Closed-form Algorithms in Mobile Positioning: Myths and Misconceptions

Niilo Sirola

***Abstract*—The iterative least squares method, or Gauss–Newton method, is a standard algorithm for solving general nonlinear systems of equations, but it is often said to be unsuited for mobile positioning with e.g. ranges or range differences or angle-of-arrival measurements. Instead, various closed-form methods have been proposed and are constantly being reinvented for the problem, claiming to outperform Gauss–Newton. We list some common conceptual and computation pitfalls for closed-form solvers, and present an extensive comparison of different closed-form solvers against a properly implemented Gauss–Newton solver. We give all the algorithms in similar notations and implementable form and a couple of novel closed-form methods and implementation details. The Gauss–Newton method strengthened with a regularisation term is found to be as accurate as any of the closed-form methods, and to have comparable computation load, while being simpler to implement and avoiding most of the pitfalls.**

I. Introduction

MOBILE positioning often boils down to solving systems of nonlinear equations, often in the least squares sense. (Notable exceptions are "RF fingerprint" or database correlation methods.) This can be done either in closed form or iteratively. The iterative approaches are often deemed impractical because they require an initial estimate to ensure global convergence, or require "too much computation". Instead, closed-form, or algebraic, methods are offered [1], [2], [3], [4], [5], [6], [7], [10] which often relax some aspects of the original problem or employ clever algebra in order to produce a simple "direct solution".

The main point of this work is to show that a carefully implemented iterative least squares method not only is more flexible and simpler to implement, but also can dominate the proposed closed-form approaches with respect to both RMS error and computation time.

Another goal of this work is to present a fair – or at least reproducible – simulation comparison of different iterative and closed form algorithms.

The disadvantages of iterative Gauss–Newton, described in Section V, versus (carefully implemented) direct methods can be summarized as:

- requires initial guess
- convergence not guaranteed
- may not handle nonlinearity well
- slightly more computation

Niilo Sirola was with Department of Mathematics, Tampere University of Technology, Finland. He is now with Taipale Telematics, PO Box 42, 33901 Tampere, Finland (e-mail: niilo.sirola@taipaletelematics.com)

Manuscript received December 1, 2009; revised March 11, 2010.

The advantages of Gauss–Newton are:

- uniform derivation for any number and any type of measurements
- simple and stable code – fewer bugs
- gives an error estimate
- minimizes the sum of squared errors – variance-optimal
- with suitable assumptions, is equivalent to a maximum-likelihood estimate or maximum-a-posteriori estimate
- no need to assume "sufficiently small" noise
- works with both correlated and independent noise
- extends to non-Gaussian and heavy-tail noises as the Reweighted Iterative Least Squares
- extends naturally to time series as the Extended Kalman Filter

II. Problem statement

Let x be the 2D or 3D coordinates of the receiver and s_i, $i = 1 \dots n$ the coordinates of the positioning beacons, be they cellular base stations, WLAN or UWB stations, pseudolites, etc.

For simplicity and just to illustrate the general point, we consider only range measurements. The inclusion of range difference measurements and angle-of-arrival measurements would in fact slant the situation towards the iterative Gauss–Newton method even more, as there are only a handful of direct methods capable of handling any combination of measurements.

The range between a beacon and the receiver is typically measured via a propagation delay in synchronised systems or round-trip time in unsynchronised systems. In both cases, the measurement equation is

$$r_i = \|s_i - x\| + v_i,$$

where v_i is the error in ith measurement, and the errors are assumed independent.

In the following, we mostly use the generalised measurement equation

$$y = h(x) + v,$$

where y denotes the measurement vector, h the vector-valued measurement function and v the error vector. This notation allows a wide range of different measurement types besides the range measurements considered. The object of a positioning method is to find a position vector x such that the error term needed to explain the measurement is as small as possible. Choosing for the solution the position x^* that minimizes $\|y - h(x)\|$, or, equivalently $\sum_i (y_i - h_i(x))^2$), is known as the *least squares solution* and is optimal in variance sense.

978-1-4244-7157-7/10/$26.00 ©2010 IEEE

III. Common pitfalls of closed-form methods

The following sections briefly list some either conceptually or computationally problematic points common to most proposed closed-form methods.

