Cooperative Localization by Time-of-Flight Trilateration – Fast, Robust or Optimal

S. Genchev

Key Words: *Absolute positioning; trilateration; cooperative robotics; numerical optimization.*

Abstract. *This paper presents several novel trilateration methodologies for localization in teams of cooperative robots based on Time-of-Flight (ToF) measurements between the robots. One real-time, low-computation technique and other two optimal, computation costly models are introduced, compared and analysed. All methods and their statistical characteristics have been presented analytically and numerically compared by graphical simulations.*

1. Introduction

Generally, one's current position could be determined by a set of measurements relative to its previous known position, or relative to other objects from the same environment, the locations of which are already known [1]. In the first case, the position information is calculated indirectly by integrating data retrieved from rapid interceptive sensors and therefore the induced error is accumulated in time. In order to correct these errors periodically by Kalman filtering [4] or other sensor fusion method an absolute source for localization,is needed e.g. a GPS device or other similar system [2,3]. But when working in unknown indoor, not-structured environments, such absolute knowledge is unavailable. A way around this limitation is the cooperative localization recently emerged from the multi-robot field. The sensing capabilities of a single robot may be very restricted, but there are multiple robots working together in the same environment, and sharing sensing information [5,6]. A team of small robots with limited capabilities could easily perform tasks, like exploration and map building [7,8], otherwise very complex for a single, spatially limited robot, no matter how advanced are its sensors. The localization in the team of robots is generally achieved by measuring their relative position and/or orientation and communicating with each other. A lot of solutions exist in the literature, but they are mostly designed only for specific tasks and special cases [9,10,11,12]. Furthermore in noisy environments most of the proposed techniques are too complex, slow or unstable.

Nowadays the trilateration (and its derivatives) is the most used technique for absolute positioning because of its simplicity and the possibility to be implemented practically in any system. While the triangulation uses angles to calculate the unknown coordinates, the trilateration requires distance measurements between the object and several previously placed beacons in the working space, which location is very accurate. The required distances are obtained by measuring the Time-of-Flight (ToF) of a wave travelling between the object and the beacons [13, 14, 15]. The major problems are the fluctuations in the wave speed during its travel, the multi-paths by obstacles and the clock synchronization of receivers and emitters. The wave speed is varying mainly because of temperature changes in the media, which could be determined and corrected by periodical wave speed calibration according to temperature measurements. The wave multi-paths are hard to detect, but their effect may be efficiently reduced by using more beacons than required for performing the trilateration. Finally the clock synchronizations may be avoided by measuring the Time-Difference-of-Arrival (TDoA) [16,17], but it is highly non-linear and cannot be modelled easily to behave properly in noisy environments.

One of the main drawbacks of the absolute positioning systems is the necessity of structured or partially-structured working areas, i.e. there must be previously build landmarks (signs for artificial vision) or beacons (antennas for trilateration or triangulation). In cooperative robotics, this limitation may be avoided by giving the role of beacons to some of the robots in the team, thus creating a movable localization framework not depending on the working area. In this scenario, a given number of well positioned robots are operating like antennas in order to navigate the rest of the team. The procedure continues until some criterion is fulfilled and other robots are chosen to continue the role of beacons or until the final goal of the team is reached. Thus many sophisticated AI related tasks for team management arise here, e.g. beacon configuration formation, dynamic trajectory building, obstacle avoidance etc. Moreover, the mobile beacon's accuracy is not as good as the accuracy of static beacons and additional error sources arise from the uncertainties in the beacon positions. These uncertainties may be reduced if the robots serving as beacons are not in motion during the trilateration process. Also as already mentioned the robustness towards the effect of all errors may be improved by increasing the number of beacons [18]. But using big number of robots has also several disadvantages, like the risk of communication or computation overload. The communication is a network protocol problem and will not be discussed here, while the computation cost is directly linked to the efficiency of the mathematical model. It has to be noted that in a big team of cooperative robots, their system resources are very limited and

Acknowledgments

This work was supported by National Ministry of Science and Education of Bulgaria under Contract BY-I-302/2007: „Audio-video information and communication system for active surveillance cooperating with a Mobile Security Robot".

the working algorithms have to be simple enough to be supported, while producing precise and fast results.

Given the above premises, the main objective of the article is to propose new methods for absolute cooperative localization that are robust in noisy environments, simple to be used in real-time by resource-limited platforms, fast enough to allow the implementation of complex high-level AI algorithms for team management and finally producing optimal results. It can be seen from the following sections that the realisation of such a method having all these qualities is rather fictional.

2. Optimal Multi-Beacon Trilateration

Consider the case where the aim is to find a point R_0 with coordinates x_0, y_0 and z_0, given the knowledge of the measured distances l_i to n known beacons with positions R_i having coordinates x_i, y_i, z_i, in the same three-dimensional Cartesian coordinate system. The mathematical relation between each distance l_i, each beacon position R_i and the unknown R_0 may be expressed by the following equation:

$$(1) \quad \|R_0 - R_i\| = l_i; \quad i = 1, 2, ..., n.$$

Because R_i and l_i are parameters with stochastic character, it is assumed that theirs measured values are actually the expected values, which firstly induce error in the model and its final results. The obtained non-linear, redundant system is resolved by optimization in most of the similar problems found in literature. The solution of the system minimizes the following error function

$$(2) \quad S(R) = \sum_{i=1}^{n} c_i \left(\|R - R_i\| - l_i \right)^2$$

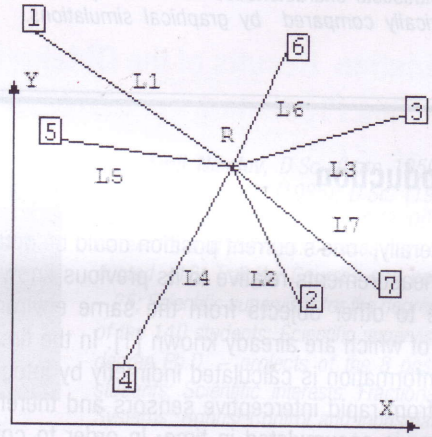
where c_i is a weight coefficient related to the beacon precision. It is necessary to ignore erroneous beacons and give priority to more precise ones. If the covariance matrix giving the probabilistic information for each beacon position and range measurements is noted by P_i then these coefficients may be:

$$(3) \quad c_i = \left(\sum_{j=1}^{n} \frac{1}{trace(P_j)} \right)^{-1} \frac{1}{trace(P_i)} \Rightarrow \sum_{i=1}^{n} c_i = 1;$$

