EFFICIENT DETECTION OF COLLISION IMMINENCE FOR MOBILE ROBOTS*

Vítor Santos[1], José Castro[2], M. Isabel Ribeiro[2]

[1]*Centro de Tecnologia Mecânica e Automação, Departamento de Engenharia Mecânica, Universidade de Aveiro, Campus Universitário, 3810-019 Aveiro, PORTUGAL*
e-mail: vitor@ua.pt
[2]*Instituto Superior Técnico/Instituto de Sistemas e Robótica, Av.Rovisco Pais 1, P 1049-001 Lisboa, PORTUGAL*
e-mail: {jcastro,mir}@isr.ist.utl.pt

Abstract: Collision avoidance is a most relevant issue in mobile robot navigation. The most efficient approach is to track or predict the short-term situations of imminent collision. This is usually done using sensors to detect obstacles that may be hit by the moving robot. When navigating in cluttered environments, the frequency of those situations may lead the robot to many interruptions, possibly unnecessary ones, as well. Abrupt accelerations generated by emergency stops can degrade odometry systems used for dead reckoning. Therefore, there is the need to efficiently detect the imminence of collision and to minimise the number of stops. This paper addresses this problem of imminent collision detection by proposing a generic solution for mobile robots using ultrasonic or similar ranging devices to sense the nearby obstacles. On the other hand, no obstacle avoidance or path recovery is proposed at this level.

Keywords: Mobile Robots, Robot Navigation, Ultrasonic Sensors, Robot Safety

1. INTRODUCTION

The problem of detecting obstacles and issuing a suited command to avoid imminent collision is ancient, common and used by most, if not all, mobile robots. The underlying principle is somehow simple and therefore seldom described: if a sensor measurement is too short (below a given threshold, even if dynamic) stop the vehicle as the simplest action, or possibly invert or modify its motion. The problem addressed in this paper concerns the situations where only collision is to be avoided; no escape solutions are to be designed at this time. This means that, if for some reason (e.g., the motion generation systems failed), the robot is lead to face a situation of imminent collision, then the first priority is to stop. Eventual further motion is to be dealt as a second phase!

When the sensors point to the direction of motion there is a clear correspondence between short measurements and the presence of an obstacle in the path. However, when a mobile robot has a set of sensors placed around its body, the evaluation whether they point to the current direction of the robot motion cannot be done without further analysis. A second issue concerns the usual limited measuring capability of sensors, such as minimal measurable distances, as is the case with some ultrasonic devices.

Traditional research on mobile robot navigation with ultrasound rarely mentions the problem of imminent collision detection and also does not describe it as a problem on its own. This is true for many of the classical works such as those of J. Crowley [1], A. Elfes [2], A. Zelinsky [3], M. Drumheller [4], J. Borenstein [5] [6], J. Leonard [7], among many others. The reason might be that the imminent collision detection was not considered as a problem separable of the navigation itself. This paper specifically addresses this particular issue. The proposed method has been integrated in the lowest level of a navigation architecture [8], where higher levels manage to avoid entering in an emergency situation, such as an imminent collision. Nonetheless, the system has the tools to detect and to handle this last condition, should it occur.

The problem of imminent collision detection can be stated as the question: knowing that the vehicle has a current velocity and that some sensor detects a nearby obstacle ("short" measurement), is there an imminent collision situation?

The problem can be decomposed into three parts: i) detection of the obstacle, i.e., a “short” measurement, ii) evaluation of the need to stop, and iii) efficient stopping upon detection. The first part depends on the efficiency of the sensors, the second is the actual motivation of this work, and the latter represents the expected results.

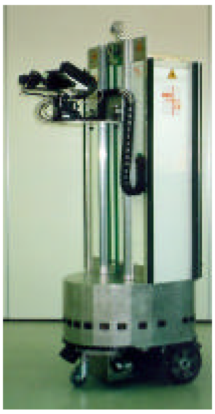
In summary, the main objective of this paper is the efficient use of raw sensorial data (in this case ultrasonic) for the detection of collision imminence based on the instantaneous robot dynamics. The efficiency stands both for the absence of collisions and for the false detection that will unnecessarily stop the

* This work was supported by the project RESOLV-Reconstruction using Scanned Laser and Video of the ACTS Programme, EU

robot. There is only the intention to address the detection of imminent collisions in an efficient way. Other navigation concepts, such as obstacle avoidance, path recovery, or path replanning, which we have been proposing as separable but hierarchically related units in the last few years [8], [9], [10], [11], are not mentioned. This strategy emphasizes the importance we place on the modularity as far as mobile robot navigation architectures are concerned.