A. Squared ranges

Whereas the Gauss–Newton method aims to minimize $\|y - h(x)\|$, closed-form methods practically always minimize $\|y^2 - h(x)^2\|$ instead because the squaring transforms the Euclidian distances in the equations into more manageable second-degree polynomials. In the case where the system has a unique solution, that is, there is exactly one x such that $y = h(x)$, these approaches give identical results (assuming the squaring artefact solutions that satisfy $y = -h(x)$ are removed).

However, in the case of overdetermined systems where no position candidate satisfies all the measurement equations exactly, the closed-form methods produce a solution that minimises, not the sum of squared errors, but the sum of the measurement errors to the *fourth* power. Thus, large errors have more influence on the solution than they would in a least-squares treatment. We call position solutions obtained this way *least-quartic solutions.*

Furthermore, whereas the least-squares solution coincides with the maximum likelihood solution under Gaussian error distributions, the least-quartic solutions do not.

B. Negative measurements

While the geometric range between two distinct points necessarily is strictly positive, a realized measurement of the range does not have to be. If the error term is Gaussian, it can take arbitrarily large negative values and with certain probability this takes the measurement to the negative side as well. In theory, range measurements derived from round trip time cannot produce negative measurements, but in case there are any systematic biases in the system that need to be estimated and removed, slight perturbations in these may cause the range measurement to become negative in this case as well.

If equations are being squared on both sides, negative values either get treated as positives, or in a more careful implementation, truncated to zero. In neither case do the implied errors then follow the intended distribution.

Note that while the treatment of squaring artifacts in range measurements is straightforward, this is not the case with for example range difference or angle-of-arrival formulations.

C. Auxiliary variables and "partial linearisation"

A common technique for deriving closed-form solutions is to put the second-degree polynomials into a linear form by auxiliary variables. As such notational tricks cannot remove the inherent nonlinearity of the problem, such treatment has to be done with care.

As an illustration of this type of difficulty, consider a system of error-free squared ranges $y_i^2 = \|s_i - x\|^2$ that expand into

$$2s_i^T x = \|x\|^2 + \|s_i\|^2 - y_i^2, \tag{1}$$

where y_i are the scalar measurements, s_i the known beacon positions and vector x the unknown position.

Here starts the thin ice. A substitution $\lambda = \|x\|^2$, allows us to write

$$2s_i^T x = \lambda + \|s_i\|^2 - y_i^2, \tag{2}$$

which seems to be linear in the variables x and λ, only λ is not a free variable but a function of x. See Section IV-C for a sound way to solve this system while respecting the coupling of x and λ.

One proposed approach is to write the system of Eq. (2) as

$$\begin{bmatrix} 2s_1^T & -1 \\ \vdots & \\ 2s_n^T & -1 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} \|s_1\|^2 - y_1^2 \\ \vdots \\ \|s_n\|^2 - y_n^2 \end{bmatrix}$$

and proceed to solve the least-squares estimates for x and λ as if they were independent variables. An ad-hoc correction step typically follows to re-couple the resulting estimates of λ and x.

Another "novel" approach is to substitute $\gamma = \|x\|$ in (1) so that we get

$$2s_i^T x - \gamma^2 = \|s_i\|^2 - y_i^2$$

and treat this as a "linear system"

$$\begin{bmatrix} 2s_1^T & -\gamma \\ \vdots & \\ 2s_n^T & -\gamma \end{bmatrix} \begin{bmatrix} x \\ \gamma \end{bmatrix} = \begin{bmatrix} \|s_1\|^2 - y_1^2 \\ \vdots \\ \|s_n\|^2 - y_n^2 \end{bmatrix}.$$

The occurrence of γ in the left-hand matrix may be hidden from view by yet another variable substitution. The system is solved by plugging in some initial guess of γ in the matrix and then solving for the two unknowns. The testing results may look somewhat odd, which prompts an additional "correction step" or some constraints on γ.

Yet another technique in this category is to treat all the unknown ranges as variables to be solved in addition to the position. This system has more unknowns than measurements, but numerical software (such as MATLAB) still happily produce a least-squares solution that is practically meaningless. The authors of such methods typically employ yet another "corrective" step to end up with presentable results.

