TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY
Julkaisu 675 • Publication 675

Niilo Sirola

Mathematical Methods for Personal Positioning and Navigation

Tampere 2007

Tampereen teknillinen yliopisto. Julkaisu 675
Tampere University of Technology. Publication 675

Niilo Sirola

Mathematical Methods for Personal Positioning and Navigation

Thesis for the degree of Doctor of Technology to be presented with due permission for public examination and criticism in Rakennustalo Building, Auditorium RG202, at Tampere University of Technology, on the 5th of October 2007, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2007

ISBN 978-952-15-1812-6 (printed)
ISBN 978-952-15-1859-1 (PDF)
ISSN 1459-2045

Abstract

Computing the position of a personal mobile device based on a mix of various types of measurements requires a wide array of mathematical concepts ranging from optimisation to robust estimation and nonlinear filtering theory. Algorithms for positioning and navigation have surfaced concurrently with the development of new measurement equipment and navigation infrastructure. However, most solutions and algorithms pertain only to certain equipment, involving just a single or few measurement sources.

This work synthesises existing techniques into a general framework covering static positioning, filtering, batch positioning and dead reckoning. Measurements are not restricted to any specific technology, equation form or distribution assumption.

The static positioning problem, deducing position from a set of simultaneous measurements, is considered first. Parallels between geometric, least squares and statistical approaches are given. The more complex problem of time series estimation can be solved by navigation filters that also make use of all past measurements and information about the system dynamics. Different filter implementations can be derived from the ideal Bayesian filter by choosing different probability density function (pdf) approximation schemes. The standard methods are briefly introduced in this context along with a novel generalisation of a piecewise defined grid filter.

Finally, given the wide variety of existing and potential filter implementations, fair and expressive methods for comparing the quality and performance of nonlinear filters are discussed.

Preface

Personal positioning involves an exciting blend of several disciplines ranging from digital signal processing through probability theory to the special theory of relativity. From a mathematician's point of view, myriads of equally interesting problems arise from the different aspects of the system.

This research, funded by Nokia Corporation, was carried out at the Personal Positioning Algorithms group at the Institute of Mathematics, Tampere University of Technology during 2000–2007. I gratefully acknowledge additional financial support from Nokia Foundation, Finnish Cultural Foundation and Suoja Foundation.

I would like to thank my colleagues in the research group and at the institute, especially Simo Ali-Löytty and Kari Heine, for enlightening discussions and instant feedback over the years I worked on this research, my supervisor Prof Robert Piché for his professional guidance and counsel and for letting me know (on several occasions) that he wasn't rushing me, Dr Jari Syrjärinne from Nokia Corporation for providing insight and interesting problems to be solved, the pre-examiners Prof Miroslav Šimandl and Prof Xiao-Wen Chang for their comments and suggestions and, most of all, my family and my dear wife Laura for being there for me and providing the much-needed distraction from work.

Tampere, August 2007,

Niilo Sirola

Contents

List of publications **vi**

Related publications **vii**

1 Introduction **1**
1 Background . . . 3
1.1 Problem statement . . . 4
2 Measurement models . . . 5
2.1 Radio navigation . . . 5
2.2 Inertial sensors . . . 9
2.3 Offline Information . . . 10
3 Static positioning techniques . . . 11
3.1 Closed-form and geometric solutions . . . 12
3.2 Residual minimisation . . . 15
3.3 Maximum likelihood and Bayesian methods . 17
3.4 Database correlation . . . 19
4 Time series estimation and navigation filters . . . 21
4.1 Kalman-type filters . . . 23
4.2 Sequential Monte Carlo filters . . . 25
4.3 Numerical filters . . . 27
5 Benchmarking navigation filters . . . 30
5.1 Comparison criteria . . . 31
5.2 Combining the criteria . . . 34
6 Conclusions and future work . . . 34

References **39**

Publications **49**

List of publications

This thesis consists of an introduction and the following publications, in chronological order:

P1. "GPS position can be computed without navigation data" (with Jari Syrjärinne) In *Proceedings of the ION GPS 2002*, Portland OR, September 2002, pages 2741–2744.

P2. "A versatile algorithm for local positioning in closed form" In *Proceedings of the 8th European Navigation Conference GNSS2004*, Rotterdam, May 2004.

P3. "Numerical integration methods in local positioning" (with Robert Piché and Henri Pesonen) In *Proceedings of the 2nd Workshop on Positioning, Navigation and Communication (WPNC05)*, Hannover, March 2005, pages 21–30.

P4. "Moving grid filter in hybrid local positioning" (with Simo Ali-Löytty) In *Proceedings of the 10th European Navigation Conference GNSS2006*, Manchester, May 2006.

P5. "Benchmarking nonlinear filters" (with Simo Ali-Löytty and Robert Piché) In *Proceedings of the IEEE Nonlinear Statistical Signal Processing Workshop NSSPW2006*, Cambridge, September 2006.

P6. "Exhaustive global grid search in computing receiver position from modular satellite range measurements" *Journal of Physics: Conference series*, 52:73–82, 2006.

P7. "Nonlinear filtering with piecewise probability densities" Research report 87, *Tampere University of Technology, Institute of Mathematics*, 2007.

Related publications

The following publications relate to the subject, but are either redundant or contain only minor contribution from the author:

R1. Niilo Sirola and Paula Syrjärinne. Solving GPS time and position without navigation data. In *Proceedings of the ENC-GNSS 2002*, 2002.

R2. Niilo Sirola, Robert Piché, and Jari Syrjärinne. Closed-form solutions for hybrid cellular/GPS positioning. In *Proceedings of the ION GPS/GNSS 2003*, pages 1613–1619, 2003.

R3. Niilo Sirola and Jari Syrjärinne. Method for performing positioning and an electronic device. US 6,784,834. Granted Aug 31, 2004.

R4. Niilo Sirola and Jari Syrjärinne. Determining the position of a receiver and/or the system time of a positioning system. US 6,865,478. Granted Mar 8, 2005.

R5. Simo Ali-Löytty, Niilo Sirola, and Robert Piché. Consistency of three Kalman filter extensions in hybrid navigation. In *Proceedings of the European Navigation Conference GNSS 2005*, July 19–22, 2005, Munchen, 2005.

R6. Simo Ali-Löytty, Niilo Sirola, and Robert Piché. Consistency of three Kalman filter extensions in hybrid navigation. *European Journal of Navigation* 4(1), 2006.

R7. Niilo Sirola and Simo Ali-Löytty. Local positioning with parallelepiped moving grid. In *Proceedings of the 3rd Workshop on Positioning, Navigation and Communications WPNC'06*, March 16, 2006.

R8. Simo Ali-Löytty and Niilo Sirola. A modified Kalman filter for hybrid positioning. In *Proceedings of the ION GNSS 2006*, September 26–29, 2006.

CHAPTER 1

Introduction

This thesis consists of an introduction and seven articles published in scientific conferences and journals. The purpose of this introductory chapter is not to repeat the derivations or results given in the publications [P1]–[P7], but rather to give a short unified background, and summarise the contribution in context.

The main contributions of the publications are:

- A global method for GPS time recovery [P1], which completes the development of the idea originating from [74] and further developed for the local case in the author's MSc thesis. Moreover, the method is generalised to a class of global optimisation problems [P6]. See Section 3.2 for an overview.
- Geometric classification of closed form positioning problems and a general solution process [P2], see Section 3.1 for details.
- A study and simulation testing of numerical integration methods suitable for Bayesian positioning computations [P3], see Section 3.3.

- A parallelepiped piecewise continuous grid filter for the personal positioning problem [P4] and its generalisation to nonlinear filtering [P7], see Section 4.3.
- Concise treatment on why it is so difficult to evaluate the quality of nonlinear filters [P5] and some new means to visually summarise the results of a large test bench, see Section 5.

The author's role in the shared publications:

- Publication P1: based on ideas originally presented in [74], the author developed the presented algorithm and wrote the code as well as the manuscript.
- Publication P3: the author composed the code and ran the tests jointly with Henri Pesonen, and wrote the manuscript except for the introduction section.
- Publication P4: the author developed the algorithm and the code, and wrote the manuscript.
- Publication P5: the author came up with the main ideas and wrote the manuscript.

1 Background

> **navigation** *n*
> the science of getting ships, aircraft, or spacecraft from place to place; *especially* : the method of determining position, course, and distance traveled [53]

Navigation means have probably been the first subject of scientific research right after fire, and involve technological breakthroughs ranging from astrolabe and compass through seaworthy chronometer (precise clock) needed for longitude determination to radio and satellite navigation systems of the 1960's.

The success of the satellite-based GPS (Global Positioning System) around the turn of the millennium has brought navigation systems from ships and aeroplanes to cars and personal hand-held devices, and the next few years will see a GPS chip finding its way into the majority of mobile phones and other wireless devices sold.

Precise and quick positioning is now globally available, provided that enough of the sky is visible in order to get enough satellites in fix. This leaves an availability gap in tightly built areas, indoors and underground that is currently being filled with more or less sophisticated solutions employing assisted GPS [1, 9, 75, 77], WLAN or WiFi signals [33], the cellular network [8, 35], or even miniaturised inertial navigation systems [19].

Still, there seems to be a lot of untapped potential for extraction of location information from any "signal of opportunity" that a mobile device might be able to receive, whether they be intended for positioning use or not. There are many exciting mathematical problems to be solved, requiring techniques from global optimisation, analytic geometry, estimation theory, nonlinear filtering, numerical analysis, information science, high-performance computing, software design, not to mention all the mathematical and statistical modelling to be done, for example, with measurement error and user dynamics models.

1.1 Problem statement

By personal positioning, we mean computing the position, velocity, heading and possibly other states of interest of a personal mobile device. The performance of especially the filtering methods depends on the motion state of the receiver, and navigation should be possible regardless of whether the device is being carried in hand, pocket or in backpack or whether the user is walking, cycling, skiing, in car, on bus, on boat, on aeroplane, or parachuting; both outdoors and indoors as well as underground.

We call the device we wish to track the "receiver" although it may or may not have also transmit capabilities. The fixed or moving signal sources relative to which measurements are made are referred to as "beacons", and their coordinates are usually known. In the scope of this work, it does not matter which of the entities initiates the positioning process, or whether the position computations take place in the receiver, beacon, both or somewhere else.

In the Bayesian positioning framework [29, 67], all unknowns of interest are stacked into the *state vector*, x_k where subscript k refers to the time instant t_k, and the system described by the equations:

$$\begin{aligned}
&\text{prior state:} && x_0 && \text{(1a)}\\
&\text{measurement model:} && y_k = h_k(x_k) + v_k && \text{(1b)}\\
&\text{dynamics model:} && x_{k+1} = f_k(x_k) + w_k && \text{(1c)}
\end{aligned}$$

The unknown state x_k cannot be measured directly; information about it is given only indirectly through measurements y_k. The prior state x_0 and the measurement and process noises v_k and w_k are usually stochastic and their distribution or some statistics known. Usually, certain independence conditions are required to simplify the computations, such as the x_0, v_k and w_k being mutually independent for all k.

A *navigation filter* uses all three pieces of information given by the equation group (1), thus fusing together all measurements gathered up and until the kth time step. We also consider algorithms that use only the measurement model, called *static positioning*

algorithms. If they use additionally the prior distribution, they are called *Bayesian static positioning*.

Not considered in this work, *batch positioning* would use just the measurement and dynamics model, and *dead reckoning* in this framework is interpreted as propagating the prior distribution with the dynamics model without any measurements.

2 Measurement models

Following is a short review of some possible sources of measurements and associated types of measurement models. For now, we just try to model the error characteristics of the measurement and how the measurement depends on the true state of the receiver, thus forming the measurement model (1b). This has to be seen as a task separate from the inverse problem of deducing state based on measurements, although technical publications usually do not distinguish between modelling the problem and solving it.

Although we concentrate on radio navigation, the same principles can readily be applied to non-radio signals, for example acoustic, light, seismic waves, or possibly even surface waves in liquid.

2.1 Radio navigation

The signals of most radio networks can be exploited for positioning, the ideal being to use any existing and available signals as they are, without requiring any new infrastructure or costly changes to existing transmitter station software. At the very least, even if the signals were not designed for positioning purposes, the existence of a signal indicates that the transmitter is near. This knowledge can be converted to information about the receiver position with the help of on-line hot-spot databases such as [63, 83].

The basic radio navigation measurement types are listed as follows.

Range between a beacon and the receiver is typically measured via a propagation delay or round-trip time.

Biased range (or pseudo-range) is the geometrical range contaminated by an additive bias common to all range measurements. A typical source of bias is the receiver clock error in propagation delay measurements. The clock bias is considered an extra "nuisance" variable to be solved.

Range difference typically results from subtracting a biased range measurement from another in order to eliminate the bias. Now the bias does not have to be solved explicitly, but the measurement errors become correlated if the same range is used in forming several differences.

Deltarange is the time derivative of a range. Depending on the implementation, it measures either the instantaneous rate of change, or accumulated change over some period of time. Deltaranges can also have a common bias that is treated similarly as with range measurements.

Received signal power depends, among other things, on distance and it can be either mapped to a distance measurement via some loss function, or used as it is along with a pre-prepared loss model.

Angle of arrival requires special antenna arrangements to detect the direction of signal. The angle measurement usually has poor accuracy but is still useful for excluding impossible locations.

Presence of signal implies receiver is within the maximum range and transmission cone of the beacon.

Absence of signal can, in some cases, be used to infer that the receiver is outside the maximum range or the transmission cone of the beacon.

An additional variation to most measurement types comes from **quantisation** of measured values that should be accounted for if rounding causes errors of magnitude comparable to measurement

noise. Another variation is **modular measurement** where only, say, the sub-millisecond part of the transmit delay can be measured so that range is measured modulo 300 km [P1].

Navigation Satellite Systems

Each navigation satellite transmits a signal that, roughly speaking, provides a means to determine the accurate time of transmission and the position of the satellite at that time. Knowing the (biased) time of reception, the (biased) range between the satellite and receiver can be computed. The model for measurement noise depends on the particulars of the satellite system and receiver used.

GPS is currently the only globally available satellite positioning system. It was built for military purposes, but thanks to publicly available signal specification [38], anyone with a suitable receiver can use it for free. For a more comprehensive treatment on the GPS, see for example [41, 54, 58]. Other satellite navigation systems are the Russian Glonass, with a chronically incomplete constellation and slightly different signalling scheme (frequency division instead of code division [41]); the upcoming European Galileo that has a signal structure compatible with GPS; and the Chinese Compass Navigation Satellite System (also known as BeiDou-2), the first generation of which features two-way communication between satellites, ground beacons and the receiver. Second-generation BeiDou will have global coverage and a passive positioning mode similar to the other satellite navigation systems [17]. At the abstraction level of this work, all the satellite systems can be treated similarly.

Due to the low power of the satellite signals, the receiver needs a direct line of sight to the satellites, which is easy in the air, open seas and deserts but often impossible indoors, near tall buildings or under trees. Even if attenuated and reflected signals can be received with a high-sensitivity receiver, the measurement errors will be difficult to model. Additionally, to extract the satellite orbit parameters, the receiver has to decode several layers of data from the signal,

which is not always possible or practical when the signal is attenuated and noisy [76].

This gives rise to modelling the range measurements as modular measurement. In the case of GPS, the length of one spreading code cycle is approximately 300 km, and the relative phase of this code between different satellites is always measurable as long as the receiver has the satellites in track. Treating the code phase as a modular range allows including it as a measurement even though the unambiguous range to the satellite is not known [P1]. Similarly, the GPS carrier phase measurements can be thought of as a range measurement modulo about 19 cm, but making use of it is much more difficult because positions agreeing with the received measurement then occur every 19 centimetres.

In addition to the actual navigation satellites, several independent satellite- and ground-based augmentation systems transmit real-time corrections to GPS measurements that enable the removal of spatially correlated errors from the measurements [80]. The corrections given by these methods can be easily incorporated in the measurement model and need not be considered separately after this point. Additional tricks like dual-frequency or RTK processing will not be considered in this work, although there is active research on exploiting them also in personal positioning [2].

Terrestrial radio networks

The main difference between terrestrial radio beacons and navigation satellites is that terrestrial beacons are much closer to the receiver and thus nonlinearity and possible multimodal solutions are a greater problem. A second practical point is that these signals usually do not carry any information about the beacon position, therefore that information has to be separately acquired. From the algorithm point of view, it does not matter where the station coordinates come from but it may have practical consequences on time-to-first-fix, possible data transmission costs for the users, and of course building and maintaining the required beacon location database.

Possible sources for measurements include cellular phone networks, wireless LAN, RFID and Bluetooth. There are also dedicated radio navigation networks such as e-Loran or the Russian Alpha and Chayka [37], but the size of the antenna required for their ultra-low frequency signals may present practical difficulties in some applications. Possibly even (digital) TV [65] and FM radio [18] signals could be used for ranging or at least presence-of-signal of measurements.

Signal propagation models for cellular channel are already being used in network planning and can be exploited also in positioning [11, 84]. The problem is that these models depend heavily on the local environment, and it probably will not be feasible to store or transfer the detailed 3D models of the surroundings to the receiver and solve the propagation equations in real time.

Indoors and built-up urban areas are challenging positioning environments in terms of radio propagation characteristics and the amount and proximity of positioning beacons. Unlike the navigation satellite signals, radio communication signals are designed to penetrate into buildings and round corners. Thus, even indoors there might be a wealth of terrestrial radio signals available for positioning, like WLAN or other intra-building networks, but at the same time the non-line-of-sight propagation makes it difficult to build proper measurement models. Even if we successfully tune the error models separately for outdoors and indoors operation, the problem remains of deciding which one is more appropriate at any given time.

2.2 Inertial sensors

An inertial measurement unit (IMU) contains a bundle of calibrated acceleration and/or angular velocity sensors. Ideally, it measures the acceleration of the unit with respect to the sensor frame and the angular velocities of the frame. Thus, starting from an initial condition, the unit's position, velocity and attitude can be solved.

The problem is that the error in initial state and the measurement errors accumulate in the solution, and there might be very

little redundancy in the measurements. The position solution of a consumer-grade unit becomes useless in minutes, depending on the quality of the gyroscopes and accelerometers used. The challenges in using IMU as a stand-alone device thus lie in finding the initial state and estimating the slowly varying biases in the measurements, called calibration or in-motion alignment [19].

There is a wealth of published algorithms, and whole sessions in navigation conferences, dedicated to optimally combining the long-term unbiasedness of GPS and short-term relative accuracy of inertial sensors [26, 30]. In the Bayesian modelling scheme, we just augment the receiver state with the necessary bias and drift variables and include the appropriate measurement equations in the system.

2.3 Offline Information

While not measurements as such, there is some additional non-real-time information that can be considered as measurements or parameters to the measurement models.

The GPS signals carry information about the beacons' positions but terrestrial radio signals usually do not. One possibility to get this information is to use online look-up databases of beacon locations for wireless LAN network and cellular network [14, 63, 83]. Open on-line databases will probably become more popular as the penetration of location capabilities in mobile devices grows and data transfer costs come down. Algorithms for maintaining such databases and ensuring the accuracy and integrity of their data should present lots of interesting problems in large-scale parameter estimation and distributed data processing.

Additionally, there are several satellite-, radio network and internet-based services providing range corrections for GPS as well as acquisition aiding, almanac, ephemeris or precise ephemeris service, either locally or globally [25, 80, 82].

For pedestrian and especially vehicle navigation, street maps can be used to restrict the solution on streets, and aid in choosing a proper

dynamics model. There are free proprietary solutions [72] as well as open ones [57]. Street maps in suitable format are also essential for routing applications, which are outside the scope of this thesis.

3 Static positioning techniques

The static positioning problem can often be posed as a system of equations

$$y = h(x) + v \tag{2}$$

where the measurement vector $y \in \mathbb{R}^n$ and the measurement equation $h : \mathbb{R}^d \to \mathbb{R}^n$ are known, and the state $x \in \mathbb{R}^d$ and error vector $v \in \mathbb{R}^n$ are to be solved for. One of the solutions to this system is the actual true state $\bar{x}$, and the realised measurement errors $\bar{v}$, in addition to which there usually is an infinite number of other solutions. Note that for arbitrary x, choosing $v = y - h(x)$ constitutes a solution. Among these state-error combinations that satisfy the equation, we are usually interested in finding one with error vector as small as possible or "likely" in some other sense.

Consider first the error-free system $y = h(x)$. Some systems have a unique solution x^*, meaning that the equation is satisfied element-wise and x^* is the only argument with which this happens. Other systems either are not satisfied for any x, or have a multitude of solutions.

Unlike in the linear case, there are no general methods for solving a non-linear system of equations, and, as carefully reasoned in [64], "it is not hard to see why (very likely) there *never will be* any good, general methods." Moreover, there is no general systematic way of learning how many solutions a system may have other than solving the system. This is unfortunate in the positioning context, as we would certainly like to know whenever the set of measurements received implies more than one possible solutions, the simplest example being two range measurements in 2D forming two circles than can intersect in one or two points, not intersect at all or even coincide.

However, a non-linear system can often be linearised locally and subsequently characterised by the properties of the linearised system $y - h(x_0) - h'(x_0)(x - x_0) = 0$. It is then possible to try iterative methods that either converge to a root very efficiently, or fail spectacularly, indicating that there might not be a root nearby [64].

3.1 Closed-form and geometric solutions

One approach to solving the system (2) is to ignore the noise and seek a closed-form solution with respect to x. This usually works only for systems with as many equations as variables, so extra equations may have to be dropped from the system.

One of the most general formulations that have a closed-form solution is a system of polynomial equations with integer coefficients. Such systems can be solved, for example, by using resultant methods to eliminate all but one variable from the system, and then finding the solution corresponding to the roots of the remaining univariate polynomial [7]. Unfortunately, there does not seem to be efficient numerical implementations of resultant methods, but the algorithms require symbolic manipulation of polynomials with up to thousands of terms [42, 51]. Polynomial systems could also be solved by reducing them to a Gröbner basis, but this approach also requires symbolic machinery [7].

A more intuitive special case – at least if the problem is in two or three dimensions – is to give the measurements a geometric interpretation in the state space. A range measurement can be identified with a sphere centred at the beacon with radius equal to the measured value, a range difference measurement with a hyperboloid, altitude measurement with a horizontal plane, etc. Any points where all such measurement surfaces intersect are then considered position solutions as shown in Figure 1. This also gives a natural interpretation for such measurement scenarios that have multiple possible solutions.

For GPS-only situation, there are well-known closed-form least squares solutions [4, 50], as well as ones that require only three

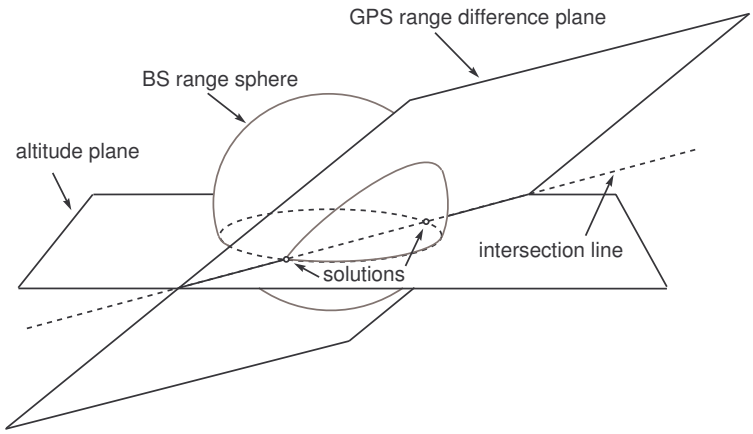


Figure 1: Geometric interpretation and multiple solution candidates

GPS measurements and an approximate altitude [62]. There are also some extensions for special cases of mixed GPS-terrestrial measurement combinations [3, 27].

The novel idea in [P2] is to combine the geometric interpretation of measurement equations with the concept of local positioning. The approach taken is to exploit the special structure of the problem, in this case the rotational symmetry of the surfaces. For example, if two quadrics of rotation have a focal point in common, their intersection can be proven to lie on a plane [50]. Then instead of solving the intersection of two quadrics, it is enough to solve the intersection of a quadric and the plane.

For further simplification, [P2] also introduces the idea of local positioning where a rough position of the receiver is already known. In the case of a cellular telephone, for example, the phone is known to lie inside the coverage area of the serving base station. This rough position information is exploited by linearising the measurements to beacons that are distant enough for the linearisation error to be negligible. A typical example of a distant beacon is a positioning satellite usually situated 20 000 to 26 000 kilometres away. Another example is the altitude that can be thought of as range measurement

to the centre of the Earth, which can be comfortably linearised in most settings.

The beacons that are not distant are called local beacons. They are relatively close to the receiver and the directions to the beacons can change drastically within the local region, and thus cannot be safely linearised.

The threshold distance dividing the beacons to local and distant ones depends on the accuracy of the rough position information and the level of linearisation error that we are ready to accept. Take, for example, position restricted inside a mobile cell with a radius of 3 km. If we allow a maximum of 10 metres of linearisation error in the measurements, then all beacons farther than 450 km away can be considered distant.

The 3D surfaces encountered in local range and range difference positioning are:

1. **plane** – defined by a range measurement to a distant beacon, a range difference between distant beacons, or an altitude measurement
2. **sphere** – defined by a range measurement to a local beacon
3. **paraboloid of revolution** – defined by a range difference between a local and a distant beacon
4. **branch of a hyperboloid of revolution** – defined by a range difference between two local beacons

All the quadrics involved can be written in the form [50]

$$Q: \quad \|s-x\| + n^T x = a \tag{3}$$

where s is a focal point of the quadric and n is called the directrix vector. In [P2], it is shown that if two quadrics of revolution share either a focal point or the directrix vector, their intersection curve is equal to the intersection of one of the quadrics with a plane. Usually, all but one of the quadrics involved can be replaced with planes with this method, and all the solutions of the resulting geometric system can be easily found in closed form.

A limitation of the method of [P2], as well as most closed-form methods, is that measurement errors are not modelled and cannot be accounted for. Moreover, the method only is able to use three measurement surfaces at a time to compute the 3D position. If there are more measurements available, the choices are either to ignore the ones with larger estimated error variances, or the ones that cannot be reduced to planes, or to compute many solution candidates from different choices of three measurements.

3.2 Residual minimisation

Usually we assume that some statistics of the error vector v in the system (2) are known. A standard technique for taking into account the measurement errors is to write the system as a minimisation problem

$$x^* = \arg\min_x \; g\left(y - h(x)\right), \tag{4}$$

where $g(v) \geq 0$ is called the *loss function* or the cost function. The loss function models our concept of "small" measurement errors, and it usually is zero only where v is zero. The quantity $v^* = y - h(x^*)$ is called the *residual* corresponding to the solution x^*.

Least squares

A common choice for the loss function, and one that is consilient with zero-mean Gaussian errors assumption, is $g(v) = v^T C^{-1} v$ where C is symmetric positive definite. Algorithms aiming to solve this problem are known as least squares methods [28]. The least squares solution is usually sought iteratively, although sometimes the structure of the system permits also a closed form solution.

We call the global and local minimisers of (4) *least squares solutions*. Note that any system, be it underdetermined or over-determined, has at least one least squares solution under mild conditions[1]. Additionally, each exact solution of the system coincides with a least

[1] Example of sufficient condition: h continuous and $\|x_k\| \to \infty \Rightarrow \|h(x_k)\| \to \infty$. [56]

squares solution. Unfortunately, there may also be additional local minima of the least squares cost function, that do not correspond to any approximate zeros of the original equation system. As an example, Figure 2 shows a portion of the least squares loss function for a system of modular GPS range measurements. There is a clear global minimum near the true position, but also numerous local minima all over the state space.

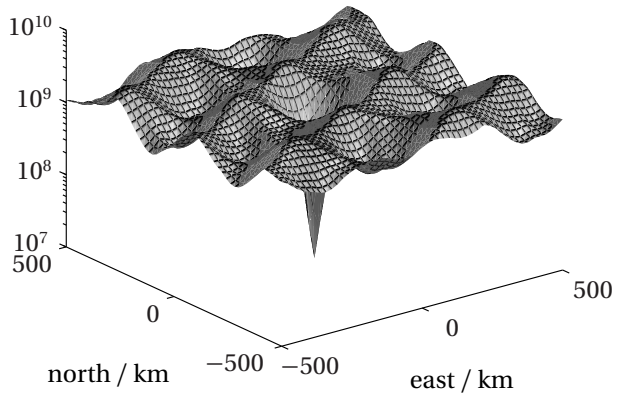


Figure 2: 2D portion of a five-dimensional least squares loss function for a system of modular GPS range equations [P1].

There is a multitude of optimisation methods for the least squares problem, the simplest and often the most useful one being the Gauss-Newton iteration [56] that starts from an initial guess x_0 and generates new iterates with

$$x \leftarrow x - \left(C^{-\frac{1}{2}} h'(x)\right) \setminus \left(C^{-\frac{1}{2}} \left(y - h(x)\right)\right) \tag{5}$$

until convergence. Here the backslash \ stands for the linear least squares solution of the related linear system and the derivative of h naturally needs to exist and be computable in all points of interest. It should be noted that such an iterative process only finds one (local) minimum, namely the one whose *attraction basin* the initial guess is in. In the case of Figure 2, iteration should be run from several starting points arranged such that at least one of them is guaranteed

to be in the attraction basin of the minimum corresponding to the true solution [P6].

The weight matrix C used in the loss function can be an arbitrary positive definite matrix. If the measurement error covariance matrix $V(v)$ is known, however, using it as the weight matrix results in a minimum variance estimate in case h is linear function, and asymptotically so even when h is nonlinear and fulfils mild regularity conditions [73].

Although state estimation is possible with very vague assumptions about the measurement noise process, error estimation requires something more specific. The Gauss-Newton iteration formula (5) can be used for *sensitivity analysis*, for if the iteration has reached a fixed point x^*, then the iteration step is, by definition, zero:

$$s(x^*, y) = \left(h'(x^*)^T C^{-1} h'(x^*)\right)^{-1} h'(x^*)^T C^{-1} \left(y - h(x^*)\right) = 0.$$

A first-order approximation of the sensitivity of the fixed point with respect to the measurements y is then

$$\frac{\partial}{\partial y} s(x^*, y) = \left(h'(x^*)^T C^{-1} h'(x^*)\right)^{-1} h'(x^*)^T C^{-1},$$

which will, if h is smooth enough near x^*, predict how the measurement noises affect the estimate x^*. Note however that this analysis only relates to the sensitivity of the estimate around the local least squares minimum defined by the realised measurements and does not necessarily say anything about the error of the estimate from the true state.

3.3 Maximum likelihood and Bayesian methods

When the measurement error distributions are known, the measurement model can be written as a conditional probability density $p(y \mid x)$, which is more flexible than the measurement equation $y = h(x) + v$. To write the density expression, it is only needed to know how measurements are distributed at each fixed x. As an example, if the measurement equation is already known, we get just

$$p(y \mid x) = p_v(y - h(x)),$$

where $p_v(\cdot)$ is the probability density of the measurement error v. The benefit of the density approach is that the errors do not have to be additive and their distribution can also depend on the state.

