Towards a Closed-Form Solution of Constraint Networks for Maximum Likelihood Mapping

Dario Lodi Rizzini
RIMLab - Robotics and Intelligent Machines Laboratory
Dipartimento di Ingegneria dell'Informazione
University of Parma, Italy
E-mail dlr@ce.unipr.it

***Abstract*— Several recent algorithms address simultaneous localization and mapping as a maximum likelihood problem. According to such formulation map estimation is achieved by finding the configuration that minimizes the error function associated to the constraint network representing the map. Almost all the proposed algorithms exploit iterative fixed-point techniques. Less attention has been paid to the evaluation of the structure of the problem.**

In this paper, we derive the closed form solution for a constraint network of planar poses when the error function is given in a particular form. The derivation of the exact solution allows a better knowledge of the problem and a more detailed investigation of the performance of existing numerical techniques. The system of equations, that are satisfied by the solution, are computed and a general algorithm to solve such equations is provided.

I. INTRODUCTION

A representation of the environment is required to achieve specific robotic tasks in several applications. Indeed, autonomously building maps of the environment has become a major problem in the robotics community. For this task the robot has at its disposal its motion information and its sensor observations, which are both affected by uncertainty. In literature, map building and localization is often referred to as *Simultaneous Localization and Mapping (SLAM)*.

Maximum Likelihood (ML) is one of the approaches developed to address this problem. According to ML, SLAM is formulated as a network of constraints. Relations of robot poses and observations are represented by constraints among nodes in a graphical model. The solution of the problem corresponds to the configuration of the graph that maximizes its likelihood or, equivalently, minimizes the least square error.

Early algorithms exploiting ML formulation of SLAM were only suitable for an offline computation of the solution. Offline methods require that all data be available at the beginning of computation. In particular, Lu and Milios [1] pioneered the maximum likelihood approach proposing a brute force technique to align range scans, once all the scans have been acquired. Gutman and Konolige [2] improved the construction of the network by introducing map patches instead of single scans and provided an effective loop detection method based on correlation. Duckett *et al.* [3] introduced Gauss-Seidel relaxation to compute the optimal solution, although in their formulation angular terms were not considered.

Recent research has focused on making these algorithms more efficient and, eventually, incremental in order to make online optimization possible. Multi-level relaxation (MLR) [4] improves the simple Gauss-Seidel relaxation by solving the network at different levels of resolution. The Treemap algorithm [5] performs efficient updates with a tree-based subdivision of the map into weakly-correlated components. The smoothing and mapping (SAM) algorithm [6] relies on a QR factorization of information matrix that allows an efficient estimation of the poses of the network nodes with an efficient back-substitution. The original algorithm has been improved [7] to integrate new observations when available. A stochastic gradient descent (SGD) technique has been proposed [8] to compute the configuration minimizing least-square error by using a representation which enables efficient updates. An incremental variant of the algorithm relies on several improvements such as tree parameterization, adaptive learning rate and selective update rules [9].

All the referred approaches adopt a specific numeric technique in order to solve the constraint network. Less interest has been devoted to evaluate of the structure of the negative likelihood function associated to the network. Few results on the existence of minima for such function and on the convergence of these methods are available.

In this paper, we discuss the derivation of a closed form solution for a general constraint network and compute a reduced set of equations depending only on angular terms. The only preliminary assumption concerns the expression of the error function. In particular, the gradient of error function is computed and the resulting system of nonlinear equations is manipulated. The position terms of the network poses are expressed with respect to the angular parameters and substituted in the remaining equations. Moreover, an algorithm to solve the angular equations is suggested.

The paper is organized as follows. Section II illustrates the general formulation of the maximum likelihood problem and the gradient of the network error function. Section III derives a simplified set of equations that are more amenable to a solution. Finally, section IV gives conclusion remarks.