D. Numerical instability

Especially when the target platform has only single precision floating point or fixed point arithmetics, numerical problems may arise in closed-form methods that involve squaring of the measurement equations.

As an example of scaling problems, consider the matrix

$$A = \begin{bmatrix} s_1^T & 1 \\ s_2^T & 1 \end{bmatrix}$$

where s_1 and s_2 are beacon position vectors. Suppose somewhere in the algorithm we have to compute the inverse

$$(A^T A)^{-1} = \begin{bmatrix} s_1 s_1^T + s_2 s_2^T & s_1 \\ s_2^T & 1 \end{bmatrix}^{-1}.$$

If the coordinates of the beacons are large numbers, this matrix may become numerically singular just because of bad scaling. GPS satellites, for example, have coordinate values of the order of 10^7 metres, and thus the above matrix will have elements differing 14 orders of magnitude, which makes it almost ill-conditioned even with double-precision presentation.

Gauss–Newton, on the other hand, has only unit vectors in the Jacobian matrix, and only has to subtract un-squared ranges and thus should be significantly more tolerant to numerical problems of this kind, assuming the vector norm computation is programmed correctly.

E. Correlated measurement errors

Measurement errors are usually assumed independent, leading to a diagonal covariance matrix. This covariance is usually assumed known, and can easily be taken optimally into account in the Gauss–Newton algorithm as well as most of the closed-form methods as a weighting matrix.

However, when dealing with squared measurements, the covariance structure becomes more complicated.

Assume that the range measurements y are unbiased, i.e. they have as expectation value the true range r, and joint covariance Σ. Writing the vector of squared ranges as $r \otimes r$, where $\otimes$ denotes element-wise product, we get

$$E(y \otimes y) = r \otimes r + \operatorname{diag}(\Sigma) \tag{3}$$

and

$$V(y \otimes y) = 4\ rr^T \otimes \Sigma + 2\ \Sigma \otimes \Sigma. \tag{4}$$

Thus, the covariance of the squared range measurements depends not only on the magnitude of the measurement noise, but also on the magnitude of the unknown ranges themselves! Algorithms often handle this by approximating the unknown values either with the measured ranges, or range estimates computed from an interim position solution. The latter case suggests an iteration step where the computed solution is used to refine this estimate of the squared range variance. The conditions for convergence of such iteration are usually not even considered.

Furthermore, the second term in Eq. (4) is usually dropped with a mention of "small error assumption". In this treatment we retain it because in our testing the measurement errors may well be of the same magnitude as the ranges.

IV. Closed-form methods for overdetermined systems

This section presents some previously proposed and some novel solvers for overdetermined systems of range measurements. All algorithms are written with similar notation to ease comparison. Whenever an algorithm offers both far-field and near-field versions, we show the near-field one. If approximation of unknown quantities with computed/measured ones is called for, we substitute them in the algorithms.

In case the reader feels some of the methods are mispresented here, the source code and the full test suite are available from the author at `www.niilosirola.fi`.

A. Baseline

As a baseline method, compute just the mean of the station coordinates:

$$\hat{x} = \frac{1}{n} \sum_{i=0}^{n} s_i$$

As this does not even depend on the measurements, any decent method should beat it, at least on average.

B. Simple intersection

Also called the "line-of-position" method, this method has been given in a more complex form at least in [4] and [9, Sec. IV B].

The method is derived by expanding the squared range equations to

$$2s_i^T x = \|x\|^2 + \|s_i\|^2 - r_i^2, \tag{5}$$

and subtracting one of the equations, say the first, from the rest. This produces a system of $N-1$ linear equations where the non-linear term $\|x\|^2$ has been eliminated.

Geometrically, this corresponds to considering each $N-1$ pairs of circles (or spheres in 3D), and replacing their intersection with the line (plane) on which the intersection lies. Note that because of elimination, this method is (asymptotically) equivalent to solving the position with one less measurement.