The function $p(y \mid x)$, when y is held constant and x as the variable, is called the *likelihood function*, and it generally does not integrate to 1 over x, thus it is not a proper probability density function. After a measurement y is realised, the *maximum likelihood* estimate is given by:

$$x^* = \arg\max_x p(y \mid x). \tag{6}$$

As an important special case, if the measurement errors are additive and normal distributed with zero mean and non-singular joint covariance Σ, the likelihood function is

$$p(y \mid x) = c \exp\left(-\frac{1}{2}\left(y - h(x)\right)^T \Sigma^{-1} \left(y - h(x)\right)\right),$$

where c is a constant coefficient. Now the likelihood is maximised exactly when the expression $\left(y - h(x)\right)^T \Sigma^{-1} \left(y - h(x)\right)$ is minimised. Thus, in this special setting, the maximum likelihood method coincides with the least squares method with Σ as the weight matrix.

Another appealing aspect of the likelihood approach is that further measurements can be incorporated optimally by simply multiplying the individual likelihoods, as long as the measurements are independent.

Bayesian estimate

Finally, for error analysis and computing confidence regions, we would preferably need the conditional distribution of the unknown state given the measurements, i.e. the *posterior density*. This is given by the Bayes' formula

$$p(x \mid y) = \frac{p(x)p(y \mid x)}{\int p(x)p(y \mid x)\,\mathrm{d}x},$$

where $p(y \mid x)$ is the likelihood function and $p(x)$ is the *prior density* that ideally should capture all the initial information about the state not present in the measurements. The prior density can conveniently include extra information in the computation, but it usually is difficult to justify any choice of $p(x)$ if "nothing" is known about the state or if the prior information is not already in probabilistic form. Additionally, the validity of the Bayesian notion of treating the true state as a random variable while it evidently is some fixed but unknown value can be debated [78, 85].

After the philosophical matters as well as the numerical challenges in computing the integral in the denominator [P3] are conquered, the posterior density can be used to compute, for example, expected value and the covariance of the state, maximum a posteriori (MAP) estimate or any confidence region required. Another important detail is that this formulation allows for recursive processing of subsequent measurements. Since the posterior represents the best knowledge about the distribution of the state, it can be substituted as the prior into the Bayes' formula when additional independent measurements arrive.

3.4 Database correlation

As a specific example of a positioning technique where the measurement model and solution methods are intertwined, we consider the database correlation methods (also known as location fingerprinting or pattern recognition methods) [43, 52, 84] for positioning with received signal strength measurements in a wireless network. The measurement model in these cases is complex and depends on the 3D environment along the signal path. It is often easier to form an empirical measurement model over the area of interest than to model the signal path in order to get an analytical measurement equation.

The suggested methods usually consist of a calibration phase where measurements are made in a great number of test locations in order to build a semi-empirical approximation of the measurement equation $h(x)$ and possible pre-processing steps to reduce the amount

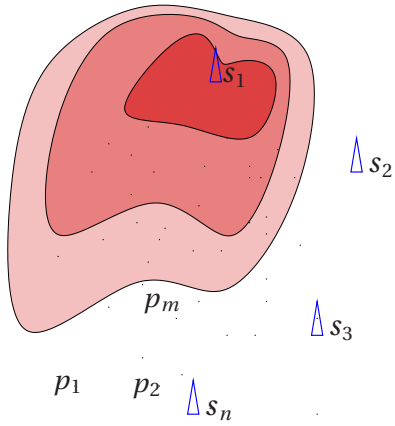


Figure 3: Database correlation: the mean reception power of each of the n beacons in m different calibration locations is measured and stored. Contours of the reception power from the first station shown.

of data stored in the database. In operation phase, when a measurement is made in an unknown location, some kind of table look-up or interpolation scheme is used to find the location (or set of locations) to best match the measurements.

In the Bayesian framework, the numerical approximation of the measurement equation can be included in the problem description as easily as any other measurement. The positioning problem then is relatively easy to formulate, but solving it will be computation intensive. Iterative least squares methods are pretty much out of the picture because the measurement equations will be multimodal and probably also discontinuous. Bayesian solution, on the other hand, requires large-scale numerical integration over the state space. The simplest method seems to be the maximum likelihood method that collapses to picking the test location with the most likely stored measurement as the estimate, but the accuracy of this approach is obviously limited by the density of test locations, and error or confidence estimates are hard to justify.

Creating or calibrating the signal strength map on-line is an interesting sub-problem as it in practice leads to an infinite-dimensional problem of estimating the measurement model $h(x)$ itself.

4 Time series estimation and navigation filters

In positioning and navigation context, time series estimation means exploiting the fact that only a small range of velocities and accelerations are possible when predicting the position for the next time instant. In the scope of this thesis, we will only study discrete-time case where measurements and state estimates are given only at certain time instants and not continuously in time.

When both the measurements and user motion are modelled as random processes, the discrete-time system is given by the two equations

$$y_k = h_k(x_k) + v_k \qquad \text{(measurement model)} \qquad (1b)$$

$$x_{k+1} = f_k(x_k) + w_k \qquad \text{(dynamics model)} \qquad (1c)$$

where $f_k : \mathbb{R}^d \to \mathbb{R}^d$ is a known state transfer function, $w_k \in \mathbb{R}^d$ is the state noise with known statistics, $h_k : \mathbb{R}^d \to \mathbb{R}^{m_k}$ is the known measurement function, and $v_k \in \mathbb{R}^{m_k}$ is the measurement noise with known statistics. The noises w_k and v_k are assumed independent temporally, of each other and of the known initial state x_0.

A nice feature of this formulation is that there is an optimal filtering algorithm, given as Algorithm 1. We use the shorthand notation $y_{1:k} = \{y_1, y_2, \ldots, y_k\}$ for the set of all measurements until time t_k, and denote the initial distribution before any measurements are received with $p(x_0) = p(x_0 \mid y_0)$.

The state transition density $p(x_k \mid x_{k-1})$ is defined by the dynamics model and the measurement likelihood $p(y_k \mid x_k)$ by the measurement model. Note that the formula for prior density could well be substituted into the formula for posterior density, resulting in a single equation, but the time step is usually split into two parts for clarity and because there often are some intermediate approximations between them.

The ideal Bayesian filter can be exactly computed only for some special cases where the number of parameters needed to describe the densities remains bounded over time [21]. Most notably, in the linear Gaussian situation the densities remain Gaussian, and

Algorithm 1: Bayesian recursive filter

1. Start with initial distribution $p(x_0 \mid y_0)$ at time t_0. Set $k = 1$.
2. Form a *prior estimate* of the state at t_k based on the state estimate at t_{k-1} and the dynamics model. The ideal *prior pdf* is

$$p(x_k \mid y_{1:k-1}) = \int p(x_k \mid x_{k-1}) p(x_{k-1} \mid y_{1:k-1}) \mathrm{d}x_{k-1}. \quad (7)$$

3. Update with measurements to get the *posterior estimate*. In ideal form, the prior pdf is multiplied by the *measurement likelihood* $p(y_k \mid x_k)$ and then normalised to get the *posterior pdf*

$$p(x_k \mid y_{1:k}) = \frac{p(x_k \mid y_{1:k-1}) p(y_k \mid x_k)}{\int p(x_k \mid y_{1:k-1}) p(y_k \mid x_k) \mathrm{d}x_k}. \quad (8)$$

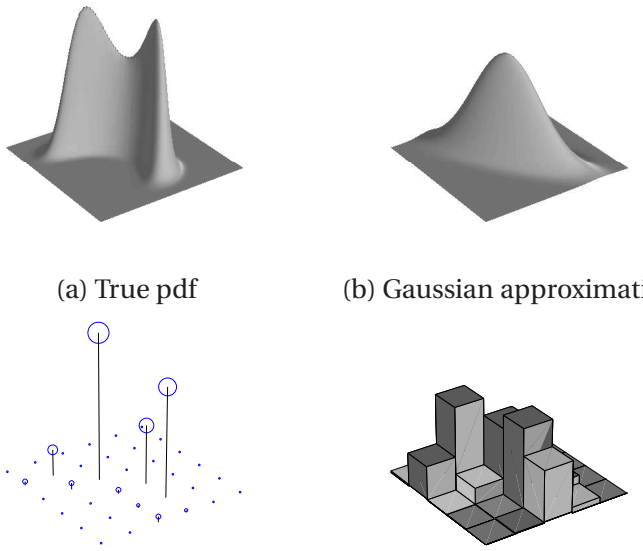
4. Increment k and repeat from 2.

the ideal filter simplifies into the Kalman filter [40]. In the general case, however, in order to be able to both evaluate the integrals and express the resulting density functions with a finite number of parameters, practical filtering algorithms resort to one or several of the following simplifications:

Approximate the system with simpler models, for example with linear ones so that the approximated problem can be solved exactly with the Kalman filter.

Approximate the pdf's with simpler ones at various stages, see Figure 4 for examples.

Approximate the integrals with some numerical method instead of computing them analytically.



(a) True pdf (b) Gaussian approximation

(c) Point-mass approximation (d) Piecewise approximation

Figure 4: A bimodal probability density in two dimensions (a) and examples of its approximations (b)-(d).

Any numerical filtering algorithm, regardless of how it is derived, can be viewed as an approximative Bayesian filter, and its properties often can be explained by the types of approximations used.

4.1 Kalman-type filters

While the Kalman filter [40] is equivalent to the ideal Bayesian filter for linear systems and Gaussian noises [16] (if the outputs of the filter are interpreted as the mean and covariance of a Gaussian posterior density), there is no unique generalisation of the idea to nonlinear systems. Nonlinear equations can be linearised in different ways so that Kalman-type equations can be applied, and the resulting algorithms can be interpreted either in the Bayesian framework as a solution of a linear approximation of the filtering problem, or deterministically as recursive least squares solvers.

Instead of starting with the linear Kalman filter and modifying it for nonlinear cases, we build on the weighted least squares estimation framework as suggested in [73] and arrive at the same algorithms with a little more generality and less "excess baggage".

Given the previous state estimate $\hat{x}_{k-1}$ and its squared error estimate matrix P_{k-1}, we can form the *a priori* estimate of current state and its approximate covariance, for example

$$\begin{aligned} x_k^- &= f_{k-1}(\hat{x}_{k-1}) \\ P_k^- &= f'_{k-1}(\hat{x}_{k-1}) P_{k-1} f'_{k-1}(\hat{x}_{k-1})^T + Q_{k-1}, \end{aligned}$$

where Q_{k-1} is the state noise covariance at t_{k-1}. Some variants like the Unscented Kalman Filter (UKF) use a numerical scheme to form better prior approximations in terms of the expected value and covariance of $f_{k-1}(x_{k-1})$.

The estimate of the current state is then found by minimising both the measurement residual and the distance from the prior estimate, using for example the least squares loss function:

$$\begin{aligned} \hat{x}_k &= \arg\min_{x_k} \left\| y_k - h_k(x_k) \right\|^2_{R_k} + \left\| x_k^- - x_k \right\|^2_{P_k^-} \\ &= \arg\min_{x_k} \left\| \begin{bmatrix} y_k \\ x_k^- \end{bmatrix} - \begin{bmatrix} h_k(x_k) \\ x_k \end{bmatrix} \right\|^2_{C_k}, \end{aligned}$$

where $C_k = \begin{bmatrix} R_k & 0 \\ 0 & P_k^- \end{bmatrix}$. If h_k is linearised about the prior estimate x_k^-, so that $h_k(x_k) \approx h_k(x_k^-) + H_k(x_k - x_k^-)$, the system becomes a linear least squares problem

$$\begin{aligned} &\min_{x_k} \left\| \begin{bmatrix} y_k - h_k(x_k^-) + H_k x_k^- \\ x_k^- \end{bmatrix} - \begin{bmatrix} H_k \\ I \end{bmatrix} x_k \right\|^2_{C_k} \\ \Leftrightarrow &\min_{x_k} \left\| C_k^{-\frac{1}{2}} \begin{bmatrix} y_k - h_k(x_k^-) + H_k x_k^- \\ x_k^- \end{bmatrix} - C_k^{-\frac{1}{2}} \begin{bmatrix} H_k \\ I \end{bmatrix} x_k \right\|^2 \\ \Leftrightarrow &\min_{x_k} \left\| b - A x_k \right\|^2, \end{aligned}$$

the least squares solution for which results in the familiar formulas for one step of the Extended Kalman Filter (EKF), also known as the Kalman-Schmidt filter [21]:

$$
\begin{aligned}
\hat{x}_k &= (A^T A)^{-1} A^T b \\
&= \left(H_k^T R_k^{-1} H_k + \left(P_k^- \right)^{-1} \right)^{-1} \left(H_k^T R_k^{-1} \left(y_k - h_k(x_k^-) + H_k x_k^- \right) - P_k^- x_k^- \right) \\
&= \ldots = x_k^- + P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1} \left(y_k - h_k(x_k^-) \right) \\
P_k &= (A^T A)^{-1} \\
&= \left(H_k^T R_k^{-1} H_k + (P_k^-)^{-1} \right)^{-1} \\
&= \ldots = \left(I - P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1} H_k \right) P_k^-
\end{aligned}
$$

The "extended Kalman filter" is not a single algorithm but a name for a collection of filter variants, obtained by for example different linearisations of h_k, iteration, different coordinate systems, higher order expansions or adding other ad-hoc tuning parameters [21]. There are also various ways to build so-called derivative-free filters [24] that estimate H_k numerically instead of analytically.

Furthermore, using loss functions other than the least squares one may result in filters better suited for non-Gaussian error distributions or more tolerant to outliers, called robust filters [60]. Further extensions to the linear filtering idea include mixture filters where the prior estimate can consist of several components or that use multiple linearisation points [15].

4.2 Sequential Monte Carlo filters

The Sequential Monte Carlo – or particle filter, or Bayesian bootstrap filter – approach follows from representing the prior and posterior distributions as a set of samples rather than a density function [23, 32, 36]. Then it is natural to approximate the integrals in the ideal filter with the Monte Carlo integration method. The sequential Monte Carlo filter, as given in Algorithm 2, propagates a set of weighted samples that are approximately distributed according to the posterior distribution.

Algorithm 2: Sequential Monte Carlo

Here we denote particles with superscript i: $x_{k|k}^{(i)}$ being the ith particle at the t_k, and $i = 1 \ldots N_k$.

1. Initialise samples from the initial distribution $x_0^{(i)} \sim p_{0|0}(x)$, and assign all weights to $w_0^{(i)} = \frac{1}{N}$. Set $k = 1$.

2. Form the prior estimate at t_k by sampling from the *proposal distribution* $q_{k|k-1}$. The proposal distribution can be almost arbitrary, a common choice being the state transition density, i.e. $x_{k|k-1}^{(i)} \sim p(x_k \mid x_{k-1}^{(i)})$.

3. Modify the weights of the samples with the measurement likelihood:

$$w_{k|k}^{(i)} \leftarrow \frac{w_{k|k-1}^{(i)} p(y_k \mid x_k^{(i)}) p(x_k \mid x_{k-1}^{(i)})}{q_k^{(i)}(x_k^{(i)})}.$$

   After normalisation, the weighted samples represent the posterior approximation.

4. Resample [66] if needed, increment k and continue from 2.

Particle filtering in its several forms has been successfully applied to a range of positioning problems, including aircraft terrain navigation [6], car navigation with map-aided dead reckoning [34], robot laser localisation [49], and simultaneous localisation and mapping [22, 55] to name just a few.

In principle, particle filtering is a very flexible method because it deals with distributions directly and any form of likelihood or state transition probability is easy to implement. Also, particle filter has proven (asymptotic) convergence properties [20], meaning that increasing the number of particles gives more accurate approximations of the optimal posterior distribution, limited only by the computation power in reserve. However, it is not easy to evaluate what is a sufficient number of particles in a given situation to guarantee reasonable accuracy.

In addition, particle filters run into remarkable difficulties whenever measurement noises are much smaller than state noises, as is the case in GPS positioning for instance [61]. In these cases, the volume of state space where the posterior distribution has non-zero density is a tiny fraction of the same for prior distribution, and thus very large number of particles (or some special proposal distribution) is needed to ensure that a representative number of particles survive with a non-zero weight. Another possibility is to treat the almost linear variables and measurements separately, leading to quasi-linear or Rao-Blackwellised particle filter [69]. Cases with singular or nearly singular likelihood distributions also require special treatment and detract from the flexibility of the method.

4.3 Numerical filters

Miscellaneous collection of filters can be arrived at with other choices of approximating density functions [16] or numerical integration methods [P3]. Once the density function family is chosen, the corresponding filter usually arises quite naturally. Additional shortcuts in computing the integrals are usually taken to reduce computation load.

Among first examples was the *point mass filter* [10] that approximates pdf's with point masses on a regular grid spanned over the region of state space where there is "significant portion" of probability. With this approximation, integrals reduce to sums and can be efficiently computed with discrete convolution.

For a long time, the computational requirements of the point-mass approach made it unpractical [10, 79]. Towards the end of the millennium, plentiful supply of computation power made this an interesting research subject, and adaptive [12] as well as anticipative [71] grid design methods have been explored.

Among the more exotic approaches to nonlinear filtering is the differential geometric filter [47] that develops a manifold structure for the space of (un-normalised) probability densities and reformulates the filtering problem as finding projections on the manifold.

One variant of the point mass approach is to approximate pdf's as piecewise constant functions instead of point masses [46]. Grid design is similar to the point-mass method, but the required integrals reduce into an efficiently computable form only in certain special cases [P4].

In [P7], the idea was further generalised to a class of nonlinear filters that approximate probability density functions with parametric functions defined piecewise over a finite of collection of *elements*. If G is a collection of non-overlapping bounded elements $\{G(i) \subset \mathbb{R}^d \mid i = 1, \ldots, n\}$ whose union adequately covers the "interesting region" of the state space, then any density $p(x)$ can be approximated as

$$\hat{p}(x) = \sum_{i=1}^{n} \chi_{G(i)}(x)\pi(i,x), \tag{9}$$

where $\chi_{G(i)}$ is the characteristic function of the element $G(i)$ and $\pi(i,\cdot) \in \mathscr{F}$ is a function from the chosen approximation family. The function family $\mathscr{F}$ could be, for example, the family of constant, linear, delta-peak, exponential or truncated Gaussian functions.

The piecewise approximation (9) is constructed by projecting the actual density function to the approximation space. Figure 5 illustrates a simple piecewise constant filter in one dimension with projection steps inserted in different points.

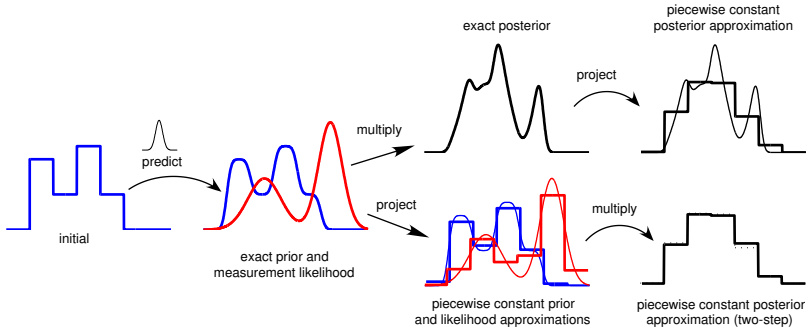


Figure 5: One-step (top) and two-step (bottom) piecewise filters.

The elements do not necessarily have to be of uniform shape or regularly arranged, but simpler choices such as a rectangular grid often result in simpler algorithms. To make computations feasible, we often use a coarse grid to track the approximate modes of the posterior and some more accurate method, such as denser local grid or an iterative local search, to pinpoint more accurate estimates.

Depending on which approximation function family and projection criteria are chosen and which steps involve intermediate projections, different approximate filters are obtained. The general formulation of a two-step piecewise variant, outlined in Algorithm 3, will lose considerable detail of the density functions in the projections, but in certain cases allows for an effective implementation as shown in [P7] for a few tractable combinations of approximation family and projection criteria.

Algorithm 3: General two-step piecewise filter

1. Start with an initial distribution $\hat{p}_{0|0}$ in the form (9). Set $k = 1$.
2. *(Prediction)* Form a new grid G_k to cover the interesting region of the state space. Approximate the prior density

$$\tilde{p}_{k|k-1}(x_k) = \sum_{i=1}^{n_{k-1}} \int_{G_{k-1}(i)} p(x_k \mid x_{k-1})\, \pi_{k-1|k-1}(i, x_{k-1})\, \mathrm{d}x_{k-1}$$

   in the form (9) with the piecewise functions defined as

$$\pi_{k|k-1}(i,\cdot) = \mathrm{proj}_{\mathscr{F}(G_k(i))}\left(\tilde{p}_{k|k-1}(\cdot)\right).$$

3. *(Update)* Compute the new posterior approximation $\pi_{k|k}(i,\cdot)$ as the product of prior and projected likelihood:

$$\pi_{k|k}(i,\cdot) \propto \pi_{k|k-1}(i,\cdot)\, \mathrm{proj}_{\mathscr{F}(G_k(i))}\left(p\left(y_k \mid \cdot\right)\right).$$

   Normalise, and compute the mean and covariance estimates.
4. Increase k and repeat from Step 2.

---

5 Benchmarking navigation filters

In this work, all numerical filters are seen as approximations to the ideal Bayesian filter (Algorithm 1). A natural criterion for such filters is the closeness of the computed posterior distribution to the ideal Bayesian posterior, expressed through some suitable pdf distance measure. In this section, we use the term filter to cover both the algorithm and its numerical implementation, so that we do not distinguish between errors caused by the theoretical algorithm, like linearisation, and the ones resulting from its implementation on a digital computer.

Figure 6 demonstrates why it is better to compare the posterior distributions instead of just the mean estimates. The ideal distribution in this case is bimodal, with one mode corresponding to the true state and the other caused by the measurement geometry. Note that without knowing the true track, there are no grounds for excluding either mode as the ideal posterior already contains all the information given by the measurements. Both filters A and B use a Gaussian approximation of the posterior; filter A gives about the correct mean and covariance while filter B is tracking one of the modes of the ideal posterior.

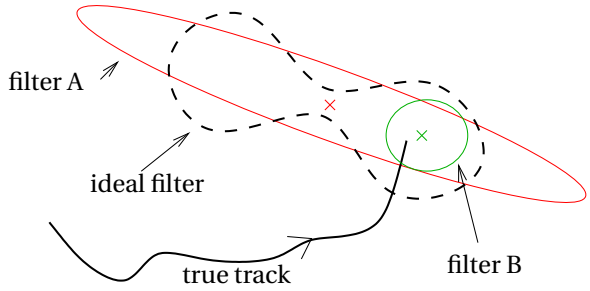


Figure 6: Hypothetical contour curves of posterior distributions given by an ideal filter and two approximate filters for a bimodal distribution.

Looking only at the error of the posterior mean, filter B seems to be doing better as its mean estimate is very close to the true state, in fact closer than that of the ideal posterior. Filter A, on the

other hand, captures the first two moments of the ideal posterior almost perfectly but still loses the information about its modes. A comparison criterion that considers the complete posterior distribution is thus needed to get a correct picture of the quality of the filters' current estimate. A third filter that could track both modes of the ideal posterior would then be rated superior to both A and B.

5.1 Comparison criteria

Commonly used pdf distance measures relate to the difference of the density or cumulative density functions of the distributions of interest [31, 39]. Other distance functions contain ratios (Kullback-Leibler, χ^2) or products (Bhattacharyya) of the densities or cumulative densities. In either case, the scalar distance between two pdf's is defined as either the supremum or an integral over a function defined over the whole state space, which presents numerical difficulties in accurate implementation. Note also that distance measures involving density functions are not directly applicable to point-mass or particle representations of pdf's.

An obvious problem with comparing to the ideal posterior distribution is that it is accurately known only in some special cases. Usually, the best we can do is compute an offline *reference solution* with, say, a particle filter using as many particles as our computational facilities can handle in reasonable time. This still leaves the practical problem of storing the (density estimates of the) reference solution, and implementing the comparison. If feasible, the test bank should also contain some cases where the optimal solution is analytically available, such as linear Gaussian cases, so that the quality of the reference solution itself can be monitored.

Another problem is that the posterior density approximation of the filter to be benchmarked is rarely explicitly available. Particle filters, for example, propagate a sample from the posterior distribution, and it is not a trivial task to derive the density function from it. The cumulative density function, though, is well defined and straightforward to compute. As an additional problem, most of the filters widely used in engineering do not even try to estimate the posterior

mean and covariance in the Bayesian sense, but might for example track the mode of the posterior and the squared error of the estimate. In these cases, we just have to interpret the results given by the filter as if it is approximating the posterior with a Gaussian distribution with the given mean and covariance.

Even if a filter propagates the mean and covariance of the posterior distribution correctly, we still do not know what the posterior is, only its first two moments. Figure 7 demonstrates density functions of some symmetric one-dimensional distributions having the same mean and variance. One can imagine more intricate examples if the symmetry is not required, or in multiple dimensions.

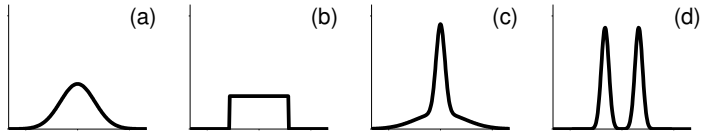


Figure 7: Examples of symmetric distributions with same mean and variance: a) normal, b) uniform, c) unimodal Gaussian mixture, d) bimodal Gaussian mixture.

To summarise, it could be said that the three problems in computing the distance of a posterior approximation to the ideal one are:

1. the (cumulative) density of the ideal posterior is intractable,
2. the (cumulative) density of the posterior approximation is intractable, and
3. even if we knew the (cumulative) densities of the two distributions, computing their distance would be intractable.

Instead of comparing the full distributions, the following figures of merit (or demerit) are easier to compute during simulation, and cover different aspects of the filter performance.

Error from true track – as well as its mean or RMS value or percentiles – can be evaluated in simulation tests where the *true track* is known. When working with real data, obtaining the ground truth for all state variables is a problem and art form in itself.

Note that the error consists of the error inherent to the data (caused by measurement errors and insufficient geometry) plus error caused by the approximations made by the filter. These two error types might even cancel each other out.

Consistency is a function of the error from true track and the filter's covariance estimate. The general inconsistency test [R6] gives a bound for the normalised error that is valid for any distribution with the given covariance:

$$\mathrm{P}\left(\left\| \hat{\Sigma}_k^{-\frac{1}{2}} (x_k - \hat{x}_k) \right\| \geq \sqrt{\frac{d}{\alpha}} \right) \leq \alpha,$$

where $\hat{x}_k$ and $\hat{\Sigma}_k$ are respectively the state and covariance estimates, d is the dimension of the state, and α is the risk level of the test. Even in most pathological cases, the fraction of errors exceeding this bound should not be larger than the risk level.

Error from reference is the distance of the estimate from a *reference solution*, an "optimal" mean estimate conditioned on the measurements. In most of the tested situations with six-dimensional state, particle filters with 1–5 million particles produced an acceptable reference solution. Note that only point estimates of the reference solution need to be stored, not the complete posterior density functions. If the reference is sufficiently close to the optimal solution, this criterion is equal to the error caused by the approximation process and is not contaminated with the problem-dependent noise like the error from true track.

Computation time is the relative time per fix of the implemented algorithm, and depends on the amount of skill and effort put in the programming, as well as to some extent on the properties of the memory management and cache structure of the platform used. The order of magnitude still gives some indication of the filter's performance.

5.2 Combining the criteria

As some filters fare better with respect to one criterion and some to the other, it is difficult to put the filters into an explicit order of quality to decide which one is the "best". In [P5], we call a filter *Pareto optimal* (with respect to a set of filters and a set of criteria) if any of the other filters that is better in some criteria does worse in some other criteria. Conversely, if a filter is not Pareto optimal, then some other filter is at least as good in all criteria. The concept of Pareto optimality might not help in deciding which filter works best, but it indicates which filters certainly do not.

Figure 8 depicts the trade-off between accuracy and computation time. The data comes from a semi-realistic MATLAB simulation testbench used in [P4, P5, R5, R6, R7, R8], where a number of different filters with varying parameters were run over a simulated dataset of 60 000 distinct time instants. The resulting square root of mean squared 2D errors (rms) is plotted against the relative computation time. Pareto optimal filters are marked with crosses. Piecewise continuous grid filter and point mass filter, although slow and with very rough accuracy, are nevertheless Pareto optimal because of their near-zero inconsistency. The figure also shows that with these simulation parameters, particle filters with varying sizes seem to give the best performance in terms of computation time versus the rms error.

6 Conclusions and future work

When building mathematical models of real-world problems, it is crucial to aim for the simplest and most general form that still enables one to draw conclusions about the model's real-world counterpart. Another important concept is to consider a question and its answer separately, for the question might be very simple even when the answer is prohibitively complex or intractable to compute.

This work began from a specific problem with a specific positioning system, but the focus expanded to nonlinear filtering as that seems

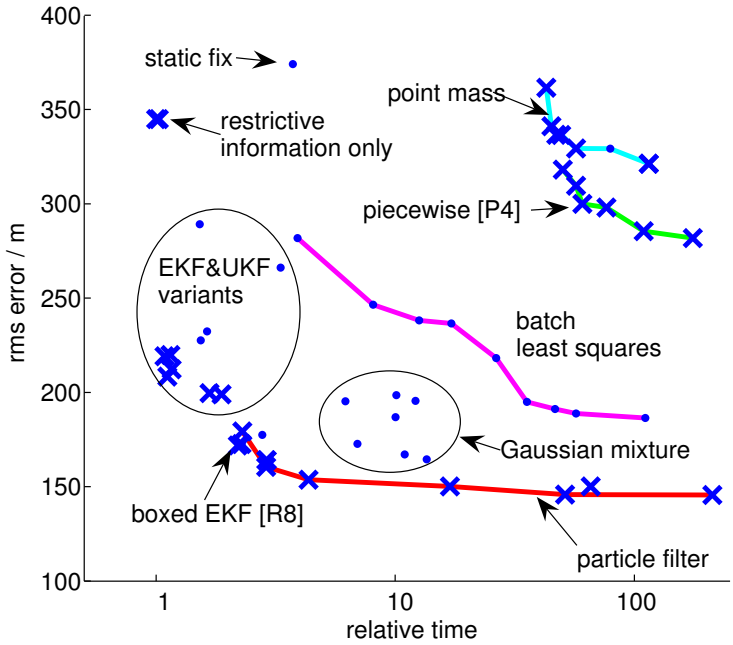


Figure 8: Mean RMS figures of different filters against computation time. Crosses denote "Pareto optimal" filters. [P5]

to contain most of the situations of interest as special cases, and has an elegant optimal yet intractable solution in the ideal Bayesian filter.

Theoretical and practical knowledge of the strong and weak points of different nonlinear filters enable the development of so-called hybrid filters that optimally combine different types or implementations of filters. For this end, a comprehensive simulation test bench was developed along with different measurement models and navigation filters.

