

# Generalized Geometric Triangulation Algorithm for Mobile Robot Absolute Self-Localization

João SENA ESTEVES<sup>1</sup>, Adriano CARVALHO<sup>2</sup>, Member, IEEE, Carlos COUTO<sup>3</sup>, Member, IEEE

<sup>1,3</sup>DEI, University of Minho, Campus of Azurém, 4800-058 GUIMARÃES, Portugal,  
e-mail : (sena<sup>1</sup> ccouto<sup>3</sup>)@dei.uminho.pt

<sup>2</sup>DEEC., Faculty of Engineering, University of Porto, R. Dr. Roberto Frias, 4200-465 PORTO, Portugal,  
e-mail: asc@fe.up.pt

**Abstract**—Triangulation with active beacons is widely used in the absolute localization of mobile robots. The Geometric Triangulation algorithm allows the self-localization of a robot on a plane. However, the three beacons it uses must be “properly ordered” and the algorithm works consistently only when the robot is within the triangle formed by these beacons. This paper presents an improved version of the algorithm, which does not require beacon ordering and works over the whole navigation plane except for a few well-determined lines where localization is not possible.

**Index Terms**—Mobile Robots, Robot Navigation, Position Location.

## I. INTRODUCTION

Localization is the process of finding both position and orientation of a vehicle in a given referential system [1], Drumheller in [2], [3]. Navigation of mobile robots indoors usually requires accurate and reliable methods of localization. Many transportation systems now using wire-guided automated vehicles may benefit from the increased layout design flexibility provided by a wire-free localization method such as triangulation with active beacons.

Triangulation with active beacons is a robust, accurate, flexible and widely used method of absolute localization [4], [5]. Several triangulation algorithms have been proposed. These are some examples: *Geometric Triangulation*, *Iterative Search*, *Newton-Raphson Iterative Search* and *Geometric Circle Intersection* [6], algorithm from the *Imperial College Beacon Navigation System* [3], [7], triangulation using three circle intersection [8], triangulation using two circle intersection [9], [10], *Position Estimator* algorithm [10]. The term *absolute localization* was defined by Drumheller [2] as “the enabling of a mobile robot to determine its position and orientation [...] in a way that is independent of assumptions about previous movements”. This is a very important task since relative localization estimated by dead reckoning methods is only reliable within a few meters of distance traveled or short periods of time. These methods are usually too inaccurate to insure safe navigation over large distances or long periods of time [1], [3], [11], [12], [13], [14], [15], [16], [17], [18]. So, absolute localization is required to perform periodic corrections of the estimated localization. According to [19], “the most common

positioning methods for indoor vehicles are odometry for relative positioning and triangulation for absolute positioning”.

Triangulation is based on the measurement of the bearings of the robot relatively to beacons placed in known positions. It differs from trilateration, which is based on the measurement of the distances between the robot and the beacons. These beacons are also called *landmarks* by some authors. According to [3], the term *beacon* is more appropriate for triangulation methods.

When navigating on a plane, three distinguishable beacons - at least - are required for the robot to localize itself (Fig. 1).  $\lambda_{12}$  is the oriented angle “seen” by the robot between beacons 1 and 2. It defines an arc between these beacons, which is a set of possible positions of the robot [20]. An additional arc between beacons 1 and 3 is defined by  $\lambda_{31}$ . The robot is in the intersection of the two arcs. Usually, the use of more than three beacons results in redundancy. In [6], triangulation with three beacons is called *three-object triangulation*.

Section II describes the restrictions that are common to all three-object triangulation algorithms. Section III presents the Geometric Triangulation algorithm described in [6]. This is done in order to clarify the improvements made to this algorithm in Section IV. Simulation results presented in that section validate these improvements.

## II. RESTRICTIONS COMMON TO ALL THREE-OBJECT TRIANGULATION ALGORITHMS

As shown in the previous section, the robot must “see” at least three beacons to localize itself in a plane. All areas of the plane with less than three visible beacons are unsuitable for robot localization. It also has been shown that those three beacons are usually enough to allow robot localization.

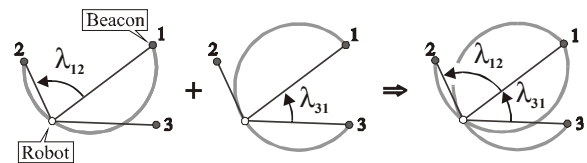


Fig. 1. Three-object triangulation

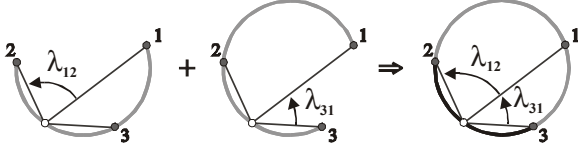


Fig. 2. Robot cannot localize itself on the circumference that contains the three beacons

However, this is not true if the robot and the beacons all lie in the same circumference (Fig. 2). The robot cannot localize itself on this circumference because the intersection of the two arcs is another arc, not a point. These two restrictions are common to all three-object triangulation algorithms. Even in an obstacle-free environment a problem arises when a beacon becomes between the robot and other beacon. In such cases, it is assumed that only the closest beacon is visible.

