An Improved Version of the Generalized Geometric Triangulation Algorithm

João Sena Esteves  
Department of Industrial Electronics  
University of Minho  
4800-058 Guimarães, PORTUGAL  
sena@dei.uminho.pt

Adriano Carvalho  
Dept. of Electrical Engineering and Computers  
Faculty of Engineering, University of Porto  
4200-465 Porto, PORTUGAL

Carlos Couto  
Department of Industrial Electronics  
University of Minho  
4800-058 Guimarães, PORTUGAL

Abstract – Triangulation with active beacons is widely used in the absolute localization of mobile robots. The original Generalized Geometric Triangulation algorithm suffers only from the restrictions that are common to all algorithms that perform self-localization through triangulation. But it is unable to compute position and orientation when the robot is over the segment of the line that goes by beacons 1 and 2 whose origin is beacon 1 and does not contain beacon 2. This paper presents an improved version of the algorithm that allows self-localization even when the robot is over that line segment.

I. INTRODUCTION

Localization is the process of finding both position and orientation of a vehicle in a given referential system [1], [2], [3], [4], [5], [6], [7], [8], [9]. Triangulation with active beacons is a robust, accurate, flexible and widely used method of absolute localization [1], [10]. Self-localization through triangulation is based on the measurement of the bearings of the robot relatively to beacons placed in known positions. When navigating on a plane, three distinguishable beacons are required - and usually enough - for the robot to localize itself. In 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 [11]. An additional arc between beacons 1 and 3 is defined by \( \lambda_{31} \). The robot is in the intersection of the two arcs. Several algorithms of self-localization through triangulation are described in [1], [2], [5], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21] and [22].

Two restrictions are common to all algorithms that perform self-localization through triangulation [2]:
1. The robot must “see” at least three distinguishable beacons to localize itself in a plane. All areas of the plane with less than three visible beacons are unsuitable for robot localization;
2. Localization is not possible if the robot is over the circumference defined by three non-collinear beacons (the intersection of the arcs shown in Fig. 1 is another arc, not a point) or over the line defined by three collinear beacons.

The Geometric Triangulation algorithm described in [17] uses three distinguishable beacons that must be ordered in a particular way. According to the authors of that paper, “the algorithm works consistently only when the robot is within the triangle formed by the three landmarks.”

The Generalized Geometric Triangulation algorithm [2] does not require beacon ordering and suffers only from the two restrictions that are common to all algorithms that perform self-localization through triangulation. But it is unable to compute position and orientation when the robot is over the segment of the line that goes by beacons 1 and 2 whose origin is beacon 1 and does not contain beacon 2. It was assumed that, when a beacon becomes between the robot and another beacon, the closest beacon hides the farther one or else the goniometer is not able of simultaneously detecting more than one beacon. Any of those situations prevents self-localization. However, the impediment is due to the technology used, not triangulation itself.

Section II presents an improved version of the Generalized Geometric Triangulation algorithm [1]. It works over the segment of the line that goes by beacons 1 and 2 whose origin is beacon 1 and does not contain beacon 2. Position and orientation errors are defined on Section III. Simulation results [1] that validate the improved version of the Generalized Geometric Triangulation algorithm are shown in Section IV. Conclusions and references are presented in Section V and Section VI, respectively.

II. THE IMPROVED GENERALIZED GEOMETRIC TRIANGULATION ALGORITHM

The Generalized Geometric Triangulation algorithm uses (Fig. 2) three distinguishable beacons, 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. \(L_{13}\) 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.

Line 14 of the algorithm was not present in the original version of the algorithm. It is required only when the robot is over the segment of the line that goes by beacons 1 and 2 whose origin is beacon 1 and does not contain beacon 2.

Beacons are also called landmarks by some authors.

III. POSITION AND ORIENTATION ERRORS

In general, the computed position does not coincide with the true position of the robot and the computed orientation also does not coincide with its true orientation. There are a position error and an orientation error, which in this paper have the following definitions:

- the position error $\Delta P_R$ (Fig. 3) is the distance between the computed position $P_R$ and the true position $P_0$;
- the orientation error $\Delta \theta_R$ is the modulus of the difference between the computed orientation $\theta_R$ and the true orientation.

Measurement errors constitute the main source of position and orientation errors, which magnitude also depends on the position of the robot relatively to the beacons.