Based on quite extensive testing, it seems that a filter that approximates posterior distributions correctly will exhibit smaller average rms error when run on enough test cases than some other filter with inferior posterior distribution approximation. It might also be that the computational work in comparing the whole distributions for a short test track is of the same order as computing the rms errors for

a large test set. Even then, checking the correctness of approximate posterior densities over a short test track gives valuable qualitative information of the behaviour of the filter that can only be guessed at by looking at the averages of a large number of runs.

While most of the treatment of nonlinear filters in this work is based on the Bayesian approach to estimation and filtering, this is not the only plausible interpretation. Similar algorithms can be developed from frequentist or deterministic viewpoint [78, 85], and with prudently chosen assumptions even algorithms equivalent to the Bayesian ones [73]. A possible weakness of the Bayesian viewpoint is that the densities of measurement and state noises are assumed known, which is not often the case with real-world data. Therefore, in addition to studying deterministic filters that require less assumptions about the noise statistics, the model mismatch properties of Bayesian filters, their robust extensions [13, 60] as well as outlier and error detection [48, 68] are an important area of future additional research.

One more thing to consider is the validity of the dynamics model. Brownian motion model is convenient for theoretical work, but pedestrians or vehicles very rarely move anything like that. Alternative dynamics models usually require additional state variables such as accelerations [44], or dismissing some of the convenient assumptions of the state model like the additivity of state noise or its independence of the state.

Some of the aspects not yet touched in this work are the Cramér-Rao bounds for sequential estimation [81], [70] and for point-mass and particle filters [6]. Firstly, the computation of the bound seems to require the accurate posterior density that is unknown in the first place. Secondly, it is not entirely clear how the Cramér-Rao approach relates to the Bayesian framework, and it might be that the idea of best possible estimation accuracy in this context is already captured by the optimal Bayesian posterior that we approximate with our reference solution. The potentially more feasible Barankin bound [5] has been applied to static positioning [45] but has not been generalised to the filtering problem yet.

For maximal use of prior information, digital street maps can be integrated into position filtering. Depending on what can be asserted about the receiver's mode of transport (e.g. on bicycle or in car), street maps may be used to considerably constrain the set of possible positions. A greater challenge would be to produce and exploit similar maps for indoor and underground environments. However, it can be assumed that once the data representation and storage problems are overcome, such information will be relatively easy to incorporate into filtering.

Another future challenge is cooperative positioning [59] where a number of mobile devices share measurements or even join forces to solve for unknown beacon positions and/or error models. This problem fits nicely into the Bayesian filtering framework, but poses major challenges in practical implementation of the algorithms as well as the implied privacy considerations.

References

[1] N. Agarwal, J. Basch, P. Beckmann, P. Bharti, S. Bloebaum, S. Casadei, A. Chou, P. Enge, W. Fong, N. Hathi, W. Mann, A. Sahai, J. Stone, J. Tsitsiklis, and B. V. Roy. Algorithms for GPS operation indoors and downtown. *GPS World Magazine*, 2002.

[2] K. Alanen, L. Wirola, J. Käppi, and J. Syrjärinne. Mobile RTK: Using low-cost GPS and internet-enabled wireless phones. *Inside GNSS*, 1:32–39, May/June 2006.

[3] M. R. Anderson, T. M. Tran, R. McCord, B. Yang, and A. Babich. Direct position solutions from mixed ranging measurements. In *Proceedings of ION GNSS 2005*, pages 1417–1424, Long Beach, 13–16 September, 2005.

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

[5] E. W. Barankin. Locally best unbiased estimates. *The Annals of Mathematical Statistics*, 20(4):477–501, 1949.

[6] N. Bergman. Posterior Cramér-Rao bounds for sequential estimation. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.

[7] V. D. Borisevich, V. G. Potemkin, S. P. Strunkov, and H. G. Wood. Global methods for solving systems of nonlinear alge-

braic equations. *Computers and Mathematics with Applications*, 40:1015–1025, 2000.

[8] J. Borkowski, J. Niemelä, and J. Lempiäinen. Performance of Cell ID+RTT hybrid positioning method for UMTS radio networks. In *Proceedings of the Fifth European Wireless Conference, Feb 24–27, 2004, Barcelona*, 2004.

[9] R. Bryant. Assisted GPS using cellular telephone networks for GPS Anywhere. *GPS World Magazine*, pages 40–45, May 2005.

[10] R. S. Bucy and K. D. Senne. Digital synthesis of non-linear filters. *Automatica*, 7(3):287–298, 1971.

[11] H. Bühler, H. Bonek, and B. Nemsic. Path tracing: estimation of time dispersion for mobile radio channels. Technical Report COST 231 TD(94) 034, European Cooperation in the Field of Scientific and Technical Research, Lisboa, 1994.

[12] Z. Cai, F. L. Gland, and H. Zhang. An adaptive local grid refinement method for nonlinear filtering. Research raport 2679, INRIA Rennes, 1995.

[13] A. Carosio, A. Cina, and M. Piras. The robust statistics method applied to the Kalman filter: theory and applications. In *Proceedings of ION GNSS 2005*, pages 525–535, Long Beach, 13–16 September, 2005.

[14] CellSpotting.com. A global location based information service. Accessed 3 Apr 2007. URL `http://www.cellspotting.com/`.

[15] R. Chen and J. S. Liu. Mixture Kalman filters. *J. Roy. Statistic Soc. B*, 62:493–508, 2000.

[16] Z. Chen. Bayesian filtering: from Kalman filters to particle filters and beyond. Technical report, Adaptive Systems Laboratory, McMasters University, 2003.

[17] Chinese Defence Today. Compass navigation satellite system (BeiDou-2), 14 April 2007. URL `http://www.sinodefence.com/strategic/spacecraft/beidou2.asp`.

[18] J. Cisneros, D. Kelley, and L. A. Greenbaum. An urban positioning approach applying differential methods to commercial FM radio emissions for ground mobile use. In *Proceedings of 50th ION Annual Meeting, June 6–8, 1994, Colorado Springs*, 1994.

[19] J. Collin. *Investigations of Self-Contained Sensors for Personal Navigation*. Dissertation, Tampere University of Technology, 2006.

[20] D. Crisan. Particle filters a theoretical perspective. In A. Doucet, N. de Freitas, and N. Gordon, editors, *Sequential Monte Carlo Methods in Practice*, pages 17–41. Springer-Verlag, 2001.

[21] F. Daum. Nonlinear filters: beyond the Kalman filter. *IEEE Aerospace & Electronical Systems Magazine*, 20(8):57–69, 2005.

[22] A. Doucet, N. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised filtering for dynamic Bayesian networks. In *Proceedins of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 176–183, 2000.

[23] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

[24] J. Duník, M. Šimandl, O. Straka, and L. Král. Performance analysis of derivative-free filters. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 1941–1946, Seville, Spain, December 12–15 2005.

[25] P. Enge, T. Walter, S. Pullen, C. Kee, Y.-C. Chao, and Y.-J. Tsai. Wide area augmentation of the Global Positioning System. *Proceedings of the IEEE*, 84(8):1063–1088, 1996.

[26] J. A. Farrell and M. Barth. *The Global Positioning System & Inertial Navigation*. McGraw-Hill, 1999.

[27] I. Fernandez-Corbaton, A. Vayanos, P. Agashe, and S. Soliman. Method and apparatus for determining an algebraic solution to GPS terrestrial hybrid location system equations. US Patent 6,289,280 B1, Sep. 11, 2001.

[28] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, Chichester, second edition, 1987.

[29] D. Fox, J. Hightower, H. Kauz, L. Liao, and D. J. Patterson. Bayesian techniques for location estimation. In *Proceedings of Workshop on Location-aware Computing, part of UBICOMP Conference, Seattle, October 2003*, pages 16–18, 2003.

[30] S. Fujioka, M. Tanikawara, M. Nishiyama, Y. Kubo, and S. Sugimoto. Comparison of nonlinear filtering methods for INS/GPS in-motion alignment. In *Proceedings of ION GNSS 2005*, pages 467–477, 2005.

[31] A. L. Gibbs and F. E. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3):419–435, 2002.

[32] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F (Radar and Signal processing)*, 140(2):107–113, 1993.

[33] F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks. *IEEE Signal Processing Magazine*, July:41–53, 2005.

[34] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation and tracking. *IEEE Transactions on Signal Processing*, 50(2), 425–437 2002.

[35] G. Hein, B. Eissfeller, V. Oehler, and J. Winkel. Synergies between satellite navigation and location services of terrestrial mobile communication. In *Proceedings of the ION GPS 2000*, pages 535–544, 2000.

[36] K. Heine. Unified framework for sampling/importance resampling algorithms. In *Proceedings of Fusion 2005, July 25-29, 2005, Philadelphia*, 2005.

[37] B. Hofmann-Wellenhof, K. Legat, and M. Wieser. *Navigation: principles of positioning and guidance*. Springer-Verlag, New York, 2003.

[38] ICD. NAVSTAR GPS space segment/navigation user interfaces. interface control document (ICD), ARINC Research Corporation, Fountain Valley, California, 2000. Rev. C (Public Release Version).

[39] D. H. Johnson and S. Sinanovic. Symmetrizing the Kullback-Leibler distance, 2001. URL `http://cmc.rice.edu/docs/docs/Joh2001Mar1Symmetrizi.pdf`. Submitted to IEEE Transactions on Information Theory.

[40] R. E. Kalman. A new approach to linear filtering and prediction. *Transactions of ASME, Journal of Basic Engineering*, 82:35–45, 1960.

[41] E. D. Kaplan, editor. *Understanding GPS: principles and applications*. Artech House, Norwood, 1996.

[42] D. Kapur, T. Saxena, and L. Yang. Algebraic and geometric reasoning using Dixon resultants. In *Proceedings of ACM International Symposium on Symbolic and Algebraic Computation '94*. ACM Press, 1994.

[43] M. Khalaf-Allah and K. Kyamakya. Database correlation using Bayes filter for mobile terminal localization in GSM suburban environments. In *Proceedings of 3rd Workshop on Positioning, Navigation and Communication 2006 (WPNC'06)*, Hannover, 16 March, 2006.

[44] S. Kitao, Y. Kubo, Y. Muto, and S. Sugimoto. Dynamical models with constraint for precise RTK positioning. In *Proceedings of ION GNSS 2005*, pages 1555–1563, Long Beach, 13–16 September, 2005.

[45] H. Koorapaty. Barankin bounds for position estimation using received signal strength measurements. In *Proceedings of the 59th IEEE Vehicular Technology Conference*, pages 2686–2690, 17–19 May 2004.

[46] S. C. Kramer and H. W. Sorenson. Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, 24(6):789–801, 1988.

[47] R. Kulhavý. Recursive nonlinear estimation: geometry of a space of posterior densities. *Automatica*, 28(2):313–323, 1992.

[48] H. Kuusniemi. *User-level reliability and quality monitoring in satellite-based personal navigation*. Dissertation, Tampere University of Technology, 2005.

[49] C. Kwok, D. Fox, and M. Meilă. Real-time particle filters. *Proceedings of the IEEE*, 92(2), 2004.

[50] J. L. Leva. An alternative closed-form solution to the GPS pseudo-range equations. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1430–1439, 1996.

[51] D. Manocha. Efficient algorithms for multipolynomial resultant. *The Computer Journal*, 36(5):485–496, 1993.

[52] M. McGuire, K. N. Plataniotis, and A. N. Venetsanopoulos. Estimating position of mobile terminals from path loss measurements with survey data. *Wireless Communications and Mobile Computing*, 3:51–62, 2003.

[53] Merriam-Webster. Merriam-Webster's Online Dictionary. Accessed 8 May 2007. URL `http://www.m-w.com/`.

[54] P. Misra and P. Enge. *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, 2nd edition, 2001.

[55] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Proceedings of the 2003 IEEE International Conference on Robotics & Automation, Taipei, Taiwan, Sept 14–19, 2003*, pages 1985–1991, 2003.

[56] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables*. Academic Press, Inc., Orlando, 1970.

[57] OSM. Openstreetmap project. Accessed 3 Apr 2007. URL `http://www.openstreetmap.org/`.

[58] B. Parkinson and J. Spilker, editors. *Global positioning system: theory and applications Volume I*. Charles Stark Draper Laboratory, Inc., Cambridge, 1996.

[59] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal. Locating the nodes. *IEEE Signal Processing Magazine*, July:54–69, 2005.

[60] T. Perälä and R. Piché. Robust Extended Kalman filtering in hybrid positioning applications. In *Proceedings of the 4th Workshop on Positioning, Navigation and Communication (WPNC'07), March 22*, pages 55–64, 2007.

[61] D. Petrovich and R. Piché. A comparison of particle filters for personal positioning. In *VI Hotine-Marussi Symposium of Theoretical and Computational Geodesy*, Wuhan, China, May 29 – June 2, 2006.

[62] M. Phatak, M. Chansarkar, and S. Kohli. Position fix from three GPS satellites and altitude: a direct method. *IEEE Transactions on Aerospace and Electronic Systems*, 36(1):350–354, 1999.

[63] Place Lab. Place lab – a privacy-observant location system. Accessed 7 May 2007. URL `http://placelab.org`.

[64] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical recipes: the art of scientific computing*. Cambridge University Press, Cambridge, 1987.

[65] M. Rabinowitz and J. J. Spilker, Jr. A new positioning system using television synchronization signals. *IEEE Transactions on Broadcasting*, 51(1):51–61, 2005.

[66] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter, Particle Filters for Tracking Applications*. Artech House, Boston, London, 2004.

[67] T. Roos, P. Myllymäki, and H. Tirri. A statistical modeling approach to location estimation. *IEEE Transactions on Mobile Computing*, 1(1):59–69, 2002.

[68] H. Sairo. *Error Detection in Personal Satellite Navigation*. Dissertation, Tampere University of Technology, 2006.

[69] T. B. Schön. *Estimation of Nonlinear Dynamic Systems: Theory and Applications*. Dissertation, Linköpings universitet, 2006.

[70] M. Šimandl, J. Královec, and P. Tichavský. Filtering, predictive, and smoothing Cramér-Rao bounds for discrete-time nonlinear dynamic systems. *Automatica*, 37:1703–1716, 2001.

[71] M. Šimandl, J. Královec, and T. Söderström. Anticipative grid design in point-mass approach to nonlinear state estimation. *IEEE Transactions on Automatic Control*, 47(4):699–702, April 2002.

[72] smart2go. Smart2go – share the world with your friends. Accessed 22 May 2007. URL `http://www.smart2go.com/`.

[73] P. Swerling. Modern state estimation methods from the viewpoint of the method of least squares. *IEEE Transactions on Automatic Control*, 16(6):707–719, December 1971.

[74] J. Syrjärinne. Possibilities for GPS time recovery with GSM network assistance. In *Proceedings of the ION GPS 2000*, pages 955–966, 2000.

[75] J. Syrjärinne. Wireless-assisted GPS, keeping time with mobiles. *GPS World Magazine*, pages 22–31, January 2001.

[76] J. Syrjärinne. *Studies of modern techniques for personal positioning*. Dissertation, Tampere University of Technology, Tampere, 2001.

[77] J. Syrjärinne and L. Wirola. Setting a new standard. Assisting GNSS receivers that use wireless networks. *Inside GNSS*, pages 26–31, October 2006.

[78] A. Tarantola. Popper, Bayes and the inverse problem. *Nature Physics*, 2:492–494, August 2006.

[79] P. Terwiesch and M. Agarwal. A discretized nonlinear state estimator for batch processes. *Computers & Chemical Engineering*, 19(2):155–169, 1995.

[80] C. Tiberius. Navigation – the accuracy game. In *Proceedings of the European Navigation Conference GNSS 2003, Apr 22-25, 2003, Graz*, 2003.

[81] P. Tichavský, C. H. Muravchik, and A. Nehorai. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing*, 46(5):1386–1396, 1998.

[82] F. Toran-Marti, J. Ventura-Traveset, E. Gonzalez, M. Toledo, A. Catalina, C. Barredo, and A. Salonico. Positioning via Internet: SISNeT catches GPS in urban canyons. *GPS World*, 15(4): 28–35, 2004.

[83] WiGLE. WiGLE wireless geographic logging engine. Accessed 3 Apr 2007. URL `http://www.wigle.net/`.

[84] G. Wölfle, R. Hoppe, D. Zimmermann, and F. M. Landstorfer. Enhanced localization technique within urban and indoor environments based on accurate and fast propagation models. In *Proceedings of European Wireless 2002*, Firenze, Feb 2002.

[85] P. Xu. Nonlinear filtering of continuous systems: foundational problems and new results. *Journal of Geodesy*, 77:247–256, 2003.

PUBLICATION 1

Niilo Sirola and Jari Syrjärinne: GPS position can be computed without navigation data. In *Proceedings of the 15th International Technical Meeting of the Satellite Division of the Institute of Navigation ION GPS 2002*, Portland, Oregon, Sept. 24–27, 2002, pages 2741–2744.

GPS Position Can Be Computed without the Navigation Data

Niilo Sirola, *Tampere University of Technology, Tampere, Finland*

Jari Syrjärinne, *Research and Technology Access, Nokia Mobile Phones, Finland*

BIOGRAPHY

Niilo Sirola is a M.Sc. student at Tampere University of Technology majoring in technical mathematics and software design. He has been researching assisted GPS positioning algorithms since 2000.

Jari Syrjärinne received his M.Sc. degree in 1996 and Doctor of Technology degree in 2001 from Tampere University of Technology, Finland. Since 1999, he has been working for Nokia Mobile Phones where he also finished his doctoral thesis about modern techniques for personal positioning. His current research topics include cellular positioning, sensor fusion, and AGPS.

ABSTRACT

This paper presents an algorithm for computing a GPS receiver's position and the current time from C/A-code phase measurements to at least six satellites and a set of valid satellite ephemerides. Reference position and reference time are not necessary. The process uses a cost function having numerous local minima in addition to the global minimum. An exhaustive search over the 5-dimensional time-position-bias search space will reliably find the global minimizer, and thus solve the approximate position and time. Introducing a reference time within one minute from the true time resulted in worst-case computation time of a few seconds. If the current pseudo-range correction parameters are available, the accuracy of this method is comparable to that of the conventional pseudo-range methods. As a conclusion, an accurate GPS position can be solved in a reasonable time even without the navigation data or precise system time.

INTRODUCTION

The method presented in this paper is intended for GPS positioning in weak signal situations where the satellite signals are so noisy that C/A codes can be tracked but the navigation messages are beyond recognition [1,2,3]. The ephemeris and pseudo-range correction parameters could be already in the receiver's memory, or they could be acquired via e.g. cellular network along with a delayed reference time [1]. A local positioning algorithm for this situation that requires initial time and position approximations has already been presented in [4]. This paper describes a global extension to the local algorithm.

BACKGROUND: THE LOCAL ALGORITHM

Use the following symbols:

t, $\mathbf{r}$, b	system time, 3D user position and user clock bias to be solved,
ϕ_i	C/A code phase measurement to ith satellite,
ε_i	approximate ranging error compensation,
$\mathbf{s}_i(t)$	position of ith satellite at system time instant t, and
τ_i	approximate time-of-travel from ith satellite to user.

The range fit to the ith satellite is defined by

$$q_i(t,\mathbf{r},b) = \mathrm{frac}_\Lambda\left(\phi_i + \varepsilon_i + b - \left\|\mathbf{s}_i(t-\tau_i) - \mathbf{r}\right\|\right) \tag{1}$$

where $\Lambda \approx 300$ km is the length of the C/A code sequence and the lambda-fraction operator is defined by

$$\mathrm{frac}_\Lambda x \triangleq x - \Lambda\,\mathrm{round}\frac{x}{\Lambda}. \tag{2}$$

Note that the user clock bias b can be restricted between $-\frac{1}{2}\Lambda$ and $\frac{1}{2}\Lambda$ since the fraction operator eliminates integer multiples of Λ.

Denoting $\mathbf{x} = \begin{bmatrix} t \\ \mathbf{r} \\ b \end{bmatrix}$ and $\mathbf{q}(\mathbf{x}) = \begin{bmatrix} q_1(\mathbf{x}) \\ \vdots \\ q_n(\mathbf{x}) \end{bmatrix}$, we can write the sum of squares cost function

$$f(\mathbf{x}) = \tfrac{1}{2}\left\|\mathbf{q}(\mathbf{x})\right\|^2, \tag{3}$$

which has a minimum at the true position $(T, \mathbf{R}, B)$. When $n > 5$ and the satellite geometry is non-pathological,

i.e. both satellite positions and velocities are non-coplanar, this minimum is the unique global minimum.

Figure 1 shows an example of the cost function projected into two dimensions. There are various local minima in addition to the global minimum at the true position. If any gradient-based minimization method is initiated far enough from the global minimum, the search will stray into one of the local minima and not be able to find the true position.

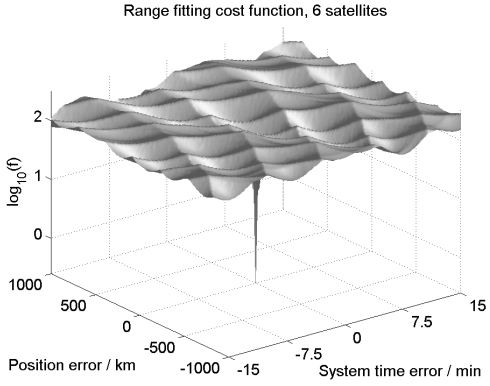


Figure 1: Two-dimensional slice of the cost function

An initial point $(t_0, \mathbf{r}_0, b_0)$ lies "close enough" to the global minimum $(T, \mathbf{R}, B)$ at least when

(4) $$\kappa|T-t_0| + \|\mathbf{R}-\mathbf{r}_0\| + |B-b_0| < \tfrac{1}{2}\Lambda ,$$

where $\kappa \approx 710$ m/s (maximum Doppler velocity) [4]. If this condition is satisfied, then the true position can always be reached with just a couple of Gauss-Newton steps given by

(5) $$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[\nabla\mathbf{q}(\mathbf{x}_k)^T \nabla\mathbf{q}(\mathbf{x}_k)\right]^{-1} \nabla\mathbf{q}(\mathbf{x}_k)^T \mathbf{q}(\mathbf{x}_k)$$

.

GLOBAL ALGORITHM

The local algorithm finds the correct solution given an initial point inside the "attraction basin" of the true solution. Next, we construct a global extension of the local method. The purpose of the global phase is to launch the local search with different starting points, and identify whether the resulting point is the global minimum or not.

Fortunately, it is easy to detect when the search is going to fail, because the cost function is smooth and convex near the true solution. Most often, when started outside the attraction basin, the Gauss-Newton search takes so large a step that it would take the solution outside the region defined by (4). This is a clear indication of failed search. Also, the cost function value at the true position is considerably smaller than in any of the local minima.

The search space

The search space is five-dimensional, consisting of the system time a.k.a. GPS time, 3D position, and user clock bias. Under the assumption that the satellite ephemeris is valid, the system time is bound within two or three hours from the time-of-ephemeris, depending on whether the ephemeris is fitted for four or six hours. The user is assumed to be on the Earth surface or within couple of kilometers above or below it, and the clock bias range equivalent is by definition inside $\pm\frac{1}{2}\Lambda$.

The brute-force approach is to span a grid over the search space such that at least one grid point necessarily satisfies the condition (4). If the system time is divided into intervals of two minutes, it suffices to divide the surface of the Earth into squares 100km across and the bias into intervals of 100km. This results in over 15 million distinct points to inspect. We can start the local search from every point and pick the resulting minimum with the smallest cost function value as the solution.

The computational load of the brute-force approach is roughly equal to 15 million point position fixes. This is hardly practical. There are two ways to accelerate the search: reduce the number of grid points, and speed up the local search.

Satellite visibility

Instead of searching the whole Earth, we can restrict the user position to the area where all the measured satellites are above the horizon. Figure 2 demonstrates the visibility region of three satellites. The point **r** on the Earth surface belongs to the visibility area roughly when

(6) $$\mathbf{s}_i^T \mathbf{r} \geq r_{\text{Earth}}^2 \quad \forall i .$$

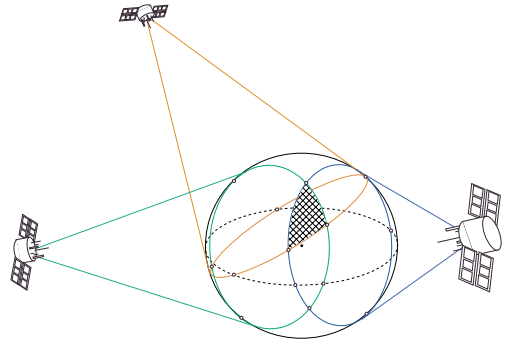


Figure 2: Satellite visibility region

In case of eight satellites, the visibility region has an average area of 30 million square kilometers, which is

about 5% of the total Earth surface area. The more satellites are available, the smaller the visibility area, and thus less grid points to search. This suggests that all the available satellites should be used in this visibility search phase, even if some of those have such a low signal-to-noise ratio that they should not be used in the actual position solution.

In practice, it is most efficient to divide the four-hour system time span into intervals of about 2 minutes and search the whole visibility region of one interval before moving to the next one.

Gradual local search

Secondly, we try to reduce the time spent inspecting one point. Instead of executing the full local search at every point, we try some heuristic methods to spot and eliminate the "bad" points as fast as possible, leaving us hopefully with just one point that satisfies all the conditions and is the true solution.

First, given time and position, we check if there is a bias value such that all range fits q_i have absolute value smaller than some pre-defined constant q_{thres}. The denser the search grid is, the lower value can be set for q_{thres}. However, if q_{thres} is too small the search might miss the true position.

If a good initial bias was found, we launch a rough local search from the obtained point. This phase uses roughly approximated cost function and takes just a few iteration steps to check whether the search converges or not. The search is terminated if the iteration step is too large and would take the solution outside of the assumed attraction basin or too far from the Earth surface.

Rough search initiated inside the attraction basin of the true solution will produce a position estimate with an accuracy of a few kilometers. However, a few additional local minima are usually found.

In the final phase, the full-precision local search is launched starting from the coarse minimum. Some of the searches fail to converge, but most of them produce a local minimizer of the cost function. Fortunately, the cost function value at the global minimum is considerably smaller than in any of the local minima. If f is below a threshold value calculated from the chi-square distribution, we have found the true position, and stop the search.

Reference position or time

It is also possible to use any kind of additional position or time information to further restrict the search space, e.g. cell coverage if cellular base station coordinated are available. Usual cell ranges are within few tens of kilometers in rural area and few kilometers in more urban areas.

Wireless assistance can also include a coarse position estimate, which naturally can also be used. In this case, the position search space reduces to just one point and only the correct time has to be found. Conversely, roughly known system time restricts the system time search to just one two-minute interval.

TESTING

The proposed methods were implemented in Matlab for testing and simulations. Actual ephemeris from the morning of May 16th 2001 was used, but all the measurements were simulated.

Figure 3 depicts a user in Tampere, Finland using satellites 1, 6, 10, 17, 24, and 30 on May 16th 2001 at 6:12 UTC. The figure shows the edge of the visibility region of the satellites at the time and the points for which the initial bias was found as red circles. The rough local search from these points produced six solution candidates, marked with blue crosses. Only one point, the actual user position, satisfied the cost function threshold after the refined local search, and it is marked in the figure as a blue asterisk.

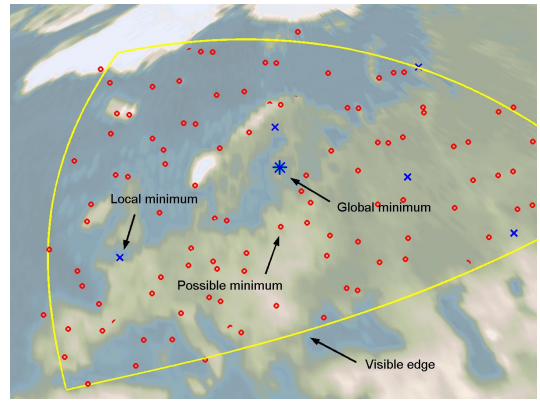


Figure 3: Visibility region, position candidates and true position (Background: Day Earth Texture Map ©2002 The Living Earth, Inc.)

We also studied the time Matlab spent computing a single position fix. While this does not predict the actual DSP load, it will give some overall insight of the complexity of the problem. Figure 4 shows statistics about the time-per-fix as a function of satellites used. The more satellites are available, the smaller the satellite visibility region gets and thus the number of possible solutions to check decreases. The position search is terminated as soon as a satisfactory solution is found, which could with good luck happen with very first points checked, or with bad luck the last one. Thus, the total computation times vary from a fraction of a second up to minutes. Overall, it seems that at least eight satellites are required to fix the position in reasonable time. The maximum observed positioning time for seven satellites was 2.6 minutes, and for six satellites over 7 minutes.

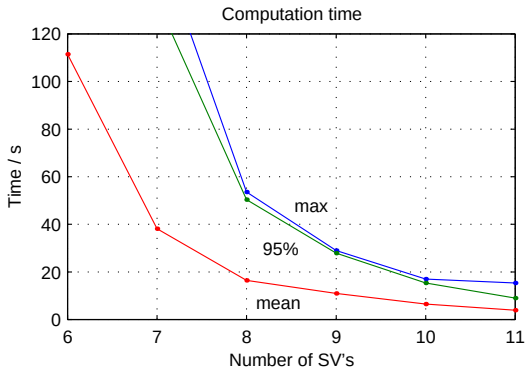


Figure 4: Computation time using just satellite visibility region

Finally, we also experimented with initial time known within one minute. This is a very modest requirement for timing assistance, and even the receiver's internal clock should be able to meet this requirement. With this information, the computation time is reduced by two orders of magnitude (Figure 5).

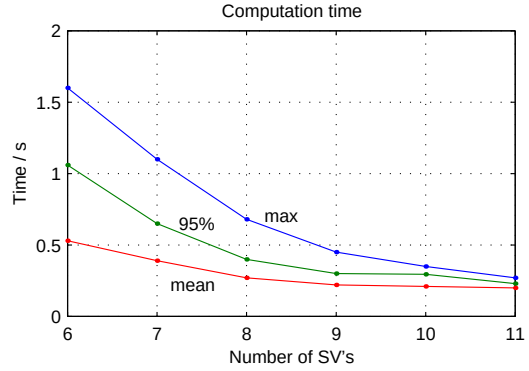


Figure 5: Computation time assuming reference time within one minute

CONCLUSIONS

As a conclusion, an accurate GPS position can be solved without the navigation data and without accurate time assistance from the network, sometimes even in a reasonable time. The receiver still needs the ephemeris parameters and the pseudo-range correction parameters, but they are valid for hours after reception and thus easy to deliver to the receiver in time. In fact, there already are cellular standards for US CDMA, US TDMA, and European GMS and UMTS systems for network assistance messages containing these information elements [5,6]. The presented method makes it possible to solve GPS position with rather modest assistance.