### III. THE GEOMETRIC TRIANGULATION ALGORITHM

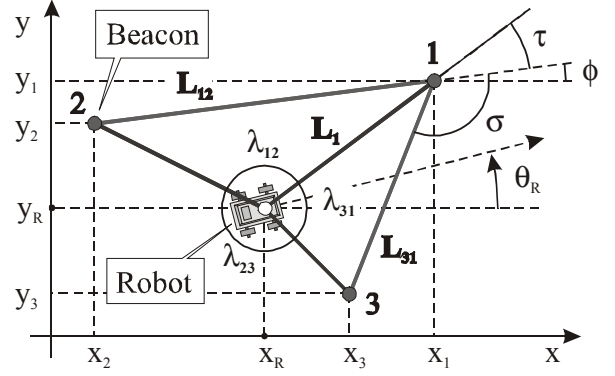
The Geometric Triangulation algorithm (Fig. 3) described in [6] uses three distinguishable beacons in a Cartesian plane, labeled 1, 2 and 3, placed in known positions  $(x_1, y_1)$ ,  $(x_2, y_2)$  and  $(x_3, y_3)$ .  $L_{12}$  and  $L_{31}$  are, respectively, the distances between beacons 1 and 2 and beacons 1 and 3.  $L_1$  is the distance between the robot and beacon 1. To determine its position  $(x_R, y_R)$  and orientation  $\theta_R$ , the robot measures angles  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  (relative beacon orientations from the robot).

#### A. Specific Restrictions of the Geometric Triangulation Algorithm

In addition to the restrictions that are common to all three-object triangulation algorithms (referred in Section II) and according to [6], the Geometric Triangulation algorithm has the following specific ones:

1. The beacons must be labeled consecutively (1, 2, 3) in counterclockwise fashion;
2. Both the angle between beacons 1 and 2 ( $\lambda_{12}$ ) and the angle between beacons 1 and 3 ( $\lambda_{31}$ ) must be less than  $180^\circ$ . If this is not true, the beacon labels must be shifted counterclockwise once or twice, until the requirement is met.

When these two requirements are met, the beacons are said to be “properly ordered”. Resulting from all restrictions (both common and specific ones), there are zones and paths in the plane where the Geometric Triangulation algorithm does not work. Moreover, still according to [6], “the algorithm works consistently only when the robot is within the triangle formed by the three landmarks [beacons]. There are areas outside the landmark [beacon] triangle where the algorithm works but these areas are difficult to determine and are highly dependent on how the angles are defined. (...) No consistent rule was found to define the angles to insure a correct execution of this method”. Such an angles definition is used in the Generalized Geometric Triangulation algorithm described in Section IV.



#### Geometric Triangulation Algorithm

1. Properly order beacons.
2. Let  $\lambda_{31} = 360^\circ + (\lambda_1 - \lambda_3)$
3. Let  $\lambda_{12} = \lambda_2 - \lambda_1$
4. Let  $\phi$  be the angle between the positive x-axis and the line formed by the points of beacons 1 and 2.
5. Let  $\sigma$  be the angle between the positive x-axis and beacons 1 and 3, plus  $\phi$ .
6. Let  $\gamma = \sigma - \lambda_{31}$
7. Let  $p = \frac{L_{31} \cdot \sin \lambda_{12}}{L_{12} \cdot \sin \lambda_{31}}$
8. Let  $\tau = \tan^{-1} \left[ \frac{\sin \lambda_{12} - p \cdot \sin \gamma}{p \cdot \cos \gamma - \cos \lambda_{12}} \right]$
9. Let  $L_1 = \frac{L_{12} \cdot \sin(\tau + \lambda_{12})}{\sin \lambda_{12}}$
10.  $x_R = x_1 - L_1 \cdot \cos(\phi + \tau)$
11.  $y_R = y_1 - L_1 \cdot \sin(\phi + \tau)$
12.  $\theta_R = \phi + \tau - \lambda_1$