Algorithm 1 Simple intersection

$$A = \begin{bmatrix} 2s_1^T \\ \vdots \\ 2s_n^T \end{bmatrix}, \qquad b = \begin{bmatrix} \|s_1\|^2 - r_1^2 \\ \vdots \\ \|s_n\|^2 - r_n^2 \end{bmatrix}$$
$$D = \begin{bmatrix} -\mathbf{1} & I_{N-1} \end{bmatrix}$$

Then the estimate is

$$\hat{x} = (A^T D^T D A)^{-1} A^T D^T D b.$$

C. Range-Bancroft

Bancroft's method for gps is applicable also to any range difference measurements. For simplicity, we derive the method for range measurements, a similar derivation applies for the biased range measurements without the use of Minkowski functionals as in the original text.

Note that similarly to the Simple Intersection method, Bancroft efficiently differentiates the measurements and thus uses one effective measurement less than the other methods.

The algorithm is derived as follows. Start by expanding the squared range equations to

$$r_i^2 = \|s_i\|^2 - 2s_i^T x + \|x\|^2.$$

Write this as

$$\begin{bmatrix} 2s_1^T \\ \vdots \\ 2s_n^T \end{bmatrix} x = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \|x\|^2 + \begin{bmatrix} \|s_1\|^2 - r_1^2 \\ \vdots \\ \|s_n\|^2 - r_n^2 \end{bmatrix}$$
$$Ax = \mathbf{1}\|x\|^2 + b$$

Multiply both sides with the Moore–Penrose pseudo-inverse of A to get

$$x = A^\dagger \mathbf{1}\|x\|^2 + A^\dagger b.$$

Note that although this is a least-squares solution of an overdetermined system of equations, the residual minimized is not the measurement residual but something completely different. However, for an exactly determined system, this solution agrees with the other methods.

Write $p = A^\dagger \mathbf{1}$ and $q = A^\dagger b$ and note that these are computable from known quantities. Then we have

$$x = p\|x\|^2 + q.$$

Taking the squared norm of both sides gives

$$\begin{aligned}\|x\|^2 &= \|p\|^2\|x\|^4 + 2p^T q\|x\|^2 + \|q\|^2 \\ \|p\|^2\|x\|^4 &+ (2p^T q - 1)\|x\|^2 + \|q\|^2 = 0\end{aligned}$$

which is quadratic in $\|x\|^2$. We thus solve

$$\|p\|^2 t^2 + (2p^T q - 1)t + \|q\|^2 = 0 \tag{6}$$

for t and get zero, one or two real roots. In most cases, we get two roots, substitute them back to $x^* = pt^* + q$ and pick the one with the smaller residual $\sum_{i=1}^n (y_i - \|s_i - x^*\|)^2$.

Algorithm 2 Range-Bancroft

$$A = \begin{bmatrix} 2s_1^T \\ \vdots \\ 2s_n^T \end{bmatrix}, \qquad b = \begin{bmatrix} \|s_1\|^2 - r_1^2 \\ \vdots \\ \|s_n\|^2 - r_n^2 \end{bmatrix}$$
$$p = (A^T A)^{-1} A^T \mathbf{1}$$
$$q = (A^T A)^{-1} A^T b$$

Solve the two roots for t from

$$\|p\|^2 t^2 + (2p^T q - 1)t + \|q\|^2 = 0,$$

and get the two solution candidates

$$\hat{x}_i = pt_i + q, \quad i = 1, 2$$

Pick the candidate with the smallest residual $\|y - h(\hat{x}_i)\|$ as the final estimate.

D. Beck et al.

Beck, Stoica and Li (2008) give a procedure for computing the exact least-quartic solution for the range equations. The procedure requires the use of bisection[1] method to find a root of a univariate strictly monotonous function, so again this is not strictly a closed-form method, but still one with guaranteed convergence. They also give a modification for the range difference case.

[1]Although even the bisection with a semi-infinite starting range is not well defined and requires some ad-hockery to get started.