II. PROBLEM FORMULATION

This section discusses the general formulation of the maximum likelihood (ML) approach and derives the equations that must be satisfied by the solution. The mapping problem is formulated as a graph, whose nodes correspond to the variables of the map and whose edges represent the constraints between pairs of these variables. The network of constraints is extracted from raw sensor data and hence after we assume that it is a given datum of the problem. Furthermore, in this paper the map consists only of robot poses according to the *delayed-state* representation [10]. The delayed-state approach corresponds to a view-based representation of the environment similar to one adopted in [1], but it can be obtained from a *feature based* map by marginalizing the poses [11].

Let $\mathbf{p} = [p_1 \ \cdots \ p_n]^T$ be the vector of robot poses. Each planar pose $p_i = [x_i, y_i, \theta_i]^T$ is usually represented by two position coordinates x_i and y_i and the orientation θ_i with respect to a global reference frame. The origin pose is conventionally noted as $p_0 = [0, 0, 0]^T$. In this paper, the pose is represented by a 4 dimensional vector $p_i = [x_i, y_i, c_i, s_i]^T$, where $c_i = \cos\theta_i$ and $s_i = \cos\theta_i$ verify equation

$$g(c_i, s_i) = c_i^2 + s_i^2 - 1 = 0 \tag{1}$$

Let $\hat{\delta}_{ij}$ and Ω_{ij} be respectively the mean and the information matrix of an observation of node j seen from node i corresponding to the constraint $\langle i, j\rangle$. Since poses are represented by 4 parameters, $\hat{\delta}_{ij} = [\hat{x}_{ij}, \hat{y}_{ij}, \hat{c}_{ij}, \hat{s}_{ij}]$ and Ω_{ij} is a 4×4 positive definite symmetric matrix. Let $f_{ij}(\mathbf{x})$ be a function that computes a zero noise observation according to the current configuration of the nodes j and i

$$f_{ij}(\mathbf{p}) = \begin{bmatrix} (x_j - x_i)\ c_i + (y_j - y_i)\ s_i \\ -(x_j - x_i)\ s_i + (y_j - y_i)\ c_i \\ c_j\ c_i + s_j\ s_i \\ s_j\ c_i - c_j\ s_i \end{bmatrix} \tag{2}$$

where $\cos(\theta_j - \theta_i) = c_j\ c_i + s_j\ s_i$ and $\sin(\theta_j - \theta_i) = s_j\ c_i - c_j\ s_i$.

The vector of error variables for constraint $\langle i, j\rangle$ is given by

$$e_{ij}(\mathbf{p}) = f_{ij}(\mathbf{p}) - \delta_{ij} \tag{3}$$

The expression of the error is different for the two parameterizations, but the second one that uses *sine* and *cosine* of orientation is safer due to the periodicity of functions. The error on constraint is a scalar obtained by weighting the error of each component of $e_{ij}(p)$

$$\chi^2_{ij}(\mathbf{p}) = e^T_{ij}(\mathbf{p})\ \Omega_{ij}\ e_{ij}(\mathbf{p}) \tag{4}$$

and the error of the whole network is

$$\begin{aligned} \chi^2(\mathbf{p}) &= \sum_{\langle i,j\rangle \in \mathcal{C}} \chi^2_{ij}(\mathbf{p}) && (5) \\ &= \sum_{\langle i,j\rangle \in \mathcal{C}} e^T_{ij}(\mathbf{p})\ \Omega_{ij}\ e_{ij}(\mathbf{p}) && (6) \end{aligned}$$

where $\mathcal{C}$ is the set of constraints. The argument $\mathbf{p}$ will be omitted when the interpretation of equations is not ambiguous.

The network is solved by finding the configuration $\mathbf{p}$ that minimizes Eq. (6). Our aim is to investigate the existence of one or more exact minima and to possibly compute their closed-form expression. The first step is the assessment of critical points of Eq. (6) corresponding to the configurations that make gradient of error null. In the next subsection, the general expression of gradient is derived.