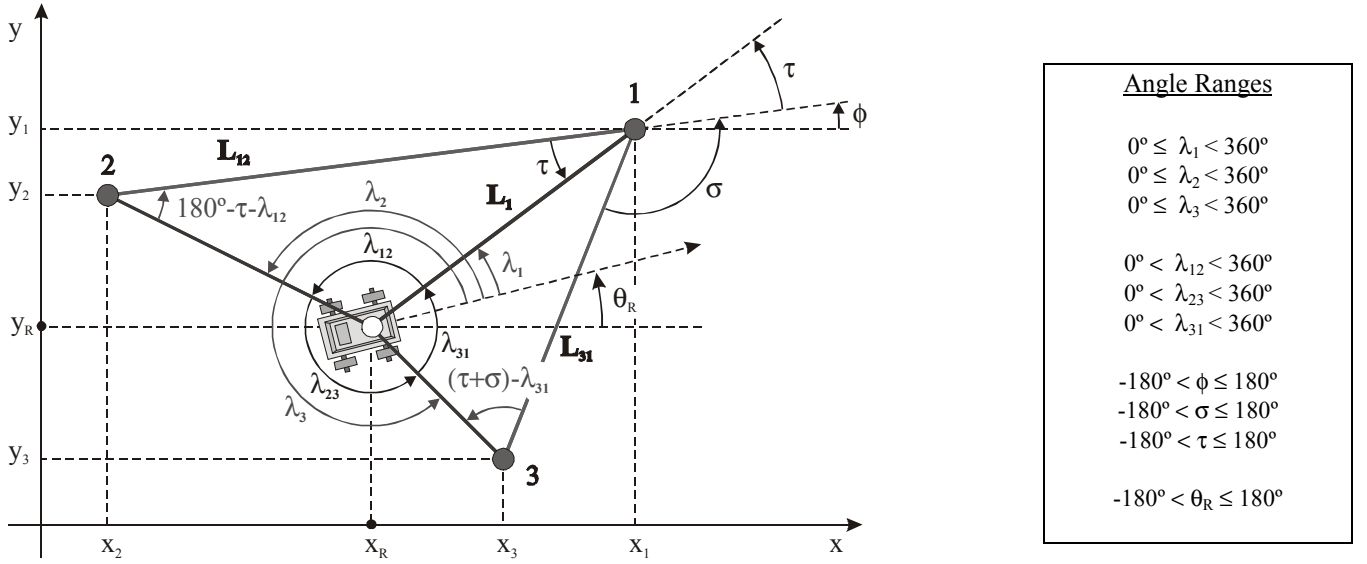
Fig. 3. Geometric Triangulation

### IV. THE GENERALIZED GEOMETRIC TRIANGULATION ALGORITHM

This section presents a new version of the Geometric Triangulation algorithm that does not require beacon ordering and works over the whole navigation plane except for a few well-determined lines where localization is not possible.

These improvements are mainly achieved through a careful definition of the angles used by the algorithm, which is only subject to the restrictions that are common to all three-object triangulation algorithms.

Consider (Fig. 4) three distinguishable beacons in a Cartesian plane, randomly labeled 1, 2 and 3, with known positions  $(x_1, y_1)$ ,  $(x_2, y_2)$  and  $(x_3, y_3)$ .  $L_{12}$  is the distance between beacons 1 and 2.  $L_{31}$  is the distance between beacons 1 and 3.  $L_1$  is the distance between the robot and beacon 1. In order to determine its position  $(x_R, y_R)$  and orientation  $\theta_R$ , the robot measures - in counterclockwise fashion - the angles  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$ , which are the beacon orientations relative to the robot heading. Algorithm lines 2 through 5 compute the oriented angles  $\lambda_{12}$  and  $\lambda_{31}$  “seen” by the robot between beacons 1 and 2 and beacons 3 and 1, respectively. Both  $\lambda_{12}$  and  $\lambda_{31}$  are always positive.



#### Generalized Geometric Triangulation Algorithm

1. If there are less than three visible beacons available, then return a warning message and stop.
2.  $\lambda_{12} = \lambda_2 - \lambda_1$
3. If  $\lambda_1 > \lambda_2$  then  $\lambda_{12} = 360^\circ + (\lambda_2 - \lambda_1)$
4.  $\lambda_{31} = \lambda_1 - \lambda_3$
5. If  $\lambda_3 > \lambda_1$  then  $\lambda_{31} = 360^\circ + (\lambda_1 - \lambda_3)$
6. Compute  $L_{12}$  from known positions of beacons 1 and 2.
7. Compute  $L_{31}$  from known positions of beacons 1 and 3.
8. Let  $\phi$  be an oriented angle such that  $-180^\circ < \phi \leq 180^\circ$ . Its origin side is the image of the positive x semi-axis that results from the translation associated with the vector which origin is (0, 0) and ends on beacon 1. The extremity side is the part of the straight line defined by beacons 1 and 2 which origin is beacon 1 and does not go by beacon 2.
9. Let  $\sigma$  be an oriented angle such that  $-180^\circ < \sigma \leq 180^\circ$ . Its origin side is the straight line segment that joins beacons 1 and 3. The extremity side is the part of the straight line defined by beacons 1 and 2 which origin is beacon 1 and does not go by beacon 2.