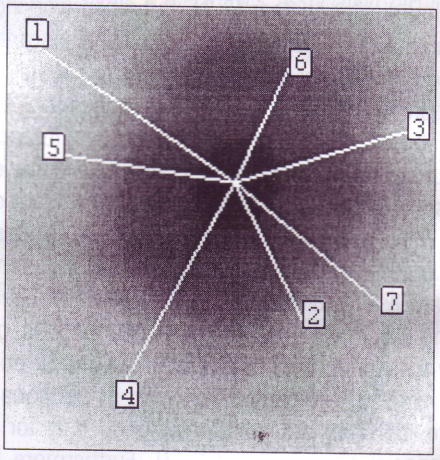
$$(4) \quad P_i = \begin{bmatrix} \sigma_{X_i}^2 & \sigma_{X_i Y_i} & \sigma_{X_i Z_i} & 0 \\ \sigma_{Y_i X_i} & \sigma_{Y_i}^2 & \sigma_{Y_i Z_i} & 0 \\ \sigma_{Z_i X_i} & \sigma_{Z_i Y_i} & \sigma_{Z_i}^2 & 0 \\ 0 & 0 & 0 & \sigma_{L_i}^2 \end{bmatrix}$$

Here position and range data are not supposed to be correlated. There may be correlation only between dimensions in the position information.

Before presenting some practical examples it has to be noted that in most of the experiments included in this paper, the beacon positions are chosen randomly in order to obtain more objective results. However in a real world application the beacon configuration is really crucial for precise and efficient trilateration and thus has to be composed in a suitable way. In fact the trajectories of the moving beacons have to be computed in accordance with the potential places where they could form better beacon framework for navigating other robots. On *figure 1a* a random configuration of seven beacons (shown by numbers enclosed in cases) is given with equal levels of uncertainty which are placed randomly on a square area with dimensions 100x100 m. *Figure 1b* shows two-dimensional *intensity map* with the associated values of S in the hole area, respecting the chosen position of R_0, which is the minimum of S (the darker zones mean lower error values, the darkest point is the minimum level).



(a)



(b)

Figure 1. a) Random two-dimensional configuration of 7 beacons (shown by numbers enclosed in cases). The distances to the unknown point R_0 are being represented by lines; **b)** Associated intensity map with the levels of the error function S given the position of R_0 (darker zones means lower error).

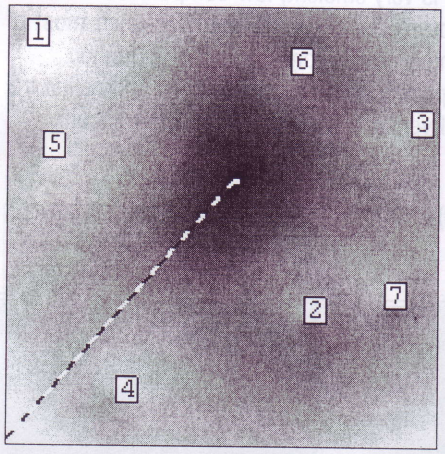
The intensity maps are just a more practical way for presenting three-dimensional experimental data. Instead of coordinate values, the third data set is shown by normalized gray intensities ranging between black and white. The mapping between the intensity and the data may be linear, logarithmic or other.

The error function has concave character around R_0 which allows to use the gradient of S that is zero in the minimum location.

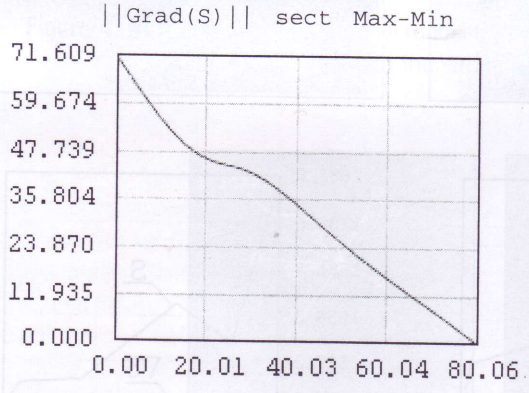
$$(5)\quad \nabla S = \left[\frac{\partial S}{\partial x}\ \frac{\partial S}{\partial y}\ \frac{\partial S}{\partial z}\right]^T = 2\sum_{i=1}^{n} c_i \left(1 - \frac{l_i}{\|R - R_i\|}\right)(R - R_i);$$

$$(6)\quad \nabla S(R_0) = 0.$$

On *figure 2a* an intensity map for the gradient module of S based on the same configuration is presented as in *figure 1*. A transversal section between the maximum and minimum values is drawn on *figure 2b*:



(a)



(b)

Figure 2. a) Intensity map for the gradient module of the error function. Transversal section between maximum and minimum values marked by doted line; **b)** Section of the gradient module between maximum and minimum values.

Unfortunately (6) is too complex to be analytically calculated and is resolved rather by non real-time numerical methods. If $J_{\nabla S}$ is the Jacobian matrix associated to the gradient of S in R_0, the value of R_0 may be approximated by performing several iterations, until some convergence criterion is reached:

$$(7)\quad R_{0,k+1} = R_{0,k} - J_{\nabla S}^{-1}(R_{0,k})\nabla S(R_{0,k});$$

$$(8)\quad J_{\nabla S}(R_{0,k}) = \left[\begin{array}{ccc} \frac{\partial^2 S}{\partial x^2} & \frac{\partial^2 S}{\partial x \partial y} & \frac{\partial^2 S}{\partial x \partial z} \\ \frac{\partial^2 S}{\partial x \partial y} & \frac{\partial^2 S}{\partial y^2} & \frac{\partial^2 S}{\partial y \partial z} \\ \frac{\partial^2 S}{\partial x \partial z} & \frac{\partial^2 S}{\partial y \partial z} & \frac{\partial^2 S}{\partial z^2} \end{array}\right]_{R_{0,k}}.$$

As convergence criteria may be used the difference between two successive steps, given the fact that it tends towards zero when approaching the searched point.

$$(9)\quad \Delta R_0 = \|R_{0,k+1} - R_{0,k}\| \xrightarrow{k\to\infty} 0.$$

Even if the convergence is very fast, the computation cost may be considerable for large number of beacons. Furthermore, the main difficulty in computing (6) arises from the fact that the error function may have several local minimums. Therefore, incorrect solutions are to be expected, if the initial guess for R_0 is too far from the real point. In this case there is a big possibility to converge towards an incorrect local minimum, or do not converge at all if the noise levels are too high. The more intuitive choice for initial guess is the previously calculated solution. Better estimations may be provided by other sensors e.g. an inertial navigation system working in parallel. *Figure 3* presents three cases of convergence using the configuration from *figure 1* starting the computation from different points around the space with convergence criterion of 1 mm to be reached. The intensity maps are from the error levels around the real solution, where is the darkest zone. The trajectory of the guessed points is marked by a line with changing intensity for better visibility. The corresponding graphs are giving an idea of the error and its gradient during the way of the approximated point towards the real value. The next case on *figure 4* shows an example for a local minimum which capture the approximating point, and the final result is not the real R_0 but a point with strong likelihood to be the solution.