In comparison to the conventional GPS and AGPS (Assisted GPS) or WAG (Wireless Assisted GPS) solutions, this method has several potential advantages. Most importantly, GPS navigation becomes possible in some bad signal conditions, even with time assistance significantly worse than the 1.5 or 3 seconds required by the earlier solutions [2, 3]. Additionally, while the conventional receiver must listen to at least a few seconds of the navigation message in order to acquire the system time, this algorithm can start calculating as soon as the assistance data is present.

The drawback is, evidently, the increased requirement of computational power at least in the case where even the coarse reference time is unavailable.

When compared to the conventional least squares pseudorange positioning method, range fitting produces somewhat less accurate position fixes due to the system time and thus satellite positions being solved inaccurately [4]. However, the accuracy difference is marginal and negligible in practice since it will be buried in multipath etc. especially in weak signal situations.

Another downside of the presented method is that exact GPS time cannot be solved thus not making it possible to acquire accurate time from GPS. On the other hand, exact time is not needed in navigation applications or in emergency call positioning in which only good coverage and rapid time-to-fix will make a difference.

ACKNOWLEDGEMENTS

The authors would like to thank Professor Robert Piché for his constructive comments and advice.

REFERENCES

[1] Syrjärinne, J. Possibilities for GPS time recovery with GSM network assistance. Proceedings of ION GPS 2000.

[2] Enge, P., Fan, R. & Tiwari, A. GPS reference network's new role: providing continuity and coverage. GPS World, July 2001.

[3] Bryant, R., Dougan, S. & Glennon, E. GPS receiver algorithms & system for weak signal operation. Proceedings of ION GPS 2001.

[4] Sirola, N. & Syrjärinne, P. GPS positioning without navigation data. Proceedings of ENC-GNSS2002, 27-30 May 2002, Copenhagen.

[5] TDMA Third generation wireless system assisted mobile positioning through satellite teleservice. SP-4027-740 Draft text. June 5, 2000.

[6] ETSI GSM LCS TS 101 527 V7.2.0 (2000-04). Mobile station (MS) - serving mobile location centre (SLMLC) radio resource LCSD protocol (RRLP). GSM 04.31 version 7.2.0.

PUBLICATION 2

Niilo Sirola: A versatile algorithm for local positioning in closed form. In *Proceedings of the 8th European Navigation Conference ENC-GNSS 2004*, May 16–19, Rotterdam, 2004.

A Versatile Algorithm for Local Positioning in Closed Form

Niilo Sirola, *Institute of Mathematics, Tampere University of Technology, Finland*

ABSTRACT

This paper presents an algorithm for computing the intersections of three surfaces that are either planes or quadrics of revolution. The algorithm is directly applicable to calculating position from various combinations of range and pseudorange measurements. With a locality assumption, the system is reduced into a simplified form that is easy to solve analytically.

The presented algorithm can be used as a preliminary step to obtain an optimal set of starting points for least-squares iteration. The testing section presents an example of how the algorithm can be used to augment the conventional iterative least-squares method in the presence of a two-fold ambiguity.

INTRODUCTION

This paper describes a versatile local closed-form point position solution that can be computed from various minimally determined combinations of range and pseudorange measurements. These measurements are interpreted as quadratic surfaces and planes, whose intersections coincide with the position solutions.

A minimally determined system of quadratic surfaces in three dimensions may have up to nine distinct solutions. In satellite positioning, only one of the different solutions is terrestrial and others can be ruled out [4]. This is not necessarily the case when some or all of the positioning stations are terrestrial.

The widely used iterative least-squares method produces just one solution, usually the one closest to the starting point of the iteration. Closed-form methods, on the other hand, calculate all solutions ensuring that the true solution is not overlooked.

There already exist various closed-form methods that can handle homogenous measurements from one type of stations [1, 5]. There also are methods that cover some specific combinations [2, 7]. The method described in this paper, however, can be applied to almost any combination of station and measurement types. A general minimally determined set of measurements is transformed into a unified form, which can be easily solved.

LOCAL POSITIONING

The novel idea in this contribution is to combine the closed-form analytic solutions with the concept of local positioning.

In the local positioning scheme, it is assumed that a rough position of the user is already known. In the case of a cellular telephone, for example, the position of the phone is known to lie inside the coverage area of the serving base station. The area inside which the user is known to lie is called the local area.

The precise 3D position of the user inside the local area is then solved from a set of measurements to positioning stations. Stations can include positioning satellites, cellular base stations, or other suitable beacons with known locations.

The stations are classified into local and distant stations. A station is considered distant if it is so far away that the direction from user to the station is constant over the whole local area. This allows for the linearization of the measurements to the distant station. A typical example of a distant station is a positioning satellite. Another example is the altitude than can be modelled as a range measurement to the center of the Earth, which is distant in most settings and can thus be linearized.

The stations that are not distant are called local stations. They are relatively close to the user and the directions to the stations can change drastically within the local region.

The threshold distance dividing the stations to local and distant ones depends on the size of the local area and the level of linearization error that we are ready to

accept. Take, for example, a mobile cell with a radius of 3 km. If we allow a maximum of 10 meters of linearization error in the measurements, then all stations farther than 450 km away can be considered distant.

MATHEMATICAL MODEL

The 3D surfaces encountered in local range and range difference positioning and examples of the measurements generating them are:

1. **plane** – defined by a range measurement to a distant station, a range difference between distant stations, or an altitude measurement
2. **sphere** – defined by a range measurement to a local station
3. **paraboloid of revolution** – defined by a range difference between a local and a distant station
4. **branch of a hyperboloid of revolution** – defined by a range difference between two local stations

In the following, we use a local coordinate frame. The calculations are, however, easily generalized to ECEF (Earth-Centered Earth-Fixed) coordinates that are often used in position calculations.

The equation of a plane is

$$P_i: \quad \mathbf{n}_i^T \mathbf{x} = a_i. \tag{1}$$

A central idea of this contribution is to model the measurements to distant stations as planes in order to make the calculations more tractable. The surfaces involving local stations can all be written in the form

$$Q_i: \quad \|\mathbf{s}_i - \mathbf{x}\| + \mathbf{n}_i^T \mathbf{x} = a_i \tag{2}$$

where $\mathbf{s}_i$ is a focal point of the quadric and $\mathbf{n}_i$ is called the directrix vector. This is the focal point/directrix representation of a branch of a quadric of revolution. The formulation is similar to that used by Leva in [5]. The type of the surface depends on the values of the parameters $\mathbf{s}$, $\mathbf{n}$, and a. Table 1 summarizes the different surface types, including the degenerate ones.

In the rest of the paper, the term quadric always refers to a surface of the form (2).

BASIC INTERSECTIONS

The minimally determined positioning problem is equivalent to solving the intersections of three surfaces that are either planes or quadratic surfaces. In the following, we present the four possible intersection types.

Table 1: Surfaces defined by $\|\mathbf{s} - \mathbf{x}\| + \mathbf{n}^T\mathbf{x} = a$

	$a < \mathbf{s}^T\mathbf{n}$	$a = \mathbf{s}^T\mathbf{n}$	$a > \mathbf{s}^T\mathbf{n}$
$\mathbf{n} = \mathbf{0}$	-	(point)	sphere
$0 < \|\mathbf{n}\| < 1$	-	(point)	(ellipsoid)
$\|\mathbf{n}\| = 1$	-	(line)	paraboloid
$\|\mathbf{n}\| > 1$	hyperboloid branch	cone branch	hyperboloid branch

All the measurements are assumed independent in the sense that any set of three normal vectors or directrix vectors is independent. This rules out some degenerate intersections.

Three planes

The intersection of three planes (1)

$$P_1 \cap P_2 \cap P_3 \tag{3}$$

can be solved as a linear system

$$\begin{bmatrix} \mathbf{n}_1^T \\ \mathbf{n}_2^T \\ \mathbf{n}_3^T \end{bmatrix} \mathbf{x} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}. \tag{4}$$

Because the vectors $\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$ are independent, this system has exactly one solution.

Line and a quadric

The intersection of two planes (1) and a quadric (2)

$$P_1 \cap P_2 \cap Q_3$$

reduces to an intersection of a line and a quadric because the two non-parallel planes intersect in a line. The system has either two distinct intersections, as in the case illustrated in Figure 1, or one intersection when the line touches the quadric, or no intersections at all.

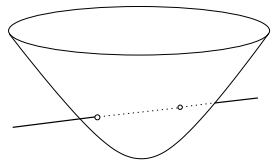


Figure 1: The two intersection points of a paraboloid and a line are marked with circles.

See the Appendix for the general solution formula.

Plane and two quadrics

A more complex case of intersecting a plane with two quadrics

$$P_1 \cap Q_2 \cap Q_3 \tag{5}$$

is illustrated in Figure 2.

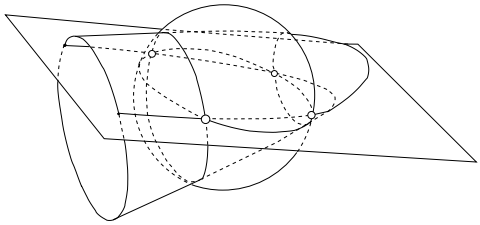


Figure 2: Intersection of a plane and two quadrics, a paraboloid and a sphere. The four intersection points are marked with circles.

This type of system can have 0 to 4 distinct solutions. The Appendix gives a solution for a plane–sphere–quadric intersection which is sufficient in most cases.

Three quadrics

The last case is that all three surfaces are quadrics.

$$Q_1 \cap Q_2 \cap Q_3 \tag{6}$$

The system may have up to nine solutions and can be solved for example with the resultant method [6]. The solution, however, requires symbolic computation and is so sensitive to measurement errors that the closed-form solution is of no use in positioning.

GEOMETRIC REDUCTION

The structure of the intersection is more complex the more quadrics it involves. In the local positioning scheme, we have already introduced the concept of distant stations and managed to model some of the measurements with planes.

Additionally, we can exploit the fact that if two quadrics of revolution intersect in a plane, one of the quadrics can be replaced with the intersection plane without affecting the solution set.

The intersection curve of two quadrics of the form (2) lies on a plane (at least) if the quadrics

1. share a focal point $\mathbf{s}$, or
2. have the same directrix vector $\mathbf{n}$.

The formulae for the intersection planes in the two cases are given in the Appendix.

Consequently, if we have control over the order in which the range differences are formed, it is beneficial to form all the differences in respect with the same station. Then the resulting quadrics share the station as their focal point and their intersection is equivalent to set planes intersecting at most one quadric.

Also, since all spheres have $\mathbf{0}$ as the directrix vector, two spheres trivially intersect in a plane (provided that they intersect at all.)

We arrive at the following algorithm for closed-form minimally determined positioning:

Algorithm 1:

1. Acquire the raw measurements and transform the measurements to a geometric form.
2. Reduce to a simpler geometry by checking each pair of quadrics and substituting the intersection plane for the other quadric if possible.
3. Solve the intersection(s).
4. Try to eliminate the incorrect solutions with the help of base station sector information, additional altitude information etc.

ON THE ITERATIVE LEAST-SQUARES METHOD

A common method for solving systems of nonlinear algebraic equations is the Gauss-Newton iteration [3]. With n quadrics (2) and m planes (1), the system to be solved is

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \|\mathbf{s}_1 - \mathbf{x}\| + \mathbf{n}_1^T\mathbf{x} - a_1 \\ \vdots \\ \|\mathbf{s}_n - \mathbf{x}\| + \mathbf{n}_n^T\mathbf{x} - a_n \\ \mathbf{n}_{n+1}^T\mathbf{x} - a_{n+1} \\ \vdots \\ \mathbf{n}_{n+m}^T\mathbf{x} - a_{n+m} \end{bmatrix} = \mathbf{0}. \tag{7}$$

Given a suitable starting point $\mathbf{x}_0$, the Gauss-Newton iteration for finding an approximate solution to the system (7) is

$$\mathbf{x}_{k+1} \leftarrow \left[J(\mathbf{x}_k)^T J(\mathbf{x}_k)\right]^{-1} J(\mathbf{x}_k)^T f(\mathbf{x}_k), \tag{8}$$

where the Jacobian matrix J is defined as

$$J(\mathbf{x}) = \begin{bmatrix} -\frac{\mathbf{s}_1-\mathbf{x}}{\|\mathbf{s}_1-\mathbf{x}\|}^T + \mathbf{n}_1^T \\ \vdots \\ -\frac{\mathbf{s}_n-\mathbf{x}}{\|\mathbf{s}_n-\mathbf{x}\|}^T + \mathbf{n}_n^T \\ \mathbf{n}_{n+1}^T \\ \vdots \\ \mathbf{n}_{n+m}^T \end{bmatrix}. \quad (9)$$

The iteration is terminated after a predefined number of steps or when $\mathbf{x}$ does not change much between iterations. The obtained point $\mathbf{x}_k$ minimizes the sum of squared residuals $\sum_{i=1}^{n+m} f_i(\mathbf{x})^2$.

The main advantage of the least-squares method is that it is very generic and can make use of an arbitrary number of measurements of any type. Including more measurements in the solution improves the accuracy when compared to the geometric solution that uses three measurements only. Additionally, the least-squares method readily provides an approximation of the covariance of the solution, $cov(\mathbf{x}) \approx \left[J(\mathbf{x}_k)^T J(\mathbf{x}_k)\right]^{-1}$.

On the other hand, the Gauss-Newton iteration step is based on linearization, and may thus perform poorly when the stations are near and the measurement equations are strongly nonlinear. Moreover, the computational load of the iterative method is sometimes an issue.

In local positioning, the main problem is the presence of multiple solutions. The least-squares iteration finds only one solution at time. Even over-determined systems often have several local minima and the iteration has to be run from multiple starting points in order to find all solution candidates. The challenge is to generate enough starting points such that all solutions are found but not too many to cause unnecessary computational load.

TESTING

The following simulation illustrates one possible application of the presented closed-form algorithm.

Figures 3 and 4 depict the results of a positioning simulation using an altitude measurement, a range measurement to a base station, a range difference to two satellites, and a range difference to two base stations. All measurements have some normally distributed noise in them, and the position was computed from 100 measurement sets both with the geometric algorithm and the iterative least-squares method.

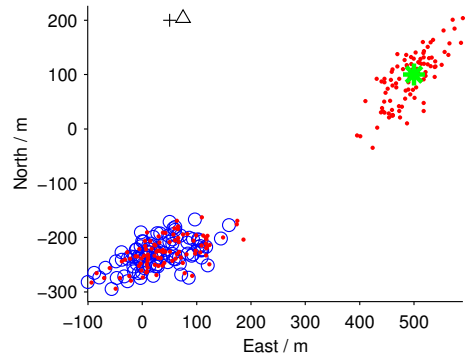


Figure 3: True position (green star), base station (black triangle), geometric solutions (red dots), LS starting point (black cross), and LS solutions (blue circles).

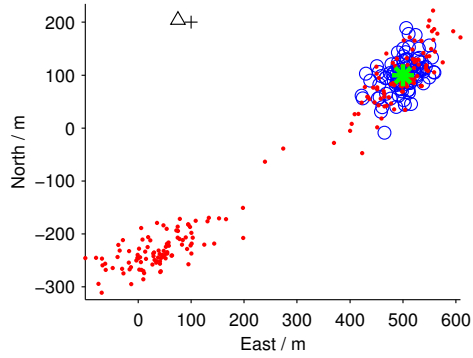


Figure 4: True position (green star), base station (black triangle), geometric solutions (red dots), LS starting point (black cross), and LS solutions (blue circles).

The closed-form geometric solution uses only the first three measurements, while the iterative least-squares method uses all four measurements. The geometric method gives two position candidates per fix, resulting in two symmetric clusters of position candidates. The iterative least-squares method produces only one solution per fix.

Although the system is over-determined, the choice of the starting point may have dramatic effect on the solutions. In Figure 3, the starting point is to the left of the base station, and all 100 position solutions end up near the ”wrong” minimum. The situation in Figure 4 is similar except for the more fortunate choice of the starting point, and the least-squares solutions end up being a little more accurate than the geometric solutions.

Finally, Figure 5 gives the results when the two geometric solutions are computed first and the least-squares iteration then started from both solutions. Of the resulting two local minima, we have here chosen the one with the smaller residual. As can be seen, this is not a 100% reliable criterion, but still only a few solutions have strayed near the false solution.

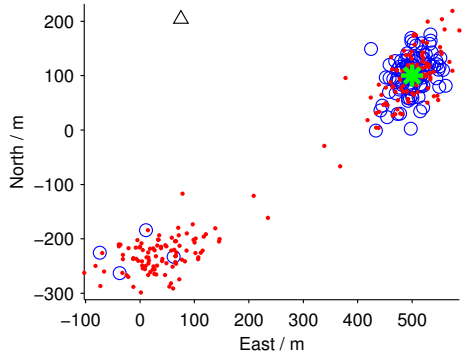


Figure 5: True position (green star), base station (black triangle), geometric solutions (red dots), and LS solutions (blue circles).

For more test results, see [8].

CONCLUSIONS

This paper presents an algorithm for computing the intersections of three surfaces that are either planes or (branches of) quadrics of revolution.

In the context of local positioning, the presented algorithm can be used as a preliminary step to obtain a set of starting points for least-squares iteration that is optimal in some sense.

As the geometries encountered in local positioning often produce more than one solution, the choice of the most likely of the solutions is an important topic for future research.

ACKNOWLEDGEMENT

This study was supported by Nokia Mobile Phones and a grant from the Nokia Foundation. The author would also like to thank Professor Robert Piché and doctor Jari Syrjärinne for their advice during the research.

REFERENCES

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

[2] I. Fernandez-Corbaton, A. Vayanos, P. Agashe, and S. Soliman. Method and apparatus for determining an algebraic solution to GPS terrestrial hybrid location system equations. US Patent 6,289,280 B1, Sep. 11, 2001.

[3] R. Fletcher. *Practical methods of optimization*. John Wiley & Sons, Chichester, second edition, 1987.

[4] E. D. Kaplan, editor. *Understanding GPS: principles and applications*. Artech House, Norwood, 1996.

[5] J. L. Leva. An alternative closed-form solution to the GPS pseudo-range equations. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1430–1439, 1996.

[6] D. Manocha and J. Demmel. Algorithms for intersection parametric and algebraic curves i: Simple intersections. *ACM Transactions on Graphics*, 13(1):73–100, 1994.

[7] M. Phatak, M. Chansarkar, and S. Kohli. Position fix from three GPS satellites and altitude: a direct method. *IEEE Transactions on Aerospace and Electronic Systems*, 36(1):350–354, 1999.

[8] N. Sirola, R. Piché, and J. Syrjärinne. Closed-form solutions for hybrid cellular/GPS positioning. In *Proceedings of the ION GPS/GNSS 2003*, pages 1613–1619, 2003.

APPENDIX

Line–quadric intersection

To solve the system of two planes and one arbitrary quadric

$$\begin{cases} \mathbf{u}_1^T\mathbf{x} &= d_1 \\ \mathbf{u}_2^T\mathbf{x} &= d_2 \\ \|\mathbf{s}-\mathbf{x}\| + \mathbf{n}^T\mathbf{x} &= a \end{cases}, \tag{10}$$

construct the parametric equation of the line that satisfies the two linear equations:

$$\mathbf{x} = \mathbf{p} + t\mathbf{w}, \quad t \in \mathbb{R}, \tag{11}$$

where $\mathbf{p}$ is any point on the line and $\mathbf{w}$ is the unit direction of the line, obtained from

$$\mathbf{w} = \frac{\mathbf{u}_1 \times \mathbf{u}_2}{\|\mathbf{u}_1 \times \mathbf{u}_2\|}. \tag{12}$$

As $\mathbf{p}$ can be any point on the line, we choose – with a little forethought – the solution of the linear system

$$\begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \mathbf{w}^T(\mathbf{n}\mathbf{n}^T - I) \end{bmatrix} \mathbf{p} = \begin{bmatrix} d_1 \\ d_2 \\ \mathbf{w}^T(a\mathbf{n} - \mathbf{s}) \end{bmatrix}. \quad (13)$$

Substitute $\mathbf{x}$ as defined in (11) into the last equation in (10) to get

$$\|\mathbf{s} - (\mathbf{p} + t\mathbf{w})\| + \mathbf{n}^T(\mathbf{p} + t\mathbf{w}) = a, \quad \text{or}$$
$$\|\mathbf{s} - \mathbf{p} - t\mathbf{w}\| = a - \mathbf{n}^T\mathbf{p} - t\mathbf{n}^T\mathbf{w}. \quad (14)$$

Squaring and rearranging, the first-order terms cancel out because of our choice of $\mathbf{p}$, and what remains is a second-degree polynomial in t:

$$\left[1 - (\mathbf{n}^T\mathbf{w})^2\right] t^2 = (a - \mathbf{n}^T\mathbf{p})^2 - \|\mathbf{s} - \mathbf{p}\|^2. \quad (15)$$

Solve this for t, substitute to (11), and get the solutions

$$\mathbf{x} = \mathbf{p} \pm \sqrt{\frac{(a - \mathbf{n}^T\mathbf{p})^2 - \|\mathbf{s} - \mathbf{p}\|^2}{1 - (\mathbf{n}^T\mathbf{w})^2}}\, \mathbf{w}. \quad (16)$$

Plane–sphere–quadric intersection

Write the system as

$$\begin{cases} \mathbf{u}^T\mathbf{x} = a, \quad \|\mathbf{u}\| = 1 \\ \|\mathbf{q} - \mathbf{x}\| = c \\ \|\mathbf{s} - \mathbf{x}\| + \mathbf{n}^T\mathbf{x} = b \end{cases} \quad (17)$$

where $\mathbf{u}$ and $\mathbf{n}$ are not parallel. The intersection of the sphere and the plane is a circle with the parametric presentation

$$\mathbf{x} = \mathbf{p} + t\mathbf{v} \pm \sqrt{1 - t^2}\mathbf{w}, \quad -1 \le t \le 1 \quad (18)$$

where

$$\mathbf{p} = \mathbf{q} + (a - \mathbf{u}^T\mathbf{q})\mathbf{u} \quad (19)$$

$$\mathbf{w} = r\frac{\mathbf{u} \times \mathbf{n}}{\|\mathbf{u} \times \mathbf{n}\|}, \quad \mathbf{v} = \mathbf{u} \times \mathbf{w} \quad (20)$$

$$r = \sqrt{c^2 - (a - \mathbf{u}^T\mathbf{q})^2}. \quad (21)$$

Substitute (18) into the last equation of (17). Rearrange and square twice to obtain a fourth-degree polynomial

$$a_1^2t^4 + 2a_1a_2t^3 + (a_2^2 + 2a_1a_3 + a_4^2)t^2 + 2a_2a_3t + a_3^2 - a_4^2 \quad (22)$$

where

$$\begin{aligned} a_1 &= (\mathbf{n}^T\mathbf{v})^2 \\ a_2 &= 2\left[\mathbf{s} + (\mathbf{n}^T\mathbf{p} - b)\mathbf{n} - \mathbf{p}\right]^T\mathbf{v} \\ a_3 &= b^2 - r^2 - \|\mathbf{s}\|^2 + \left[2\mathbf{s} + (\mathbf{n}^T\mathbf{p} - 2b)\mathbf{n} - \mathbf{p}\right]^T\mathbf{p} \\ a_4 &= 2(\mathbf{s} - \mathbf{p})^T\mathbf{w}. \end{aligned} \quad (23)$$

Solve this and substitute the real roots into (18) to get 0–4 solutions.

Focal presentation of a hyperboloid branch

Starting from the range difference equation

$$\|\mathbf{s}_1 - \mathbf{x}\| - \|\mathbf{s}_2 - \mathbf{x}\| = d, \quad (24)$$

we can write

$$\begin{aligned} \|\mathbf{s}_1 - \mathbf{x}\| &= \frac{d + \|\mathbf{s}_1 - \mathbf{x}\| + \|\mathbf{s}_2 - \mathbf{x}\|}{2} \\ &= \frac{d^2 + d(\|\mathbf{s}_1 - \mathbf{x}\| + \|\mathbf{s}_2 - \mathbf{x}\|)}{2d} \quad d \neq 0 \\ &= \frac{d^2 + \|\mathbf{s}_1 - \mathbf{x}\|^2 - \|\mathbf{s}_2 - \mathbf{x}\|^2}{2d} \\ &= \frac{d^2 + \|\mathbf{s}_1\|^2 - 2(\mathbf{s}_1 - \mathbf{s}_2)^T\mathbf{x} - \|\mathbf{s}_2\|^2}{2d} \end{aligned} \quad (25)$$

and rearrange this to

$$\|\mathbf{s}_1 - \mathbf{x}\| + \frac{(\mathbf{s}_1 - \mathbf{s}_2)^T}{d}\mathbf{x} = \frac{d^2 + \|\mathbf{s}_1\|^2 - \|\mathbf{s}_2\|^2}{2d} \quad (26)$$

which is of the form (2).

First type of reduction

Start with two quadrics sharing a focal point

$$\begin{cases} \|\mathbf{s} - \mathbf{x}\| + \mathbf{n}_1^T\mathbf{x} = a_1 \\ \|\mathbf{s} - \mathbf{x}\| + \mathbf{n}_2^T\mathbf{x} = a_2 \end{cases} \quad (27)$$

and subtract the second equation from the first one to see that the intersection lies on the plane

$$(\mathbf{n}_1 - \mathbf{n}_2)^T\mathbf{x} = a_1 - a_2. \quad (28)$$

Second type of reduction

Start with two quadrics sharing the directrix vector

$$\begin{cases} \|\mathbf{s}_1 - \mathbf{x}\| + \mathbf{n}^T\mathbf{x} = a_1 \\ \|\mathbf{s}_2 - \mathbf{x}\| + \mathbf{n}^T\mathbf{x} = a_2 \end{cases} \quad (29)$$

and subtract the second equation from the first to get

$$\|\mathbf{s}_1 - \mathbf{x}\| - \|\mathbf{s}_2 - \mathbf{x}\| = a_1 - a_2. \quad (30)$$

This is the equation of a hyperboloid branch, which we transform according to Eqs. (24) and (26) to

$$\|\mathbf{s}_1 - \mathbf{x}\| + \frac{(\mathbf{s}_1 - \mathbf{s}_2)^T}{a_1 - a_2}\mathbf{x} = \frac{(a_1 - a_2)^2 + \|\mathbf{s}_1\|^2 - \|\mathbf{s}_2\|^2}{2(a_1 - a_2)} \quad (31)$$

Substitute $\|\mathbf{s}_1 - \mathbf{x}\| = a_1 - \mathbf{n}^T\mathbf{x}$ to get

$$a_1 - \mathbf{n}^T\mathbf{x} + \frac{(\mathbf{s}_1 - \mathbf{s}_2)^T}{a_1 - a_2}\mathbf{x} = \frac{(a_1 - a_2)^2 + \|\mathbf{s}_1\|^2 - \|\mathbf{s}_2\|^2}{2(a_1 - a_2)} \quad (32)$$

and simplify to

$$2\left[(a_2 - a_1)\mathbf{n} + \mathbf{s}_1 - \mathbf{s}_2\right]^T\mathbf{x} = a_2^2 - a_1^2 + \|\mathbf{s}_1\|^2 - \|\mathbf{s}_2\|^2 \quad (33)$$

which is the equation of the intersection plane.

PUBLICATION 3

Niilo Sirola, Robert Piché and Henri Pesonen: Numerical integration methods in local positioning. In *Proceedings of the 2nd Workshop on Positioning, Navigation and Communication (WPNC05)*, March 17, 2005, Hannover, pages 21–30, 2005.

PROCEEDINGS OF THE 2nd WORKSHOP ON POSITIONING, NAVIGATION AND COMMUNICATION (WPNC'05)
& 1st ULTRA-WIDEBAND EXPERT TALK (UET'05)

Numerical Integration Methods in Local Positioning

Niilo SIROLA[a], Robert PICHÉ, Henri PESONEN,
Tampere University of Technology, Tampere, Finland

[a]corresponding author, e-mail: niilo.sirola@tut.fi

Abstract - A Bayesian approach to position estimation requires the evaluation of multidimensional integrals which, except for special cases, must be evaluated numerically. In this paper different numerical integration methods are compared: nodal quadrature with uniform grid, Monte Carlo with random and quasi-random nodes, and adaptive cubature. The methods are used to compute a number of two- and three- dimensional integrals having features typical of "local" positioning. It is found that adaptive cubature methods outperform the other approaches both in terms of accuracy and computation time, at least in low dimensions and when the probability densities are smooth.

1 Local Positioning

One of the challenges of positioning and tracking is that the number and locations of the reference signal sources can be, at best, barely sufficient. Under these circumstances, the estimation algorithm should use all the available information as effectively and efficiently as possible. In *local* positioning and tracking, which is based on reference signals from nearby sources such as Bluetooth, WLAN, acoustic sensors, or cellular networks, the measurement geometry is strongly non-linear and the measurement errors are strongly non-normal, with multiple modes and curved ridges. These characteristics can seriously degrade the accuracy and reliability of conventional least-squares methods, which are optimal for linear geometry and normal error distributions.

The Bayesian approach[2] offers a general framework for positioning and tracking for arbitrary geometries and probability distributions. Many conventional methods can be formulated as special cases of Bayesian positioning and tracking. A Bayesian position estimate is based on the *posterior* probability distribution of the position. By Bayes' theorem, the posterior distribution is proportional to the product of the *measurement likelihood function* and the *prior* distribution. A Bayesian model of a typical positioning problem is presented in section 2.

The position estimate can be derived from the posterior probability distribution in different ways. The *maximum likelihood estimate* is the point at which the posterior distribution is maximum. When the measurement errors have normal distributions, the computation of the maximum likelihood estimate is equivalent to the conventional iterative least squares method. For general multimodal distributions, however, numerically finding the global maximum can be challenging.

An alternative estimate is the *mean value* of the posterior distribution. In the special case of a unimodal and symmetric posterior distribution, the maximum likelihood estimate coincides with the mean. Apart from some special cases, computing the mean value requires a numerical method to approximate integrals in several dimensions. The challenge is then to find a numerical method to compute these integrals with reasonable speed and accuracy, especially taking into account the limited computing power available in mobile positioning settings.

In section 3, we describe three standard numerical integration methods:

Monte Carlo integration is a popular choice for multidimensional integrals. The samples can be from a random number generator or from a quasi-random sequence (lattice). The error of the approximation can be evaluated using standard variance estimation formulas.

Nodal quadrature with a uniform rectangular grid is a generalization of the trapezoid rule; error can be estimated by combining solutions from different grid refinements.

Adaptive quadrature methods combine sophisticated subregion grid refinement and integration formulas of different orders. Schürer [4] reports that such methods can be competitive with Monte Carlo methods even for problems of up to 100 dimensions.

In section 4, we report on a series of numerical tests of these methods applied to two- and three-dimensional integrals arising in a typical position estimation problem.

The paper closes with a summary of our conclusions and an outline of further investigations.

2 Model

In this work we consider only positioning with range measurements, but any types of measurements with a known error model, such as range difference measurements, can be modeled similarly and incorporated to the model.