2. FRAMEWORK

This work has been developed within the RESOLV project (**RE**construction using **S**canned **L**aser and **V**ideo) aiming at 3D environment reconstruction using laser and video data, where a robot is supposed to navigate within environments partially or totally unknown [12] to support the reconstruction operation. This robot, altogether with the laser scanner and the video camera, is called the AEST (Autonomous Environment Sensor for Telepresence), and is shown in Figure 1. A multi-loop navigation architecture for the mobile robot has been developed to fit the sought requirements [8].



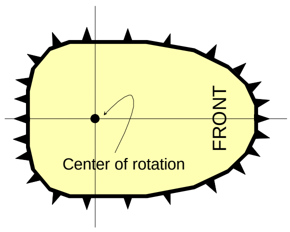


Figure 1 - The AEST (left) and a top view of the oval shape robot base (above)

Within the navigation architecture, and integrated in its lower level (reflexive loop), there is a module named Emergency Handling, which, among other roles, is responsible for efficient Imminent Collision Detection (ICD). It counts mainly on raw ultrasound data, the current velocities of robot wheels and the layout of sensors around the robot base.

The emergency procedures should rarely be active since higher-level navigation modules, namely the local navigation mode [9] [10], are expected to detect and to drive the robot away from the obstacles. However, environments can be unknown and obstacles around the robot may change very rapidly. Also, the density or shape of obstacles may induce the local navigation to failure since it uses integrated sensorial data, and it may not act as promptly as required. Another reason for this inefficiency is the fact that the local navigation component requires fair computational costs, and sometimes its output may not be fast enough to avoid more demanding situations of cluttered environments or navigation near obstacles. Therefore, the navigation architecture should provide an efficient detection of the need to stop. Furthermore, the particular geometry of the robot (Figure 1), tailored by constraints imposed by the perception and the 3D reconstruction modules with obvious consequences on the platform weight distribution and dimensions, claims for caution dynamics, namely limited accelerations and decelerations.

3. KINEMATICS MODEL

A solution for the problem is supported on the model of the robot's instantaneous motion, mainly at the kinematics level as described in this section.

The derivation will be based on an oval-shaped robot, such as the AEST developed for the RESOLV project. However, as it will be clear, the method can be extended to vehicles of any shape, in particular those having two motorised wheels. Extensions to other types of driving are also possible since only the velocity of the sensor relative to world frame is needed at any time.

The reasoning is based on the kinematics of the rigid body, which is the robot structure. Each sensor attached to the structure is characterised by the geometric parameters $P_S(S_x, S_y, b)$ where S_x and S_y are the co-ordinates of the sensor on the robot reference frame and b is the angle of sensor orientation in that frame (Figure 2). The robot reference frame is chosen is such a way that one axis fits upon the line that connects the two wheels, but this is arbitrary.

The ICD algorithm requires information on the sensor velocity, which is related to the robot wheel velocities, as described further. Moreover, and as expected, the importance of a measurement depends also on the direction it has been made. Therefore, the sensor velocity is decomposed in two components, one that points to the direction of measurement and sets the importance of the sensor in the imminent collision detection, and an orthogonal component. The algorithm's main principle is the following: if the robot is moving **too fast** in a direction where measurements are **too short**, then collision preventing actions have to be carried on.

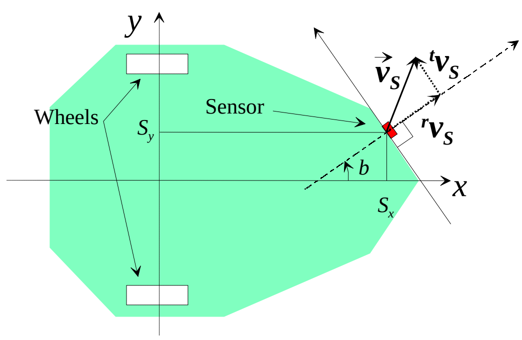


Figure 2 - Velocity components of one sensor

The first step of the algorithm is to determine the sensor velocity in the world frame (observer frame) and to decompose it into components relative to the instantaneous frame located along the longitudinal axis of the sensor beam (Figure 2):

- The component along the sensor axis will be called the radial velocity of the sensor, $^r v_s$. It is the component of sensor's current velocity that points the same direction as the sensor beam.