Even if the iteration is successful we still do not have any statistical information about the obtained solution. Such knowledge is required for sensor fusion with other sources of position information and thus it's crucial to estimate it correctly. All the techniques based to minimization of S share almost the same statistical behavior. In order to study it consider the errors due to the beacon uncertainties in the already considered case of error minimization. Denote by δR the solution error and by δB_i

the input information error from each beacon:

$$(10)\quad \delta R = R_{estim} - R_{real} \Rightarrow R_{real} = R_{estim} - \delta R\ ;$$

$$(11)\quad \delta B_i = B_{i,measure} - B_{i,real}\ ;$$

$$(12)\quad B_i = \begin{bmatrix} x_i & y_i & z_i & l_i \end{bmatrix}^T .$$

Because of the presence of δB_i, the Taylor series of (5) is modified with one additional term, which is the solution error, after convergence of (7) is achieved.

$$(13)\quad \Rightarrow \nabla S(R) = \nabla S(R_0) + J_{\nabla S}(R_0)(R - R_0) + \sum_{i=1}^{n} \left.\frac{\partial \nabla S}{\partial B_i}\right|_{R_0} \delta B_i$$

$$(14)\quad \left.\frac{\partial \nabla S}{\partial B_i}\right|_{R_0} = \begin{bmatrix} \frac{\partial^2 S}{\partial x \partial x_i} & \frac{\partial^2 S}{\partial x \partial y_i} & \frac{\partial^2 S}{\partial x \partial z_i} & \frac{\partial^2 S}{\partial x \partial l_i} \\ \frac{\partial^2 S}{\partial y \partial x_i} & \frac{\partial^2 S}{\partial y \partial y_i} & \frac{\partial^2 S}{\partial y \partial z_i} & \frac{\partial^2 S}{\partial y \partial l_i} \\ \frac{\partial^2 S}{\partial z \partial x_i} & \frac{\partial^2 S}{\partial z \partial y_i} & \frac{\partial^2 S}{\partial z \partial z_i} & \frac{\partial^2 S}{\partial z \partial l_i} \end{bmatrix}_{R_0}$$

$$(15)\quad \Rightarrow R = R_0 - J_{\nabla S}^{-1}(R_0)\nabla S(R_0) + \sum_{i=1}^{n} J_i \delta B_i\ ;$$

$$(16)\quad J_i = -J_{\nabla S}^{-1} \left.\frac{\partial \nabla S}{\partial B_i}\right|_{R_0} .$$

Finally after substitutions, the solution error and its covariance matrix are well estimated approximately by

$$(17)\quad \delta R \approx \sum_{i=1}^{n} J_i \delta B_i$$

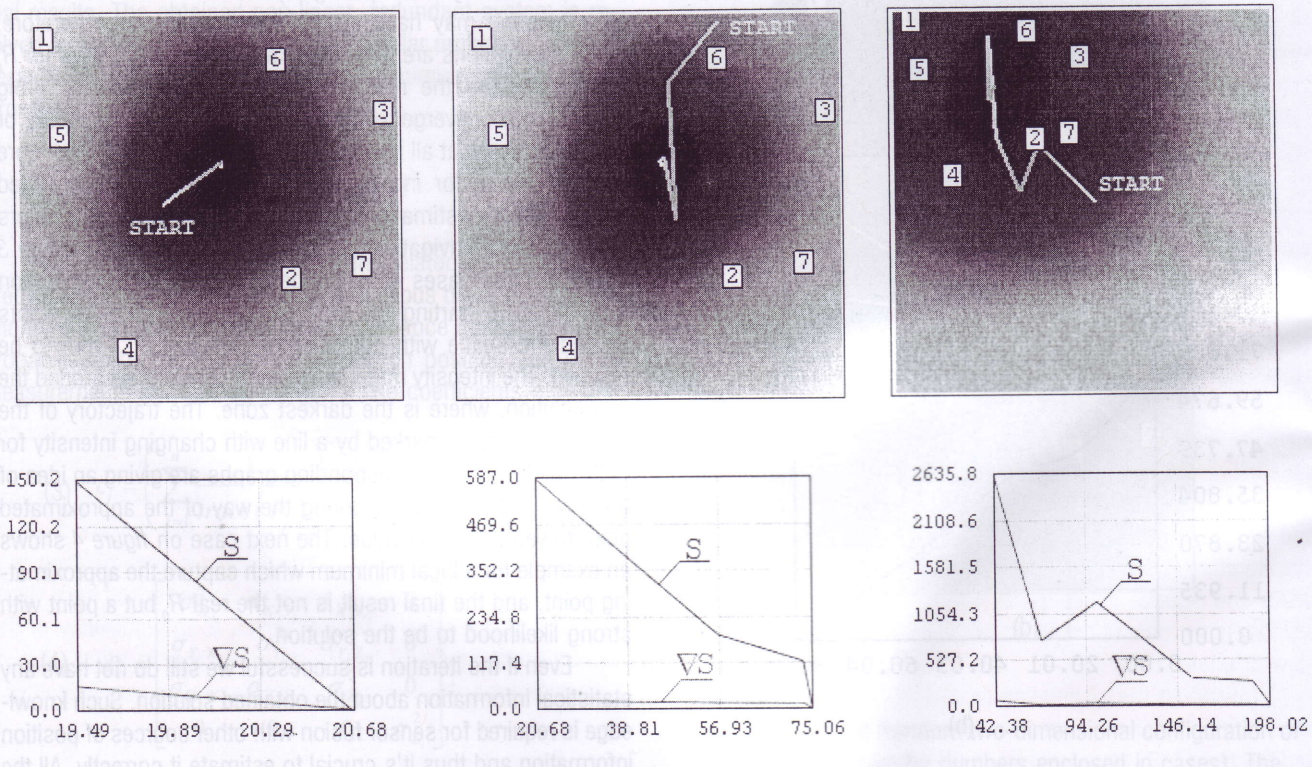
$$(18)\quad P \approx \sum_{i=1}^{n} J_i P_i J_i^T$$

2.1. Angle-Walk Trilateration

The angle-walk trilateration is one of the new techniques introduced for the first time with this paper (as far as is known). It is very efficient, low-computation solution for finding the location of the error minimum R_0, when most of the used beacons have very wrong position information.