A. Error Gradient Expression

Since the orientation parameters are subjected to Eq. (1), the addressed optimization problem is constrained and the computation of critical points is performed exploiting the method of *Lagrange multipliers*. Thus, the function to optimize consists of both the error function Eq. (6) and the conditions provided by Eq. (1)

$$E(\mathbf{p}) = \chi^2(\mathbf{p}) - \sum_{i=1}^{n} \lambda_i\ g_i(c_i, s_i). \tag{7}$$

Then, the target function $E(\mathbf{p})$ is a sum of terms and each term depends only on the parameters of few poses. The expression of a row block of gradient $\nabla_i E = \partial E(\mathbf{p})/p_i$ is a function of the variables of the pose p_i and of the poses p_j connected to node i by a constraint. Let $\mathcal{I}_i$ and $\mathcal{O}_i$ be respectively the sets of the indices connected to pose p_i by an incoming or an outgoing constraint, or formally

$$\begin{aligned} \mathcal{I}_i &= \{j |\ \langle j, i\rangle \in \mathcal{C}\} && (8) \\ \mathcal{O}_i &= \{j |\ \langle i, j\rangle \in \mathcal{C}\} && (9) \end{aligned}$$

Therefore, the gradient row block $\nabla_i \chi$ has the following expression

$$\begin{aligned} \nabla_i E &= \sum_{j \in \mathcal{I}_i} \nabla_i \chi^2_{ji} + \sum_{j \in \mathcal{O}_i} \nabla_i \chi^2_{ij} + \lambda_i\ \nabla_i g_i(c_i, s_i) && (10) \\ &= \sum_{j \in \mathcal{I}_i} 2 \frac{\partial e^T_{ji}}{\partial p_i}\ \Omega_{ji}\ e_{ji} + \sum_{j \in \mathcal{O}_i} 2 \frac{\partial e^T_{ij}}{\partial p_i}\ \Omega_{ij}\ e_{ij} \\ &\quad + 2\lambda_i\ [0, 0, c_i, s_i]^T && (11) \end{aligned}$$

Hence after, the constant 2 is omitted since it does not affect the computation of critical points. Eq. (11) contains the Jacobians of constraint errors e_{ij} and e_{ji} with respect to p_i. The value of Jacobian is different when the constraint is incoming or outgoing pose p_i. Therefore, the expressions of incoming Jacobian and outgoing Jacobian are given by

$$\begin{aligned} \frac{\partial e_{ij}}{\partial p_j} &= \begin{bmatrix} c_i & s_i & 0 & 0 \\ -s_i & c_i & 0 & 0 \\ 0 & 0 & c_i & s_i \\ 0 & 0 & -s_i & c_i \end{bmatrix} && (12) \\ \frac{\partial e_{ij}}{\partial p_i} &= \begin{bmatrix} -c_i & -s_i & (x_j - x_i) & (y_j - y_i) \\ s_i & -c_i & (y_j - y_i) & -(x_j - x_i) \\ 0 & 0 & c_j & s_j \\ 0 & 0 & s_j & -c_j \end{bmatrix} && (13) \end{aligned}$$

The expression of gradient row block is obtained by substituting the expressions of Jacobians and of the constraint error vectors e_{ij}.

B. Gradient of Simplified Error Function

In the general case, the expression of gradient terms $\nabla_j \chi^2_{ij}$ and $\nabla_i \chi^2_{ij}$ are complex and difficult to manipulate. The constraint error functions χ^2_{ij} represent the Mahalanobis distance between the value of constraint with current configuration $\mathbf{p}$ and the required value. When $\Omega_{ij} = I_4$ for each constraint $\langle i, j \rangle$, this distance is equal to the Euclidean one. Such distance mixes metric and angular terms which are not homogeneous, but its expression can be computed and manipulated to achieve some results. Even though such an approach may be questionable, it is worthwhile investigating it in order to gain further insight into the structure of the solution. Gradient terms become:

$$\nabla_j \chi^2_{ij} = \begin{bmatrix} x_{ij} - \hat{x}_{ij} c_i + \hat{y}_{ij} s_i \\ y_{ij} - \hat{x}_{ij} s_i - \hat{y}_{ij} c_i \\ c_j - \hat{c}_{ij} c_i + \hat{s}_{ij} s_i \\ s_j - \hat{c}_{ij} s_i - \hat{s}_{ij} c_i \end{bmatrix} \quad (14)$$

$$\nabla_i \chi^2_{ij} = \begin{bmatrix} -x_{ij} + \hat{x}_{ij} c_i - \hat{y}_{ij} s_i \\ -y_{ij} + \hat{x}_{ij} s_i + \hat{y}_{ij} c_i \\ x^2_{ij} c_i + y^2_{ij} c_i - \hat{x}_{ij} x_{ij} - \hat{y}_{ij} y_{ij} \\ \quad + c_i - \hat{c}_{ij} c_j - \hat{s}_{ij} s_j \\ x^2_{ij} s_i + y^2_{ij} s_i + \hat{y}_{ij} x_{ij} - \hat{x}_{ij} y_{ij} \\ \quad + s_i + \hat{s}_{ij} c_j - \hat{c}_{ij} s_j \end{bmatrix} \quad (15)$$

In particular, Eq. (14) and Eq. (15) represent respectively the gradient of error on the constraint $\langle i, j \rangle$. The two equations depend on the relative position displacements $x_{ij} = x_j - x_i$ and $y_{ij} = y_j - y_i$ and on the angular parameters. Observe that the gradient terms achieved by differentiating with respect to position parameters x_i, y_i, x_j and y_j are linear and equal except for the sign.

III. SOLUTION OF CONSTRAINT NETWORK

In the previous section, the expression of gradient terms has been computed under a specific hypothesis about the information matrices of the constraints. Then, the aim is the computation of the solutions, which in this context correspond to the values of the pose vector $\mathbf{p}$ that make the error gradient equal to zero. A solution is not necessarily a minimum of error function Eq. (6). Furthermore, the existence of one or more solutions has not been fully investigated. The system of equations will be manipulated in the following in order to give a constructive answer to these questions.

Let $(\mathcal{N}, \mathcal{C})$ be the connected directed graph representing the map, $n = |\mathcal{N}| - 1$ the number of nodes and $e = |\mathcal{C}|$ the number of constraints or edges. The constraint networks that are considered in this paper must satisfy few assumptions mainly depending on the nature of the mapping problem.

- The constraints $\langle i, i+1 \rangle$, with $i = 1, \ldots, n$, belong to the network. This property holds because the graph has been built by a robot exploring the environment and a proper node numbering that respects the sequential growth of the network is chosen like in [8]. The direction of the constraint from the pose with a smaller index to the pose with a larger index is the result of next rule.
- Each constraint is expressed from the point of view of the pose with smaller index. Thus, constraint $\langle i, j \rangle$ contains the parameters of the frame change from i to j with $i < j$. In general, the direction of a constraint $\langle i, i+1 \rangle$ does not affect the result, since information matrix Ω_{ij} can be adjusted. However, the hypotheses previously made on Ω_{ij} do not allow the inversion of a constraint without changing the error function. Even so, the proposed rule on direction is basically consistent with the usual structure of the network.

In order to achieve a solution of $\nabla E(\mathbf{p}) = 0$, the components of each gradient row block Eq. (11) are divided in two parts. The first two equations of each row block are linear with respect to both position and orientation parameters. The last two equations are obtained from the differentiation of error function with respect to the angular parameters and depend also on the Lagrange multipliers. Hence after, the first ones are called *position equations* and the latter *angular equations*. The solution algorithm initially operates on position equations in order to substitute the position terms in angular equations. Then, the angular parameters are solved.