Algorithm 3 Beck et al. [3]

$$A = \begin{bmatrix} 2s_1^T & -1 \\ \vdots & \\ 2s_n^T & -1 \end{bmatrix}, \qquad b = \begin{bmatrix} \|s_1\|^2 - r_1^2 \\ \vdots \\ \|s_n\|^2 - r_n^2 \end{bmatrix}$$
$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad q = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{2} \end{bmatrix}$$

Let

$$\hat{z}(\lambda) = (A^T A + \lambda P)^{-1}(A^T b - \lambda q)$$

Find the zero of

$$\phi(\lambda) = \hat{z}(\lambda)^T P \hat{z}(\lambda) + 2q^T \hat{z}(\lambda)$$

by bisection, the function is known to be strictly decreasing in $[-1/\lambda_1,\ \infty]$, where

$$\lambda_1 = \max \text{eig } (A^T A)^{-\frac{1}{2}} P (A^T A)^{-\frac{1}{2}}.$$

The final position estimate $\hat{x}$ is the first m components of $\hat{z}(\lambda^*)$.

E. Cheung et al.

Cheung et al. [6] give a constrained weighted least squares solution for range measurements. The unknowns to be solved for are the position and squared distance to first beacon.

The original paper does not give any means for solving the five-root equation. For efficient implementation in this test, the five-root equation was manipulated into a fifth-degree polynomial on λ with coefficients

$$\begin{aligned}p_5 =& -\frac{1}{2}c_3 g_3 \gamma_1^2 \gamma_2^2 \\ p_4 =& c_3 f_3 \gamma_1^2 \gamma_2^2 + (-\frac{1}{4}c_2 g_2 - c_3 g_3)\gamma_1^2 \gamma_2 \\ &+ (-\frac{1}{4}c_1 g_1 - c_3 g_3)\gamma_1 \gamma_2^2 \\ p_3 =& (\frac{1}{2}c_2 f_2 + 2c_3 f_3 - \frac{1}{2}e_2 g_2)\gamma_1^2 \gamma_2 \\ &+ (\frac{1}{2}c_1 f_1 + 2c_3 f_3 - \frac{1}{2}e_1 g_1)\gamma_1 \gamma_2^2 \\ &+ (-\frac{1}{2}c_2 g_2 - \frac{1}{2}c_3 g_3)\gamma_1^2 + (-\frac{1}{2}c_1 g_1 - \frac{1}{2}c_3 g_3)\gamma_2^2 \\ &+ (-\frac{1}{2}c_1 g_1 - \frac{1}{2}c_2 g_2 - 2c_3 g_3)\gamma_1 \gamma_2 \\ p_2 =& e_2 f_2 \gamma_1^2 \gamma_2 + e_1 f_1 \gamma_1 \gamma_2^2 \\ &+ (c_2 f_2 + c_3 f_3)\gamma_1^2 + (c_1 f_1 + c_3 f_3)\gamma_2^2 \\ &+ (-e_1 g_1 - e_2 g_2 + c_1 f_1 + c_2 f_2 + 4c_3 f_3)\gamma_1 \gamma_2 \\ &+ (-\frac{1}{4}c_1 g_1 - c_2 g_2 - c_3 g_3)\gamma_1 \\ &+ (-c_1 g_1 - \frac{1}{4}c_2 g_2 - c_3 g_3)\gamma_2\end{aligned}$$

$$
\begin{aligned}
p_1 =&(2e_1f_1 + 2e_2f_2)\gamma_1\gamma_2 \\
&+ (\frac{1}{2}c_1f_1 + 2c_2f_2 + 2c_3f_3 - \frac{1}{2}e_1g_1)\gamma_1 \\
&+ (2c_1f_1 + \frac{1}{2}c_2f_2 + 2c_3f_3 - \frac{1}{2}e_2g_2)\gamma_2 \\
&- \frac{1}{2}c_1g_1 - \frac{1}{2}c_2g_2 - \frac{1}{2}c_3g_3 \\
p_0 =& e_1f_1\gamma_1 + e_2f_2\gamma_2 + c_1f_1 + c_2f_2 + c_3f_3.
\end{aligned}
$$

The roots of the polynomial can be found with standard numerical methods, and the one closest to zero picked from those.

