Low cost self-localization system with two beacons

Héber Sobreira, A. Paulo Moreira, João Sena Esteves

Abstract— **The use of mobile robots for advertisement or entertainment purposes has been subject to several approaches without great success, mainly due to the high cost of the robots and their maintenance. This paper presents the self-localization system of an easy to use, low cost yet robust robotic platform which, expectedly, will have an increased value in the market. The low cost self-localization system of the platform is based on the detection of two infrared-emitting beacons (using a simple infrared sensor) and odometry. It allows computing both the position and the orientation of the robot in an external referential. To achieve this, an algorithm based on geometric considerations has been developed. Since it does not require too much computing power, it is implementable in low cost computing platforms such as small microcontrollers.**

I. INTRODUCTION

This paper describes the self-localization system developed and implemented on a small size service robot. The robot is being developed under a project called CleverRob (http://fe.up.pt/groundsys), which aims at the development of a very low cost and easy to use robot for advertising campaigns. It is intended to move autonomously, localizing itself within an area defined by beacons, with a localization system that does not require considerable computing power. One of the main requirements of the project is to achieve an easy to use robot. So, the end user should not have to place beacons in specific, carefully defined, positions and configure the robot with these positions. Instead, a single device with two beacons, capable of defining the navigation area of the robot by itself, is the only piece of equipment that has to be installed.

II. LOCALIZATION PROBLEM DEFINITION

The localization problem to be solved has the following definition:

- *Compute the position of the robot (x_r, y_r) and the orientation of the robot (θ_r) in an external referential defined in the navigation plane by two beacons (*Fig. 1*). The xx axis contains the known positions of the beacons and the yy axis is equidistant to these positions. θ_R is the angle formed by the positive xx semi-axis of the referential and the positive x_rob semi-axis fixed in the robot.*

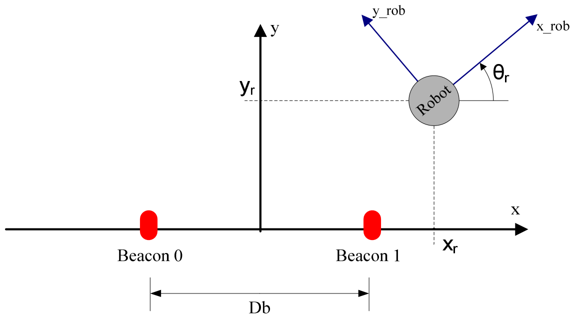

Fig. 1. Localization problem definition: compute the position of the robot (x_r, y_r) and the orientation of the robot (θ_r) in an external referential defined in the navigation plane by two beacons.

III. HARDWARE DESCRIPTION

The robot has a diameter of 33cm, a height of 15cm and differential traction, with two traction wheels and a free wheel. Regarding its self-localization, it is equipped with two encoders attached to the traction wheels and a sensor capable of detecting two infrared-emitting beacons placed on the navigation plane. This sensor (beacon sensor) is a remote-control type infrared receiver, placed inside a tube that is opaque to infrared radiation. This way, the sensor becomes directional: a beacon is only detected (and identified) when it lies on (or very close to) the longitudinal axis of the tube.

The beacons used are two sets of infrared emitters, placed on an aluminum bar, separated by a fixed distance Db (the results presented in this paper were obtained using Db = 97.5cm). In order to make the beacons distinguishable, one beacon (Beacon 0) emits a 1900Hz signal and the other (Beacon 1) emits a 950Hz signal. Both signals modulate a 44kHz carrier.

The left side of Fig. 2 represents the hardware used by the localization system: two encoders attached to the robot traction wheels, a beacon sensor on the robot and a set of two beacons placed on the navigation plane.

In the right side of Fig. 2, $\xi + 90^o$ is the orientation of Beacon 0 (it might have been Beacon 1 instead of Beacon 0) in relation to the positive x_rob semi-axis fixed in the robot. ξ is measured using the beacon sensor and odometry data obtained during a rotation performed by the robot.

H. Sobreira and A. P. Moreira are with the Department of Electrical and Computer Engineering, Faculty of Engineering, University of Porto, and with INESC-Porto, Portugal [dee09025, agm]@fe.up.pt.

J. S. Esteves is with the Department of Industrial Electronics, Engineering School, University of Minho, Portugal, sena@dei.uminho.pt.