10.  $\gamma = \sigma - \lambda_{31}$
11.  $\tau = \tan^{-1} \left[ \frac{\sin \lambda_{12} \cdot (L_{12} \cdot \sin \lambda_{31} - L_{31} \cdot \sin \gamma)}{L_{31} \cdot \sin \lambda_{12} \cdot \cos \gamma - L_{12} \cdot \cos \lambda_{12} \cdot \sin \lambda_{31}} \right]$
12. If  $\begin{cases} \lambda_{12} < 180^\circ \\ \tau < 0^\circ \end{cases}$  then  $\tau = \tau + 180^\circ$
13. If  $\begin{cases} \lambda_{12} > 180^\circ \\ \tau > 0^\circ \end{cases}$  then  $\tau = \tau - 180^\circ$
14. If  $|\sin \lambda_{12}| > |\sin \lambda_{31}|$  then  $L_1 = \frac{L_{12} \cdot \sin(\tau + \lambda_{12})}{\sin \lambda_{12}}$
15. else  $L_1 = \frac{L_{31} \cdot \sin(\tau + \sigma - \lambda_{31})}{\sin \lambda_{31}}$
16.  $x_R = x_1 - L_1 \cdot \cos(\phi + \tau)$
17.  $y_R = y_1 - L_1 \cdot \sin(\phi + \tau)$
18.  $\theta_R = \phi + \tau - \lambda_1$
19. If  $\theta_R \leq -180^\circ$  then  $\theta_R = \theta_R + 360^\circ$
20. If  $\theta_R > 180^\circ$  then  $\theta_R = \theta_R - 360^\circ$

Fig. 4. Generalized Geometric Triangulation

According to  $\lambda_{12}$  it is possible to divide the plane in two zones (Fig. 5). The same is valid to  $\lambda_{31}$ , and this results in the plane divisions shown in Fig. 6.

Applying the law of sines to the triangles formed by the robot and the beacons in each zone of the plane, we obtain the following expressions (for Zone I,  $0^\circ < \sigma < 180^\circ$ , refer to Fig. 4).

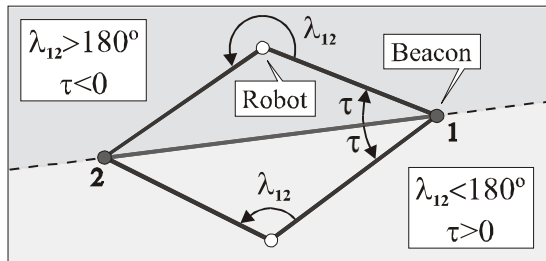


Fig. 5. Division of the plane according to  $\lambda_{12}$

$$\text{Zone I: } \begin{cases} \frac{L_{31}}{\sin \lambda_{31}} = \frac{L_1}{\sin(\tau + \sigma - \lambda_{31})} \\ \frac{L_{12}}{\sin \lambda_{12}} = \frac{L_1}{\sin(180^\circ - \tau - \lambda_{12})} \end{cases} \quad (1)$$

$$\text{Zone II: } \begin{cases} \frac{L_{31}}{\sin \lambda_{31}} = \frac{L_1}{\sin(\tau + \sigma - \lambda_{31})} \\ \frac{L_{12}}{\sin(360^\circ - \lambda_{12})} = \frac{L_1}{\sin(-180^\circ + \tau + \lambda_{12})} \end{cases} \quad (2)$$

$$\text{Zone III: } \begin{cases} \frac{L_{31}}{\sin(360^\circ - \lambda_{31})} = \frac{L_1}{\sin(\lambda_{31} - \tau - \sigma)} \\ \frac{L_{12}}{\sin \lambda_{12}} = \frac{L_1}{\sin(180^\circ - \tau - \lambda_{12})} \end{cases} \quad (3)$$

$$\text{Zone IV: } \begin{cases} \frac{L_{31}}{\sin(360^\circ - \lambda_{31})} = \frac{L_1}{\sin(\lambda_{31} - (\tau + \sigma) - 360^\circ)} \\ \frac{L_{12}}{\sin(360^\circ - \lambda_{12})} = \frac{L_1}{\sin(-180^\circ + \tau + \lambda_{12})} \end{cases} \quad (4)$$