A. Position Equations

The position equations are obtained by combining the terms in the first two rows of Eq. (14) and Eq. (15) according to the graph topology. Since the same operations are performed on the equations for coordinate x and for coordinate y, a vectorial notation is adopted

$$\sum_{j \in \mathcal{I}_i} (-d_{ji} + l_{ji}) + \sum_{j \in \mathcal{O}_i} (d_{ij} - l_{ij}) = 0 \quad (16)$$

where d_{ij} and l_{ij} are defined as

$$d_{ij} = \begin{bmatrix} x_{ij} \\ y_{ij} \end{bmatrix} = \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \quad (17)$$

$$l_{ij} = \begin{bmatrix} c_i & -s_i \\ s_i & c_i \end{bmatrix} \begin{bmatrix} \hat{x}_{ij} \\ \hat{y}_{ij} \end{bmatrix} \quad (18)$$

The system of position equations is given by Eq. (16), one for each pose p_i. Our aim is to compute the value of position terms d_{ij} with respect to angular terms l_{ij}, which are considered known terms of the linear system.

An interesting result is achieved by splitting the terms d_{ij} into the position coordinate vectors $d_i = [x_j, y_j]^T$ and $d_j = [x_i, y_i]^T$. The matrix of the new linear system is the *adjacency matrix* of the constraint network [12], after removing node p_0. Thus, the solution of the system can be computed by multiplying the inverse of adjacency matrix and the fixed terms. The values of position displacement x_{ij} and y_{ij} is then obtained by the difference of the positions of the two poses.

However, it would be better to estimate directly the x_{ij} and y_{ij} in order to point out the dependence on angular terms d_{ij}. Constraints are ordered starting from the constraints in the form $\langle i, i+1 \rangle$ (called *spanning constraints* hence after) and then adding the other constraints $\langle i, j \rangle$ with $j > i + 1$ in lexicographical order (called *loop constraints*). Figure 1 illustrates the difference between the two types of constraints with an example. The *incidence matrix* of the network is the matrix of the linear system of equations. Since the rank of incidence matrix of a connected graph is $|\mathcal{N}| - 1 = n$ [12],

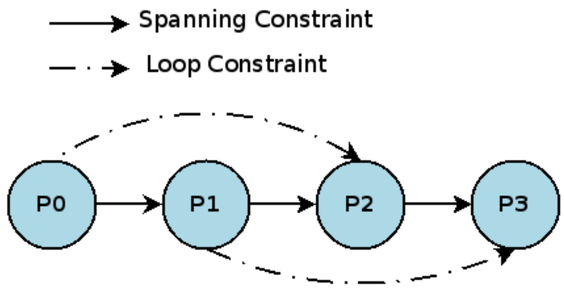

Fig. 1. An example of network with spanning constraints and loop constraints. Observe that $\kappa(0,2,1,3)=1$, since $\langle 0,2\rangle$ and $\langle 1,3\rangle$ share only the constraint $\langle 1,2\rangle$.

the missing $n_L = e - n$ equations are provided by loop equations:

$$d_{ij} = \sum_{k=i+1}^{j} d_{k-1,k} \tag{19}$$

Through simple manipulation of the system of equations, it can be shown that each spanning constraint depends on its angular parameters and on the loop constraints that "include" the spanning constraint

$$d_{i-1,i} = l_{i-1,i} + \sum_{\langle g,h\rangle \in \mathcal{C}_{i-1,i}} \kappa(i-1,i,g,h)\ (d_{gh} - l_{gh}) \tag{20}$$

while loop constraints are computed by solving the linear system of equations derived by each loop equation (19)

$$\begin{aligned} &(j-i+1)\ d_{ij} + \sum_{\langle g,h\rangle \in \mathcal{C}_{i,j}} \kappa(i,j,g,h)\ d_{gh} = \\ &\quad (j-i)\ l_{ij} + \sum_{k=i+1}^{j} l_{k-1,k} + \sum_{\langle g,h\rangle \in \mathcal{C}_{i,j}} \kappa(i,j,g,h)\ l_{gh} \end{aligned} \tag{21}$$

In the above equations, the following symbols have been used

$$\mathcal{C}_{ij} = \{\langle g,h\rangle \in \mathcal{C} | g+1 < h, \langle g,h\rangle \neq \langle i,j\rangle\} \tag{22}$$

$$\kappa(i,j,g,h) = \max\{\min\{j,h\} - \max\{i,g\}, 0\} \tag{23}$$

where $\mathcal{C}_{ij}$ is the set of loop constraints different from $\langle i,j\rangle$ and $\kappa(i,j,g,h)$ counts the number of edges shared by $\langle i,j\rangle$ and $\langle g,h\rangle$.