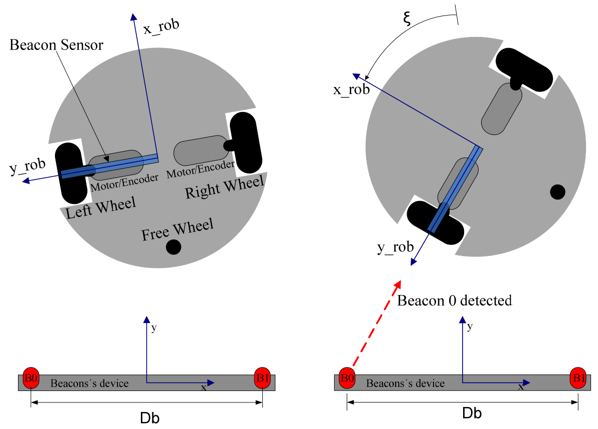

Fig. 2 Hardware used by the localization system localization (left image) and measurement of the orientation of Beacon 0 in relation to the positive x_rob semi-axis fixed in the robot (right image).

IV. LOCALIZATION APPROACH

The robot localization system has two modes of operation. The first mode uses only odometry data to compute the robot position. The second mode corrects this computed position periodically, using not only odometry data but also the beacon detector. The following sections describe only the second mode of operation of the localization system.

A *line of position* is the geometrical place of all points of the navigation plane where the robot may be according to a certain measurement. The localization approach used is based on the intersection of the lines of position obtained with the measurement of the distance between the robot and one beacon and the measurement of the angle between the line segments that join the robot and the beacons, as suggested by João Sena Esteves in [1].

A. Measuring the distance between the robot and a beacon

The distance between the robot and a beacon is measured indirectly. In order to compute its distance from beacon i (i may be either 0 or 1), the robot uses measurements made at two positions, P1 and P2 (Fig. 3). P2 is the actual position of the robot and P1 is a previous position.

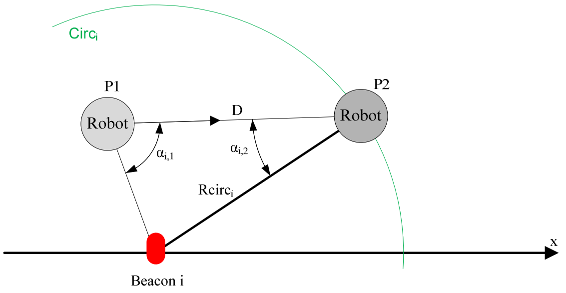

Fig. 3. Measuring the distance between the robot and beacon i.

The robot performs the following operations:

- At position P1, it performs a rotation (it is assumed that the robot position remains unchanged during this task) and detects beacon i using the beacon detector;
- It moves to a new position P2;
- At position P2, it performs another rotation and detects beacon i using the beacon detector.

Then, using odometry data, it is possible to compute D, $\alpha_{i,1}$ and $\alpha_{i,2}$ (Fig. 3). These quantities have the following definitions:

- $\alpha_{i,1}$ is the smallest angle defined by the line segments that join P1 to beacon i and P1 to P2;
- D is the distance between P1 and P2;
- $\alpha_{i,2}$ is the smallest angle defined by the line segments that join P2 to beacon i and P1 to P2.

To compute Rcirc$_i$, which is the distance between P2 and beacon i, the following equations (derived from the Law of Tangents [2]) are used:

$$K_i = \frac{tan[0.5*(180-2*\alpha_{i,1}-\alpha_{i,2})]}{tan[0.5*(180-\alpha_{i,2})]} \qquad \textbf{(1)}$$

$$Rcirc_i = D * \frac{1-K_i}{1+K_i} \qquad \textbf{(2)}$$

Rcirc$_i$ defines the line of position Circ$_i$, which is a circumference with radius Rcirc$_i$ and center at the position of beacon i.

B. Measuring the angle formed by the line segments joining the robot to each beacon

In order to measure the angle β (Fig. 4), formed by the line segments joining the robot to each beacon, the robot uses both odometry and the beacon sensor while performing a 360º rotation. The line of position defined by the measure of β is an arc of the circumference Circ$_2$, which contains the beacons positions and the robot position. Circ$_2$ has a radius Rcir$_2$ and center at (x_2, y_2).

By definition, x_2 = 0. To compute y_2 and Rcirc$_2$, the following equations (derived from the Law of Sines [2]) are used:

$$Rcirc_2 = \frac{Db}{2*\sin(\beta)},\ \beta \in [0°, 180°] \qquad \textbf{(3)}$$

$$If\ \ \beta \in [0°, 90°[\ \ then\ \ y_2 = \sqrt{Rcirc_2{}^2 - \left(\frac{Db}{2}\right)^2} \qquad \textbf{(4)}$$

$$If\ \beta \in [90°, 180°]\ \ then\ \ y_2 = -\sqrt{Rcirc_2{}^2 - \left(\frac{Db}{2}\right)^2} \qquad \textbf{(5)}$$

C. *Computing the robot position*

The position of the robot is at the intersection of $Circ_i$ and $Circ_2$ ($Circ_i$ may be either $Circ_0$ or $Circ_0$), which is computed using the method found in [3]. Fig. 5 shows an example using the intersection of $Circ_1$ and $Circ_2$.