Algorithm 4 Cheung et al. [6]

$$
A = \begin{bmatrix} 2s_1^T & -1 \\ \vdots & \\ 2s_n^T & -1 \end{bmatrix}, \qquad b = \begin{bmatrix} \|s_1\|^2 - r_1^2 \\ \vdots \\ \|s_n\|^2 - r_n^2 \end{bmatrix}
$$

$$
B = \begin{bmatrix} r_1 & & \\ & \ddots & \\ & & r_n \end{bmatrix}
$$

$$
\Psi = B\Sigma B
$$

$$
P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad q = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{2} \end{bmatrix}
$$

Compute U and Λ from the eigenvalue decomposition

$$
(A^T\Psi^{-1}A)^{-1}P = U \begin{bmatrix} \gamma_1 & & \\ & \gamma_2 & \\ & & 0 \end{bmatrix} U^{-1}
$$

and get

$$
\begin{aligned}
c &= 2U^Tq \\
g &= 2U^{-1}(A^T\Psi^{-1}A)^{-1}q \\
e &= (\Psi^{-1}AU)^Tb \\
f &= U^{-1}(A^T\Psi^{-1}A)^{-1}A\Psi^{-1}b.
\end{aligned}
$$

Find the root λ^* closest to zero of the following five-root equation

$$
\begin{aligned}
c_3f_3 - \frac{\lambda}{2}c_3g_3 + \sum_{i=1}^{2}\frac{c_if_i}{1+\lambda\gamma_i} - \frac{\lambda}{2}\sum_{i=1}^{2}\frac{c_ig_i}{1+\lambda\gamma_i} \\
+ \sum_{i=1}^{2}\frac{e_if_i\gamma_i}{(1+\lambda\gamma_i)^2} - \frac{\lambda}{2}\sum_{i=1}^{2}\frac{(e_ig_i + c_if_i)\gamma_i}{(1+\lambda\gamma_i)^2} \\
+\frac{\lambda^2}{4}\sum_{i=1}^{2}\frac{c_ig_i\gamma_i}{(1+\lambda\gamma_i)^2} = 0.
\end{aligned}
$$

Compute the estimate of the augmented state as

$$
\hat{z} = (A^T\Psi^{-1}A + \lambda^*P)^{-1}(A^T\Psi^{-1}b - \lambda^*q)
$$

and take the first two elements as the position estimate $\hat{x}$.

V. Gauss–Newton

Weighted Iterative Least Squares (WLS), given in Algorithm 5 is called in this paper as "the" Gauss–Newton method. It is a numerically robust method that can easily make use of almost any kind of measurement found in mobile location setting, and also assign the appropriate weighting, at least in the case of additive noise with finite variance.

A slight disadvantage is the greater computational cost than the simplest closed-form methods, the need for an initial starting position (although we can use for example the result of the Simple Intersection method for this), and that in the case of multiple ambiguous solutions only one of the candidates is found.

The algorithm is however very straightforward to implement as long as basic linear algebra operations are available, and does not require for example solving fifth-degree polynomials. The only hurdle with more complex measurement models is the need to explicitly compute the Jacobian matrix, or the derivative of the measurement equation. In case of range measurements, it just consists of the unit vectors pointing from the current position estimate to the beacons.

In the algorithm listing the backslash operator $\backslash$ is used as a shorthand for solving a linear system in least squares sense, i.e. $x = A\backslash b$ means "let x be the least-squares solution to $Ax = b$".

Algorithm 5 Iterative Weighted Least Squares

1) Choose initial guess x_0 and stopping tolerance δ. Set $k = 0$.
2) Compute
$$
J_k(x) = \begin{bmatrix} \frac{s_1 - x}{\|s_1 - x\|}^T \\ \vdots \\ \frac{s_n - x}{\|s_n - x\|}^T \end{bmatrix}
$$
3) Set $x_{k+1} = x_k + \Delta x_k$, where
$$
\Delta x_k = -(\Sigma^{-\frac{1}{2}}J_k)\backslash\left(\Sigma^{-\frac{1}{2}}\left(h(x_k) - r\right)\right)
$$
4) If stopping condition $\|\Delta x_k\| < \delta$ is not satisfied and $k \leq k_{\max}$, increment k and repeat from Step 2.

A. Regularised Gauss–Newton

The divergence problems of the vanilla Gauss–Newton can be corrected with the use of *prior information*. As a hint of what type of prior information to use, the study of numerical results of the closed-form methods indicated that the methods have some built-in preference to positions near the beacon coordinates over positions farther away. This is reasonable behaviour since a receiver is usually more likely to hear beacons close-by than those far away. This kind of prior assumption can easily be incorporated in the Gauss–Newton algorithm as a regularisation term. A kind of theoretical justification for this is that the resulting algorithm is equivalent to a Bayesian maximum-a-posteriori algorithm with prior distribution chosen around the center point of the beacons.