Suppose there is a guess for R_0 and define the following directional vector associated with each beacon.

$$(19)\quad \Omega_i = \frac{R_0 - R_i}{\|R_0 - R_i\|} .$$



a) b) c)

Figure 3. Successful convergences towards the real point and the associated error and gradient values in the iteration steps; **a)** three steps, error~0; **b)** five steps, error $2.93.10^{-14}$; **c)** seven steps, error $4.62.10^{-9}$.

Based on that angle information, each beacon is making its private guess for the true position of R_0 and its covariance matrix there:

$$\breve{R}_{0,i} = R_i + l_i\Omega_i; \tag{20}$$

$$\breve{P}_{0,i} = E\left(\breve{R}_{0,i} - E\left(\breve{R}_{0,i}\right)\right) = P_{R_i} + \sigma_{l_i}^2\Omega_i\Omega_i^T. \tag{21}$$

Finally, a new estimate for R_0 is found by static sensor fusion - a linear combination of all beacon guesses, multiplied by weight modifiers w_i based on their guessed conditional precision:

$$R_0 = \sum_{i=1}^{n} w_i\breve{R}_{0,i}; \tag{22}$$

$$w_i = \left(\sum_{i=1}^{n}\breve{P}_{0,i}^{-1}\right)^{-1}\breve{P}_{0,i}^{-1}; \tag{23}$$

$$\Rightarrow R = \left(\sum_{i=1}^{n}\breve{P}_{0,i}^{-1}\right)^{-1}\sum_{i=1}^{n}\breve{P}_{0,i}^{-1}\breve{R}_{0,i}. \tag{24}$$

These few steps have to be repeated until the orientation vectors are adjusted and R_0 stabilizes according to the application requirements. It takes more steps than using the error gradient described previously, but the steps are simpler. The process is more stable and performs well in presence of noisy beacons. On *figure 5* a case is presented where the error gradient cannot be used if the beacon uncertainties are high, but which do not pose any problems to the angle-walk. Furthermore, when the beacon uncertainties increase, the iterations between the starting point and R_0 decrease. In the first scenario, the beacon positions are contaminated with normally distributed noise with variance of 1 meter, and the algorithm needs about 60 iterations to reduce the iteration error bellow 1mm. In the second scenario, the variance is increased to 10 meters and the required iterations are halved to 30. In the last case, the variance is 50 meters and the algorithm is missing the right position some of the time, converging to other minimums. The statistical analysis is also failing in the last case, because of the wrong convergence.

2.2. Net-Walk Trilateration

The net-walk trilateration is another novel technique in this paper. The algorithm is from the family of the Simplex method, and allows fast numerical optimization of S in the presence of local minimums. When S is not strongly concave,

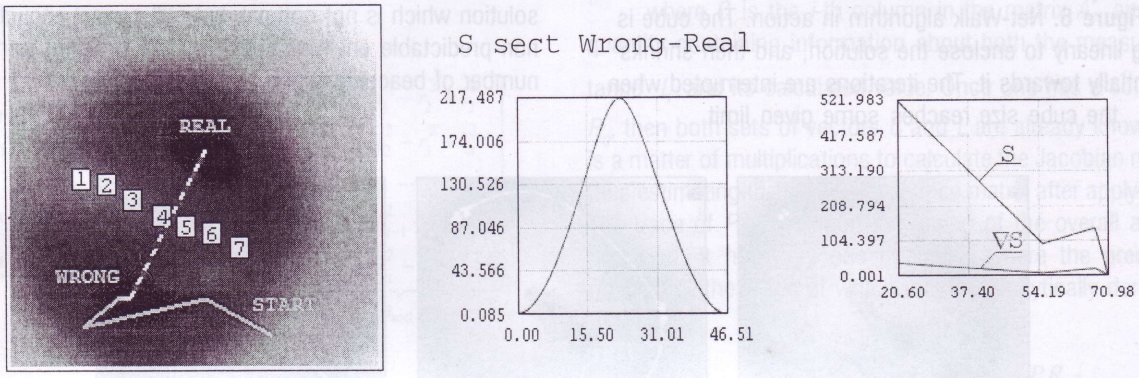
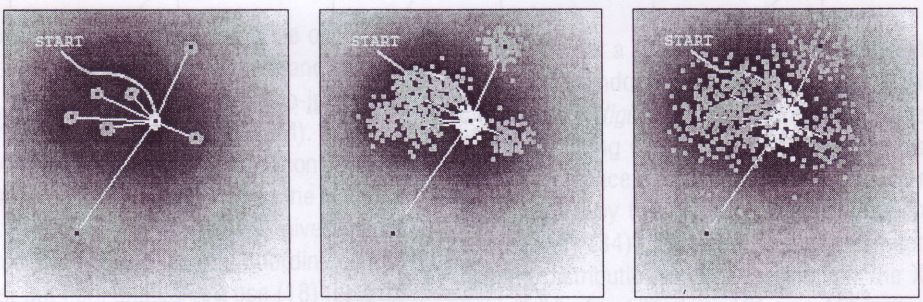


Figure 4. a) A non successful convergences towards a wrong point; **b)** transversal section of the error levels between the real and the wrong points; **c)** error and gradient on the way.



(a) Var = 1 (b) Var = 10 (c) Var = 50

Figure 5. Angle-Walk example in the presence of beacon position errors with different variances; **a)** V(Ri)=1 m, 60 steps; **b)** V(Ri)=10 m, 30 steps; **c)** V(Ri) = 50 m, wrong convergence detected.

because of some specific beacon arrangement (e.g. too close to each other or in line formation) the previously described methods of the error gradient and the angle-walk have the tendency to fall into these wrong solutions. That is why we need a method that performs multiple, but still intelligent computations in different points of the space simultaneously. For that purpose, the net-walk algorithm uses a cube (or quadrant for two dimensional applications) with variable side, the centre and the vertices of which are the points where error computing is performed using (2). If the lower level is laying on some of the vertices, then the cube is moved so the point coincides with its centre. Otherwise, the size of the cube is half shrinked around the centre. Then the procedure is repeated until the cube size become small enough to meet the convergence requirements. The final result is lying in the cub's centre. To reduce the number of required iterations, the initial cube side have to be chosen big enough to cover the area where the real solution is. It's contradictory, but that is because the cube shrinks exponentially, while moving linearly (*figure 6*).