- The orthogonal component, ${}^t v_s$, can be ignored since its physical meaning is of reduced importance in this approach.

For a robot with two parallel motorised wheels with velocities (v_L, v_R), the entire body (robot) will move around a fixed point, *C*. In general, if two points rigidly linked travel with linear values v_L and v_R, those points will both describe arcs of circumferences with a common centre. Let the centre be the mentioned point *C* with co-ordinates (C_x, C_y).

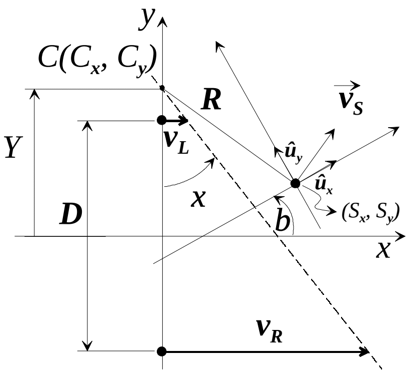


Figure 3 - Centre of rotation (C) of two rigid points moving with parallel velocity, and sensor co-ordinates (S_x, S_y).

Considering the reference frame *xy* as illustrated in Figure 3, C_x becomes 0 and C_y can be determined using simple trigonometric rules:

$$\tan x = \frac{v_L}{C_y - D/2} = \frac{v_R}{C_y + D/2} \qquad (1)$$

where D is the distance between the motorised wheels, and C_y, v_R, v_L are taken with their algebraic values (negative, positive or null), ensuring therefore a general formula. Using Ψ to alternatively designate C_y, it then comes successively that:

$$C_y v_L + \frac{D}{2} v_L = C_y v_R - \frac{D}{2} v_R \qquad (2)$$

$$\Psi = C_y = \frac{D}{2} \frac{v_R + v_L}{v_R - v_L} \qquad (3)$$

This is the well-known expression related to the radius of the arc described by a mobile robot. It is clear that if velocities have opposite signs (one positive and the other negative) Ψ will fall in the region between the wheels. Actually, as the robot is a rigid structure, all points will describe circumferences with the centre in *C*, but with radius *R*, which depends on the point co-ordinates (S_x, S_y) on the robot reference frame (Figure 3). We can also derive the angular velocity (the same for the entire robot), which is obtained using also the value of Ψ from (3):

$$w = \frac{v_L}{\Psi - D/2} = \frac{v_R}{\Psi + D/2} = \frac{v_R - v_L}{D} \qquad (4)$$

We shall now derive the geometric and kinematics relations for a generic point $P_S(S_x, S_y, b)$ on the robot. First, we identify from Figure 3 the following points/vectors:

$$P_S = \begin{bmatrix} S_x \\ S_y \end{bmatrix},\ C = \begin{bmatrix} 0 \\ \Psi \end{bmatrix},\ \vec{R} = \begin{bmatrix} S_x \\ S_y - \Psi \end{bmatrix}$$

It must be also noticed that the sensor velocity $\vec{v}_S$ is a vector perpendicular to $\vec{R} = P_S - C$. $\mathrm{R} = \left\| \vec{\mathrm{R}} \right\|$ is the radius of the rotation around C. The perpendicular vector is obtained with a 90º rotation (clockwise or counter clockwise depending on the sense of the angular velocity). Furthermore, to obtain the correct length of the vector $\vec{v}_S$, appropriate rescaling has to be done. The new vector must be divided by $\left\| \vec{R} \right\|$ and multiplied by the intensity of the sensor velocity $\left\| \vec{v}_S \right\|$, that is *wR*, which gives:

$$\vec{v}_S = \frac{wR}{R} Rot(90) \begin{bmatrix} S_x \\ S_y - \Psi \end{bmatrix} \qquad (5)$$

and, finally,

$$\vec{v}_S = w \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \times \begin{bmatrix} S_x \\ S_y - \Psi \end{bmatrix} = w \begin{bmatrix} \Psi - S_y \\ S_x \end{bmatrix} \qquad (6)$$