Algorithm 6 shows that the only modification required is in the computation of Δx. We used the mean point of the beacon coordinates as the regularisation point x_r, and the

regularisation coefficient $c = 10^{-4}$, which corresponds to prior distribution centered at x_r, with a standard deviation of 10 kilometres.

Algorithm 6 Regularised Gauss–Newton

1) Choose initial guess x_0 and stopping tolerance δ. Set $k = 0$.
2) Compute
$$J_k(x) = \begin{bmatrix} \frac{s_1 - x}{\|s_1 - x\|}^T \\ \vdots \\ \frac{s_n - x}{\|s_n - x\|}^T \end{bmatrix}$$
3) Set $x_{k+1} = x_k + \Delta x_k$, where
$$\Delta x_k = -(\Sigma^{-\frac{1}{2}} J_k + cI) \backslash \left(\Sigma^{-\frac{1}{2}} \left(h(x_k) - r \right) + c(x - x_r) \right)$$
4) If stopping condition $\|\Delta x_k\| < \delta$ is not satisfied and $k \leq k_{\max}$, increment k and repeat from Step 2.

VI. Testing

The simulation setup follows that of [7]. Real range measurements from cellular stations are not readily available, and if they were, the details of the channel model and forming the measurement equations would draw the attention away from the choice of positioning algorithm.

The five base stations are at coordinates $[0, 0]$ m, $[3000\sqrt{3}, 3000]$ m, $[0, 6000]$ m, $[-3000\sqrt{3}, 3000]$ m, and $[-3000\sqrt{3}, -3000]$ m. In [7], the user position was fixed at $[x, y] = [1000, 2000]$ m, but instead of that we consider user positions randomly placed in the whole 12 km by 12 km region the base stations cover.

The measurement errors in this test are taken to be independent and identically distributed, as not all the closed-form methods could be trivially generalised to the correlated noise case. The standard deviation of the measurement noise sweeps from 1 m to 10 km in 30 steps, and for each noise level 1000 sets of random true positions and measurements were generated and each algorithm ran on the same data.

Fig. 1 shows the RMS error of each solver normalised by the noise level:

$$e_{\text{norm}} = \frac{1}{n} (\bar{x} - \hat{x})^T \left(J(\bar{x})^T \Sigma^{-1} J(\bar{x}) \right)^{-1} (\bar{x} - \hat{x})^T,$$

where $\hat{x}$ is the estimate and $\bar{x}$ the true position. This is connected to the first-order approximation of the Cramer–Rao lower bound such that a Cramer–Rao optimal solver whould have the mean normalised error of 1. With noise standard deviation less than 200–300m, all the solvers give nearly identical and optimal results, except for the Simple intersection, Range-Bancroft, and the baseline solver (that just uses the mean of station coordinates). However, at noise level of about 2km, the unregularised Gauss–Newton starts diverging (note that this is usually seen as the main weakness of this method!), while the regularised Gauss–Newton still works as intended. It could be argued that noise levels as large

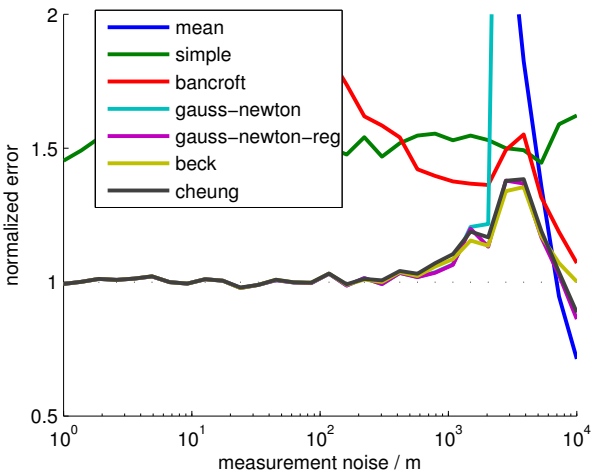

Fig. 1. Results of the noise sweep tests with independent Gaussian measurement errors

as this represent somewhat pathological case, as the beacons are only a few kilometres apart[2].