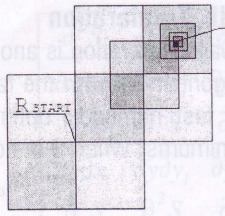
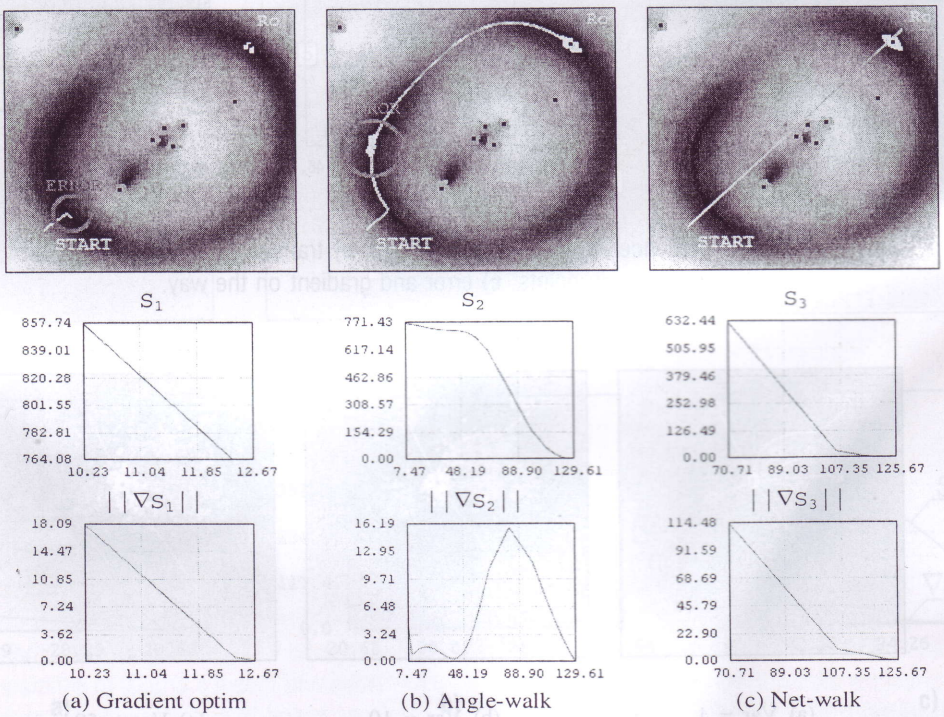


Figure 6. Net-Walk algorithm in action. The cube is moving linearly to enclose the solution, and then shrinks exponentially towards it. The iterations are interrupted when the cube size reaches some given limit

In *figure 7* a case is presented where the gradient method fails absolutely, the angle-walk is still working most of the time and the net-walk is operating without fault. The convergence criterion is chosen to be one micron while all beacon position and range variances are one meter, without inter-dimension correlations. The intensity maps in the figures show the error gradient, while the graphs contain the error and its gradient module during the iterations in the three cases. The net-walk iterates 55 times without falling in the several local minimums present in the map, while the angle-walk needs 572 iterations and in some of the realizations cannot converge properly. Three zones with error gradient close to zero are evident from the gradient graph.

3. Real-Time Trilateration Without Optimization

All of the considered techniques in the previous sections are based on iterative search for solution, which in some cases is yielding totally wrong results. Moreover, computing by iteration could take a lot of time and computational resources. Finally, one of the main assertions in the optimization process is the proper knowledge for the input uncertainties, which in unpredictable unknown environment is not possible, therefore reducing the overall robustness of the mentioned methods. Because of these arguments, a need arises for fast real-time solution which is not optimal, but still correct enough in noisy non-predictable environments and still profiting from the big number of beacons in use.



(a) Gradient optim (b) Angle-walk (c) Net-walk

Figure 7. A case where only the net-walk is working properly; **a)** gradient optimization - fails most of the time; **b)** Angle-walk - most of the realizations are successful (with 572 iterations); **c)** Net-walk - performs properly (55 iterations)

If the lengths of the radius vectors of R_0 and R_i are respectively r_0 and r_i, then by reconsidering equation (1) one obtains

$$(25) \qquad x_0x_i + y_0y_i + z_0z_i = \frac{r_0^2 + r_i^2 - l_i^2}{2}$$

By substitution and subtraction of the equations, one gets rid of the square of r_0, thus obtaining linear relationships for x_0, y_0 and z_0 of the form

$$(26) \qquad x_0(x_i - x_j) + y_0(y_i - y_j) + z_0(z_i - z_j) = \frac{1}{2}\left(r_i^2 - l_i^2 - r_j^2 + l_j^2\right)$$

This kind of linearization is not precision costly, but at the expense of a part of the redundant input information instead, i.e. the subtracting equation. In our case it works very well and the obtained system of linear equations does not present any abnormalities. However, this is not a straightforward methodology, and cannot be applied successfully for any similar redundant nonlinear problem, because of the potential singularities. In the case, where all beacons are characterized by similar uncertainties, we have to build a symmetric model that gives no priority to the precision of any of them. By choosing i and j to be consecutive indexes and using matrix formulation for (26) we obtain the following linear matrix equation

(27)

$$\underbrace{\begin{bmatrix} x_1 - x_2 & y_1 - y_2 & z_1 - z_2 \\ x_2 - x_3 & y_2 - y_3 & z_2 - z_3 \\ \cdots & \cdots & \cdots \\ x_{n-1} - x_n & y_{n-1} - y_n & z_{n-1} - z_n \\ x_n - x_1 & y_n - y_1 & z_n - z_1 \end{bmatrix}}_{A_{n\times 3}} \underbrace{\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}}_{R_0} = \frac{1}{2}\underbrace{\begin{bmatrix} r_1^2 - l_1^2 - r_2^2 + l_2^2 \\ r_2^2 - l_2^2 - r_3^2 + l_3^2 \\ \cdots \\ r_{n-1}^2 - l_{n-1}^2 - r_n^2 + l_n^2 \\ r_n^2 - l_n^2 - r_1^2 + l_1^2 \end{bmatrix}}_{B_{n\times 1}};$$

$$(28) \qquad A^* = \left(A^T A\right)^{-1} A^T;$$

$$(29) \qquad \Rightarrow R_0 = A^* B$$

where A^* is the pseudo-inverse of A. Because of the redundant input information, the matrix A is not square and in order to find a solution for R_0 we are going to use pseudo-inversion, which is equivalent to the Least Square Method (LSM). It should be noted, that the matrix A^* depends only on the beacon configuration and therefore has to be calculated only when the beacons change their positions. The final expression (29) gives a solution for R_0 in function of the actual beacon's coordinates and the measured distances. But here one cannot use (18) for error analysis because this method is not based on minimizing the error function like in algorithms from the previous sections. It has been found that in the considered case when working with beacons with similar uncertainties, relationship (29) is statistically close to linear, i.e. the probability distribution of the input variables is mostly transferred to the output. For example if the input errors are supposed to be normally distributed, such a distribution to the output is expected. Considering the above premises, the analysis here is based on the same first-order error propagation law as in the optimization case