Expression (6) is valid only when $\omega \neq 0$. In the particular circumstance of $\omega = 0$ we have $\vec{v}_S = \vec{v}_R = \vec{v}_L$. Now, by defining a reference frame located on the sensor, but with the main axis along the direction of sensor measurement (angle β), the sensor velocity can be expressed in this reference frame whose canonical base is defined by the vectors $\hat{u}_x$ and $\hat{u}_y$ (expressed in the world or observer frame):

$$\hat{u}_x = Rot(b) \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \cos b \\ \sin b \end{bmatrix},\ \hat{u}_y = Rot(b) \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -\sin b \\ \cos b \end{bmatrix} \qquad (7)$$

The radial and tangential sensor velocities are obtained as projections of the sensor velocity $\vec{v}_S$ on $\hat{u}_x$ and $\hat{u}_y$ respectively. The radial component sensor velocity is therefore:

$${}^r v_S = w \begin{bmatrix} \Psi - S_y \\ S_x \end{bmatrix} \cdot \begin{bmatrix} \cos b \\ \sin b \end{bmatrix} = w(\Psi - S_y)\cos b + w S_x \sin b \qquad (8)$$

Since $w = (v_R - v_L)/D$, for $v_R > v_L$ we get $\omega > 0$, and the centre of rotation is always on the left side of the vehicle. On the other hand, for $v_R < v_L$, the centre of rotation is on the right side of the robot and $\omega < 0$. This yields the appropriate relations for the radial and tangential components of sensor velocity.

$${}^r v_S = \frac{v_R - v_L}{D} \left[(\Psi - S_y)\cos b + S_x \sin b \right] \qquad (9)$$

Knowing the expression for the radial velocity of each sensor, that is, the component of velocity in the direction of measurement, the ICD algorithm can be formulated, based only on the robot physical properties and the instantaneous velocity of its two wheels.

4. THE IMMINENT COLLISION DETECTION ALGORITHM

The main idea of the ICD algorithm is to look for sequences of adjacent sensors that have measurements below a given threshold. The sequence may even consist of one sensor only, thus making the system much safer, but possibly too active.

The referred threshold should be a mere indicator for a worst case. It serves solely to elect a sensor as a candidate to trigger an emergency stop. It is therefore a threshold of **level 0**. Further analysis on sensor radial velocity will decide whether it is a real emergency by calculating the threshold of **level 1**. This level simply verifies whether velocity satisfies a criterion of minimal intensity. Beyond this level, there is the **level 2** threshold, which will take into account relations between velocities as explained further. Figure 4 illustrates the main steps of the algorithm.

4.1. Level-0 threshold

Due to performance reasons, not all sensors are checked in the ICD algorithm and a criterion is defined to elect the sensors to be checked. This criterion evaluates the maximal robot instantaneous velocity (in absolute value) and verifies, for that velocity, whether sensor measurement is too short according to the values shown in Table 1. This means that for each possible interval of robot instantaneous velocity, there is a corresponding minimal measurement for any sensor. The threshold is defined for the sensor measurement; if for a given velocity the sensor threshold is reached then elect that sensor for the next level.

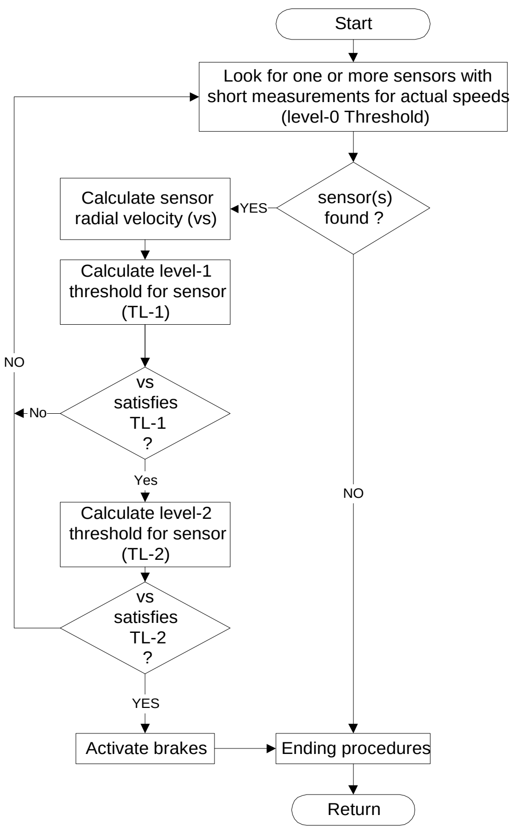


Figure 4 - Algorithm to detect imminent collisions. (TL= threshold level).