The robot position is obtained from the intersection of two circumferences which, in general, consists of two points. This leads to two possible mathematical solutions for the robot position.

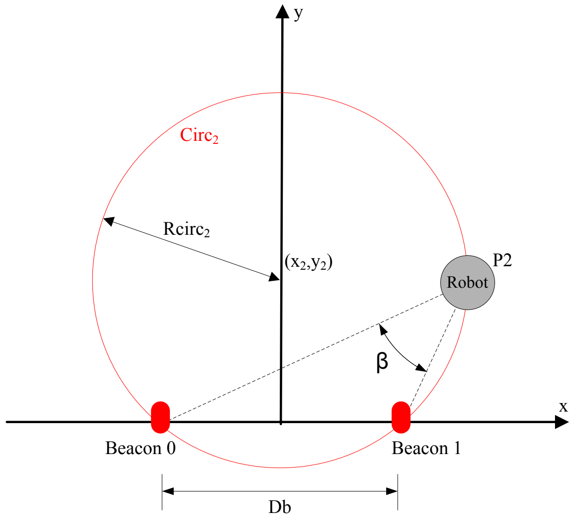

Fig. 4. Measuring the angle formed by the line segments joining the robot to each beacon.

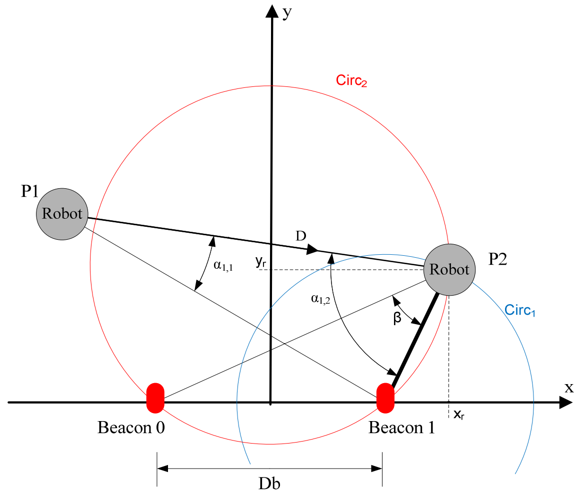

Fig. 5. Computing the robot position using the distance between the robot and Beacon 1.

There are some techniques that may be used in order to overcome such ambiguity:

- Considering the navigation restrictions of the robot, it is possible to reject any mathematical solution such that $y_R<0$.
- One of the two circumferences is the line of position given by the distance between the robot and a beacon. It is possible to use, also, the circumference given by the distance between the robot and the other beacon. Given two possible position solutions, this circumference will generally pass only at (or close from) the true position of the robot.
- Using odometry data, the robot may compute an estimate of its position. Circumferences intersections occurring too far from the computed position may be excluded [1].

D. *Computing the robot orientation*

After the robot position has been computed, its orientation θ_r may be obtained from the measurement of the angle θ_{ri} (Fig. 6) formed by the positive x_rob semi-axis and the line segment that joins the robot and beacon i (i may be 0 or 1) [1, 4, 5, 6], using the following equations:

$$\theta_r = \theta_{ri} - \theta_i \tag{6}$$

$$\theta_i = \mathrm{Atan2}(-\mathrm{y_r}, \mathrm{x_{bi}} - \mathrm{x_r}) \tag{7}$$

Atan2(y,x) implements Arctangent function regarding the quadrants: it distinguishes τ from τ+π.

V. SIMULATION RESULTS AND REAL WORLD IMPLEMENTATION

The results shown in this section were obtained with the *SimTwo* simulation platform [7], developed by Paulo Costa from *5dpo Robotic Team* [8]. Both the robot movements and its sensors readings are simulated by the platform. Encoders have finite resolution and beacons are not punctual.

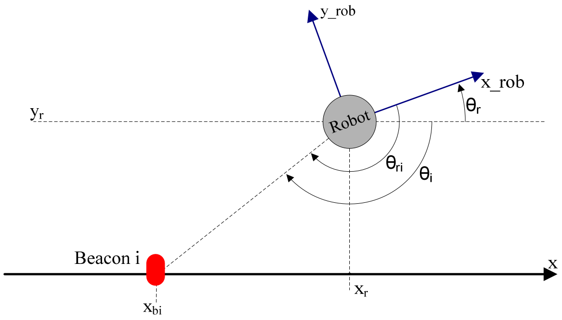

Fig. 6. Computing the robot orientation.