$$(30) \qquad P_{OUT} = JP_{IN}J^T = \sum_{i=1}^{n} J_i P_i J_i^T$$

where P_{IN} *and* P_{OUT} are the full input and output covariance matrices. J denotes the full Jacobian matrix, relating the output variations with the variations of each input parameter. It is calculated in R_0 which is assumed to be the expected value of the real solution. The Jacobian matrices are obtained after differentiation of (29)

$$(31) \qquad dR_0 = A^*\left(dB - dAR_0\right);$$

$$(32) \qquad J_i = \left[\frac{\partial R_0}{\partial x_i} \; \frac{\partial R_0}{\partial y_i} \; \frac{\partial R_0}{\partial z_i} \; \frac{\partial R_0}{\partial l_i}\right] = \left(C_{i-1} - C_i\right)L_i;$$

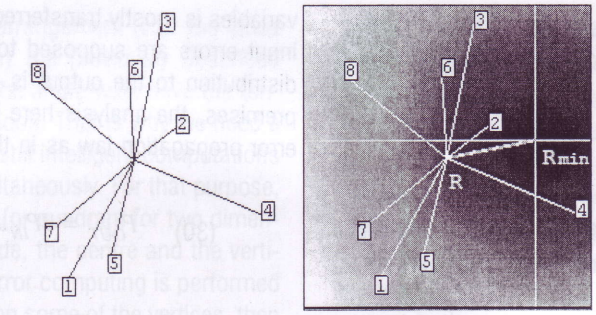
$$(33) \qquad L_i = \left[x - x_i \;\; y - y_i \;\; z - z_i \;\; l_i\right]; \quad C_i = \left[a_{1i}^* \;\; a_{2i}^* \;\; a_{3i}^*\right]^T$$

where C_i is the i-th column in the matrix A^*, and L_i is a quantity containing information about both the measured distance l_i and its calculated value. Once one has a solution for R_0, then both sets of vectors C and L are already known and it is a matter of multiplications to calculate the Jacobian matrices, thus estimating the output covariance matrix after applying (30). The trace of P_{out} is a good evaluation of the overall algorithm precision. It has only one minimum where the precision is maximum, the place of which may be analytically derived by

$$(34) \qquad R_{\min} = \left(\sum_{i=1}^{n} m_i P_i\right)^{-1} \sum_{i=1}^{n} m_i P_i R_i;$$

$$(35) \qquad m_i = \left\|C_{i-1} - C_i\right\|^2.$$

Consider a random beacon configuration with random positions and random uncertainties and correlations, governed by a normal law (*figure 8a*). On *figure 8b* an intensity map is shown for the resulting variance levels (darker zones mean lower values) in the space enclosing the beacons. The minimum location R_{min}, marked by two crossing lines, has been analytically estimated with (34). On *figure 1* are presented four probability density distributions for the errors on the X axis, calculated numerically for 10^2, 10^3, 10^4 and 10^5 error samples respectively, and compared with the estimated normal distribution, presented with doted line. These graphs clearly display that the numerically obtained distribution law converges to something very close to the analytic one as the size of the sample population grows.



(a) (b)

Figure 8. a) Experimental configuration; b) Variance intensity map

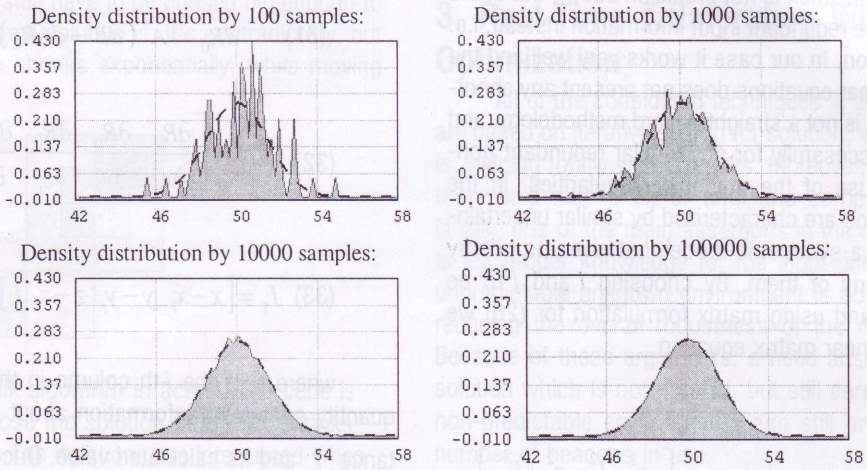
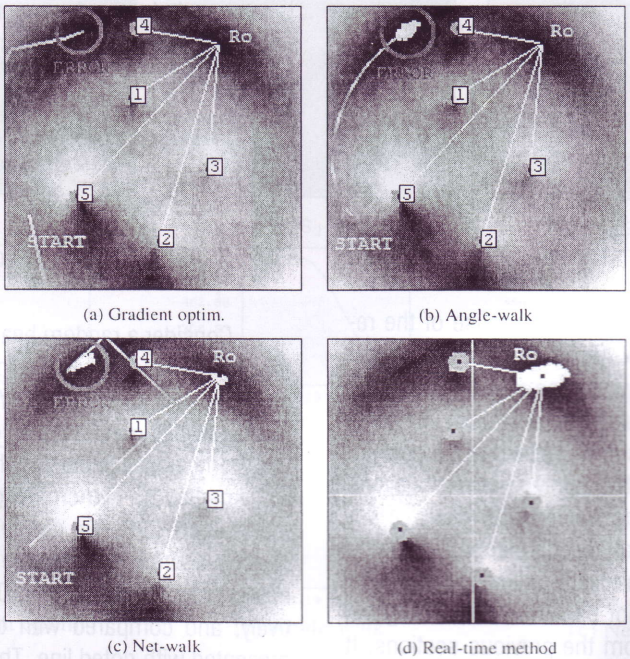


Figure 9. Analytic (doted line) and numerical (solid line) density distributions for the error on the X axis. The numerical one is being calculated for different number of samples: 10^2, 10^3, 10^4 and 10^5. The analytic is supposed to be normal as the input errors.



(a) Gradient optim. (b) Angle-walk

(c) Net-walk (d) Real-time method

Figure 10. The initial guess is too far away from the solution: a) The gradient search is totally wrong; b) The angle-walk is failing in all cases; c) The net-walk succeeds in some of the realizations; d) The real-time trilateration is giving good enough results.