Equations Eq. (21) (for each $\langle i,j\rangle$) show that position parameters d_{ij} of a loop constraint $\langle i,j\rangle$ are affected by the angular parameters of all the constraints, which belong to the same loop or to another loop that intersects the given loop. This dependence is reflected by spanning constraints after the substitution in Eq. (20).

B. Angular Equations

For each pose p_i, a pair of angular equations is defined. These equations combine the terms of incoming and outgoing constraint gradient (respectively Eq. (14) and Eq. (15)) and the terms related to Lagrange multipliers:

$$\begin{aligned} &\sum_{j\in\mathcal{I}_i} (c_i - \hat{c}_{ji} c_j + \hat{s}_{ji} s_j) \\ &+ \sum_{j\in\mathcal{O}_i} (x_{ij}^2 c_i + y_{ij}^2 c_i - \hat{x}_{ij} x_{ij} - \hat{y}_{ij} y_{ij} \\ &\quad + c_i - \hat{c}_{ij} c_j - \hat{s}_{ij} s_j) - \lambda_i c_i = 0 \end{aligned} \tag{24}$$

$$\begin{aligned} &\sum_{j\in\mathcal{I}_i} (s_i - \hat{c}_{ji} s_j - \hat{s}_{ji} c_j) \\ &+ \sum_{j\in\mathcal{O}_i} (x_{ij}^2 s_i + y_{ij}^2 s_i + \hat{y}_{ij} x_{ij} - \hat{x}_{ij} y_{ij} \\ &\quad + s_i + \hat{s}_{ij} c_j - \hat{c}_{ij} s_j) - \lambda_i s_i = 0 \end{aligned} \tag{25}$$

The above equations are difficult to manipulate due to the dependence on Lagrange multipliers and on the square of position displacements x_{ij}^2 and y_{ij}^2. Equations (24) and (25) can be combined together in order to achieve a simplified relation. In general, given the polynomials $f_1, f_2, g_1, g_2 \in \mathbb{R}[\mathbf{p}, \lambda]$, if $[\mathbf{p}^*, \lambda^*]^T$ is a zero both of f_1 and f_2, then $[\mathbf{p}^*, \lambda^*]^T$ is also a zero of $g_1 f_1 + g_2 f_2$. Thus, equation $g_1 f_1 + g_2 f_2 = 0$ can be solved and the achieved solutions are verified in the original equations.

A new equation is obtained by adding side by side the Eq. (24) multiplied by $-s_i$ and the Eq. (25) multiplied by c_i

$$\begin{aligned} &\sum_{j\in\mathcal{I}_i} (\hat{c}_{ji} c_{ji} - \hat{s}_{ji} s_{ji}) - \sum_{j\in\mathcal{O}_i} (\hat{c}_{ij} s_{ij} - \hat{s}_{ij} c_{ij}) \\ &+ \sum_{j\in\mathcal{O}_i} (\hat{x}_{ij} x_{ij} s_i + \hat{y}_{ij} y_{ij} s_i + \hat{y}_{ij} x_{ij} c_i - \hat{x}_{ij} y_{ij} c_i) = 0 \end{aligned} \tag{26}$$

where

$$c_{ji} = c_{ij} = c_j c_i + s_j s_i \tag{27}$$

$$s_{ji} = -s_{ij} = s_j c_i - c_j s_i \tag{28}$$

$$\sigma_{ji} = \hat{c}_{ji} s_{ji} - \hat{s}_{ji} s_{ji} \tag{29}$$

From the discussion on previous section the position parameters x_{kh} and y_{kh} have the following expression

$$x_{ij} = \sum_{\langle k,h\rangle} \alpha_{kh} (c_k \hat{x}_{kh} - s_k \hat{y}_{kh}) \tag{30}$$

$$y_{ij} = \sum_{\langle k,h\rangle} \alpha_{kh} (s_k \hat{x}_{kh} + c_k \hat{y}_{kh}) \tag{31}$$