The *range measurement* to the ith fixed station can be modeled as

$$r_i = \|\mathbf{s}_i - \mathbf{x}\| + \varepsilon_i$$

where $\mathbf{s}_i$ is the known station position, $\mathbf{x}$ is the unknown user position, and ε_i is a random variable that represents the measurement error. Denoting the measurement error probability density as ϕ_i, the measurement likelihood function can be defined as

$$p(r_i \mid \mathbf{x}) = \phi_i(\|\mathbf{s}_i - \mathbf{x}\| - r_i).$$

This formula can be interpreted as the probability density of the measurement given a particular user position. For example, if the measurement error has normal distribution with zero mean and variance σ_i^2, then the measurement likelihood would be

$$p(r_i \mid \mathbf{x}) = \frac{1}{\sigma_i\sqrt{2\pi}} e^{-(\|\mathbf{s}_i-\mathbf{x}\|-r_i)^2/(2\sigma_i^2)}.$$

In general, however, the measurement error distribution need not be normal, or even symmetric or zero-mean. In the case of several independent measurements $r_1, \ldots, r_n$, stacked into a vector $\mathbf{r}$, we have

$$p(\mathbf{r} \mid \mathbf{x}) = \prod_{i=1}^{n} \phi_i(\|\mathbf{s}_i - \mathbf{x}\| - r_i).$$

Bayes' formula gives the posterior distribution, modulo a normalising factor, as

$$p(\mathbf{x} \mid r) \propto p(\mathbf{x}) p(r \mid \mathbf{x}),$$

where $p(\mathbf{x})$ is the prior density for $\mathbf{x}$. A position estimate is then given by the mean value of the posterior density,

$$\hat{\mathbf{x}} = \frac{\int \mathbf{x} p(\mathbf{x}) p(r \mid \mathbf{x})\, \mathrm{d}\mathbf{x}}{\int p(\mathbf{x}) p(r \mid \mathbf{x})\, \mathrm{d}\mathbf{x}}. \tag{1}$$

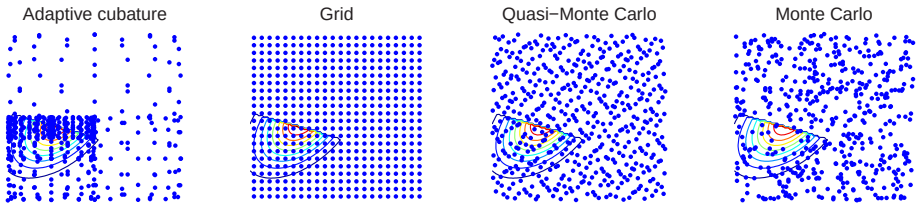


Figure 1: Sample points chosen by different numerical integration methods for $N = 500$.

In the scope of this work, we take the prior to be constant in a given bounded area of interest Ω and zero outside this area, although other sensible choices for the prior also exist. With this prior the position estimate (1) reduces to

$$\hat{\mathbf{x}} = \frac{\int_\Omega \mathbf{x} p(r \mid \mathbf{x}) \, \mathrm{d}\mathbf{x}}{\int_\Omega p(r \mid \mathbf{x}) \, \mathrm{d}\mathbf{x}}.$$

It is interesting to note that the Bayesian approach provides a position estimate even when the number of range measurements is too small to determine a unique geometric position fix or least squares position estimate.

3 Integration methods

All the numerical integration methods discussed here can be written in the form

$$\int_\Omega f(\mathbf{x}) d\mathbf{x} \approx V \frac{1}{N} \sum_{k=1}^{N} f(\mathbf{x}_k)$$

where the number of integration points N and the integration points $\mathbf{x}_k$ are either decided beforehand independent of the integrand, or adaptively during the integration. The integration region Ω is taken to be a multidimensional cube and V denotes the volume of Ω.

In choosing a suitable integration method for this particular application, there are several considerations. As opposed to asymptotic error analysis, we are interested in the accuracy gained with a minimum amount of work. The measurements usually have errors in the order of tens to hundreds of meters in them, so that it does not make sense to look for more than two or three decimals of accuracy. A more important quality of an integration method is to find a coarse solution with as few function evaluations as possible. In addition, the method should come with a robust error estimate, but not an overly pessimistic one.

Figure 1 illustrates how the three considered integration schemes distribute the integration points in the case of a 2-dimensional example integrand. The distribution produced by the plain Monte Carlo is also given for reference.

3.1 Nodal quadrature

Nodal quadrature ("grid method") is a generalization of the trapezoid rule to multiple dimensions. The integration nodes $\mathbf{x}_k$ are chosen to form an equispaced rectangular grid over the integration domain. For irregular regions a triangular or tetrahedral grid may be used. The integral over each element is then approximated as the element area or volume times the average

of the integrand values at the vertices. The error of the nodal quadrature can be estimated by extrapolation of the solutions for two grids of different densities, using the fact that the error is proportional to the square of the element dimension.

3.2 Monte Carlo and Quasi Monte Carlo Integration

In Monte Carlo approach, the integration points are chosen at random. In this application, we choose the points from a uniform distribution over the integration domain.

The variance of the Monte Carlo estimate is

$$\frac{V^2}{N}\sum_{k=1}^{N} f^2(\mathbf{x}_k) - \frac{V^2}{N^2}\left(\sum_{k=1}^{N} f(\mathbf{x}_k)\right)^2, \tag{2}$$

which is easy to compute along with the integral. The integration can be run until desired accuracy is reached. The error estimate is only probabilistic, though. A variant of the method is called quasi-Monte Carlo. Quasi-random numbers are a deterministic sequence that has the properties of a uniform random sequence, but the points are "more uniformly" distributed in space. The rightmost plot in Figure 1 illustrates how the uniformly random points can sometimes miss the important regions even in two dimensions.

3.3 Adaptive Cubature

In an adaptive scheme, the integration nodes are concentrated to where the integrand is changing most rapidly. For testing, we used an experimental CUBPACK package[1] that performs adaptive subregion division and provides error estimates.

Figure 1 illustrates how the adaptive method concentrates its effort around a discontinuous edge in the integrand since that is where integrand is changing most rapidly. This may not be preferable behavior for this application.

4 Numerical testing and results

In the testing phase, we generated a large amount of test cases where range measurements from two stations are received. In addition, linear constraints may be imposed, simulating the effect of sector boundaries.

The measurement errors are assumed normal, and thus the test problems are of the form

$$p(\mathbf{r} \mid \mathbf{x}) \propto I_{A,b}(\mathbf{x}) \exp\left(-\frac{1}{2}\sum_{i=1}^{2} \frac{(\|\mathbf{s}_i - \mathbf{x}\| - r_i)^2}{\sigma_i^2}\right) \tag{3}$$

where the indicator function $I_{A,b}(\mathbf{x}) = 1$ if $A\mathbf{x} \geq b$ and 0 otherwise.

The integration methods are used to compute the mean value of the integrand over a 1 km by 1 km domain. The integrands in this case may have either one or two peaks in the domain, and depending on the linear constraints, they may have discontinuous edges.

We chose the parameters $\mathbf{s}_i$, r_i, σ_i, A, and b for each test case such that we get an equal number of each of the three types of test functions:

Type 1 - unimodal continuous

Type 2 - multimodal continuous

Type 3 - unimodal with a discontinuity

Type 4 - multimodal with one or more discontinuities

The measurement errors have standard deviation σ between 50 and 150 meters, which could corresponds to either good-quality base station measurements or bad-quality satellite measurements (without multipath).

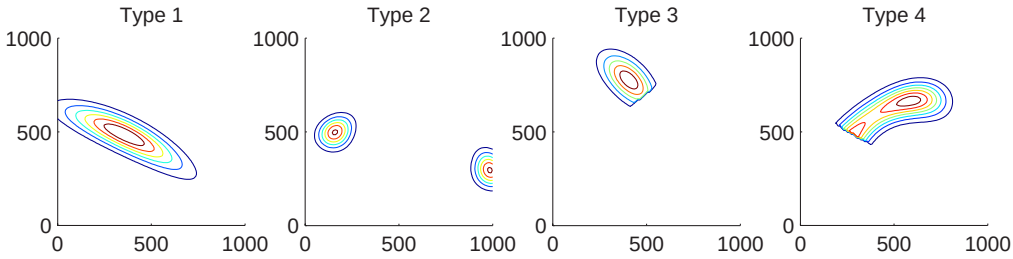


Figure 2: Examples of the different types of test functions.

Figure 2 shows contour plots of some example functions of each type.

For each test case, a reference solutions was sought using Cubpack with a maximum of 150 000 integrand evaluations allowed. In most cases, this took the accuracy of the reference solution to the centimeter-level.

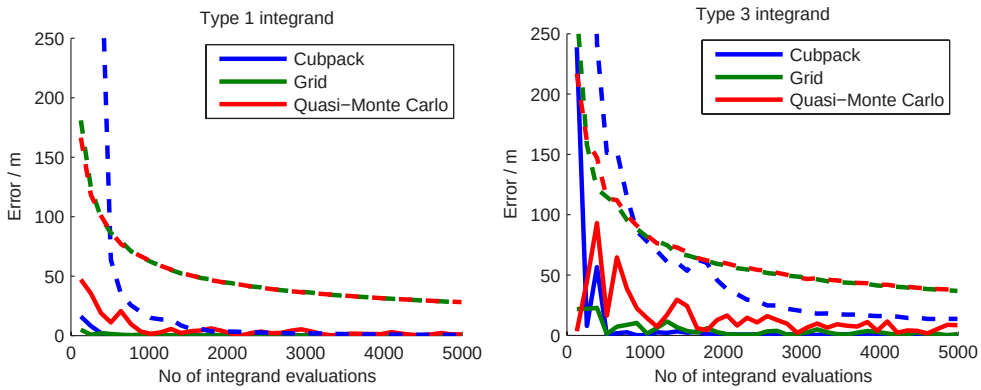


Figure 3: Integration errors (solid lines) and error bounds (dashed lines) with different numbers of integration points.

Obviously, performing this many integrand evaluations is hardly practical, when compared to the iterative point position methods that rarely need to evaluate the cost function more than ten times. Figure 3 illustrates the behavior of the methods with a low number of integration points. In the first plot, the integrand is a well-behaved one and all methods settle to a couple of meters accuracy if more than 1000 integration points are used. Cubpack's error bound comes down rather quickly, but the slack probabilistic error estimates of the other two methods would require a lot more integration points to reach, say, 10 meters.

The second plot in Figure 3 shows an example of a more difficult discontinuous integral. In this plot, it is seen how the quasi-Monte Carlo method produces notably different methods with different amount of integration points although the points are assigned deterministically.

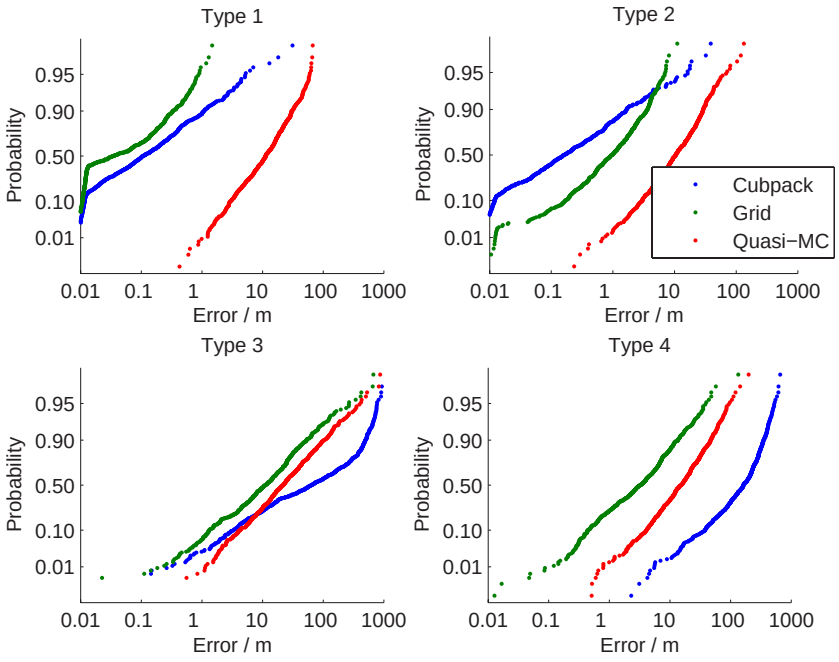


Figure 4: N=500.

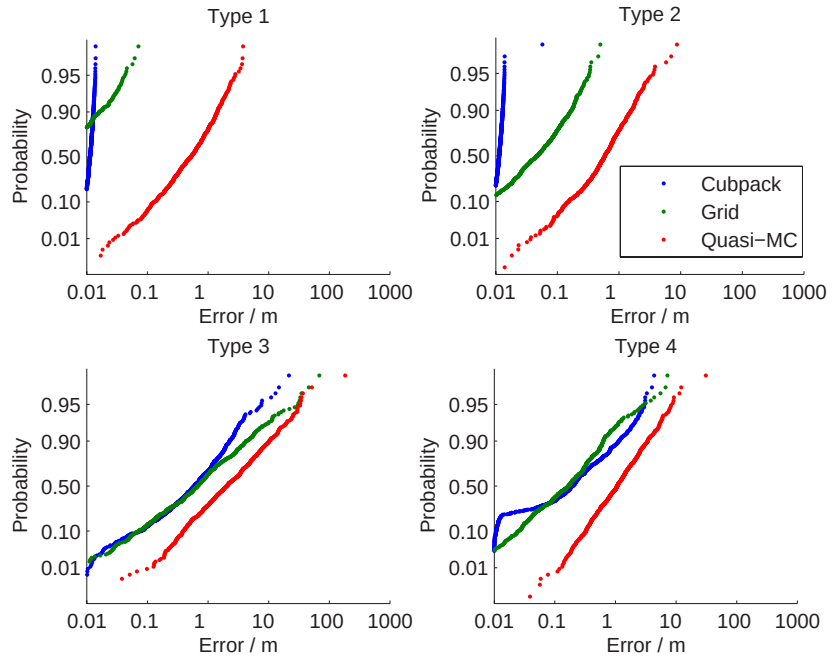


Figure 5: N=10000.

Next, we ran a larger test set using 500, 1 000, 5 000, and 10 000 integration points and compared the accuracies. Figures 4 and 5 break down the results for the different integrand types. These plots represent the distribution of the integration errors, and thus the higher a curve is the better. From Figure 4, it is interesting to notice that with 500 integration points the grid method performs best in all cases but the Type 2 (multimodal continuous). The quasi-Monte Carlo methods is clearly worse than the two others in the continuous cases (Types 1 and 2), but in the discontinuous cases it comes second between the grid method and Cubpack. It is worth to note that Cubpack, being a general-purpose package, needs more points than this to be effective.

As can be seen in Figure 5, when 10 000 integration points are used, Cubpack dominates the results in the continuous cases and reaches centimeter accuracy almost certainly. In the discontinuous tests, Cubpack and the grid method perform comparably. The quasi-Monte Carlo seems to give almost an order of magnitude worse results in all problem types.

$N = 500$	Type 1	Type 2	Type 3	Type 4	Total
Cubpack	41	77	21	8	35
Grid	73	24	56	89	61
Quasi-MC	3	4	23	14	11

$N = 1000$	Type 1	Type 2	Type 3	Type 4	Total
Cubpack	91	96	59	52	74
Grid	57	7	32	45	35
Quasi-MC	10	5	12	14	10

$N = 5000$	Type 1	Type 2	Type 3	Type 4	Total
Cubpack	100	98	69	69	84
Grid	83	16	32	35	42
Quasi-MC	36	31	14	27	27

$N = 10000$	Type 1	Type 2	Type 3	Type 4	Total
Cubpack	100	98	55	56	77
Grid	93	28	48	55	56
Quasi-MC	54	50	26	40	46

Table 1: How many times (%) each method gave the best answer.

The Table 1 gives another perspective to the comparison. The table shows how often a particular method found the best solution to the problem. The columns do not add to 100% because often more than one method produced the same solution (within a tolerance). Again, the table shows that apart from the case of only 500 integration points allowed, the adaptive Cubpack is most often the most accurate of the tested methods.

4.1 A 3D case

For comparison, we did small-scale testing on tree-dimensional test functions also. The 3D test case was formed by adding an "altitude measurement" to the 2D case. The test function was formed by multiplying Eq. (3) by

$$\exp -\frac{1}{2}\frac{x_3^2}{\sigma_a^2},$$

simulating a normally distributed zero-altitude measurement with variance σ_a^2.

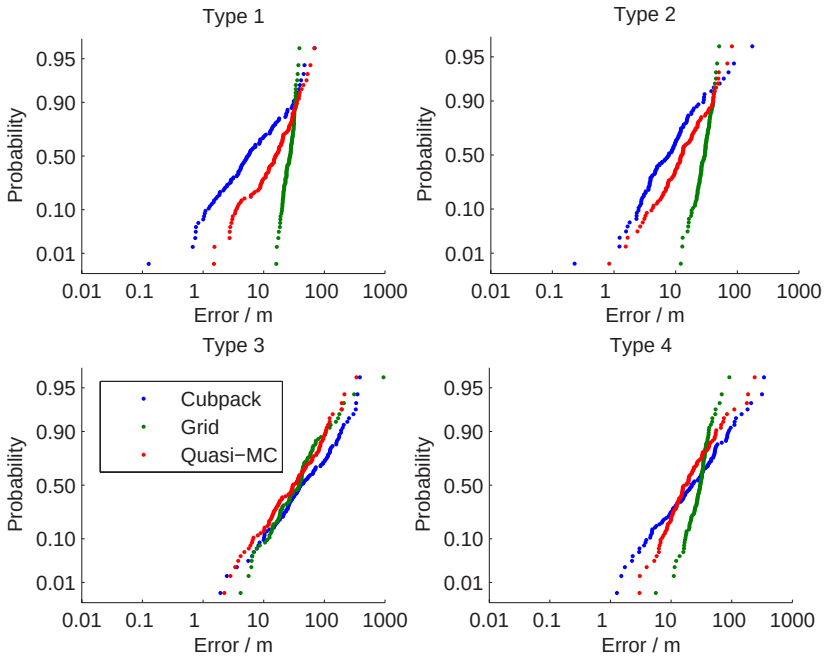


Figure 6: 3D with N=1000.

Figure 6 again shows the superiority of the adaptive method in continuous cases 1 and 2. In the discontinuous cases 3 and 4, the quasi-Monte Carlo method is for the first better than the two others in some regions.

$N = 1000$	Type 1	Type 2	Type 3	Type 4	Total
Cubpack	73	75	23	36	52
Grid	7	1	43	21	18
Quasi-MC	69	74	67	75	71

Table 2: How many times (%) each method gave the best answer in 3D.

Table 2 also shows how the quasi-Monte Carlo is more feasible in a higher dimension.

5 Conclusion

It is found that adaptive cubature methods outperform the Monte Carlo approach both in terms of accuracy and computation time, at least in low dimensions and when the measurement error densities are smooth. With discontinuous integrands, however, the advantage of the adaptive cubature is not so obvious. This can, however, be improved in the future by using polyhedral integration domains with the discontinuities placed at the edges of the domain.

At this point, we assumed that the computational load of each method depends only on the number of integration nodes, ignoring the overhead in the algorithms themselves. More detailed

performance analysis would call for a more thorough implementation of the different methods, otherwise we are comparing implementations instead of algorithms.

Future research topics include more detailed error analysis, where the analytical properties of the family of integrand functions is taken into account. Furthermore, the adaptive strategy can also be applied to Monte Carlo type methods, yielding very promising results[3].

Acknowledgements

This study was funded by Nokia Corporation. Niilo Sirola acknowledges the financial support of the Nokia Foundation.

References

[1] R. Cools and A. Haegemans, "Algorithm 824: CUBPACK: a package for automatic cubature; framework description" *ACM Trans. Math. Software*, vol.29, no.3, pp. 287-296, 2003

[2] D. Fox, J. Hightower, L. Liao, D. Schulz and G. Borriello, "Bayesian Filters for Location Estimation" *IEEE Pervasive Computing*, vol.2, no.3, pp. 24-33, 2003

[3] A. Karaivanova and I. Dimov, "Error analysis of an adaptive Monte Carlo method for numerical integration" *Mathematics and Computers in Simulation*, vol.47, no.2-5, pp. 201-213, 1998

[4] R. Schürer, "A comparison between (quasi-)Monte Carlo and cubature rule based methods for solving high-dimensional integration problems" *Mathematics and Computers in Simulation*, vol.62, no.3-6, pp. 509-517, 2003

PUBLICATION 4

Niilo Sirola and Simo Ali-Löytty: Moving grid filter in hybrid local positioning. In *Proceedings of the 10th European Navigation Conference ENC-GNSS 2006,* May 9–10, 2006.

Moving Grid Filter in Hybrid Local Positioning

Niilo Sirola and Simo Ali-Löytty
Institute of Mathematics, Tampere University of Technology, Finland

Abstract

A novel moving grid filter, a generalization of the point-mass filter, is given for the hybrid local positioning problem. Preliminary test results are given to show that the algorithm is computationally feasible, and produces accuracy of the same order as the grid separation.

1 Introduction

As opposed to single-fix position algorithms where only current measurements are used in computations, position filters try to extract information about the user's current state (position, velocity, heading, etc.) by using all data measured up and until current time. In the Bayesian filtering framework [Fox et al., 2003, Roos et al., 2002], the density function of the state is computed conditional on all current and past measurement data. If the state and measurement processes are modelled as Markov chains, the Bayesian posterior density can be computed recursively, so that all the information in the past measurements is fused into the state probability density, and there is no need to store the old measurements explicitly.

With a navigation filter, we want to estimate the state as given by the discrete-time system defined by the *motion model*:

$$x_{k+1} = f_k(x_k) + w_k,$$

and the *measurement model*:

$$y_k = h_k(x_k) + v_k,$$

where f_k and h_k are known *transition functions* and *measurement functions*, respectively. Assuming noises v_k,w_k independent in time, from each other and from initial state, the optimal algorithm for recursive Bayes estimation is:

1. Start with initial distribution $p(x_0 \mid y_0)$ and $k = 1$
2. Prediction: compute the prior distribution $p(x_k|y_{k-1}) = \int_{\mathrm{R}^d} p(x_{k-1}|y_{k-1})p(x_k|x_{k-1})dx_{k-1}$
3. Correction: compute the posterior distribution $p(x_k \mid y_k) \propto p(x_k \mid y_{k-1})p(y_k \mid x_k)$
4. Output posterior mean $E(x_k)$ and variance $V(x_k)$.
5. Increase k and repeat from 2.

Analytical solutions to the Bayesian filtering problem are known only for a few special cases. In the general case, it is intractable to solve the filtering densities exactly. Instead, the density functions and sometimes also the motion and measurement models are approximated appropriately to make the computation feasible. For example, with linear-Gaussian approximation, the optimal filter reduces to the Extended Kalman filter. Approximating the densities with a set of points distributed accordingly leads to a particle filter. In this paper, we show how a piece-wise constant approximation of the density functions can be used to derive a moving grid filter.

As an additional problem, the output desired from a filter is usually a point estimate, posterior mean, instead of the full posterior distribution. The mean error of the point estimate does not necessarily correlate at all with the quality of the estimated posterior distribution [Duník et al., 2005]. The practical problem of implementing and testing a position filter is thus two-fold. The first problem is to find the optimal filtering solution or at least derive some of its properties like the Cramér-Rao lower bound [Bergman, 2001]. The second problem is to try to approximate the optimal filter in some computationally feasible way. Neither of these problems has been satisfactorily solved.

1.1 Hybrid positioning problem

In the hybrid positioning scheme, measurements and other information from different systems are fused together. One of the challenges is to provide accurate or at least correct position information in urban areas or indoors. The behaviour of satellite-based systems such as GPS is unpredictable at best when used indoors in high-sensitivity mode. Local wireless networks, such as the cellular network, WLAN [Syrjärinne, 2001], or Bluetooth [Kotanen et al., 2003], offer some positioning capacity but with accuracy inferior to GPS. Another possible component of a personal navigator are the on-board sensors such as accelerometers, barometers or digital compasses.

Combining the various measurement sources is difficult because of different error characteristics, unpredictable distortions, or systematic errors in measurements, strong nonlinearity, complex time-dependencies, and missing data. It is not simple to explicitly model all the cases in a general way, let alone solve the models accurately. Even with correct models, the commonly used Kalman filter and its nonlinear extensions can fail without warning [Ali-Löytty et al., 2005].

Table 1 gives some examples of measurements and corresponding measurement equations. The clock bias β and drift γ present in the pseudorange and deltarange measurements can be either estimated as two additional states, or preferably eliminated by differencing.

Each measurement together with the distribution of the measurement error defines a likelihood function $L_i(x \mid m_i)$, and the likelihood implied by several simultaneous measurements is found by multiplying the individual likelihoods. The measurement error ε need not be normal. Ideally, we would use an empirically determined distribution that matches the real situation, as is done in the location fingerprinting methods for mobile phone or WLAN positioning [McGuire et al., 2003, Teuber and Eissfeller, 2006]. From the algorithm point of view, the measurement likelihood can be a "black box" function, as only the function values in certain points are needed.

Table 1: Examples of measurement equations

range	$\|s - x_{1:3}\| = m + \varepsilon$
maximum range	$\|s - x_{1:3}\| \leq m$
pseudorange	$\|s - x_{1:3}\| + \beta = m + \varepsilon$
quantized range	$\mathrm{round}_k \|s - x_{1:3}\| = m + \varepsilon$
sector	$\angle(x_{1:2} - s_{1:2}) \in [m - \theta/2,\ m + \theta/2]$
altitude	$x_3 = m + \varepsilon$
deltarange	$\left(\frac{s - x_{1:3}}{\|s - x_{1:3}\|}\right)^T (s' - x_{4:6}) + \gamma = m + \varepsilon$
heading	$\angle x_{4:5} = m + \varepsilon$

2 The grid mass filter

Model the density functions as piecewise constant over a finite collection of bounded *elements*. The element size is ideally of the same order as the expected accuracy. A coarser grid may be used to detect the approximate local maxima of the likelihood and some more accurate method, such as denser local grid or an iterative search, to pinpoint the exact maximum likelihood estimates.

As suggested in [Kramer and Sorenson, 1988] and refined in [Sirola and Ali-Löytty, 2006], approximate density functions with

$$\hat{p}(x) = \sum_{i=1}^{n} \chi_{G(i)}(x) \frac{\pi(i)}{|G(i)|}$$

where $G(i)$ is the ith parallelepiped element and $\chi_{G(i)}$ its characteristic function, $|G(i)|$ is the volume of the ith element, and $\pi(i) \in [0, 1]$ is the probability of the state being inside the ith element. The density function is thus constant $\frac{\pi(i)}{|G(i)|}$ inside each element.

An evenly spaced d-dimensional parallelepiped grid can be uniquely described with vectors $e \in \mathrm{R}^d$, $n \in \mathrm{Z}_+^d$ and a non-singular basis matrix $E \in \mathrm{R}^{d \times d}$. The centre points of the elements are

$$c(i) = e + Ei, \quad 0 \leq i \leq n,$$

where $i \in \mathrm{Z}_+^d$. The element with multi-index i is then defined as

$$G(i) = \left\{ c(i) + E\gamma \mid \gamma \in \left(-\frac{1}{2}, \frac{1}{2}\right]^d \right\}.$$

All elements have an equal volume of $|\det E|$. It is easy to see that the elements are disjoint and cover a parallelepiped region.

2.1 Algorithm

To derive the moving grid algorithm, we approximate each step of the optimal Bayesian recursive filter with its grid counterpart. Having chosen suitable grids, we only have to compute and

update the probabilities of the state being inside each element. We assume the motion model is linear, that is $f(x) = Tx$.

1. Approximate the initial distribution $p_{0|0}$ by computing the element probabilities $\pi_{0|0}(i)$, see Figure 1a. Set $k = 1$.

2. Prediction step: propagate the grid basis with $E_k = TE_{k-1}$ and compute new element probabilities for the *prior distribution* (Figure 1b) from

$$\pi_{k|k-1}(i) = \sum_{j=0}^{n_{k-1}} \pi_{k-1|k-1}(j)\ \tau_{k|k-1}(i-j)$$

   which is fast to compute as a discrete linear convolution. Here $\tau_{k|k-1}(i-j)$ is the transition probability from jth element of $(k-1)$th grid to ith element of kth grid, and it depends on the motion model and the relative positions of the grids (see below).

3. Correction step: use the current measurements to update element probabilities as

$$\pi_{k|k}(i) = \pi_{k|k-1}(i) \int_{G_k(i)} L_k(x_k|y_k)\ dx_k.$$

   This step requires either the analytic or numerical integration of the likelihood function (Figure 1c) over each element. Because the integrands are often discontinuous, the quasi Monte Carlo method seems best suited for this. After update, normalize the element probabilities so that they sum to 1, to form the *posterior distribution* shown in Figure 1d.