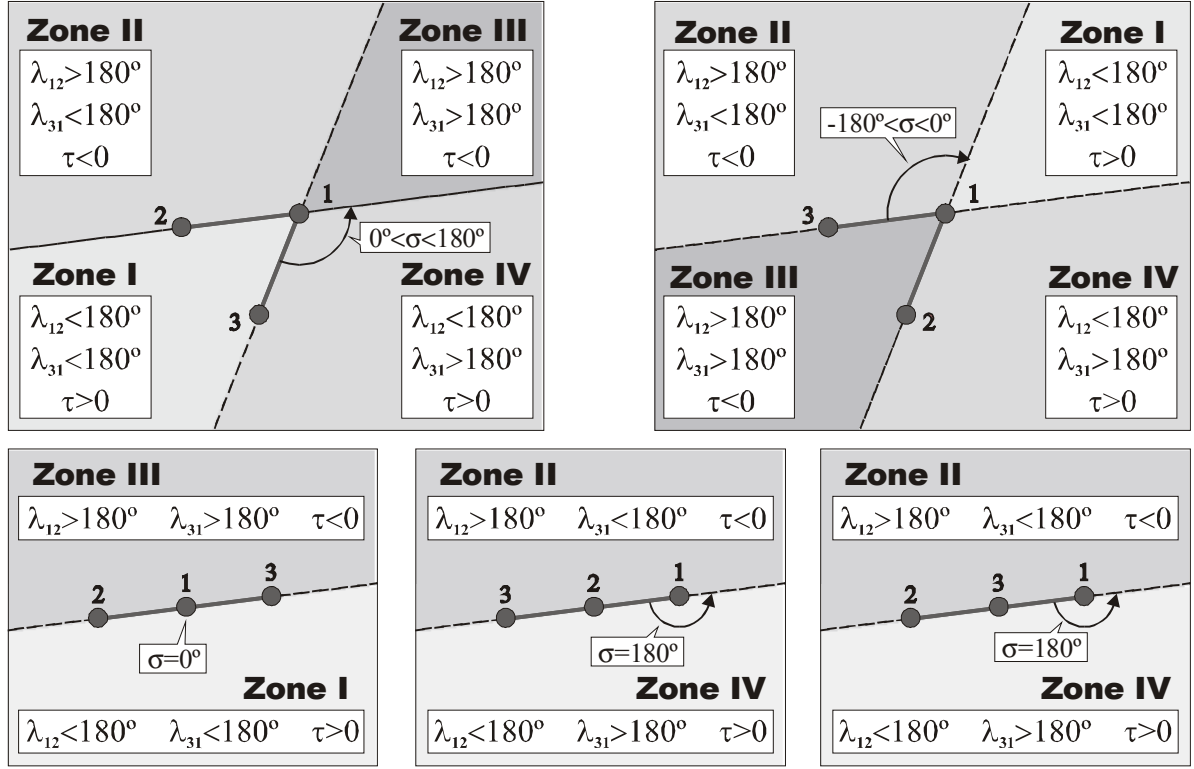


Fig. 6. Divisions of the plane according to  $\lambda_{12}$ ,  $\lambda_{31}$ ,  $\sigma$  and the relative positions of the beacons

Since, by definition,

$$\gamma = \sigma - \lambda_{31}, \quad (5)$$

solving (1), (2), (3) and (4) to find  $\tau$  and  $L_1$ , the same result is obtained in each zone:

$$\tau = \tan^{-1} \left[ \frac{\sin \lambda_{12} \cdot (L_{12} \cdot \sin \lambda_{31} - L_{31} \cdot \sin \gamma)}{L_{31} \cdot \sin \lambda_{12} \cdot \cos \gamma - L_{12} \cdot \cos \lambda_{12} \cdot \sin \lambda_{31}} \right] \quad (6)$$

$$L_1 = \frac{L_{12} \cdot \sin(\tau + \lambda_{12})}{\sin \lambda_{12}} \quad (\text{valid if } \sin \lambda_{12} \neq 0) \quad (7)$$

$$L_1 = \frac{L_{31} \cdot \sin(\tau + \sigma - \lambda_{31})}{\sin \lambda_{31}} \quad (\text{valid if } \sin \lambda_{31} \neq 0) \quad (8)$$

For beacon configurations with  $\sigma = 0^\circ$  or  $\sigma = 180^\circ$ , if  $\lambda_{12} = 180^\circ$  or  $\lambda_{31} = 180^\circ$  then the robot is only able to “see” two beacons and  $L_1$  cannot be computed. For other configurations, to enable localization if  $\lambda_{12} = 180^\circ$  or  $\lambda_{31} = 180^\circ$ , in the computation of  $L_1$  the algorithm chooses (lines 14 and 15), from expressions (7) and (8), the one with larger denominator.