IV. SIMULATIONS RESULTS

In order to validate the improved version of the Generalized Geometric Triangulation algorithm, five tests were made in a simulation environment. The code was written in Java 2. It was used the Java 2 SDK, Standard Edition (version 1.3 for Windows), upgraded with Java 3D (version 1.2.1 Beta, for Win32/DirectX), on a personal computer equipped with an Intel Pentium III (995MHz) processor and running Windows XP (version 5.1.2600). Graphics were plotted with Matlab (version 5.2).

![Fig. 1. Geometric Triangulation algorithm.

Fig. 2. Generalized Geometric Triangulation.

2 The position of a robot with non-negligible dimensions is the position of one of its points.
Simulations are performed in a square shaped area of the navigation plane, using the following beacon configurations:
- non-collinear beacons ordered in counterclockwise fashion;
- non-collinear beacons ordered in clockwise fashion;
- collinear beacons, beacon 1 between beacons 2 and 3;
- collinear beacons, beacon 3 between beacons 1 and 2;
- collinear beacons, beacon 2 between beacons 1 and 3.

In each test, three beacons labeled 1, 2 and 3 are placed in known positions of a Cartesian plane. Beacon positions are printed close to the results of each test. A robot is placed at the origin of the referential system. Its orientation is arbitrarily set to a value between -180° and 180°. Then, a four-step sequence (Fig. 4) is performed:

I. Angles $\lambda_1$, $\lambda_2$ and $\lambda_3$ are computed from a priori known beacons and robot positions.
II. Angles $\lambda_1$, $\lambda_2$ and $\lambda_3$ are rounded to integers, simulating the outputs of a digital goniometer with resolution $\rho$ equal to 1° (measurement uncertainty $\pm \Delta \lambda$ equal to $\pm0.5^\circ$).
III. A priori known beacon positions and the rounded values of $\lambda_1$, $\lambda_2$ and $\lambda_3$ are used as inputs of the Generalized Geometric Triangulation algorithm, which computes both position and orientation of the robot.
IV. Position and orientation errors are computed from a priori known robot position and orientation and their computed values.

The four steps are repeated for robot positions covering a 100 x 100 square. Position increments of 0.1 are made in both $x$ and $y$ directions. In each point, robot orientation is arbitrarily set to a value between -180° and 180°. Double precision (64 bits) has been used in all four steps.

Position and orientation errors obtained in each position are plotted in 2D graphics (Fig. 5) using a grayscale such that a point becomes darker as the error verified in that position increases. The axes of position error graphics are labeled in the same length units. The z-axes of orientation error graphics are labeled in degrees.

Simulations results show that the improved version of the Generalized Geometric Triangulation algorithm works with the five beacon configurations used. As expected, significant position and orientation errors occur when the robot is over or close to the circumference defined by three non-collinear beacons or the line defined by three collinear beacons.

V. CONCLUSIONS

Two restrictions are common to all algorithms that perform self-localization through triangulation:
1. The robot must “see” at least three distinguishable beacons to localize itself in a plane. All areas of the plane with less than three visible beacons are unsuitable for robot localization;
2. Localization is not possible if the robot is over the circumference defined by three non-collinear beacons or over the line defined by three collinear beacons.

The Generalized Geometric Triangulation algorithm suffers only from the two restrictions mentioned above. But it is unable to compute position and orientation when the robot is over the segment of the line that goes by beacons 1 and 2 whose origin is beacon 1 and does not contain beacon 2. It was assumed that, when a beacon becomes between the robot and another beacon, the closest beacon hides the farther one or else the goniometer is not able of simultaneously detecting more than one beacon. Any of those situations prevents self-localization, but the impedance is due to the technology used, not triangulation itself.

An improved version of Generalized Geometric Triangulation algorithm was presented. It works over the segment of the line that goes by beacons 1 and 2 whose origin is beacon 1 and does not contain beacon 2.

Simulations were performed using non-collinear beacons ordered in counterclockwise fashion, non-collinear beacons ordered in clockwise fashion, collinear beacons with beacon 1 between beacons 2 and 3, collinear beacons with beacon 3 between beacons 1 and 2, and collinear beacons with beacon 2 between beacons 1 and 3. Results show that the improved version of the Generalized Geometric Triangulation algorithm works with the five beacon configurations used. Significant position and orientation errors only took place when the robot was over or close to the circumference defined by three non-collinear beacons or the line defined by three collinear beacons.
Fig. 5. Position and orientation errors.
VI. REFERENCES