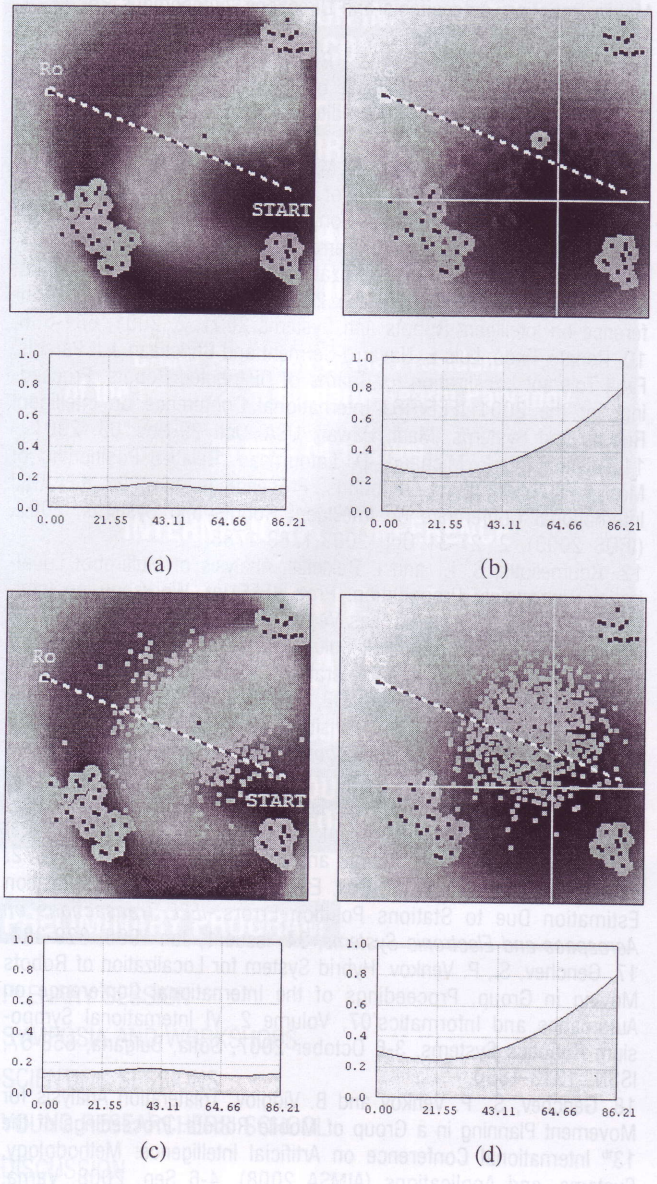


Figure 11. Varying considerably the uncertainty of a single beacon when in the presence of a large number of beacons. Intensity maps present the variance levels in the space around the beacons. Transversal sections are marked by doted lines and plotted in the associated graphs bellow the maps. **a)** Optimal method before the variation; **b)** Real-Time method before the variation; **c)** Optimal method after the variation; **d)** Real-Time method after the variation.

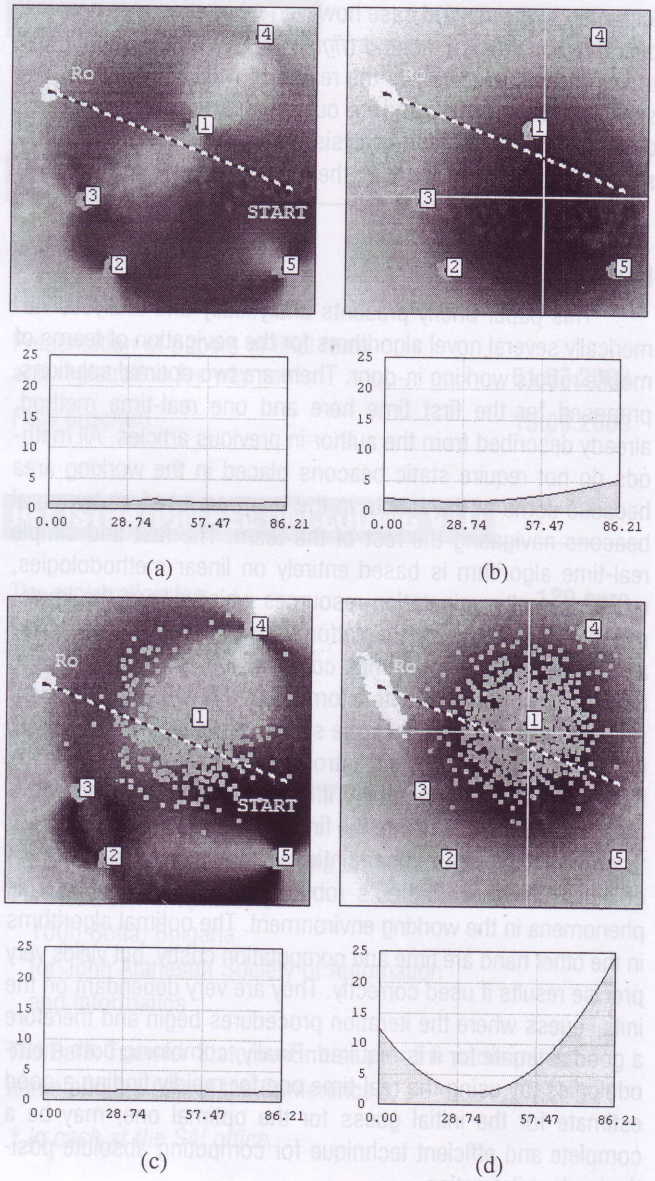


Figure 12. Varying considerably the uncertainty of a single beacon when in the presence of a low number of beacons. Intensity maps present the variance levels in the space around the beacons. Transversal sections are marked by doted lines and plotted in the associated graphs bellow the maps. **a)** Optimal method before the variation; **b)** Real-Time method before the variation; **c)** Optimal method after the variation; **d)** Real-Time method after the variation.

In the next example, demonstrated on *figure 10*, a case is shown where the initial guess is too far away and only the real-time algorithm is able to resolve the real solution. The corresponding intensity maps are from the error gradient. The netwalk manages to reach the true point in some of the realizations, while the other optimal methods fail completely. In the last example the errors between the real-time and the optimal solutions are compared before and after increasing vastly the uncertainties of one of the beacons. There are two chosen formations, one containing large number of erroneous beacons (*figure 11*), and one with only few (*figure 12*). The intensity maps for this experimentation are from the output variance levels in the working area. On the real-time map, the variance minimum location have been estimated by (33) and marked by two crossed lines. For the first, redundant configuration with large number of beacons, both the optimal (*figure 11a, c*) and real-time (*figure 11b, d*) methods do not exhibit notable changes in its output precision for R_0, or for any other point in the space around according to the transversal sections trough the variance maps. It is also obvious that the overall precision of the optimal algorithm is a

lot better. In the second case however, where there are only few beacons, the optimal method (*figure* 12a, c) is still ignoring the totally wrong beacon while the real-time (*figure* 12b, d) is very sensitive to its errors, and its output precisions is greatly reduced. For more detailed analysis and experimentation on this subject please see [18] where the method was firstly proposed.