4. Output current mean and variance estimates

$$\mu_k = \sum_{i=0}^{n} \pi_{k|k}(i) c_k(i)$$

$$V_k = \sum_{i=0}^{n} \pi_{k|k}(i) c_k(i) c_k(i)^T - \mu_k \mu_k^T + \frac{1}{12} E_k E_k^T.$$

5. Increase k and repeat from 2.

2.2 Computing the transition probabilities

The general transition probability from jth element to ith element is given by

$$\Gamma_{k+1|k}(i|j) = \int_{G_{k+1}(i)} \left(\int_{G_k(j)} \phi_{k+1|k}(x \mid z)\ dz \right) dx.$$

With the linear motion model $f(x) \equiv Tx$ and that fact that the new and old grid orientations are related by $E_{k+1} = TE_k$, the transition probability can be simplified to

$$\Gamma_{k+1|k}(i|j) = \int_{[-\frac{1}{2},\frac{1}{2}]^d} p_{w_k}(E_{k+1}(i-j) + e_{k+1} - Te_k + E_{k+1}\lambda)\ \ d\lambda,$$

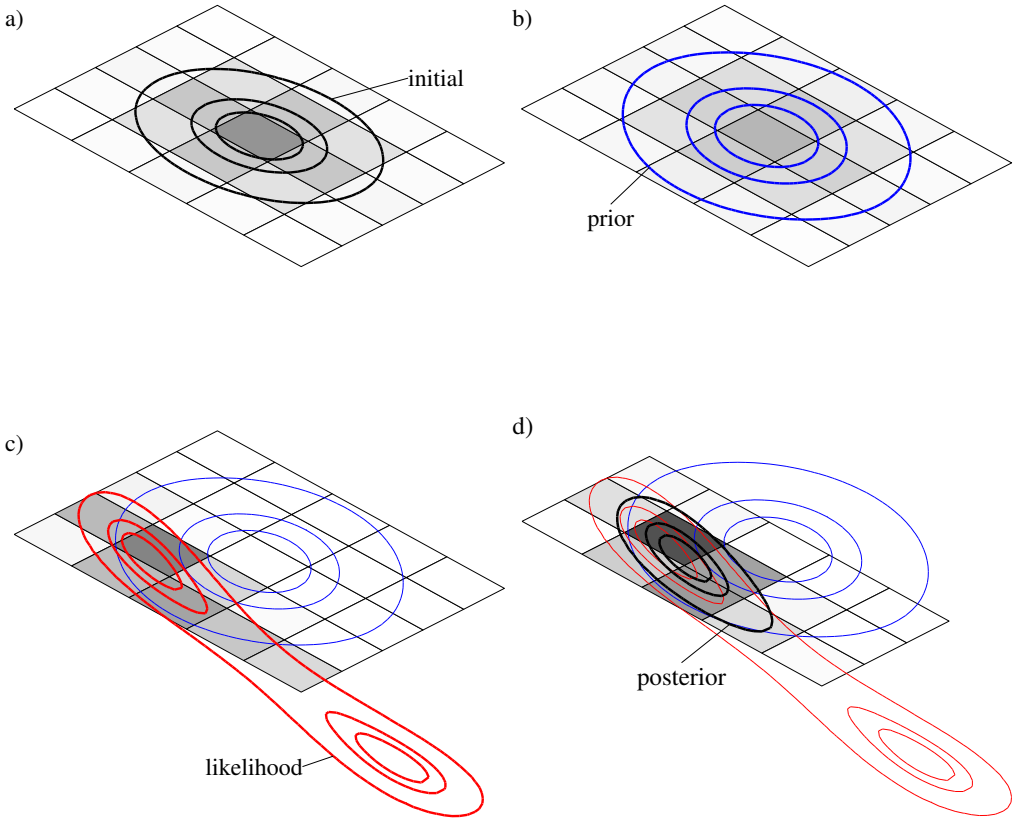


Figure 1: Steps of the moving grid algorithm, showing the ideal distributions with contour lines, and corresponding grid approximations with shaded elements.

where p_{w_k} is the density function of the noise w_k. Note that $\Gamma_{k+1|k}(i|j)$ now depends only on the difference $i - j$, not on i and j separately. Thus, we write $\tau_{k+1|k}(i - j) = \Gamma_{k+1|k}(i|j)$, and further

$$\tau_k(i - j) = \int_K p_{w,i-j}(\lambda)\, d\lambda$$

where $p_{w,i-j}$ is shorthand for the modified process noise probability density function. Specifically, if $w \sim N(0, Q)$, then

$$w_{i-j} \sim N\left(j - i - E_{k+1}^{-1}(e_{k+1} - Te_k),\ E_{k+1}^{-T} Q E_{k+1}^{-1}\right).$$

Then the transition probability for each value of $i - j$ is just multinormal probability in a hyperbox and can be computed efficiently [Genz, 1992].

2.3 Projection between grids

Because the consecutive grid bases are related by $E_{k+1} = TE_k$, the elements will eventually become "sheared" along the velocity dimension. In practice, it is best to straighten the grid by

projecting the density to a new grid every once in a while. Figure 2 illustrates the projection process. For accurate projection, the mean density in the new element is computed by taking intersection with every element in the old grid and summing the weighted densities. Accurate projection is computationally expensive in the general multidimensional case, and is still work in progress. With a large number of small elements, the projection method does not have to be very sophisticated, as long as it is asymptotically correct. With a few large elements, on the other hand, it is important to lose as little information as possible in the projection.

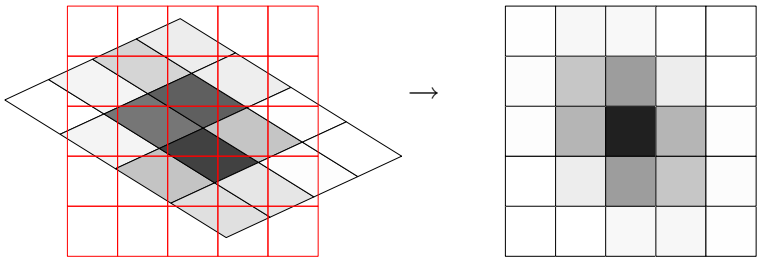


Figure 2: Straightening the grid (left) by projecting to a rectangular grid (right)

2.4 Grid design

By grid design we mean deciding what area the grid should cover and how large grid elements to use. One extreme choice is to generate massive amount of very small elements. Then the approximation will be asymptotically accurate even if the $\pi(i)$ are chosen sub-optimally, for example using just the density value in the element centre as is done in the classical point-mass filter [Bucy and Senne, 1971].

Another extreme is to use just a few very large elements. Then it is essential that the mean densities in the elements be computed as accurately as possible. Most of the structure of the pdf is however lost when large parts of the pdf are approximated with constant patches. Fortunately, it usually is not necessary to use grid spacing much denser than the expected positioning accuracy.

In the implementation it is thus possible to try to strike a balance between computation load and accuracy. Optimally, the element should not be much smaller than the finest features of the posterior distribution. There are some analytic results on how to anticipatively design 1D and 2D grids for the point-mass method [Šimandl et al., 2002], but these are yet to be generalised for the multidimensional piecewise constant grids.

3 Testing

The moving grid method was implemented in MATLAB along with a test bench for comparison between other nonlinear filters.

The simulation test bench was designed to produce dynamic test data similar to what could be expected in real-world personal positioning scenario. The main difference from the real data is that in the simulation the true track and correct measurement and motion models are available for verification of the results. The testing process consists of first generating a true track of 120 points with one second intervals with a velocity-restricted random walk model, then generating the base station along the track with maximum ranges set so that one to three stations can be heard from every point on the track. A GPS constellation is then simulated with an elevation mask and shadowing profile set so that only 1–4 satellites are visible at a time. Finally, noisy measurements are generated for each time step from the visible satellite ranges and delta-ranges, base stations ranges and sector information, and optionally also compass and altitude measurements.

Several track and measurement sets were generated with different parameters for the user motion model, available measurement sources, and measurement noises. The testing scenarios range from positioning with just one or two base stations with up to 500 m ranging errors through a hybrid case with a couple of base stations and a two-three satellites to an over-determined satellite-only case.

The test tracks were run through the moving grid filter, and the mean and covariance of the posterior distribution recorded at each time step. For comparison, the data was also processed with an extended Kalman filter (EKF) [Ali-Löytty et al., 2005], and a 2-million point "bootstrap" particle filter, which requires about hundred times more computation but is what we think the closest we can get to the optimal solution without spending months of CPU time. The particle filter solution is used as reference in the testing.

The focus of these preliminary results is on gaining insight into the problem and the behaviour of the grid solver by inspecting the results of some individual runs. Quantitative results would not make much sense at this point because, firstly, the data is simulated with arbitrarily chosen parameters so we can expect arbitrary results, and secondly, there is not yet a sound and fair way of comparing different filters either to each other or to the true/optimal solution.

Figure 3 shows an example run using just two base station range measurements. The ellipses represent the one-sigma approximation of the estimated posterior distributions. In the last part of the track (bottom left), the optimal posterior distribution is bi-modal, and the reference estimate travels somewhere in between the two. It can be seen how the EKF strayed onto the wrong branch of the posterior and gives over-optimistic variance estimates.

The 2D error of the grid filter is compared to the element radius in Figure 4. The error of the estimated mean from the reference mean, that is taken to be quite close to the optimal posterior mean, is indeed always smaller than the element radius. Thus, the grid filter works as well as is to be expected. The accuracy can be improved by either using more elements, which results in more computation, or to use a tighter significant domain, in which case there is a risk of leaving too much of the posterior density outside the grid.

4 Discussion

Personal positioning often is required although there are only a few measurement sources available that might have large errors with unusual distributions. In these cases, it is essential that the

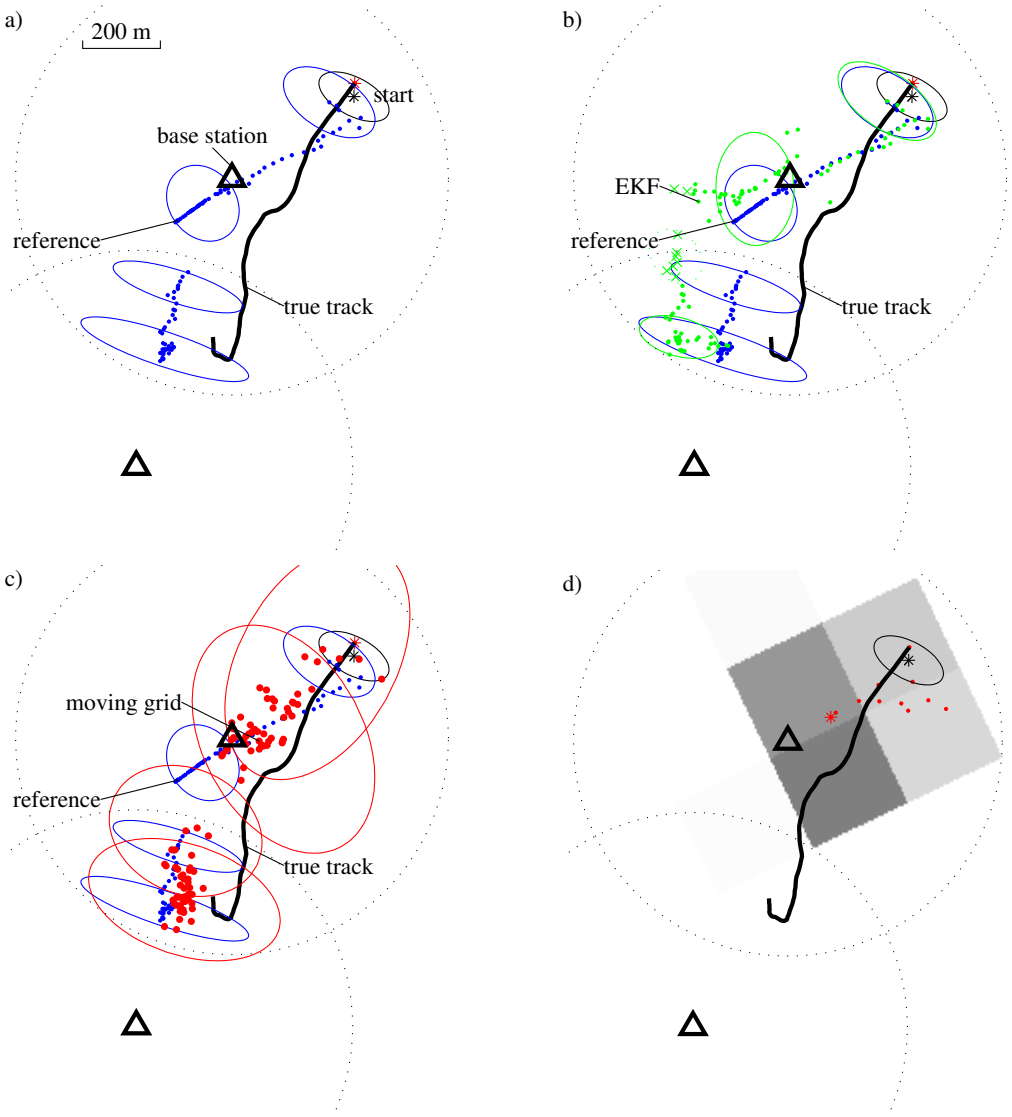


Figure 3: An example run using range measurements from two base stations. Plot a) shows the true track and the 2-million point bootstrap particle filter reference solution, plot b) presents the strayed EKF solution. Plot c) shows the moving grid solution with 200 elements, and plot d) the 2-D projection of the grid estimate at $t = 10$.

maximum amount of information be extracted from every measurement. One way of achieving this is to approximate the ideal Bayesian filter as accurately as possible.

The moving grid filter presented in this paper is a conservative approach to Bayesian filtering. A coarse grid may not be able to present the finer features of posterior density function accurately, but it should at least keep track of the shape of the posterior and not drop any its peaks. Preliminary testing of the MATLAB implementation shows that although accurate solution would require huge number of grid elements, the filter can be run with very coarse grids and it succeeds

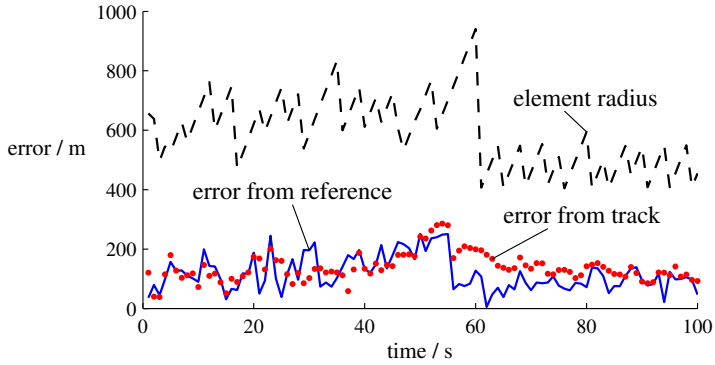


Figure 4: The 2D error of the moving grid filter mean estimate compared to the element radius

in keeping the magnitude of the error about the same as the grid element radius.

This work also underlines the question of how to evaluate the accuracy of nonlinear filters. The most common approach is to evaluate the errors between posterior mean and the true track. However, this error consists of two components: the error of the optimal posterior mean estimate from the true track, and the error of the approximated posterior mean from the optimal posterior mean.

Especially in bimodal cases, it often happens that the optimal mean estimate is in between the two peaks, and thus far away from the true track. In these cases, unimodal filters such as EKF produce posterior distributions that are clearly wrong (unimodal vs. bimodal), but their mean estimate still may be much closer to the true track than that of the optimal filter, depending on which of the peaks they happen to follow.

In this work, we compared the filtered mean not only with the true track but also with the means of the classical EKF and that of a 2-million point particle filter. Ideally, we would like to compare the approximated posterior distribution to the true posterior distribution, but as of yet no feasible numerical method of doing so is known. This is a topic for future research.

Acknowledgments

The authors would like to thank Robert Piché for his comments and corrections. This study was funded by Nokia Corporation. Both authors were supported in part by the Nokia Foundation. The particle filter we used for reference was implemented by Duane Petrovich.

References

[Ali-Löytty et al., 2005] Ali-Löytty, S., Sirola, N., and Piché, R. (2005). Consistency of three Kalman filter extensions in hybrid navigation. In *Proceedings of the European Navigation Conference GNSS 2005, July 19-22, 2005, Munchen*.

[Bergman, 2001] Bergman, N. (2001). Posterior Cramér-Rao bounds for sequential estimation. In Doucet, A., de Freitas, N., and Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York.

[Bucy and Senne, 1971] Bucy, R. S. and Senne, K. D. (1971). Digital synthesis of non-linear filters. *Automatica*, 7(3):287–298.

[Duník et al., 2005] Duník, J., Šimandl, M., Straka, O., and Král, L. (2005). Performance analysis of derivative-free filters. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005*, pages 1941–1946, Seville, Spain.

[Fox et al., 2003] Fox, D., Hightower, J., Kauz, H., Liao, L., and Patterson, D. J. (2003). Bayesian techniques for location estimation. In *Proceedings of Workshop on Location-aware Computing, part of UBICOMP Conference, Seattle, October 2003*, pages 16–18.

[Genz, 1992] Genz, A. (1992). Numerical computation of multivariate normal probabilities. *Journal of Computational and Graphical Statistics*, 1(1):141–149.

[Kotanen et al., 2003] Kotanen, A., Hännikäinen, M., Leppäkoski, H., and Hämäläinen, T. D. (2003). Experiments on local positioning with Bluetooth. In *Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC'03)*, pages 297–303.

[Kramer and Sorenson, 1988] Kramer, S. C. and Sorenson, H. W. (1988). Recursive Bayesian estimation using piece-wise constant approximations. *Automatica*, 24(6):789–801.

[McGuire et al., 2003] McGuire, M., Plataniotis, K. N., and Venetsanopoulos, A. N. (2003). Estimating position of mobile terminals from path loss measurements with survey data. *Wireless Communications and Mobile Computing*, 3:51–62.

[Roos et al., 2002] Roos, T., Myllymäki, P., and Tirri, H. (2002). A statistical modeling approach to location estimation. *IEEE Transactions on Mobile Computing*, 1(1):59–69.

[Šimandl et al., 2002] Šimandl, M., Královec, J., and Söderström, T. (2002). Anticipative grid design in point-mass approach to nonlinear state estimation. *IEEE Transactions on Automatic Control*, 47(4):699–702.

[Sirola and Ali-Löytty, 2006] Sirola, N. and Ali-Löytty, S. (2006). Local positioning with parallelepiped moving grid. In *Proceedings of 3rd Workshop on Positioning, Navigation and Communication 2006 (WPNC'06)*, pages 179–188, Hannover.

[Syrjärinne, 2001] Syrjärinne, J. (2001). *Studies of modern techniques for personal positioning*. Dissertation, Tampere University of Technology, Tampere.

[Teuber and Eissfeller, 2006] Teuber, A. and Eissfeller, B. (2006). WLAN indoor positioning based on Euclidian distances and fuzzy logic. In *Proceedings of 3rd Workshop on Positioning, Navigation and Communication 2006 (WPNC'06)*, pages 159–168.

PUBLICATION 5

Niilo Sirola, Simo Ali-Löytty and Robert Piché: Benchmarking nonlinear filters. In *Proceedings of IEEE Nonlinear Statistical Signal Processing Workshop NSSPW 2006*, Cambridge, 13–15 September 2006.

BENCHMARKING NONLINEAR FILTERS

Niilo Sirola, Simo Ali-Löytty, Robert Piché

Institute of Mathematics, Tampere University of Technology, Finland

ABSTRACT

Algorithm developers need relevant and practical criteria to evaluate and compare the performance of different discrete-time filters or filter variants. This paper discusses some pitfalls in different approaches and proposes a combination of criteria on which to base comparisons. A comparison of eight filters for a class of hybrid personal positioning problems is presented as an example.

1. INTRODUCTION

Hybrid personal positioning means sequentially estimating the three-dimensional position and velocity of a mobile terminal using various types of discrete-time measurements such as ranges to cellular base stations, satellite pseudoranges and delta ranges, and barometric altitude. Typically there are just a few measurements at each time step (not enough for a static position fix) and the posterior distribution has ridges or multiple peaks. Our research group has investigated several different kinds of nonlinear filters, including several flavours of Kalman-type filters [1, 2], mixture filters, particle filters [3], and grid-based filters [4].

In order to identify promising and relevant filtering techniques we have developed a software test bench in which all the filters can be simulated on a wide range of fairly realistic hybrid positioning scenarios. Even with the "level playing field" provided by our test bench, the question of finding fair and meaningful ways to benchmark and rank different filters still presents many conceptual and practical difficulties.

All the filters we study can be cast as numerical approximations of the ideal discrete-time Bayesian filter, which we summarize in Section 2. We therefore seek to compare the sequence of estimates computed by the approximate filter to the sequence of posterior probability distribution functions (pdf's) that would be produced by the ideal Bayesian filter. This differs from the usual practice for simulation tests in the literature which present computed measures of difference between true and estimated trajectories, from which it is impossible to distinguish the contributions of estimation error and numerical approximation.

In Sections 3–5 we discuss some issues related to devising relevant and computable numerical performance indicators based on pdf's, and we propose a combination of comparison criteria. These are applied in Section 6 to compare filters for the personal positioning problem.

This study was funded by Nokia Corporation. Additionally, Simo Ali-Löytty recognises the financial support from the Tampere Graduate School in Information Science and Engineering (TISE) and the Nokia Foundation.

2. FILTERING IN PERSONAL POSITIONING

Consider a discrete-time stochastic system whose state $x_k \in \mathbb{R}^d$ and measurements $y_k \in \mathbb{R}^{m_k}$ at time instant t_k are described by the two equations:

$$x_k = f_{k-1}(x_{k-1}) + w_{k-1} \qquad \text{(dynamic model)} \quad (1)$$

$$y_k = h_k(x_k) + v_k \qquad \text{(measurement model)} \quad (2)$$

where $f_k : \mathbb{R}^d \to \mathbb{R}^d$ is a known function, $w_k \in \mathbb{R}^d$ is the state noise with known distribution, $h_k : \mathbb{R}^d \to \mathbb{R}^{m_k}$ is the known measurement function, and $v_k \in \mathbb{R}^{m_k}$ is the measurement noise with known distribution. The noises w_k and v_k are assumed white and independent of each other and of the initial state x_0.

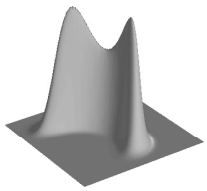
The dynamic model (1) can be used to define the state transition probability, i.e. the probability of moving from state x_{k-1} to x_k

$$\phi_k(x_k \mid x_{k-1}) = p_{w_k}\left(x_k - f_{k-1}\left(x_{k-1}\right)\right).$$

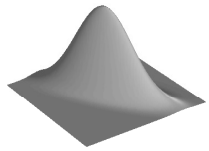
Similarly, the measurement model (2) defines the likelihood function

$$L_k(y_k \mid x_k) = p_{v_k}\left(y_k - h_k\left(x_k\right)\right).$$

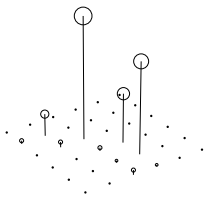
The ideal recursive Bayesian algorithm computes the probability distribution function of the state x_k at each time step conditioned on a known initial state distribution and all measurements up to that time. Denote the set of all measurements up to kth time instant $\{y_1, y_2, \ldots, y_k\}$ with the shorthand notation $y_{1:k}$.



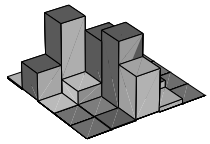
(a) True pdf



(b) Gaussian approximation



(c) Point-mass approximation



(d) Piecewise approximation

Fig. 1. A bimodal probability density in two dimensions (a) and examples of its approximations (b)-(d).

The algorithm for recursive Bayesian estimation is:

1. Start with initial distribution $p_{0|0}(x_0)$ and $k = 1$.
2. Compute the **prior pdf**:

$$p_{k|k-1}(x_k \mid y_{1:k-1}) = \int_{\mathbb{R}^d} p_{k-1|k-1}(x_{k-1} \mid y_{1:k-1}) \phi_k(x_k \mid x_{k-1}) dx_{k-1}. \quad (3)$$

3. Compute the **posterior pdf**:

$$p_{k|k}(x_k \mid y_{1:k}) \propto p_{k|k-1}(x_k \mid y_{1:k-1}) L_k(y_k \mid x_k).$$

4. Output mean and variance of $p_{k|k}(x_k \mid y_{1:k})$.
5. Increase k and repeat from 2.

3. PARAMETRIC PDF REPRESENTATIONS

The integral in Eq. (3) makes the recursive Bayesian algorithm difficult to implement in practice. In order to perform the computations in finite time and memory, the filters must store the densities in some parametric form with a finite number of parameters. Except for some special cases, it is impossible to preserve the form of the distributions exactly, as illustrated in Figure 1.

Most algorithms traditionally used for the filtering problem can be cast as approximations to the Bayesian filter. The type of pdf approximation determines the properties of the filter. Some examples of parametric forms and implied filter families are:

Gaussian – (Extended) Kalman filter and variants: second order EKF, unscented Kalman filter, etc.

Mixture of Gaussians – Gaussian mixture filter

Sample – particle filter (i.e. sequential Monte Carlo filter)

Point mass on a grid – point mass filter [5, 6]

Piecewise constant function – grid mass filter [7, 4]

4. COMPARISON CRITERIA

The ultimate goal is to compare the approximate posterior distribution given by the filter to the optimal distribution obtained with the ideal Bayesian recursive algorithm. Table 1 lists some distance functions for two pdf's f and g with support in A. Most of the distance alternatives require the computation of an integral over the region of interest, while infinity norm requires "only" finding the maximum absolute difference. None of the listed distance measures is appropriate for point-mass or particle representations.

Table 1. Probability density distances

	name	$d_A(f,g)$
1)	infinity norm	$\sup_{x \in A} \lvert f(x) - g(x) \rvert$
2)	Lissack-Fu	$\int_A \lvert f(x) - g(x) \rvert^p dx$
3)	Bhattacharyya	$-\log \int_A \sqrt{f(x) g(x)} dx$
4)	Kullback-Leibler	$\int_A f(x) \log \frac{f(x)}{g(x)} dx$
5)	Simandl [6]	$1 - \int_A \min\left(f(x), g(x)\right) dx$

It should be noted that our Bayesian interpretation of the filtering problem presents some practical and conceptual difficulties. Not all filters seek to estimate the posterior mean and variance, they may instead use, for example, the posterior mode and the estimate squared error.

Even if a filter reports a posterior mean and variance, this does not suffice to specify a general posterior pdf. Even in one dimension, symmetric pdf's having the same mean and covariance can differ significantly (Figure 2). The pdf distance measures can detect differences that are not evident from only the first two moments. For example, a biased normal distribution $\mathcal{N}(0.5, 1)$ is actually more similar to the standard normal distribution $\mathcal{N}(0, 1)$ with respect to any of the probability density measures in Table 2 than the other pdf's in Figure 2 that have the correct mean and variance.

Another example where the posterior density is not known explicitly is the particle filter. Particle filters propagate a sample from the posterior distribution, and it is not a trivial task to derive the density function from it.

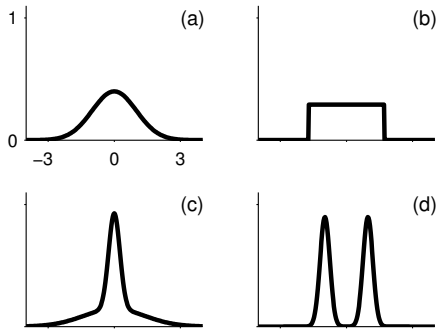


Fig. 2. Examples of symmetric distributions with mean 0 and variance 1: a) normal distribution, b) uniform between $-\sqrt{3}/2$ and $\sqrt{3}/2$, c) unimodal Gaussian mixture, d) bimodal Gaussian mixture.

Table 2. Distances of densities in Fig. 2 from $\mathcal{N}(0,1)$ with respect to distance measures 1)-5)

	(a)	(b)	(c)	(d)	$\mathcal{N}(0.5,1)$
1)	0	0.20	0.53	0.65	0.12
2)	0	0.04	0.13	0.42	0.03
3)	0	0.07	0.05	0.31	0.03
4)	0	2.8	0.19	2.7	0.11
5)	0	0.20	0.27	0.55	0.20

An alternative comparison strategy, studied in [3], is to test statistically whether samples from the approximate distribution and the reference distribution indeed come from the same distribution. This technique applies mainly to comparing two particle filters, as samples from the estimated posteriors are readily available.

The state space is partitioned into hyper-rectangular bins so that each bin contains roughly an equal fraction of the posterior probability. If the two samples under comparison are from the same distribution, they should have roughly the same expected frequency in each bin. A standard two-sample χ^2 test is then employed on each bin to check this.

The adaptive binning method could be generalized to work with the integrals of the posterior approximations over the bins. Some practical difficulties still remain.

A practical difficulty in comparing pdf's is that the reference posterior (that would be produced by the ideal Bayesian filter) is usually unavailable. The best we can do is compute an offline reference solution with, say, a particle filter with as many particles as our computational facilities can handle in reasonable time. This still leaves the practical problem of storing the reference solutions, and implementing the comparison.

5. PRACTICAL CRITERIA

Instead of comparing the full distributions, the following criteria are easier to compute in the test bench, and cover different aspects of the filter performance:

Error from true track is simple to calculate in pure simulation tests where the true track is known. We compute the mean error, RMS error, and 95% percentile.

When comparing different solvers, note that this error consists of the error inherent to the data (caused by measurement errors and insufficient geometry) plus error caused by the approximations made by the filter. These two error types might even cancel each other out.

Circular error percentiles describe how often the estimate is within a predetermined distance from the true state. The smaller the distance, the more this criterion favours filters that seek the posterior mode instead of mean.

Consistency measures the conservativeness of covariance estimates. According to the general inconsistency test [1]

$$P\left(\|C_k^{-\frac{1}{2}}(x_k - \hat{x}_k)\| \geq \sqrt{\frac{d}{\alpha}}\right) \leq \alpha,$$

where $\hat{x}_k$ and C_k are respectively the state and covariance estimates d is the dimension of the state, and α is the risk level of the test. The more estimates during the simulation satisfy the inequality, the more inconsistent the solver is.

Error from reference is the distance to a pre-computed "optimal" mean estimate conditioned on the measurements.

Note that this requires storing only the point estimates of the reference solution, not the whole particle swarms. If the reference is sufficiently close to the optimal solution, this criterion is equal to the error caused by the approximation process and is not contaminated with the problem-dependent noise. Alas, in some situations the solution apparently does not converge after even increasing the number of particles to ten million.

Computation time is just the time per fix in our MATLAB implementation, and obviously depends on the amount of skill and effort put in the implementation of the algorithms. The order of magnitude still gives some indication of the solver's performance.

As there are multiple criteria, it is difficult to put different solvers into order, as naturally some fare better with respect to one criterion and some to the other. We call a solver **Pareto optimal** with respect to a set of criteria if there is no solver that would improve one or more criteria without weakening any of the others. Conversely, a solver will not be Pareto optimal if some other solver is at least as good in all criteria. The

concept of Pareto optimality will not help in deciding which solver is the "best", but it indicates which solvers certainly are not.

6. APPLICATION: PERSONAL POSITIONING

For the test bench, we generated 500 different tracks, each consisting of 120 time instants at one-second intervals, and measurements generated for each time instant. For simplicity, the motion model is linear Gaussian and the measurement model consists of linear and nonlinear measurements with additive Gaussian noise, and additionally restrictive information, such as maximum ranges and mobile cell sector bounds. The resulting optimal posteriors contain "banana" and "donut" shaped densities, bimodal densities, almost degenerate distributions (very large variance in some direction and small in others) and sharp edges.

We chose to generate only one set of measurements per track, and to run each track only once. As there are 500 different tracks, the statistics computed from the whole bench have some meaning, but there is still the possibility to examine individual cases qualitatively.

Figure 3 presents an example of the output, plotting the mean error from the reference track of each solver against the relative computation time. There is a distinct trade-off between computation time and performance. Solvers towards the bottom left of the figure achieve the best results with the least computation.

7. CONCLUSIONS

Numerical comparison of nonlinear filters remains an essential although maybe an overlooked problem. We have laid some theoretical framework in which to perform the comparison, but we fear we have raised more questions than answers, and found more pitfalls than solutions. The individual comparison criteria proposed in Section 5 are not themselves novel, but we believe that a systematic use a diverse set of criteria gives a more complete picture of the strengths and weaknesses of different solvers, as well as the hardness of different filtering problems.

Several open questions remain, ranging from the issue of balancing the test bank so that it does not favour any particular type of solver, to the validity of the Bayesian approach itself [8]. Future work also includes taking model mismatch and blunder measurements into account.

8. REFERENCES

[1] S. Ali-Löytty, N. Sirola, and R. Piché, "Consistency of three Kalman filter extensions in hybrid navigation," *European Journal of Navigation*, vol. 4, no. 1, 2006.