For all beacon configurations except those with  $\sigma = 0^\circ$  or  $\sigma = 180^\circ$ , if  $\lambda_{12} = 180^\circ$  then the robot is over the line segment that joins beacons 1 and 2 and

$$L_1 = \frac{L_{31} \cdot \sin(\sigma - \lambda_{31})}{\sin \lambda_{31}} \quad (9)$$

Since  $\tau = 0^\circ$ , to compute  $L_1$  it is possible to use expression (8) instead of expression (9). Similarly, for all beacon configurations except those with  $\sigma = 0^\circ$  or  $\sigma = 180^\circ$ , if  $\lambda_{31} = 180^\circ$

then the robot is over the line segment that joins beacons 1 and 3 and expression (7) can be used to compute  $L_1$ .

The value returned by function  $\tan^{-1}$  is in the range of  $-90^\circ$  to  $90^\circ$ . So, algorithm lines 12 and 13 are required to compute values of  $\tau$  falling outside that interval. Similarly, algorithm lines 19 and 20 ensure that  $-180^\circ < \theta \leq 180^\circ$ . They may be omitted if  $\theta$  is allowed to fall outside this interval.

#### A. Specific Restrictions of the Generalized Geometric Triangulation Algorithm

The Generalized Geometric Triangulation algorithm suffers from the restrictions that are common to all three-object triangulation algorithms. However, the specific restrictions of the Geometric Triangulation algorithm do not further apply. In fact, the modified algorithm has the following features:

1. The three beacons may be randomly labeled 1, 2, and 3 (there is no need for consecutive labeling in counterclockwise fashion);
2. The three beacons may be placed anywhere in the plane (as long as two beacons do not share the same position);
3. Both the angle between beacons 1 and 2 ( $\lambda_{12}$ ) and the angle between beacons 3 and 1 ( $\lambda_{31}$ ) may be equal or greater than  $180^\circ$ ;
4. The algorithm works consistently inside, outside or over the triangle formed by the three beacons (except where any restriction common to all three-object triangulation algorithms apply).

The Generalized Geometric Triangulation algorithm works over the whole navigation plane, except for the few lines plotted in Fig. 7.

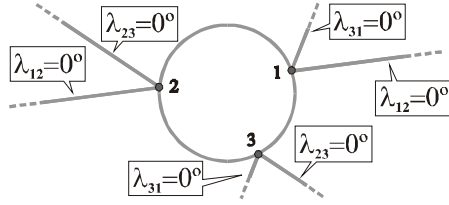


Fig. 7. Robot is unable to localize itself on these lines of the navigation plane

### B. Simulation Results

In order to validate the improved features of the Generalized Geometric Triangulation algorithm, two tests were made in a simulation environment. The code was written in Java 2 and compiled with the Java Development Kit (version 1.3) on a personal computer equipped with a Pentium III processor and running Windows 2000 Professional. Computations were performed using IEEE 754 Double floating-point format (64 bits). Graphics were plotted with Matlab (version 5.2).

In the first test, three beacons placed in known positions of a Cartesian plane are labeled 1, 2 and 3 in counterclockwise fashion. Beacon positions and resulting values of  $\sigma$  and  $\phi$  are shown in Fig. 8. The robot is initially placed at the origin of the referential system. Its heading is arbitrarily set to  $0^\circ$ . Angles  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  are computed, rounded to integers (to simulate the outputs of an instrument unable to measure angle increments below  $1^\circ$ ) and used as inputs of the Generalized Geometric Triangulation algorithm. This algorithm then computes both position and orientation of the robot. The distance between computed position and actual position of the robot is the position error. The absolute difference between computed orientation and actual orientation of the robot is the orientation error. The whole procedure is repeated for robot positions covering a  $100 \times 100$  square. Position and orientation errors obtained in each position are displayed using a grayscale such that a point becomes darker as the error verified in that position increases.

The second test is similar to the first one and the beacons are placed in the same positions of the Cartesian plane. However, they are labeled 1, 2 and 3 in clockwise fashion. Beacon positions and resulting values of  $\sigma$  and  $\phi$  are shown in Fig. 9.