The substitution of each element of the above sums in Eq. (26) introduces the following terms

$$\begin{aligned} &\hat{x}_{ij} x_{ij} s_i + \hat{y}_{ij} y_{ij} s_i + \hat{y}_{ij} x_{ij} c_i - \hat{x}_{ij} y_{ij} c_i = \\ &\sum_{\langle k,h\rangle} \alpha_{kh} ((\hat{y}_{ij} \hat{x}_{kh} - \hat{x}_{ij} \hat{y}_{kh})(c_i c_k + s_i s_k) + \\ &\quad (\hat{x}_{ij} \hat{x}_{kh} - \hat{y}_{ij} \hat{y}_{kh})(s_i c_k - c_i s_k)) \end{aligned} \tag{32}$$

It should be observed that the above expression contains terms c_{ki} and s_{ki}. Thus, the resulting system of equations is linear with respect to parameters c_{ki} and s_{ki}. Further equations should be added using angular relations. Furthermore, the obtained solutions may not verify the original equations.

IV. CONCLUSION

In this paper, we have derived the closed form solution for a constraint network of planar poses when the error function is given in a particular form. The gradient of error function is computed and the resulting system of nonlinear equations is manipulated in order to achieve a simplified set of relations. The first result is the position components of network configuration with respect to angular parameters. The position terms can then be substituted into the remaining equations. Moreover, an algorithm to solve the angular equations has been suggested.

In future works, we will investigate the details of the solution of angular equations, that have been only outlined in this paper, and the conditions for the existence of one or more solutions. Furthermore, an implementation of the presented algorithm will be developed in order to compare the results of the algorithm with the existing network solvers.

V. ACKNOWLEDGMENTS

The author gratefully thanks Stefano Caselli for the suggestions and the discussion of the results presented in the paper. This reasearch is partially supported by laboratory AER-TECH of Regione Emilia-Romagna, Italy.

REFERENCES

[1] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Journal of Autonomous Robots*, vol. 4, pp. 333–349, 1997.

[2] J.-S. Gutmann and K. Konolige, "Incremental mapping of large cyclic environments," in *Proc. of the IEEE Int. Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 1999, pp. 318–325.

[3] T. Duckett, S. Marsland, and J. Shapiro, "Fast, on-line learning of globally consistent maps," *Journal of Autonomous Robots*, vol. 12, no. 3, pp. 287 – 300, 2002.

[4] U. Frese, P. Larsson, and T. Duckett, "A multilevel relaxation algorithm for simultaneous localisation and mapping," *IEEE Transactions on Robotics*, vol. 21, no. 2, pp. 1–12, 2005.

[5] U. Frese, "Treemap: An $O(logn)$ algorithm for indoor Simultaneous Localization and Mapping," *Journal of Autonomous Robots*, vol. 21, no. 2, pp. 103–122, 2006. [Online]. Available: http://www.informatik.uni-bremen.de/ ufrese/published/fresear06b.pdf

[6] F. Dellaert and M. Kaess, "Square root sam: Simultaneous location and mapping via square root information smoothing," *Int. Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1204, 2006.

[7] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Fast incremental smoothing and mapping with efficient data association," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2007.

[8] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2006, pp. 2262–2269.

[9] G. Grisetti, D. Lodi Rizzini, C. Stachniss, E. Olson, and W. Burgard, "Online constraint network optimization for efficient maximum likelihood map learning," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2008, pp. 1880–1885.

[10] R. Eustice, H. Singh, and J. Leonard, "Exactly sparse delayed-state filters," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2005, pp. 2428–2435.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: MIT Press, 2005.

[12] B. Bollobas, *Modern Graph Theory*. Springer Verlag, 1998.