acknowledges the support given by FCT (POSI/1999/SRI/33594).

References

[1] J. Borenstein and Y. Koren, "The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots", *IEEE Transactions on Robotics and Automation*, pp.278-288, **7(3)**, June (1991).

[2] O. Brock and O. Khatib, "High-Speed Navigation Using the Global Window Approach", *in IEEE Intl. Conf. on Robotics and Automation ICRA'99*, Detroid, Michigan, pp.341-346, May (1999).

[3] K.Dietmayer, J.Sparbert and D.Streller, "Model Based Object Classification and Tracking in Traffic Scenes from Range Images", *in IEEE Intelligent Vehicle Symposium IV2001*, Tokyo, Japan, May, 14th-17th (2001).

[4] W. Feiten, R. Bauer and G. Lawitzky, "Robust Obstacle Avoidance In Unknown and Cramped Environments", in *IEEE Intl. Conf. on Robotics and Automation ICRA'94*, San-Diego, pp.2412-2417, May (1994).

[5] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments using Velocity Obstacles", *Intl. Journal on Robotics Research*, **17 (7)**, pp.711-727, (1998).

[6] D. Fox, W. Burgard. and S. Thrun, "The Dynamic Window Approach to Collision Avoidance", *IEEE Robotics and Automation Magazine*, **4(1)**, pp.23-33, March (1997).

[7] D. Fox, W. Burgard., S. Thrun and A.Cremers, "A Hybrid Collision Avoidance Method for Mobile Robots", *in IEEE Intl. Conf. On Robotics and Automation ICRA'98*, pp.1238-1243, (1998).

[8] H. Hu and M. Brady, "A Bayesian Approach to Real-time Obstacle Avoidance for Mobile Robots", *Autonomous Robots*, **1**, pp.69-92, (1994).

[9] M. Khatib and R. Chatila, "An Extended Potential Field Approach for Mobile Robot Sensor-based Motions", *In Intl. Conf. On Intelligent Autonomous Systems IAS'4*, (1995).

[10] O. Khatib, "Real-time Obstacle Avoidance for Manipulators and Mobile Robots", *The Intl. Journal of Robotics Research*, **5(1)**, pp.90-98,(1986).

[11] J-C. Latombe, "*Robot Motion Planning*", Kluwer Academic Publishers, Boston, (1991).

[12] SICK Inc., "Laser Measurement System LMS200", Product Information.

[13] R. Simmons, “The Curvature-Velocity Method for Local Obstacle Avoidance”, *In IEEE Intl. Conf. on Robotics and Automation ICRA’96*, pp.2275-2282,Minneapolis, April (1996).

[14] J. Sparbert, K.Dietmayer and D. Streller, “Lane Detection and Street Type Classification using Laser Range Images”, in *IEEE Intl. Conf. on Robotics and Automation*, Seoul, Korea, May 21th-26th (2001).