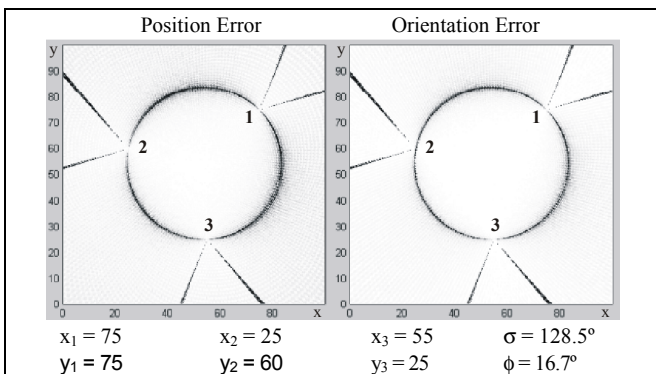


Fig. 8. Simulation results for beacons labeled in counterclockwise fashion

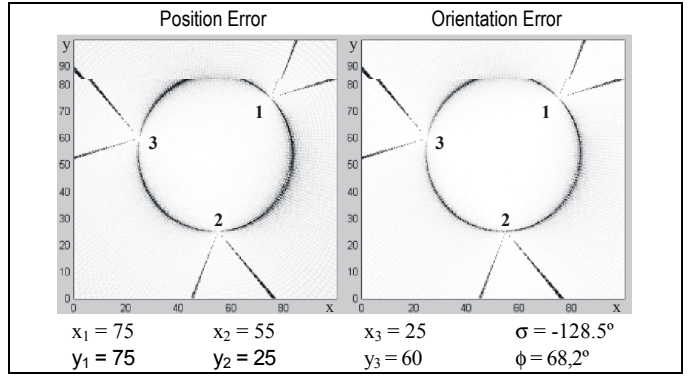


Fig. 9. Simulation results for beacons labeled in clockwise fashion

Results displayed in Fig. 8 and Fig. 9 agree with the analysis made in Section 4. The Generalized Geometric Triangulation algorithm works for both counterclockwise and clockwise beacon labeling and the areas unsuitable for robot localization are almost reduced to the few lines shown in Fig. 7. The dark straight stripes are formed by points of the square where the robot is unable to “see” three beacons. Every time this happens during the simulations, the action taken by the algorithm consists of returning a position error arbitrarily set to 20 and an orientation error arbitrarily set to  $30^\circ$ . Instead of being half straight lines, these stripes widen as the distance from the beacons increases. This is due to the errors contained in angles  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$ . It also can be seen that significant localization errors occur when the robot is over or close to the circumference defined by the three beacons. If angles  $\lambda_1$ ,  $\lambda_2$  and  $\lambda_3$  did not contain errors and computations were performed with an infinite number of significant digits, a  $0/0$  indetermination in the computation of  $\tau$  would occur when the robot is over the circumference but there would be no localization errors in its surroundings.

### C. Comparison with other triangulation algorithms

Table 1 summarizes some disadvantages of other triangulation algorithms when compared to the Generalized Geometric Triangulation Algorithm.

Table 1. Disadvantages of other triangulation algorithms

	Requires beacon ordering or special beacon configurations	Needs to solve equation systems	Has multiple solutions	Does not compute orientation	Requires position and orientation initial estimates
Iterative Search			x		
Newton-Raphson Iterative Method		x			x
Two Circle Intersection		x		x	
Three Circle Intersection		x		x	
Geometric Circle Intersection	x				
Position Estimator		x			
Imperial College B.N.S.	x			x	
Geometric Triangulation	x				



## V. CONCLUSIONS

All three-object triangulation algorithms for absolute self-localization of mobile robots navigating on a plane are subject to the following restrictions:

1. At least three distinguishable beacons must be "seen" by the robot;
2. Localization fails when the robot is over the circumference defined by the three beacons.

The Geometric Triangulation algorithm uses three distinguishable beacons and, in addition to the restrictions that are common to all three-object triangulation algorithms, has three important limitations:

1. The beacons must be labeled consecutively (1, 2, 3) in counterclockwise fashion;
2. The angle between beacons 1 and 2 ( $\lambda_{12}$ ) and the angle between beacons 1 and 3 ( $\lambda_{31}$ ) must be less than  $180^\circ$ . If this is not true, then the beacon labels must be shifted counterclockwise once or twice, until the requirement is met;
3. It works consistently only when the robot is within the triangle formed by the three beacons.

The Generalized Geometric Triangulation algorithm has been presented. It suffers from the restrictions that are common to all three-object triangulation algorithms but is free from the specific limitations of the Geometric Triangulation algorithm. Improvements are mainly achieved through a careful definition of the angles used by the new algorithm, which has the following features:

1. The three beacons may be randomly labeled 1, 2, and 3 (there is no need for consecutive labeling in counterclockwise fashion);
2. The three beacons may be placed anywhere in the plane (as long as two beacons do not share the same position);
3. Both the angle between beacons 1 and 2 ( $\lambda_{12}$ ) and the angle between beacons 3 and 1 ( $\lambda_{31}$ ) may be equal or greater than  $180^\circ$ ;
4. The algorithm works consistently inside, outside or over the triangle formed by the three beacons (except where any restriction common to all three-object triangulation algorithms apply).