Table 1 - Values for level-0 thresholds: velocity ranges versus minimum sensor measurements.

$\max(\lvert v_L\rvert, \lvert v_R\rvert)$ (m/s)	Minimum Sensor Measurement (m)
< 0.2	0.40
0.2 - 0.3	0.60
0.3 - 0.4	0.75
0.4 - 0.5	0.80
> 0.5	0.90

The definition of Table 1 was based on the experimental capacity of the robot to stop within its free space, since a full model of robot dynamics was not possible to define. This table also takes into account the worst cases of sensor delay for the data rate performance. This component of the algorithm is very fast, since only simple comparisons are performed.

4.2. Level-1 threshold

After passing level 0, the sensor should now pass the test of radial velocity as defined in Section 0. This test compares the radial velocity component to a dynamic threshold, named as level-1 threshold, according to the following criteria. If the sensor radial velocity is below or equal to a given value (${}^{r}v_{Smin}$), then discard measurement. Radial sensor velocities can be positive, null or negative, meaning that the sensor is moving towards its measuring target, perpendicular to it, or moving apart from it. The two last situations imply that no collision will, at any time, occur with that sensor. The ICD implemented condition at this level is:

if ${}^{r}v_S \leq {}^{r}v_{Smin}$ **then** NO_COLLISION

The ${}^{r}v_{Smin}$ threshold parameter allows the elimination of small velocities. It was set to zero in the implemented version.

4.3. Level-2 threshold

At last, there is the level-2 threshold. If the sensor radial velocity is too small when compared to the robot "average" velocity (r_{av}), the sensor measurement is ignored as no imminent collision is occurring. The r_{av} indicator is defined as the average of the magnitude of both wheel speeds:

$$r_{av} = \left(\lvert v_L\rvert + \lvert v_R\rvert\right) / 2 \tag{10}$$

The r_{av} parameter is always positive unless the robot is stopped. A threshold coefficient for sensor S was defined as:

$$z_S = {}^{r}v_S / r_{av} \tag{11}$$

if $z_S \leq \zeta_{Smin}$ **then** NO_COLLISION

This parameter (level-2 threshold) is a measure of how important is the sensor radial velocity, when compared to the robot velocity itself. It may serve essentially for situations where the robot is moving very slowly (typically 1 to 10 cm/s) and applies only to sensors that are not pointing in the main direction of motion. A value of ζ_{Smin}= 0.5 was found as being too reactive leading to the detection of non-existing emergencies. On the other hand, ζ_{Smin} = 0.75 yielded better experimental results. This parameter can disqualify a sensor as an imminent collision detector even if the sensor has a positive radial velocity component. The overall result is that this threshold allows some movements that nonetheless present a small amount of risk, but it overcomes some limitations of level-1 threshold that states that radial velocity should be equal or less than zero for no imminent collision.

4.4. System refinement

Taking into account the following issues the ICD methodology can be further refined:

- Number of adjacent sensors,
- Conic nature of measuring beam,
- Dynamic considerations–maximal accelerations,
- Introduction of memory (data integration).

The first two issues were actually implemented by changing the corresponding parameters. The first concerns the possibility of considering imminent collision only when two or more adjacent sensors indicate an imminent collision. This can compensate for eventual sensor failure or interference, but should be avoided in environments where obstacles have very small dimensions (<10 cm in the AEST robot) and navigation has to be done often at close distances (<1 m) from obstacles. That is why we used, most of the times, one sensor only. The conic nature of ultrasound beams is also easy to implement and is explained below. On the other hand, the introduction of dynamic considerations could be done in order to smooth or optimise the stopping procedure aiming at minimising the deceleration to impose to the robot. This was however not explored on our robot due to some mechanical limitations of the braking system. Finally, the introduction of memory could further refine data failure. Each sensor could be monitored and the evolution of its recent measurements would refine the decision of detecting an imminent collision. That could be associated to a further level of decision (such as a level-3). To this date this feature was not implemented since special data filtering is necessary due to the somehow unpredictable individual ultrasonic sensor measurements, mainly specularity in unknown environments.