Fig. 7 and Fig. 8 were obtained from the graphical interface of the robot control program. They correspond to the results achieved in a displacement from P1 to P2. The navigation area of the robot is represented. Each small square represents 100cm^2.

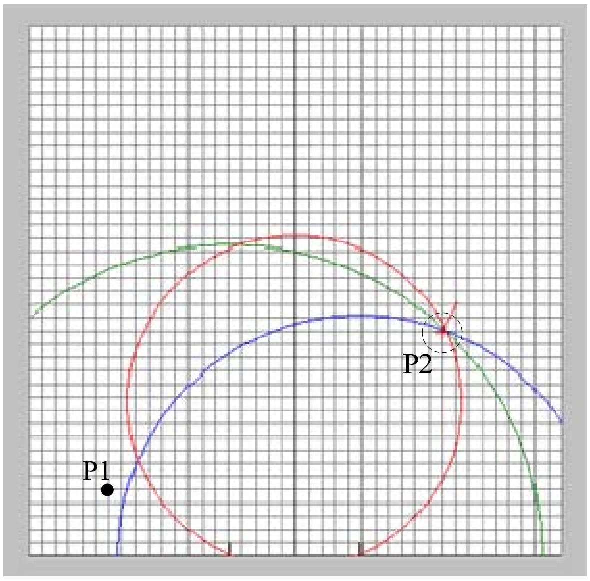

Fig. 7. Position lines obtained when the robot moves from P1 (x=-140cm, y=50cm) to P2 (x=110cm, y=170cm).

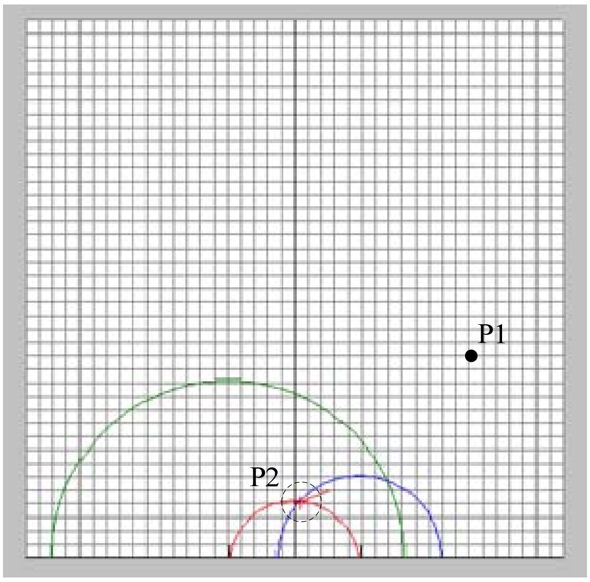

Fig. 8. Position lines obtained when the robot moves from P1 (x=130cm, y=150cm) to P2 (x=5cm, y=40cm).

The green and blue lines of position (Circ$_0$ and Circ$_1$, respectively) are obtained from the measures of the distance between the robot and Beacon 1 and the distance between the robot and Beacon 2, respectively. The red lines of position (Circ$_2$) are obtained from the measure of β.

The lines of position Circ$_0$, Circ$_1$ and Circ$_2$ in Fig. 7 have an intersection close to the true position of the robot.

In Fig. 8 there is a noticeable error in the measurement of the distance between the robot and Beacon 0. This is due to the fact that it is not possible to compute the distance between the robot and a beacon for all pairs of P1 and P2 points in the navigation plane. It is not possible to perform such a computation if P1, P2 and the beacon position are collinear. Moreover, even if this situation does not occur, computing the robot position using the distance between the robot and one of the beacons does not lead to the solution obtained using the distance between the robot and the other beacon. This is due to measurement errors, which produce computed position errors that depend not only on measurement errors values but also on the positions of the navigation plane where the measurements are made [1,4]. It is important to quantify measurement uncertainties in order to obtain the uncertainties associated with the computed values of position and orientation of the robot. Position and orientation uncertainties also depend both on measurement uncertainties and on the positions of the plane where the measurements are made [1].

A prototype of CleverRob has already been implemented. It uses the described localization approach and a method for estimating uncertainties (not explained in this paper). The prototype has proven its functionality in several events, namely in *Portugal Tecnológico* (Technological Portugal) exhibition that took place in *FIL – Feira Internacional de Lisboa* (Lisbon International Fair), October 7 – 10, 2009 (Fig. 9). The localization system allowed the robot to navigate autonomously within a limited rectangular area with 4m x 3m, defined by the beacons, without ever crossing its boundary. In laboratory experiments, the errors in the computed values of position and orientation never exceeded the respective uncertainties estimated by the localization system.