Finally, at very extreme noise levels of 6–10 km, some of the solvers interestingly give better than optimal results, replicating the behaviour seen in [7]. Note that the baseline solver is then actually the most accurate, indicating that with these noise levels the best strategy is to ignore all measurement and use just the prior information and assume the user is somewhere close to the beacons heard.

Several other test runs with different measurement geometries, different random number sequences or choices of noise levels yielded similar results. The quality of error estimates given by the algorithms were not evaluated because so few of the closed-form solvers offer any.

The computation times for the MATLAB implementation on a laptop running on 1.4 GHz Celeron M processor ranged from 0.3 ms per fix for the Simple Intersection method to 1.1 ms per fix for Cheung to 2 ms per fix for the regularised Gauss–Newton method. The exact timings depend on the maturity of the implementation as well as the properties of the MATLAB Just-In-Time optimisation engine. A fair attempt was made to optimise all the algorithms to the same extend, for example converting the five-root equation of Cheung into polynomial form, resulting in several orders of magnitude of speed-up.

VII. Conclusions

Even with this fairly limited choice of problem, with simple noise assumptions chosen to favour closed-form algorithms, iterative least squares algorithm was found to be a competitive solution. A rather rudimentary regularisation term removes the divergence problem previously seen especially if using unrealistic high noise levels.

This study also highlights the importance of using mature and numerically stable subroutines instead of complicated ad-hockery. Especially when coupled with limited testing and

[2]Additionally, real-world noise levels this high would probably not be independent and Gaussian but dominated by temporally and spatially highly correlated non-line-of-sight components.

poorly implemented benchmark solutions, it can lead to weak "novel" algorithms for problems that have already been solved.

VIII. Acknowledgements

This research was conducted within the Personal Positioning Algorithms group at TUT/Math, funded by Nokia Corporation. The author thanks prof. Robert Piché, Matti Raitoharju, and the anonymous reviewers for their comments and suggestions.

References

[1] Awange, J. L., and Fukuda, Y. Resultant optimization of the three-dimensional intersection problem. *Survey Review 39*, 304 (April 2007), 100–108.

[2] Bancroft, S. An algebraic solution of the GPS equations. *IEEE Transactions on Aerospace and Electronic Systems 21*, 7 (1986), 56–59.

[3] Beck, A., Stoica, P., and Li, J. Exact and approximate solutions of source localization problems. *IEEE Transactions on Signal Processing 56*, 5 (May 2008), 1770–1778.

[4] Caffery, Jr., J. J. A new approach to the geometry of TOA location. In *Proceedings of Vehicular Technology Conference VTC 2000* (2000), pp. 1943–1949.

[5] Chan, Y. T., and Ho, K. C. A simple and efficient estimator for hyperbolic location. *IEEE Transactions on Signal Processing 42*, 8 (1994), 1905–1915.

[6] Cheung, K. W., Sho, H. C., Ma, W.-K., and Chan, Y. T. Least squares algorithms for time-of-arrival-based mobile location. *IEEE Transactions on Signal Processing 52*, 4 (2004), 1121–1128.

[7] Cheung, K. W., Sho, H. C., Ma, W.-K., and Chan, Y. T. A constrained least squares approach to mobile positioning: algorithms and optimality. *EURASIP Journal on Applied Signal Processing* (2006), 1–23.

[8] Foy, W. H. Position-location solutions by Taylor-series estimation. *IEEE Transactions on Aerospace and Electronic Systems 12*, 2 (1976), 187–194.

[9] Holger, K., and Willig, A. *Protocols and Architectures for Wireless Sensor Networks.* John Wiley & Sons, Chichester (2005).

[10] Sirola, N. A versatile algorithm for local positioning in closed form. In *Proceedings of the 8th European Navigation Conference GNSS 2004, May 16-19, Rotterdam* (2004).

[11] Yan, J., Tiberius, C., Bellusci, G., and Janssen, G. Feasibility of Gauss-Newton method for indoor positioning. In *Proceedings of PLANS 2008* (2008).