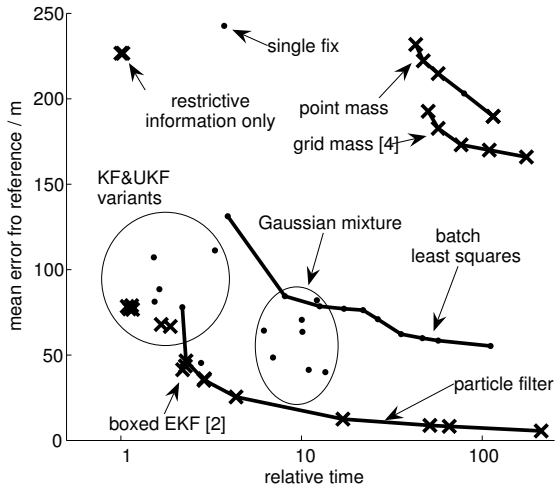


Fig. 3. Solver performance versus computation time. Crosses mark Pareto optimal solvers. Particle, point mass, and grid mass filters have been run with different numbers of particles/elements, batch least squares filter has been run with various window lengths.

[2] S. Ali-Löytty and N. Sirola, "A modified Kalman filter for hybrid positioning," to appear in *Proceedings of ION GNSS 2006*, September 2006, (accepted).

[3] D. Petrovich and R. Piché, "A comparison of particle filters for personal positioning," in *VI Hotine-Marussi Symposium of Theoretical and Computational Geodesy*, Wuhan, China, May 29 – June 2, 2006.

[4] N. Sirola and S. Ali-Löytty, "Local positioning with parallelepiped moving grid," in *Proceedings of 3rd Workshop on Positioning, Navigation and Communication 2006 (WPNC'06)*, Hannover, March 16th 2006, pp. 179–188.

[5] R. S. Bucy and K. D. Senne, "Digital synthesis of nonlinear filters," *Automatica*, vol. 7, no. 3, pp. 287–298, 1971.

[6] M. Šimandl, J. Královec, and T. Söderström, "Advanced point-mass method for nonlinear state estimation," *Automatica*, vol. 42, pp. 1133–1145, 2006.

[7] S. C. Kramer and H. W. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximations," *Automatica*, vol. 24, no. 6, pp. 789–801, 1988.

[8] P. Xu, "Biases and accuracy of, and an alternative to, discrete nonlinear filters," *Journal of Geodesy*, vol. 73, pp. 35–46, 1999.

PUBLICATION 6

Niilo Sirola: Exhaustive global grid search in computing receiver position from modular satellite range measurements. *Journal of Physics: Conference series*, 52:73–82, 2006.

Available online at `http://stacks.iop.org/conf/52/73`
Journal's home page: `http://www.iop.org/journals/conf`

Institute of Physics Publishing
doi:10.1088/1742-6596/52/1/007
Journal of Physics: Conference Series **52** (2006) 73–82
Mathematics for Industry in Denmark

Exhaustive global grid search in computing receiver position from modular satellite range measurements

Niilo Sirola

Mathematics, Tampere University of Technology, PO Box 553, FI-33101 Tampere, Finland

E-mail: niilo.sirola@tut.fi

Abstract. A global grid search algorithm with an application in weak signal satellite positioning is implemented and tested numerically. The algorithm consists of local Gauss-Newton search and a global starting point chooser, and it can be generalized as a global optimization method for functions with an attraction basin of computable minimum radius. The method is shown to find the global minimum in a bounded region in predictable time.

1. Introduction

The Global Positioning System (GPS) is a satellite positioning system that was built for military purposes, but can be used for free by anyone with a suitable receiver [1]. Currently, the number of civil and commercial GPS users is growing faster than ever.

The GPS position is computed as a solution to a nonlinear optimization problem. The basic GPS algorithms have been used for decades and found to perform well. However, with GPS receivers becoming more common, there is a growing demand for precise positioning in environments very different to those for which the GPS system was originally designed. While the satellites and the signals they are broadcasting have remained essentially unchanged for the last three decades, the continuing development of receivers and navigation algorithms has enabled significant improvements in accuracy and availability.

Due to the low power of the satellite signals, the GPS receiver needs a direct line of sight with several satellites, which is easy in the air, open seas, and deserts but often impossible indoors, near tall buildings or under trees. The receiver has to decode several layers of data from the signal, which is not always possible or practical when the signal is attenuated and noisy [2].

In this paper, we study a case where only a part of the necessary information for GPS positioning is available, and the problem becomes a mixed integer nonlinear optimization problem. We eliminate the integer variables and devise a global optimization algorithm to solve the problem.

2. Background

Each GPS satellite transmits a signal that provides a means to determine the position of the satellite and the distance between the satellite and the receiver. The position of the receiver in n dimensions can be computed from n distance measurements.

We model the GPS positioning problem as the equation system

$$\rho_i = \|\mathbf{s}_i - \mathbf{r}\|_2 + \beta, \quad i = 1 \dots n \tag{1}$$

where $\rho_i \in \mathbb{R}$ is called the pseudo-range. Pseudo-range is equal to the Euclidean distance between the ith satellite $\mathbf{s}_i \in \mathbb{R}^3$ and the receiver $\mathbf{r} \in \mathbb{R}^3$ plus an additive bias β that is common to all the pseudo-ranges. The bias results from the measurement process and is treated as the fourth unknown of the positioning problem. This model has been simplified by ignoring various errors and corresponding corrections that must be considered in actual GPS computations. These include the signal travelling in the atmosphere, Earth rotation, instability of satellite and receiver clocks, special relativity, and satellite orbit prediction errors, and their corrections. For a more comprehensive treatment on the GPS, see [1] for example.

Let the pseudoranges be measured such that $\hat{\rho}_i = \rho_i + \varepsilon_i$, where ε_i are independent measurement errors with zero mean. Given four pseudo-range measurements and the corresponding satellite positions, we can solve the system (1) and obtain estimates for the receiver position and bias. When $n > 4$, the system does not necessarily have an exact solution and a least-squares estimator is needed. Several closed-form [3, 4] as well as iterative [5, 6] estimators have been published for this problem.

The existence and uniqueness of the solution have been studied through geometric interpretation, and it has been found that in case $n = 4$, there may be two distinct solutions or no solutions at all [7]. For overdetermined cases, it is noted that a solution always exists in least squares sense and is unique "in all but degenerate cases", but the cost function is often bimodal, the other minimum fortunately always lying somewhere in outer space.

There are no convergence proofs published for the iterative methods, at least known to the author. The typically used Gauss-Newton iteration [8] seems to converge to the correct solution from practically any starting point on Earth, and it is widely agreed among GPS practitioners that the lack of convergence proofs is no problem.

3. Problem formulation

In this paper, we consider the case where the pseudo-ranges are measured modulo a constant Λ (≈ 300 km), and the satellite positions are known as functions of time but the exact time is not known. This corresponds to the situation where the satellite signals are very weak or are observed only for a short period [9].

We may write the pseudo-ranges in terms of whole and fractional multiples of Λ, such that $\rho_i = \Lambda N_i + \phi_i$ where N_i is a positive integer and $-\frac{1}{2}\Lambda \leq \phi_i < \frac{1}{2}\Lambda$. Now the system (1) becomes

$$N_i\Lambda + \phi_i = \|\mathbf{s}_i(t) - \mathbf{r}\|_2 + \beta, \quad i = 1 \dots n \tag{2}$$

where $\mathbf{s}_i(t) \in \mathbb{R}^3$ is the position of ith satellite as the function of time. The functions $\mathbf{s}_i$ are assumed known and the code phases ϕ_i can be measured. The magnitude of the measurement errors is negligible when compared to the other quantities involved, and we develop the theory assuming errorless measurements.

The system (2) has unknowns t, $\mathbf{r}$, β and additionally the integers N_i, which we eliminate from the system rather than solving them explicitly. Define the lambda-fraction operator as

$$\mathrm{frac}_\Lambda(x) = x - \Lambda \cdot \mathrm{round}(\frac{x}{\Lambda}) \tag{3}$$

where round is the usual rounding to the nearest integer with half-integers rounded up.

Now denote $\mathbf{x} = [t \;\; \mathbf{r}^T \;\; \beta]^T \in \mathbb{R}^5$ and assume $n \geq 5$. Let $\mathbf{x}_\mathrm{t}$ be the true solution.

The system

$$\mathbf{q}(\mathbf{x}) = \begin{bmatrix} q_1(\mathbf{x}) \\ \vdots \\ q_n(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \mathrm{frac}_\Lambda(\|\mathbf{s}_1(t) - \mathbf{r}\|_2 + \beta - \phi_1) \\ \vdots \\ \mathrm{frac}_\Lambda(\|\mathbf{s}_n(t) - \mathbf{r}\|_2 + \beta - \phi_n) \end{bmatrix} = \mathbf{0}. \tag{4}$$

is equivalent to the system (2), and the minimizer of $\min \frac{1}{2}\|\mathbf{q}(\mathbf{x})\|_2^2$ is the best estimate for $\mathbf{x}_\mathrm{t}$.

A global optimization method is called hybrid if it has a global stage to find a coarse approximation to the global minimum and a local search that is guaranteed to converge quickly when started sufficiently close to the optimum [10].

If the local search has an attraction basin with known minimum radius, the global search then only has to find at least one feasible point that is inside the attraction basin and then start the local search to yield the global minimum. This idea is the basis for the algorithm we propose to solve the positioning problem.

4. Local search

We seek the approximate solution for $\mathbf{q}(\mathbf{x}) = \mathbf{0}$ with the Gauss-Newton iteration [8, Section 8.5]

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \left[J(\mathbf{x}_k)^T J(\mathbf{x}_k)\right]^{-1} J(\mathbf{x}_k)^T \mathbf{q}(\mathbf{x}_k) \tag{5}$$

where the Jacobian of $\mathbf{q}(\mathbf{x})$ is

$$J(\mathbf{x}) = \begin{bmatrix} \mathbf{u}_1^T \mathbf{s}_1'(t) & -\mathbf{u}_1^T & 1 \\ & \vdots & \\ \mathbf{u}_n^T \mathbf{s}_n'(t) & -\mathbf{u}_n^T & 1 \end{bmatrix}, \quad \mathbf{u}_i = \frac{\mathbf{s}_i(t) - \mathbf{r}}{\|\mathbf{s}_i(t) - \mathbf{r}\|_2}. \tag{6}$$

The iteration is terminated when $k > k_{\max}$ or $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 < \epsilon$, where $k_{\max}$ and ϵ are some predetermined values.

This iteration converges given a starting point $\mathbf{x}_0$ sufficiently close to the minimum when $\mathbf{q}''(\mathbf{x})$ is sufficiently small [11, Chapter 6]. The solution minimizes $\|\mathbf{q}(\mathbf{x})\|_2$. The rate of convergence, when convergence occurs, is no worse than linear.

Definition 1 *Let $\mathbf{x}^*$ be a solution to $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. Let $B(\mathbf{x}^*, r) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{x}^*\| < r\}$. If the Gauss-Newton iteration converges to $\mathbf{x}^*$ from any starting point inside an open ball $B(\mathbf{x}^*, r)$, then this ball is an **attraction basin** of $\mathbf{x}^*$.*

For the unknown vector $\mathbf{x} = [t \quad \mathbf{r}^T \quad \beta]^T \in \mathbb{R}^5$ in the system (4), we find it advantageous to use the following norm instead of the Euclidean one.

Definition 2 *Define κ-**norm** as*

$$\|\mathbf{x}\|_\kappa = \kappa\,|t| + \|\mathbf{r}\|_2 + |\beta| \tag{7}$$

where κ is a positive constant dependent on the relative geometry between the user and the satellite system:

$$\kappa = \max_{i,t,\mathbf{r}} \left| \mathbf{s}_i'(t)^T \frac{\mathbf{s}_i(t) - \mathbf{r}}{\|\mathbf{s}_i(t) - \mathbf{r}\|_2} \right| \tag{8}$$

Note that in the general case $\kappa = \max \|\mathbf{s}'(t)\|_2$. When the receiver is terrestrial, however, the velocities of the satellites are almost perpendicular to the line-of-sight vector $\mathbf{s}_i(t) - \mathbf{r}$ and κ is thus much smaller than the maximum velocity of the satellite. For example, for the GPS system and user restricted on the surface of the Earth, $\kappa \approx 830$ m/s.

We denote the open ball B_κ with

$$B_\kappa(\mathbf{p}, r) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{p}\|_\kappa < r\}. \tag{9}$$

Next, we establish some properties of the system (4).

Lemma 1 *If* $\mathbf{x} \in B_\kappa(\mathbf{x}_\mathrm{t}, \frac{\Lambda}{2})$, *then*

$$|q_i(\mathbf{x})| \leq \|\mathbf{x} - \mathbf{x}_\mathrm{t}\|_\kappa \tag{10}$$

Proof. Denote $\mathbf{x} = \mathbf{x}_\mathrm{t} + \Delta\mathbf{x}$. Then consider the Taylor expansion

$$q_i(\mathbf{x}_\mathrm{t} + \Delta\mathbf{x}) = \underbrace{q_i(\mathbf{x}_\mathrm{t})}_{=0} + q_i'(\mathbf{x}_\mathrm{t} + \lambda\Delta\mathbf{x})\Delta\mathbf{x} \tag{11}$$

for some $\lambda \in (0, 1)$. From

$$q_i'(\mathbf{x}) = \begin{bmatrix} \dfrac{\mathbf{s}_i(t) - \mathbf{r}}{\|\mathbf{s}_i(t) - \mathbf{r}\|_2}^T \mathbf{s}_i'(t) & -\dfrac{\mathbf{s}_i(t) - \mathbf{r}}{\|\mathbf{s}_i(t) - \mathbf{r}\|_2}^T & 1 \end{bmatrix} \tag{12}$$

we notice that $q_i'(\mathbf{x}_\mathrm{t} + \lambda\Delta\mathbf{x})$ is also of the form $[\mathbf{y}^T\mathbf{s}_i'(\tau)) \quad -\mathbf{y}^T \quad 1]$ where $\mathbf{y}$ is a unit vector. Thus

$$\begin{aligned} |q_i(\mathbf{x})| = |q_i(\mathbf{x}_\mathrm{t} + \Delta\mathbf{x})| &= |q_i'(\mathbf{x} + \lambda\Delta\mathbf{x})\Delta\mathbf{x}| \\ &\leq \left|\mathbf{y}^T\mathbf{s}_i'(\tau)\Delta t\right| + \left|\mathbf{y}^T\Delta\mathbf{r}\right| + |\Delta\beta| \\ &\leq \kappa\,|\Delta t| + \|\Delta\mathbf{r}\|_2 + |\Delta\beta| = \|\mathbf{x} - \mathbf{x}_\mathrm{t}\|_\kappa \end{aligned} \tag{13}$$

because, by definition (8), $\kappa \geq \left|\mathbf{y}^T\mathbf{s}_i'(\tau)\right|$. ■

Lemma 2 *The least-squares fit function* $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{q}(\mathbf{x})\|_2^2$ *is smooth and has a unique minimum in* $\Omega = B_\kappa(\mathbf{x}_\mathrm{t}, \frac{\Lambda}{2})$ *if* $\mathbf{q}'(\mathbf{x})$ *has full rank and* $\|(\mathbf{q}'(\mathbf{x})^T\mathbf{q}'(\mathbf{x}))^{-1}\sum_{i=1}^n q_i(\mathbf{x})q_i''(\mathbf{x})\| < 1$ *for all* $\mathbf{x} \in \Omega$.

Proof. If f is convex and has a critical point $\mathbf{x}^*$ in convex region Ω, then $\mathbf{x}^*$ is a global minimum of f. If f is strictly convex, the minimum is unique. [11, Theorem 9.4.1]

(i) The function f is smooth in Ω if $\mathbf{q}(\mathbf{x})$ is. The Euclidean norm as well as the satellite orbit functions are smooth, and lambda-fraction is smooth everywhere except when $|q_i(\mathbf{x})| = \Lambda/2$. But, when $\mathbf{x} \in \Omega = B_\kappa(\mathbf{x}_\mathrm{t}, \frac{\Lambda}{2})$, then $|q_i(\mathbf{x})| < \Lambda/2$ by Lemma 1, and thus $\mathbf{q}$ is smooth in Ω.

(ii) Function f is strictly convex in Ω if f'' exists and is positive definite in Ω [8, 3.4.6].

$$f''(\mathbf{x}) = \mathbf{q}'(\mathbf{x})^T\mathbf{q}'(\mathbf{x}) + \sum_{i=1}^n q_i(\mathbf{x})q_i''(\mathbf{x}) \tag{14}$$

where the second term is very small. The first term is positive definite when $\mathbf{q}'(\mathbf{x})$ has full rank. Then $f''(\mathbf{x})$ is non-singular because $\|(\mathbf{q}'(\mathbf{x})^T\mathbf{q}'(\mathbf{x}))^{-1}\sum_{i=1}^n q_i(\mathbf{x})q_i''(\mathbf{x})\| < 1$ [12, Theorem 2.3.4]. Moreover, since the second term is symmetric and not large enough to reduce any of the eigenvalues of $f''(\mathbf{x})$ to zero, it cannot make any of them negative either, thus $f''(\mathbf{x})$ is positive definite.

(iii) The function f has a critical point $\mathbf{x}_\mathrm{t} \in \Omega$ because $f'(\mathbf{x}_\mathrm{t}) = \mathbf{q}'(\mathbf{x}_\mathrm{t})^T\mathbf{q}(\mathbf{x}_\mathrm{t}) = \mathbf{q}'(\mathbf{x}_\mathrm{t})^T\mathbf{0} = \mathbf{0}$.

(iv) The region Ω is a unit ball and thus convex. ■

Although convexity or unique minimum do not guarantee the convergence of the Gauss-Newton iteration (5), numerical testing corroborates the conjecture that Ω is an attraction basin for $\mathbf{x}_\mathrm{t}$. This is because the q_i are locally almost linear. The local search thus virtually guarantees good solution given a starting point $\mathbf{x}_0 \in \Omega$. Although there are several convergence theorems for Newton-like methods [13], none of them was found to guarantee convergence in whole Ω in this case, so we will have to settle on numerical evidence until further study.

5. Global search

We will now have to assume that the true solution $\mathbf{x}_t$ is the global minimum of f over the whole feasible region A. This is not true in the general case. Since each $q_i(\mathbf{x})$ has infinite number of roots, it is easy to construct satellite geometries where $f(\mathbf{x}) = 0$ for several $\mathbf{x}$. This happens also in the real world, but the probability is small at least when $n \geq 6$.

The role of the global stage of the algorithm is to find the feasible region $A \subset \mathbb{R}^5$, generate starting points in A, run the local search for each starting point, and determine which of the resulting local minima is the solution.

The global method could be a stochastic method such as Monte-Carlo or simulated annealing. In general case, these do not guarantee global convergence [10]. Instead, we develop a deterministic global stage feasible point finder that is guaranteed to find at least one point in an attraction basin of given size. First, define a covering.

Definition 3 *A point set P **covers** a region A with **covering radius** r if for any $\mathbf{x} \in A$ there is $\mathbf{p} \in P$ such that $\mathbf{x} \in B(\mathbf{p}, r)$.*

*The set $\{B(\mathbf{p}, r) \mid \mathbf{p} \in P\}$ is called the **covering** of A. It is called the **optimal covering** if there is no covering with smaller cardinality.*

It is relatively easy to find a covering, but the optimal coverings are known only for a few simple cases [14], and there are no general algorithms for finding the sharp covering radius of a given point set. Fortunately, we can quite easily generate a point set with covering radius at most a given r, a trivial example being a rectangular lattice with spacing $2r/\sqrt{p}$, where p is the number of dimensions.

Combining the concepts of attraction basin and covering radius gives the following convergence theorem for global optimization.

Theorem 1 *Let $f : A \subset \mathbb{R}^n \to \mathbb{R}$ have a global minimum $\mathbf{x}^*$. Let the open ball $B(\mathbf{x}^*, d)$ be an attraction basin of $\mathbf{x}^*$. Let P be a point set that covers A with covering radius smaller than d.*

Then the Gauss-Newton method converges to $\mathbf{x}^*$ from at least one point in P.

Proof. Because P has covering radius less than d in A, there exists by Definition 3 some $\mathbf{p} \in P$ such that $\|\mathbf{p} - \mathbf{x}^*\| < d$. Then $\mathbf{p} \in B(\mathbf{x}^*, d)$, which is an attraction basin for $\mathbf{x}^*$, and the Gauss-Newton iteration converges by Definition 1. ■

Note that if A is bounded, then P has a finite number of points, and the global minimum is found by starting the local search from each point and choosing the smallest of the resulting minima. This requires a total of at most $k_{\max} \cdot |P|$ Gauss-Newton iterations (5) and the basic algorithm is as follows.

Algorithm 0:

generate P that covers the feasible region A with radius smaller than d
for each $\mathbf{x} \in P$
  run the local search with starting point $\mathbf{x}$
  if the search converged, add the obtained minimum $\mathbf{x}^*$ to solution list X
end for
the global optimum is the local minimum with the smallest cost function value $f(\mathbf{x}^*)$

Returning to our particular problem, the feasible region A has several natural constraints. The system time t is restricted within the time of validity of the satellite orbit functions, which is about four hours. The receiver position $\mathbf{r}$ lies on or near the Earth surface, between some predetermined minimum and maximum altitudes. This forms a spherical shell. Finally, the bias can be restricted between $-\Lambda/2$ and $\Lambda/2$. Thus, the feasible region A is bounded and Algorithm 0 can be used.

We arrange the search grid as a Cartesian product of the time grid T, position grid R and bias grid B. The most natural time and bias grids are evenly spaced with ΔT and ΔB as the differences between elements. We construct the position search grid on the surface of the Earth analogous to a rectangular grid with spacing ΔR.

Taking into account that receiver position may be up to, say, 8 kilometers above the Earth, the point set $P = T \times R \times B$ then covers the feasible region A with covering radius at most $d = \frac{1}{2}\|[\Delta T\ \Delta R\ \Delta R\ 8\text{km}\ \Delta B]\|_\kappa$. This is not an optimal covering but fast to generate. The grid spacing parameters ΔT, ΔR, and ΔB can be freely chosen as long as $d < \Lambda/2$.

When using, for example, $\Delta T = 120$ s, $\Delta R = 100$ km, and $\Delta B = 60$ km, we get about 2.5 million points in P. Set $k_{\max} = 8$. Since one iteration of (2) takes about one millisecond (on a 800MHz Pentium III), the whole search would last less than six hours.

The computation can be made faster by restricting the feasible region and speeding the local search. In this paper we concentrate on restricting the feasible region.

Lemma 3 *If a terrestrial receiver sees satellites* $\mathbf{s}_1 \dots \mathbf{s}_n$ *simultaneously, then the receiver position* $\mathbf{r}$ *and time* t *must satisfy the visibility conditions*

$$\mathbf{s}_i(t)^T \mathbf{r} \geq r_{\text{Earth}}^2, \quad i = 1 \dots n \tag{15}$$

Proof. For a satellite $\mathbf{s}_i$ to lie over the horizon when viewed from point $\mathbf{r}$ on Earth, the line-of-sight vector $(\mathbf{s}_i - \mathbf{r})$ must lie on or above the local horizontal plane. The horizontal plane has normal vector parallel to $\mathbf{r}$, so the line-of-sight vector must satisfy $0 \leq \mathbf{r}^T(\mathbf{s}_i - \mathbf{r}) = \mathbf{r}^T\mathbf{s}_i - \|\mathbf{r}\|_2^2$. ∎

The visibility condition typically excludes over 90 percent of the search space. The more satellites, the smaller is the visible area on average.

Rather than checking each of the 2.5 million points against the visibility condition, it is faster to compute the visibility region explicitly at each t and cover only the visible area of Earth with R.

Furthermore, if we construct the search grid with covering radius d considerably smaller than $\Lambda/2$, we may reject the starting points with too large residuals because, by Lemma 1, $|q_i(\mathbf{x})| \leq \|\mathbf{x} - \mathbf{x}_{\text{t}}\|_\kappa < d$. These ideas give the following improved algorithm.

Algorithm 1:

```
choose a radius d and generate a search grid T × R × B with covering radius at most d
for each t ∈ T
   if the visible region V(t) is empty, proceed to next t
   generate R_t that covers V(t)
   for each r ∈ R_t and b ∈ B
      if for any i, |q_i(x)| ≥ d, proceed to next point
      run the local search with starting point x
      if the search did not converge in k_max iterations, proceed to next point
      if the altitude of the solution x* is out of bounds, proceed to next point
      add x* to solution list X
   end for
end for
if X is nonempty, choose the solution with smallest cost function value f(x)
```

How should we choose ΔT, ΔR, and ΔB for maximum efficiency? Large values minimize the number of points to search grid, but on the other hand the local search has to be started from all points. Choosing very small values produces very dense search grid, but most of the points can be rejected based on large residuals without having to start the local search. The

optimal choice lies somewhere between the extremities and depends on the relative efficiently of the different parts of the implementation. One possibility is to use heuristic optimization with the computation time of a fixed set of positioning cases as the cost function to find the optimal parameters.

6. Numerical testing

Because of the lack of a theoretical proof of $\Omega = B_\kappa(\mathbf{x}_\mathrm{t}, \frac{\Lambda}{2})$ being an attraction basin of the true solution, it is important to verify this assumption at least numerically. The model and algorithms were implemented in Matlab. Using actual GPS orbit data for satellite tracks, a million sets of pseudo-range measurements to 5–11 satellites were generated for randomly chosen true solutions. For each true solution, a random starting point was generated such that about half of the starting points lay inside the assumed basin of attraction Ω and about half outside, and the position estimate computed with the iteration (5).

The results, as listed in Table 1, show that the search converged from all starting points in Ω. Convergence occurred also from some points outside Ω, even as far as 2Λ away from the minimum. Inside Ω it took a maximum of five iterations for the search to converge and one iteration took less than one millisecond of computation time.

Table 1. Convergence tests from inside and outside attraction basin Ω

	$\mathbf{x}_0 \in \Omega$	$\mathbf{x}_0 \notin \Omega$
converged to $\mathbf{x}_\mathrm{t}$	543686	155918
converged to other minimum	0	277814
diverged	0	22582

A similar simulation was also run with Gaussian noise added to the measurements. The noise had one sigma of 15 meters, which is a typical value for a GPS receiver in poor signal conditions. Although these errors are small when compared with the other quantities in the equations, they can cause very large errors in the position solution when the satellite geometry is near degenerate. Therefore an additional test for the condition of the approximate Hessian $J^T J$ (see Eq. 5) was added and iterations finishing with an ill-conditioned Hessian were considered divergent. As seen in Table 2, bad geometries happen also inside the basin of attraction, but still no false solutions are found.

Table 2. Convergence tests with noisy measurements

	$\mathbf{x}_0 \in \Omega$	$\mathbf{x}_0 \notin \Omega$
converged to $\mathbf{x}_\mathrm{t}$	519050	142469
converged to other minimum	0	276288
diverged	24300	37893

The global algorithm could not be tested as extensively because of much longer computation times. Setting $\Delta T = 60$ s, $\Delta R = 80$ km, and $\Delta B = 33$ km reduced the execution time to about half when compared to the initial choices, but not much more tweaking was done on the Matlab

code itself. However, the computation time could be drastically reduced by implementing the inner loops with a compiled language instead of Matlab. Preliminary results of this approach have been presented in a conference paper [15].

When there is no initial information about time t other than the four-hour period, a single position fix takes an average of 10 minutes to compute with Algorithm 1. The couple of hundred position fixes computed indicate that the correct solutions are found when $n \geq 7$, and most of the time when $n = 6$.

When an approximate time t_0 is known within one minute of the true time, the search is very fast because the solver can reduce the time search grid T to just one point. The simulation consisted of 10000 position fixes from random coordinates, of which about 93% produced the correct position estimate. As expected, all of the failed searches occurred when using six satellites. In average, one search took only a few seconds, the worst-case computation time being 31 seconds for six satellites and 4.8 seconds for eleven.

7. A numerical example

For a practical example with real-world data, we took a set of actual GPS measurements and chopped them to modulo Λ to get the measurement

$$\phi = [-99575 \;\; 124457 \;\; 3316 \;\; 106688 \;\; -60690 \;\; -129977 \;\; -17471 \;\; 143623]^T \text{ m}.$$

These measurements were made on June 18th 2003 at about 12:12:46 UTC (GPS week 199, GPS time 303166 s) to GPS satellites 1, 25, 13, 4, 11, 30, 7, and 20, in that order.

In addition to the modular range measurements, the input information consists of the orbit functions for the eight satellites. The orbit functions actually used in GPS positioning have sixteen parameters, but for illustrative purposes we use just a four-parameter truncation: radius of the orbit r_i, inclination of orbit I_i, longtitude of the ascending node Ω_i, and argument of perigree ω_i. The position of ith satellite at time t can then be computed from

$$\mathbf{s}_i(t) = r_i \begin{bmatrix} \cos\phi_i \cos\Omega_i - \sin\phi_i \cos I_i \sin\Omega_i \\ \cos\phi_i \sin\Omega_i + \sin\phi(t) \cos I_i \cos\Omega_i \\ \sin\phi_i \sin I_i \end{bmatrix}, \quad \phi_i = \sqrt{\frac{\mu}{r_i^3}} t + \omega_i \tag{16}$$

where $\mu = 3.986005 \cdot 10^{14} \; \frac{\text{m}^3}{\text{s}^2}$ is the Earth's universal gravitational constant. The values of the satellite orbit parameters for the eight satellites are given in Table 3.

Table 3. Example satellite orbit parameters

i	r_i / m	I_i / rad	Ω_i / rad	ω_i / rad
1	26669227	0.97473256	2.75165527	0.52532410
2	26840334	0.94188151	-2.58537518	0.65523150
3	26506596	0.97894959	2.73303135	0.43003718
4	26567035	0.96493395	0.65864692	1.45288770
5	26604895	0.91225596	0.54902289	3.05310217
6	26385686	0.94217798	-1.48942712	1.98092506
7	26614305	0.93954499	-0.45657542	2.52472498
8	26499187	0.96450580	1.67606780	2.02825701

The time of validity for the satellite orbit functions is $t \in [296010, 310410]$ s.

We proceed according to Algorithm 1 by first choosing the search grid $\Delta T = 60$ s, $\Delta R = 80$ km, and $\Delta B = 33$ km.

The four-hour time search space is divided to 240 time steps, of which 145 have a non-empty visibility region, containing a total of 2326980 points. The condition $|q_i(\mathbf{x})| < d$ holds for 77421 points, and the local search is started from each one. 68049 of the searches converge to some local minimum within eight iterations, and after pruning out the solutions at too high or too low altitude, 2739 solution candidates remain and are added into the solution list.

The solution list entry with the smallest smallest objective function is chosen as the solution, and is in this case

$$t \approx 303165.9 \text{ s}, \quad \mathbf{r} \approx [2795073 \quad 1236237 \quad 5579648] \text{ m}, \quad \beta \approx -59556 \text{ m}$$

with the objective function value $f(\mathbf{x}) \approx 682.2$.

There are 22 other solution candidates within 10^{-4} seconds and 10 cm of this one, with objective function values less than 683. This is because the dense search grid contains multiple points within the attraction basin of the true solution. The next smallest objective function value, however, is over $2 \cdot 10^8$ and the corresponding position thousands of kilometers away.