Results of two tests made in a simulation environment agree with the previously made analysis: the Generalized Geometric Triangulation algorithm works for both counterclockwise and clockwise beacon labeling. Though the angles used as inputs of the algorithm contain errors, the areas of the plane unsuitable for robot localization are almost reduced to a few lines. Significant localization errors occur when the robot is over or close to the circumference defined by the three beacons. This suggests the need of providing the algorithm with a way of detecting these situations and also to develop a localization strategy that includes alternatives to simple three-object triangulation.

## VI. REFERENCES

- [1] McKerrow, P. J., *Introduction to Robotics*. Addison-Wesley Publishing Company, Inc., 1991.
- [2] Leonard, John J. and Durrant-White, Hugh F., *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, 1992.
- [3] Borenstein, J., Everett, H. R. and Feng, L., *Where am I? Sensors and Methods for Mobile Robot Positioning*, Technical Report, The University of Michigan, 1996.
- [4] NDC Automation Inc., *Advantage Lazerway™ - Laser Guidance vs. Inertial Guidance*, Technology Brief, USA (980129), 1998.
- [5] Mentor AGVS, *AGV Laser Guidance*. 1999.  
<http://www.mentoragvs.com/laser.html>
- [6] Cohen, Charles and Koss, Frank V., "A Comprehensive Study of Three Object Triangulation", *Mobile Robots VII*, SPIE Vol. 1831, 1992.
- [7] Everett, H.R., *Sensors for Mobile Robots*, A K Peters, June 1995.
- [8] Fuentes, O. et al., *Mobile Robotics 1994*, Technical Report 588, The University of Rochester Computer Science Department, Rochester, New York 14627, June 1995.
- [9] Stella, E. et al., "An Autonomous System for Indoor Structured Environment", *Intelligent Autonomous Systems*, U. Rembold et al. (Eds.), IOS Press, 1995.
- [10] Betke, Margrit and Gurvits, Leonid, "Mobile Robot Localization Using Landmarks", *IEEE Transactions on Robotics and Automation*, Vol. 13, No 2, April 1997.
- [11] Turenout, P. van and Honderd, G., "Navigation of a Mobile Robot", *Robotic Systems - Advanced Techniques and Applications*, Tzafestas, S. G. (ed.), p. 415-422. Kluwer Academic Publishers, 1992.
- [12] Frappier, G., Lemarquand, P. and Van den Bogaert, T., "Navigation and Perception Approach of PANORAMA Project", *Robotic Systems - Advanced Techniques and Applications*, Tzafestas, S. G. (ed.), p. 391-398. Kluwer Academic Publishers, 1992.
- [13] Borenstein, J., "Internal Correction of Dead-reckoning Errors With the Smart Encoder Trailer", *International Conference on Intelligent Robots and Systems (IROS'94) - Advanced Robotic Systems and the Real World*, Munich, Germany, September 12-16, 1994, pp 127-134.
- [14] Borenstein, J. and Feng, L., "UMBmark - A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots", Technical Report, The University of Michigan UM-MEAM-94-22, December 1994.
- [15] Crowley, James L., "Mathematical Foundations of Navigation and Perception For an Autonomous Mobile Robot", *Workshop on Reasoning With Uncertainty in Robotics*, University of Amsterdam, The Netherlands. 4 to 6 December 1995.
- [16] Borenstein, Johann and Feng, Liquiang, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots", *IEEE Transactions on Robotics and Automation*, Vol. 12, No 5, October 1996.
- [17] Borenstein, J., Everett, H. R., Feng, L. and Wehe, D., "Mobile Robot Positioning - Sensors and Techniques", *Journal of Robotic Systems*, Special Issue on Mobile Robots. Vol. 14, No. 4, pp. 231 - 249, 1997.
- [18] Soares, Manuel G. and Restivo, Francisco J., "Precisão em Navegação Contínua Terrestre e Marítima com DGPS e GPS", *3º Encontro Nacional do Colégio de Engenharia Electrotécnica*, Porto (Exponor), 5 e 6 de Junho de 1997.
- [19] Shoval, S., Zeitoun, I. and Lenz, E., "Implementation of a Kalman Filter in Positioning for Autonomous Vehicles, and its Sensitivity to the Process Parameters (abstract)", *Int. J. Adv. Manuf. Tecno.*, Vol. 13 (10), pp. 738-746, 1997.
- [20] Araújo, Paulo Ventura, *Curso de Geometria*, Gradiva, 1998.