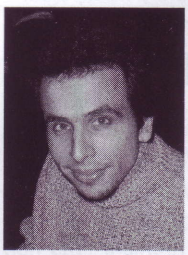
Conclusion

This paper briefly presents analytically and analyses numerically several novel algorithms for the navigation of teams of mobile robots working in-door. There are two optimal solutions, proposed for the first time here and one real-time method, already described from the author in previous articles. All methods do not require static beacons placed in the working area because some of the robots in the team are used as temporal beacons navigating the rest of the team. The fast and simple real-time algorithm is based entirely on linear methodologies, requires small computation resources which allows the implementation in low cost robot platforms. However, it has a remarkable drawback - it does not consider the precision of each beacon when computing its information. It is a fast and efficient solution when the beacons have similar uncertainties. Otherwise, if there are some beacons introducing far greater errors, the system would not ignore them if not enough beacons are available and they will dominate the final result. The main reason not to consider the input uncertainties in the calculations is to increase the overall method's robustness toward unpredictable phenomena in the working environment. The optimal algorithms in the other hand are time and computation costly, but yields very precise results if used correctly. They are very dependant on the initial guess where the iteration procedures begin and therefore a good estimate for it is required. Finally, combining both methodologies, by using the real-time one for rapidly finding a good estimate for the initial guess for the optimal one, may be a complete and efficient technique for computing absolute positioning by trilateration.

References

1. Borenstein, J., H. R. Everett, L. Feng, and D. Wehe. Mobile Robot Positioning: Sensors and Techniques.-Invited paper for the *Journal of Robotic Systems*, Special Issue on Mobile Robots, 14. No.4, 231-249.
2. Kim, J., G. Jee, J. Lee. A Complete GPS/INS lintegration Technique Using GPS Carrier Phase Measurements. IEEE Position Location and Navigation Symposium, 1998, 20-23 Apr. 1998, 526-533.
3.Ohno, K., T. Tsubouchi, B. Shigematsu, S. Yuta. Differential GPS and Odometry-based Outdoor Navigation of a Mobile Robot.-*Advanced Robotics*, 18, Issue 6, 611-635.
4.Brown, Robert G., Patrick Y.C. Hwang. Introduction to Random Signals And Applied Kalman Filtering. Second edition, ISBN 0471-52573-1, John Wiley & Sons, 1992.
5. Parker, L. E. Current State of the Art in Distributed Autonomous Mobile Robotics.-*Distributed Autonomous Robotic Systems*, L. E. Parker, G. Bekey, and J. Barhen, Eds., 4, 2000, 3-12 Springer, Tokio.
6. Uny Cao, Y., Alex S. Fukunaga, Andrew B. Kahng. Cooperative Mobile Robotics: Antecedents and Directions, Autonomous Robots, 4, 1997,1-23.
7. Hayes, A. T., A. Martinoli & R. M. Goodman. Distributed Odor Source Localisation, Special Issue on Artificial Olfaction, Nagle H. T., Gardner J. W., and Persaud K., editors,-*IEEE Sensors Journal*, 2, No. 3, 2002, 260-271.
8. Rekleitis, I. M., G.Dudek, E. E. Milios, Multi-robot Collaboration for Robust Exploration. Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA'00.,4, 2000, 3164 - 3169.
9. Spletzer, J., A. K. Das, R. Fierro, C. J. Taylor. V. Kumar, and J. P. Ostrowski. Cooperative Localization and Control for Multi-Robot Manipulation. Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems 2001, 2, 2001, 631-636.
10. Renato Tinos, Luis E. Navarro-Serment and Christiaan J.J. Paredis. Fault Tolerant Localization for Teams of Distributed Robots. Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, Hawaii, USA, Oct. 29-Nov. 03, 2001.
11. Bisson, J., F. Michaud, D. Letourneau. Relative Positioning of Mobile Robots Using Ultrasounds. Proceedings on 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003. (IROS 2003), 2, 27-31 Oct. 2003,1783-1788.
12. Roumeliotis, S. I. and I. Rekleitis. Analysis of Multirobot Localization Uncertainty Propagation. Proc. IEEE Int. Workshop on Intelligent Robots and Systems, Las Vegas, NV, USA, 2003, 1763-1770.
13. Manolakis, D. E. Efficient Solution and Performance Analysis of 3-D Position Estimation by Trilateration.- *IEEE Transactions on Aerospace and Electronic Systems*, 32, Issue 4, Oct 1996, 1239-1248.
14. Thomas, F. and L. Ros. Revisiting Trilateration for Robot Localization.-*IEEE Transactions on Robotics*, 21, Issue 1, Feb.2005, 93-101.
15. Grabowski, R. and P. Khosla. Localization Techniques for a Team of Small Robots. Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2, 2001,1067-1072.
16. Manolakis, D. E., M. E. Cox. Effect in Range Difference Position Estimation Due to Stations Position Errors.-*IEEE Transactions on Aerospace and Electronic Systems*, 34, Issue 1, Jan 1998, 329-334.
17. Genchev, S., P. Venkov. Hybrid System for Localization of Robots Moving in Group, Proceedings of the International Conference on Automatics and Informatics'07, Volume 2, VI International Symposium Robotics Systems, 3-6 October 2007, Sofia, Bulgaria, 653-57, ISSN: 1313-1850.
18. Genchev, S., P. Venkov and B. Vidolov. Trilateration Analysis for Movement Planning in a Group of Mobile Robots. Proceedings of the 13th International Conference on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA 2008), 4-6 Sep. 2008, Varna, Bulgaria, 353-364, Springer LNCS, 5253, ISBN 978-3-540-85775-4.

Manuscript received on 09.07.2008



Svetoslav Genchev *received his engineering degree from the Technical University of Sofia in 2005, currently is preparing for defending his PhD Thesis in navigation of cooperative robots. His work was mainly helped by a French scholarship program allowing him to perform some of his research in the Technological University of Compiegne, France.*

Contacts:
French Language Faculty of Electrical Engineering,
Technical University of Sofia,
8 Kliment Ohridski St.,
1000 Sofia, Bulgaria
e-mail: genchevs@abv.bg