VI. CONCLUSIONS AND FUTURE WORK

A low-cost and simple to install self-localization system for mobile platforms has been presented. The system is based on the robot odometry, an infrared sensor and two beacons which are two sets of infrared emitting LEDs.

Robot position is obtained via the measurement of the distance between the robot and one beacon and the measurement of the angle between the line segments that join the robot and the beacons. The distance between the robot and one beacon is measured indirectly.

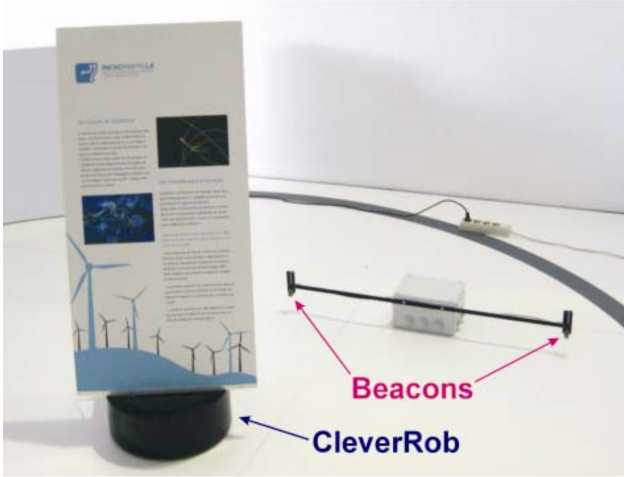

Fig. 9. CleverRob at *Portugal Tecnológico* (Tchnological Portugal) exhibition, in 2009.

Robot orientation is obtained from the computed position and the measurement of the angle formed by a reference semi-axis fixed on the robot and the line segment that joins the robot and one of the beacons. The localization algorithm that has been developed is suitable to be implemented on a low-cost computing platform such as a small microcontroller.

The robot needs to perform measurements between two positions of the navigation plane in order to compute its distance to a beacon and considerable distances must be traveled in order to achieve a low error in this computation. Future work includes the study of trajectory management when the robot is localizing itself, in order to achieve a faster and more robust localization process.

Future work also includes analyzing the possibility of making the beacon sensor capable of measuring the strength of the received signal. This will give the localization system additional information about the distance to the beacons, making the localization process faster. Robustness of this process will also be increased since it will be possible to detect multipath propagation.

VII. REFERENCES

[1] João Sena Esteves; *Metodologia de Autolocalização Absoluta em Ambientes Quase-Estruturados;* Ph.D Thesis, University of Minho, Guimarães, July 28, 2005.

[2] Steven George Krantz; *Dictionary of Algebra, Arithmetic, and Trigonometry – a volume in the CRC Comprehensive Dictionary of Mathematics;* CRC Press, 2001.

[3] Paul Bourke; *Intersection of Two Circles;* April 1997 [online, accessed on June 20, 2009]. http://local.wasp.uwa.edu.au/~pbourke/geometry/2circle/

[4] João Sena Esteves, Adriano Carvalho and Carlos Couto; *Position and Orientation Errors in Mobile Robot Absolute Self-Localization Using an Improved Version of the Generalized Geometric Triangulation Algorithm* (full text); ICIT 2006 – 2006 IEEE International Conference on Industrial Technology, Mumbai, India, December 15-17, 2006.

[5] João Sena Esteves, Adriano Carvalho and Carlos Couto; *An Improved Version of the Generalized Geometric Triangulation Algorithm* (full text); II SELASI – II European-Latin-American Workshop on Engineering Systems, Faculdade de Engenharia da Universidade do Porto, Porto, June 21-23, 2006.

[6] João Sena Esteves, Adriano Carvalho and Carlos Couto; *Generalized Geometric Triangulation Algorithm for Mobile Robot Absolute Self-Localization* (full text); ISIE 2003 – 2003 IEEE International Symposium on Industrial Electronics, Rio de Janeiro, Brasil, June 9-12, 2003.

[7] Paulo Costa; *SimTwo* [online, accessed on January 14, 2010]. http://paginas.fe.up.pt/~paco/wiki/index.php?n=Main.SimTwo

[8] *5dpo Robotic Team;* Department of Electrical and Computer Engineering (DEEC) of the Faculty of Engineering from the University of Porto (FEUP) and the Institute for Systems and Robotics – Porto (ISR-P), Portugal [online, accessed on January 14, 2010]. http://paginas.fe.up.pt/~robosoc/en/doku.php