If the sensor beam covers an aperture α around the main direction β, additional procedures can be taken, since the obstacle can be located on a wider region, not a single direction. Therefore, the radial component of the sensor velocity can take a range of values given by the following expression, which can be obtained after expression (9):

$$^{r}v_S = \frac{v_R - v_L}{D}\left[\left(\Psi - S_y\right)\cos\left(b \pm \frac{a}{2}\right) + S_x \sin\left(b \pm \frac{a}{2}\right)\right] \quad (12)$$

For the ultrasonic sensors used, α is about 20º. That value did not affect considerably the variations of previous expression, and this ability was disabled most of the time to spare computing resources.

4.5. Virtual sensors

The described approach also copes with a multi-sensor system by introducing the concept of virtual sensor. In the particular case of this work, the robot has several bumpers that in some specific navigation conditions may be activated. To keep simplicity and generality in the system software, each bumper was associated to a range of influence overlapping the areas already covered by ultrasound sensors. The sonar's data is overridden by bumper data before being used by the ICD algorithm. Moreover, bumper data must be debounced (or artificially delayed in time) so the robot has the chance to refrain from remaining glued to a particular obstacle. This leads to the definition of virtual sensors that are sources of data obtained by merging sensorial information. In our approach, the virtual sensors have been placed exactly on the same position as the ultrasound devices, but their data takes into account both the real ultrasound measurements and bumper state.

5. RESULTS

The results of the method are quite interesting especially in the case where the robot does not stop upon visual apparent imminent collisions, which is precisely what was supposed to be since there is no real imminent collision. The next three figures illustrate examples extracted from real data where the robot only stops when it is actually needed.

On each situation illustrated in Figure 5, Figure 6, and Figure 7, the lines drawn from each sensor indicate the radial component of sensor velocity in the world reference frame. If the line points outwards of the robot body, then it has a positive value, otherwise it is negative or null if it is not illustrated at all. In each case we indicate the wheels' speed, and the instantaneous path that results thereof. Potentially emergency triggering sensors are also indicated, that is, sensors that measure a too short distance to allow the robot to continue moving.

Only in the example shown in Figure 6 there is a real imminent collision, and the robot was stopped. In any of the three situations those sensors indicate the same measurements (level-0 threshold) but the need to stop exists only in one of those cases as mentioned. In Figure 5 we can however verify that one of those sensors (the 3rd on the right of the enclosed group) has a positive small value, which would trigger imminent collision detection. The level-2 threshold helped to solve that situation and allowed motion to continue. A path less curved would not allow this to occur and imminent collision would be notified.

With this method we achieved nearly zero collision motion. Only two or three minor touches on the obstacles occurred during a period of hundred of hours of navigation in cluttered environments, but they were caused by ultrasound occasional failure.

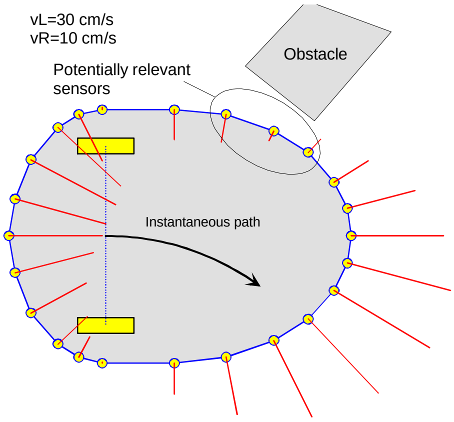


Figure 5 - No imminent collision detection.

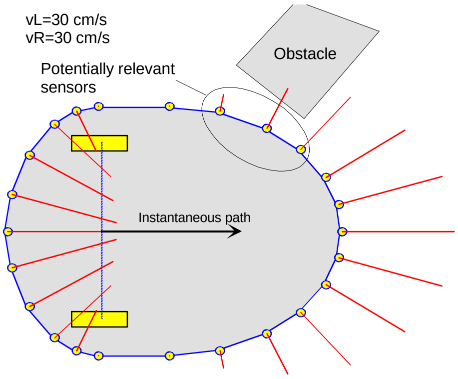


Figure 6 - Imminent collision detected.

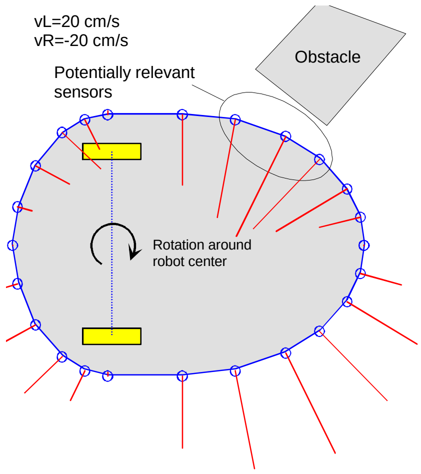


Figure 7 - No imminent collision detected

6. CONCLUSIONS

This paper described a general imminent collision detection system applied to a real specific mobile robot. The system is efficient since it detects imminent collisions, but above all, it avoids unnecessary stopping of the vehicle, which is important mainly when navigating in cluttered environments or near obstacles. The proposed solution is generic in the sense that it can be easily adapted for other robot shapes as only few geometric parameters are needed, such as the position of sensors on the robot.

The computational cost of the algorithm is perfectly within the requirements of the robot real time system. Performance is only affected by the sensor data rate, which in this case was normally not better than 2.5 Hz. This algorithm was run during the highest priority "Emergency" real time process on the robot, and took less than 1 real time clock unit (1 tick, 10 ms) every 10 ticks. The actual duration of the process was not directly measurable, but some statistical tests pointed to even less than 0.5 ticks (less than 5% of useful CPU time).

The concept of virtual sensor was successfully introduced to allow the easy integration of several types of sensors on a same algorithmic framework.

The structure of successive levels of decision (0, 1 and 2, so far), each more refined than the previous ones, ensures robustness (very few collisions) and also the mentioned efficiency (no unnecessary stops).

The global conclusion is that this relatively simple kinematics approach definitely improved collision avoidance in our robot, and made it easier for the entire navigation procedure in relatively cluttered environments.

7. REFERENCES

[1] J. L. Crowley – Navigation for an Intelligent Mobile Robot, *IEEE Journal of Robotics and Automation*, vol. RA-1(1), pp. 31-41, March 1985.

[2] Alberto Elfes – Using Occupancy Grids For Mobile Robot Perception and Navigation, *IEEE Computer*, pp. 46-57, June 1989.

[3] A. Zelinsky – Environment Mapping with a Mobile Robot Using Sonar, *Proc. of the Australian Joint Artificial Intelligence Conference – AI 88*, pp. 373-388, November 1988.

[4] Michael Drumheller – Mobile Robot Localization Using Sonar, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 9, n. 2, March 1987.

[5] J. Borenstein , Yorem Koren – Real-Time Obstacle Avoidance for Fast Mobile Robots, *IEEE Trans. on Systems, Man and Cybernetics*, vol. 19, n. 5, September/October 1989.

[6] J. Borenstein, Yoram Koren – The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots, *IEEE Trans. on Robotics and Automation*, vol. 7, n. 3, June 1991.

[7] John J. Leonard, Hugh F. Durrant-Whyte – Directed Sonar Sensing for Mobile Robot Navigation, *Kluwer Academic Publishers*, 1992.

[8] J. Castro, V. Santos, M. I. Ribeiro – A Multi-loop Navigation Architecture for Mobile Robots, *Int. Conf. of IEEE on Robotics and Automation*, Leuven, Belgium, May 1998, pp. 970-975.

[9] V. Santos, J. Gonçalves, F. Vaz – Perception Maps for the Local Navigation of a Mobile Robot: a Neural Network Approach, *IEEE Int. Conf. on Robotics and Automation*, San Diego, USA, pp. 2193-2198, May 1994.

[10] V. Santos, J. Gonçalves, F. Vaz – Local Perception Maps for Autonomous Robot Navigation, *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS96*, pp.821-827, Osaka, Japan, 4-8 Nov 1996.

[11] V. Santos, J. Castro, M. I. Ribeiro – *Navigation Architecture for the RESOLV Mobile Robot*,, 3rd Portuguese Conference on Automatic Control, CONTROLO98, pp. 805-810, Coimbra, Portugal, 9-11 September 1998.

[12] D. Leevers, P. Gil, F. M. Lopes, J. Pereira, J. Castro, J. Gomes-Mota, M. I. Ribeiro, J. Gonçalves, V. Sequeira, E. Wolfart, V. Dupourque, V. Santos, S. Butterfield, D. Hogg, Kia Ng – An Autonomous Sensor for 3D Reconstruction, *3rd European Conference on Multimedia Applications, Services and Techniques*, ECMAST98, Berlin, Germany, 26-28 May 1998.