In this experiment, the true position was known to be around [2795076 1236239 5579613], and the resulting positioning error 35 meters is well within the expected GPS accuracy. The search took little less than 14 minutes.

8. Conclusions

A solution method was developed for a special case of GPS positioning problem with minimal initial information required. The proposed method is applicable also to other satellite positioning systems, such as the European Galileo, and even to terrestrial positioning systems where distances are measured by repeating codes and positioning stations are relatively far away from the receiver.

Most of the previous GPS algorithms assume that the exact time can be deciphered from the signal and satellite positions can thus be accurately determined. The concept of time recovery, where the time has to be solved along with receiver position and bias was first suggested in [9] but with the requirement of an initial point sufficiently close to the solution. The modular GPS range problem has also been addressed by several authors, but the solutions assume that an initial position estimate within 150 km ($\approx \Lambda/2$) from the true position is known [6]. The present work gives bounds for a feasible initial estimate also in terms of time and bias, and provides an algorithm for positioning that requires the time to be known only within four hours.

Acknowledgments

This work was funded by Nokia Mobile Phones. The author would like to acknowledge Jari Syrjärinne at Nokia Mobile Phones for the original problem formulation and sharing his insight to the problem, and professor Robert Piché at Tampere University of Technology for his advice.

References

[1] Kaplan E D (ed.) 1996 *Understanding GPS: principles and applications* (Norwood: Artech House)
[2] Syrjärinne J 2001 *Studies of modern techniques for personal positioning* Dissertation, Tampere University of Technology
[3] Bancroft S 1986 An algebraic solution of the GPS equations *IEEE T. Aero. Elec. Sys.* **21** 56–9
[4] Leva J L 1996 An alternative closed-form solution to the GPS pseudo-range equations *IEEE T. Aero. Elec. Sys.* **32** 1430–9
[5] ICD-GPS-2000 2000 *NAVSTAR GPS space segment/navigation user interfaces, interface control document (ICD), Rev. C (Public Release Version)*, (Fountain Valley: ARINC Research Corporation)
[6] Dailey D J and Bell B M 1996 A method for GPS positioning, *IEEE T. Aero. Elec. Sys.* **32** 1148–54

[7] Abel J S and Chaffee J W 1991 Existence and uniqueness of GPS solutions *IEEE T. Aero. Elec. Sys.* **27** 952–6
[8] Ortega J M and Rheinboldt W C 1970 *Iterative solution of nonlinear equations in several variables* (Orlando: Academic Press)
[9] Syrjärinne J 2000 Time recovery through fusion of inaccurate network timing assistance with GPS measurements *Proc. 3rd International Conference on Information Fusion*, volume II, WeD5-3 – WeD5-10
[10] Xu P 2003 A hybrid global optimization method: The multi-dimensional case *J. Comput. Appl. Math.* **155** 423–46
[11] Fletcher R 1987 *Practical methods of optimization* (Chichester: John Wiley & Sons)
[12] Golub G H and Van Loan C F 1989 *Matrix Computations* 2nd edition (The Johns Hopkins University Press)
[13] Galántai A 2000 The theory of Newton's method *J. Comput. Appl. Math.* **124** 25–44
[14] Conway J H and Sloane N J A 1993 *Sphere packings, lattices and groups* 2nd ed (New York: Springer-Verlag)
[15] Sirola N and Syrjärinne J 2002 GPS position can be computed without the navigation data *Proc. ION GPS 2002 conference*, Portland, Oregon

PUBLICATION 7

Niilo Sirola: Nonlinear filtering with piecewise probability density approximations. Research report 87, Tampere University of Technology, Institute of Mathematics, 2007.

Nonlinear Filtering with Piecewise Probability Densities

Niilo Sirola

***Abstract*— A general framework for discrete nonlinear filtering using piecewise defined approximating probability density functions is presented. The filtering algorithm is based on projecting the predictive density functions and likelihood functions to a family of parametric functions defined over bounded elements.**

Implementation examples are given for some specific approximation function families and projection criteria. A numerical example on a well-known two-dimensional radar tracking problem is presented.

I. Introduction

Consider a discrete-time stochastic system whose state $x_k \in \mathbb{R}^d$ and measurements $y_k \in \mathbb{R}^{m_k}$ at time instant t_k are given by the equations:

$$x_k = f_{k-1}(x_{k-1}) + w_{k-1} \qquad \text{(dynamic model)} \qquad (1)$$

$$y_k = h_k(x_k) + v_k \qquad \text{(measurement model)} \qquad (2)$$

where $f_k : \mathbb{R}^d \to \mathbb{R}^d$ is a known state transfer function, $w_k \in \mathbb{R}^d$ is the state noise with known distribution, $h_k : \mathbb{R}^d \to \mathbb{R}^{m_k}$ is the known measurement function, and $v_k \in \mathbb{R}^{m_k}$ is the measurement noise with known distribution. The noises w_k and v_k are assumed white and independent of each other and of the initial state distribution $p_{0|0}(x_0)$.

The object of study is to find the conditional probability density of the unknown state x_k given measurements up to time t_k. The recursive Bayesian filter, given in Algorithm 1, is the theoretical solution to this problem. In the algorithm, the dynamic model (1) is rewritten as the *state transition density* $\phi_k(x_k \mid x_{k-1}) = p_{w_k}(x_k - f_{k-1}(x_{k-1}))$, where p_{w_k} is the state noise density function. Similarly, the measurement model (2) is contained in the *measurement likelihood function* $L_k(y_k \mid x_k) = p_{v_k}(y_k - h_k(x_k))$. The set of measurements up to kth time instant has been denoted with the shorthand notation $y_{1:k} = \{y_1, y_2, \ldots, y_k\}$.

Algorithm 1 Ideal Bayesian filter

1) Start with initial distribution $p_{0|0}(x_0)$ and $k = 1$.
2) Compute the **prior density**:
$$p_{k|k-1}(x_k \mid y_{1:k-1}) = \int_{\mathbb{R}^d} \phi_k(x_k \mid x_{k-1}) p_{k-1|k-1}(x_{k-1} \mid y_{1:k-1}) \mathrm{d}x_{k-1}. \qquad (3)$$
3) Compute the **posterior density**:
$$p_{k|k}(x_k \mid y_{1:k}) \propto p_{k|k-1}(x_k \mid y_{1:k-1}) L_k(y_k \mid x_k).$$
4) Output mean and covariance of $p_{k|k}(x_k \mid y_{1:k})$.
5) Increase k and repeat from 2.

The practical difficulties with the ideal Bayesian filter are evaluating the integrals exactly and storing the intermediate probability density functions. Except for some special cases, practical implementations rely on analytic and/or numerical approximations of both the density functions and the required integrals.

The most often used approximative filter is the extended Kalman filter and its variants that use local linearization or series expansion of the measurement and dynamic models and are thus prone to errors when the models are not "almost linear". The point-mass filter introduced in [1], on the other hand, works with global approximations of the measurement model and the state distribution, and produces more reliable results in certain situations, although at prohibitive computational cost. Improvements on the point-mass filter include anticipative grid design [2], and multi-grid extensions [3], [4].

This paper generalizes on the concept of piecewise constant parallelepiped grid filter introduced in [5] and further developed by the author in [6].

II. General piecewise filter

We develop a class of nonlinear filters by approximating probability density functions with parametric functions defined piecewise over a finite collection of bounded *elements*.

Consider a single time instant and drop the subscript k for clarity. Let G be a collection of n non-overlapping bounded elements $\{G(i) \subset \mathbb{R}^d \mid i = 1, \ldots, n\}$ whose union adequately covers the "interesting region" of the state space. The simplest example of such a collection is the rectangular grid, but in general we do not require the elements to be of the same shape, regularly arranged, or even connected. With G given, the density $p(x)$ is approximated as

$$\hat{p}(x) = \sum_{i=1}^{n} \chi_{G(i)}(x) \pi(i, x) \qquad (4)$$

where $\chi_{G(i)}$ is the characteristic function of the element $G(i)$ and $\pi(i, \cdot) \in \mathcal{F}(G(i))$ is a function defined over the ith element. The approximation function family $\mathcal{F}$ could be, for example, the family of constant, linear, delta-peak, polynomial, exponential, or truncated Gaussian functions. The approximation (4) is constructed by projecting the actual density function to the approximation space. Using this method in storing the intermediate densities in the ideal Bayesian filter yields Algorithm 2. A variant given in Algorithm 3 makes separate projections on the prediction and update steps. This two-step variant will lose more detail of the density functions in projections, but in some cases allows for a more efficient implementation.

Algorithm 2 General one-step piecewise filter

1) Start with an initial distribution $\hat{p}_{0|0}$ given in the form (4). Set $k = 1$.
2) Compute the new significant domain S_k and form a new grid G_k that covers S_k.
Given the previous posterior approximation $\hat{p}_{k-1|k-1}$ and the dynamic model, the accurate expression for the new posterior density is
$$\tilde{p}_{k|k}(x) \propto L_k(x) \sum_{i=1}^{n_{k-1}} \int_{G_{k-1}(i)} \phi_{k-1}(x \mid \xi)\ \pi_{k-1|k-1}(i, \xi)\ \mathrm{d}\xi. \qquad (5)$$
Approximate this in the form (4) with the piecewise functions defined as
$$\pi_{k|k}(i, \cdot) \propto \mathrm{proj}_{\mathcal{F}(G_k(i))}\left(\tilde{p}_{k|k}(\cdot)\right).$$
Then normalize the density.
3) Compute and output the mean and covariance estimates.
4) Increase k and repeat from Step 2.

A. The significant domain

Computing integrals is the heart of any nonlinear filtering method. In some cases, it is possible to analytically solve the integrals, but the more general methods use numerical integration that usually operate over a bounded subset of the state space. The time-honored convention of modeling real-world noise with Gaussian distributions presents a problem of all probability density functions extending to infinity. This is in contrast to real processes that, in a sense, always

Algorithm 3 General two-step piecewise filter

1) Start with an initial distribution $\hat{p}_{0|0}$ given in the form (4). Set $k = 1$.
2) *(Prediction)* Compute the new significant domain S_k and form a new grid G_k that covers S_k.
   Given the previous posterior approximation $\hat{p}_{k-1|k-1}$ and the dynamic model, the accurate expression for the prior density is
   $$\tilde{p}_{k|k-1}(x) = \sum_{i=1}^{n_{k-1}} \int_{G_{k-1}(i)} \phi_{k-1}(x \mid \xi)\ \pi_{k-1|k-1}(i,\xi)\ \mathrm{d}\xi. \quad (6)$$
   Approximate this in the form (4) with the piecewise functions defined as
   $$\pi_{k|k-1}(i,\cdot) = \mathrm{proj}_{\mathcal{F}(G_k(i))}\left(\tilde{p}_{k|k-1}(\cdot)\right).$$
3) *(Update)* Compute the new posterior approximation $\pi_{k|k}(i,\cdot)$ by projecting the product of prior and likelihood functions to the approximation space:
   $$\pi_{k|k}(i,\cdot) \propto \mathrm{proj}_{\mathcal{F}(G_k(i))}\left(\pi_{k|k-1}\ (i,\cdot)\ L_k\ (\cdot)\right).$$
   Then normalize the density.
4) Compute and output the mean and covariance estimates.
5) Increase k and repeat from Step 2.

stay "on the map" and real measurement devices whose output range is usually limited.

The *significant domain* S is the region in $\mathbb{R}^d$ in which the solution probability is non-negligible [7]. The numerical solution process then only needs to cover S, which we assume to be bounded.

B. Grid design

We have not yet fixed any structure for the grid, but for an efficient implementation it is often unavoidable. Most common choices are evenly spaced rectangular or parallelepiped grids, but also more exotic structures like hexagonal or triangle mesh could be implementable.

The goal of grid design is to cover the significant domain as efficiently as possible. The approximation should be able to capture the smallest features of the probability density functions, which is accomplished either by using a grid with many small elements, or larger elements but a more intricate approximation function family.

C. Projection criteria

Piecewise filters use the conceptually simple idea of projecting density functions to the chosen piecewise approximation function family $\mathcal{F}$. Using a function distance measure $d_A(f,g)$, the projection operator can be defined as

$$\mathrm{proj}_{\mathcal{F}(A)}(g) = \arg\min_{f\in\mathcal{F}(A)} d_A(f,g)$$

so that the projection is the function from family $\mathcal{F}$ that is closest to the original function g within the region A. Table I gives some examples of density distance functions, not all of which are proper metrics. Luckily, metric properties are not needed for the projections, although they might be useful in further investigations (see [8]).

III. Prediction step under some projection criteria and approximation families

In the following, we give formulas for a couple of simple choices of distance functions and approximation families. It is a subject of further study if any other combinations yield practical filter implementations.

TABLE I
Examples of density function distance measures

name	$d_A(f,g)$
infinity norm	$\max_{x\in A}\lvert f(x)-g(x)\rvert$
Lissack-Fu	$\int_A \lvert f(x)-g(x)\rvert^p \mathrm{d}x$
Kullback-Leibler	$\int_A f(x)\log\frac{f(x)}{g(x)}\mathrm{d}x$
Simandl [4]	$1-\int_A \min(f(x),g(x))\,\mathrm{d}x$
Bhattacharyya	$-\log\int_A \sqrt{f(x)g(x)}\mathrm{d}x$
mass-matching	$\lvert\int_A f(x)\mathrm{d}x-\int_A g(x)\mathrm{d}x\rvert$
mean-matching	$\lvert\int_A f(x)x\mathrm{d}x-\int_A g(x)x\mathrm{d}x\rvert$

A. Mass-matching

Given that we are projecting probability densities in a bounded area, it would seem reasonable to require that the total probability inside the element be conserved in the projection.

Consider the prediction step (6) in the two-step filter. We find an approximation $\hat{p}_{k|k-1}$ within the chosen approximation function family so that the probability in the element is conserved. The formula for this can be found by integrating the accurate prediction $\tilde{p}_{k|k-1}$ and the approximation over the one element at a time, setting the integrals equal, and solving the parameters of the approximating function from the equation.

Piecewise constant approximation: Let $\mathcal{F}$ be the family of constant functions, so that $\pi(i,\cdot) = \pi(i)$. Then the prediction formula is found to be

$$\pi_{k|k-1}(i) = |G_k(i)|^{-1} \sum_{j=1}^{n_{k-1}} \pi_{k-1|k-1}(j)\Gamma_{k|k-1}(i,j) \quad (7)$$

where $\Gamma_{k|k-1}(i,j) = \int_{G_k(i)}\int_{G_{k-1}(j)} \phi_{k-1}(x_k,x_{k-1})\mathrm{d}x_{k-1}\mathrm{d}x_k$ is the transition probability from jth element of the old grid to the ith element in the new grid, and $|G_k(i)| = \int_{G_k(i)}\mathrm{d}x$ is the volume of the element.

Delta function approximation: Another easy choice of approximation family is the weighted point-mass functions: $\pi(i,x) = \pi(i)\delta(x-c(i))$, where $c(i)$ is the mass center of the ith element. The corresponding prediction formula is

$$\pi_{k|k-1}(i) = \sum_{j=1}^{n_{k-1}} \pi_{k-1|k-1}(j)\phi_{k|k-1}(c_{k+1}(i), c_k(j)).$$

This is equivalent to the formula used in the conventional point-mass method [1]. The difference from the piecewise constant approximation is that the state transition density is evaluated only in the element centers instead of computing the element-to-element transition densities.

Piecewise linear: As our third simplest choice, we try a linear approximation, that is, $\pi_{k|k-1}(i,x) = a_k(i)^T x + b_k(i)$. Matching element probabilities gives a condition which leaves d degrees of freedom for the choice of approximation parameters $a(i)$ and $b(i)$:

$$\begin{aligned}&|G_k(i)|\left[a_k(i)^T c_k(i) + b_k(i)\right]\\ &= \sum_{j=1}^{n_{k-1}} \left(a_{k|k-1}(j)^T \Psi_{k|k-1}(i,j) + b_{k|k-1}(j)\Gamma_{k|k-1}(i,j)\right).\end{aligned}$$

where

$$\Psi_{k|k-1}(i,j) = \int_{G_k(i)}\int_{G_{k-1}(j)} \phi_{k-1}(x_k \mid x_{k-1})\ x_{k-1}\ \mathrm{d}x_{k-1}\mathrm{d}x_k$$

can be thought of as the first moment of the element-to-element transition probability.

The other d degrees of freedom can be fixed with boundary conditions, for example. In the one-dimensional case, the boundary conditions can be used to make the approximation continuous. In higher dimensions, there are no obvious candidates for boundary conditions and the mass-matching criterion cannot be sensibly applied to piecewise linear projections.

B. Lissack-Fu distance

We repeat the computations using Lissack-Fu distance with $p = 2$ (or L2 distance). The criterion is more complex than mass-matching, requiring the density functions to match point-wise. Point mass representation cannot deal with this kind of distance.

The parameters of the approximating functions are found by minimizing the distance function, which can lead to lengthy but straightforward computations.

Piecewise constant approximation: The L2 distance of the piecewise constant approximation $\pi_{k|k-1}$ from the accurate prediction is

$$\begin{aligned} &d_{G_k(i)}(\tilde{p}_{k|k-1}, \hat{p}_{k|k-1}(i)) \\ &= \int_{G_k(i)} |\tilde{p}_{k|k-1}(x) - \pi_{k|k-1}(i)|^2 \mathrm{d}x. \end{aligned}$$

The approximation has just one free parameter, $\pi_{k|k-1}(i)$. We find the projection by symbolically differentiating the distance with respect to $\pi_{k|k-1}(i)$ and setting the derivative to zero. The resulting prediction formula is

$$\pi_{k|k-1} = |G_k(i)|^{-1} \sum_{j=1}^{n_{k-1}} \pi_{k-1|k-1}(j) \Gamma_{k-1}(i \mid j)$$

which is equivalent to the formula (7) derived from the mass-matching criterion. This does not generalize to all distance candidates, for example the Kullback-Leibler criterion will give a different approximation and thus does not conserve element probabilities.

Piecewise linear approximation: Whereas the mass-matching criterion failed to produce unique projection for the family of linear functions, L2 criterion will. The computations for finding the zeros of the derivative of $d_{G_k(i)}(\tilde{p}_{k|k-1}, \hat{p}_{k|k-1}(i))$ with respect to the parameters $a_{k|k-1}(i)$ and $b_{k|k-1}(i)$ are lengthy but straightforward. The result is that for each i, the new approximation parameters are solved from the linear system

$$\begin{aligned} &\begin{bmatrix} V_k(i) & c_k(i) \\ c_k(i)^T & 1 \end{bmatrix} \begin{bmatrix} a_{k|k-1}(i) \\ b_{k|k-1}(i) \end{bmatrix} \\ &= |G_k(i)|^{-1} \sum_{j=1}^{n_{k-1}} \begin{bmatrix} \Theta_{k|k-1}(i,j) \\ \Psi_{k|k-1}(i,j) \end{bmatrix} a_{k-1|k-1}(j) + \begin{bmatrix} \Lambda_{k|k-1}(i,j) \\ \Gamma_{k|k-1}(i,j) \end{bmatrix} b_{k-1|k-1}(j), \end{aligned} \tag{8}$$

where the functions depending only on the grid geometry and motion model are

$$\begin{aligned} c_k(i) &= |G_k(i)|^{-1} \int_{G_k(i)} x_k \, \mathrm{d}x \\ V_k(i) &= |G_k(i)|^{-1} \int_{G_k(i)} x_k x_k^T \, \mathrm{d}x \\ \Gamma_{k|k-1}(i,j) &= \int_{G_k(i)} \int_{G_{k-1}(j)} \phi_{k-1}(x_k \mid x_{k-1}) \;\; \mathrm{d}x_{k-1} \mathrm{d}x_k \\ \Lambda_{k|k-1}(i,j) &= \int_{G_k(i)} \int_{G_{k-1}(j)} \phi_{k-1}(x_k \mid x_{k-1}) \; x_k \; \mathrm{d}x_{k-1} \mathrm{d}x_k \\ \Psi_{k|k-1}(i,j) &= \int_{G_k(i)} \int_{G_{k-1}(j)} \phi_{k-1}(x_k \mid x_{k-1}) \; x_{k-1}^T \; \mathrm{d}x_{k-1} \mathrm{d}x_k \\ \Theta_{k|k-1}(i,j) &= \int_{G_k(i)} \int_{G_{k-1}(j)} \phi_{k-1}(x_k \mid x_{k-1}) \; x_k x_{k-1}^T \; \mathrm{d}x_{k-1} \mathrm{d}x_k. \end{aligned}$$

Computing all these integrals ($3n$ d-dimensional integrals and $4n^2$ $2d$-dimensional ones) to sufficient accuracy will probably not be tractable in the general multi-dimensional case.

Another problem with the piecewise linear approximation is that it may well get negative values unless a more complicated constrained minimization process is used.

C. Projections in the one-step algorithm

In the one-step method given in Algorithm 2, the function to be projected (5) depends on the measurement likelihood function $L_k(x)$:

$$\tilde{p}_{k|k}(x) = C_k L_k(x) \sum_{j=1}^{n_{k-1}} \int_{G_{k-1}(j)} \phi_{k-1}(x \mid \xi) \; \pi_{k-1|k-1}(j, \xi) \; \mathrm{d}\xi$$

where C_k is the normalization factor. Whereas the approximation function family $\mathcal{F}$ is chosen by the algorithm designer and is thus known analytically, we would prefer to work with a "black-box" likelihood. This approach prevents us from deriving simplified projection formulas as we did for the two-step algorithm, but allows for more flexible measurement likelihoods.

The projections are done by numerical minimization of $d(\tilde{p}_{k|k}, \hat{p}_{k|k})$. In the simplest case of piecewise constant functions and mass-matching criterion, we use the procedure similar to the two-step case to get the combined prediction-update step

$$\pi_{k|k}(i) = |G_k(i)|^{-1} C_k \sum_{j=1}^{n_{k-1}} \pi_{k-1|k-1}(j) \int_{G(i)} L_k(x) \gamma_{k|k-1}(x, j) \mathrm{d}x.$$

The computation of the integrals requires more effort than in the two-step version because the likelihood term is inside the integral. Additionally, we will not be able to use certain fast techniques for computing the prediction. Still, in some cases, the smaller approximation error of the one-step method outweighs the performance penalty.

IV. Piecewise constant parallelepiped grid filter

As a practical example, we develop a specific instance of the general two-step grid algorithm (Algorithm 3) that approximates density functions in piecewise constant form

$$\hat{p}(x) = |G(i)|^{-1} \sum_{i=1}^{n} \chi_{G(i)}(x) \pi(i).$$

An evenly spaced d-dimensional parallelepiped grid is described with vectors $e \in \mathbb{R}^d$, $n \in \mathbb{Z}_+^d$ and basis matrix $E \in \mathbb{R}^{d \times d}, \det E \neq 0$. The element centers are computed from

$$c(i) = e + Ei, \quad 0 \leq i < n$$

where the multi-indices $i, n \in \mathbb{Z}_+^d$, and the ith element is given as

$$G(i) = \left\{ c(i) + E\gamma \mid \gamma \in \left(-\tfrac{1}{2}, \tfrac{1}{2}\right]^d \right\}.$$

Note that rectangular grid is also covered by this definition as a special case.

A. Prediction step

As derived in (7), the prediction equation is

$$\pi_{k+1|k}(i) \approx |G(i)|^{-1} \sum_{j=1}^{n} \pi_{k|k}(j)\Gamma_{k+1|k}(i,j) \tag{9}$$

where $\Gamma_{k+1|k}(i,j)$ is the transition probability from jth element to ith element. Another choice would have been to write this in terms of the pre-image of the target element, as is done in [5].

Computing the transition probabilities $\Gamma_{k+1|k}(i,j)$ efficiently is crucial. This term does not depend (directly) on measurements or current state, but only on the state model and the topology of the grids. We aim to make the transition probability depend on only $i-j$ instead of i and j separately, so that the prediction can be computed efficiently with discrete convolution.

Thus assume a linear state model $f(x) \equiv Tx$, and that the consecutive grid bases are compatible so that $E_{k+1} = TE_k$. Then $\phi_k(x_{k+1} \mid x_k) = p_{w_k}(x_{k+1} - Tx_k)$, where p_{w_k} is the density function of the state noise. Now the element-to-element transition probability is

$$\Gamma_{k+1|k}(i,j) = \int_{G_k(j)} \int_{G_{k+1}(i)} p_{w_k}(x_{k+1} - Tx_k)\ \mathrm{d}x_{k+1}\mathrm{d}x_k$$

Make changes of variables $x_k = e_k + E_k(j+\gamma)$ and $x_{k+1} = e_{k+1} + E_{k+1}(i+\lambda)$, and use $E_{k+1} = TE_k$ to get

$$\begin{aligned} &p_{w_k}(x_{k+1} - Tx_k) \\ &= p_{w_k}(e_{k+1} - Te_k + E_{k+1}(i-j) + E_{k+1}(\lambda - \gamma)) \\ &= p_{w_{k,i-j}}(\lambda - \gamma) \end{aligned}$$

where $p_{w,i-j}$ is shorthand for the modified process noise probability density function. Denote $K = [-\frac{1}{2}, \frac{1}{2}]^d$ to get

$$\begin{aligned} \Gamma_{k+1|k}(i,j) &= \int_K \int_K p_{w_{k,i-j}}(\lambda - \gamma)\ |E_{k+1}|\mathrm{d}\lambda |E_k| d\gamma \\ &= \tau_{k+1|k}(i-j). \end{aligned} \tag{10}$$

Specifically, if $w_k \sim \mathcal{N}(0, Q_k)$, then

$$w_{k,i-j} \sim \mathcal{N}\left(j - i - E_{k+1}^{-1}(e_{k+1} - Te_k),\ (E_{k+1}^{-1})^T Q_k E_{k+1}^{-1}\right).$$

The transition probability (10) is then computed by integrating multinormal probability in a $2d$-dimensional hyper-box using, for example, [9]. Moreover, only a few small values of $i-j$ have to be considered because the probability of larger transitions is negligible.

Note that propagating the element basis matrix through $E_{k+1} = TE_k$ repeatedly may lead to very "skewed" grids, so it is in practice best to straighten the grid and use the general prediction step instead of the convolutionized one every once in a while.

B. Update step

In general, we find the likelihood within each element by numerical integration over the element rather than hard-coding any specific likelihood equations. Thus, the integral

$$L_k(i) = \int_{x_k(i)+EK} L(x, y_k)\ \mathrm{d}x \tag{11}$$

has to be computed numerically via, for example, cubature or quasi Monte Carlo integration [10]. As a special case, quasi Monte Carlo with just one point means evaluating the likelihood only in the center of each element center, which is equal to the update in the point-mass approach.

Algorithm 4 Piecewise constant two-step parallellepiped grid filter

1) Span an initial grid G_0 and approximate the initial distribution as
$$\hat{p}_{0|0}(x) = |E_0|^{-1} \sum_{i=1}^{n_0} \chi_{G(i)}(x)\pi_{0|0}(i)$$
Set $k = 1$.
2) *(Prediction)* Compute the new significant domain S_k.
(2a) If the propagated element basis $T_{k-1}E_{k-1}$ is not too skewed, use it as basis for the new grid G_k that covers S_k. The grid anchor e_k can be chosen arbitrarily. Compute the element-to-element relative transition probabilities $\tau_{k|k-1}(i-j)$ from (10) for suitable values of $i-j$, and approximate the prior density as
$$\pi_{k|k-1}(i) \approx |E_{k-1}|^{-1} \sum_{j=1}^{n_{k-1}} \pi_{k-1|k-1}(j)\tau_{k|k-1}(i-j).$$
using multi-dimensional convolution.
(2b) Else generate new grid G_k to cover S_k and approximate the prior density using the general form
$$\pi_{k|k-1}(i) \approx |E_{k-1}|^{-1} \sum_{j=1}^{n_{k-1}} \pi_{k-1|k-1}(j)\Gamma_{k|k-1}(i|j).$$
3) *(Update)* Compute the new posterior approximation $\pi_{k|k}(i,\cdot)$
$$\pi_{k|k}(i) \propto \pi_{k|k-1}(i)L_k(i),$$
where the likelihood over ith element $L_k(i)$ is computed numerically according to (11). Normalize the density so that $\sum_{i=1}^{n_k} \pi_{k|k} = 1$.
4) Output the mean and covariance of the piecewise distribution
$$\begin{aligned} \hat{x}_k &= \sum_{i=1}^{n_k} \pi_{k|k}(i)c_k(i) \\ \Sigma &= \sum_{i=1}^{n_k} \pi_{k|k}(i)c(i)c(i)^T - \hat{x}_k\hat{x}_k^T + \frac{1}{12}E_kE_k^T \end{aligned}$$
5) Increase k and repeat from Step 2.

C. Algorithm

The steps (9), (10), and (11) are combined with some logic to manage the significant domains and grid design to get Algorithm 4.

V. Numerical example

For numerical simulation, we use the simple two-dimensional passive tracking problem from [1], [2], [5], where

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 0.9 & 0 \\ 0 & 1 \end{bmatrix} x_k + w_k, \quad w_k \sim \mathcal{N}\left(0, \begin{bmatrix} 0.1 & 0.05 \\ 0.05 & 0.1 \end{bmatrix}\right) \\ y_k &= \tan^{-1} \frac{x_k^{(2)} - \sin(k)}{x_k^{(1)} - \cos(k)} + v_k, \quad v_k \sim \mathcal{N}(0, 0.1). \end{aligned}$$

with $x_0 \sim \mathcal{N}\left(\begin{bmatrix} 5 & 0 \end{bmatrix}^T, \mathrm{diag}(9,4)\right)$. Note that it is impossible to replicate the results given in [1], [5] because they do not mention their choice of x_0. On the other hand, the starting point $[27, 0]^T$ from [2] makes the measurement model almost linear and as such is not a good test for a general nonlinear filter.

A MATLAB simulation was run for a hundred cases consisting of a simulated track and measurement realizations. Each case was solved with an extended Kalman filter (EKF), Unscented Kalman Filter (UKF), sequential Monte Carlo filter (SMC) with different numbers of particles, point mass filter (PM), and a piecewise constant

grid filter (PCG) from Algorithm 4 with various element densities. The results are given in Table II.

TABLE II
SIMULATION RESULTS

	time per fix (ms)	RMSE	inconsistent	reference error (95%)	pdf quality
EKF	0.3	2.7	8 %	6.6	14
UKF	0.5	2.0	2.0 %	4.0	13
SMC_{100}	0.6	2.0	1.4 %	1.0	103
SMC_{1000}	2	1.4	0.1 %	0.3	0.27
PM_{25}	10	3.2	0 %	5.9	7.7
PM_{100}	20	2.1	0 %	3.6	3.5
PM_{1000}	90	1.4	0.1 %	0.4	0.25
PCG_{25}	20	3.4	0 %	6.4	2.8
PCG_{100}	70	1.5	0 %	1.2	0.84
PCG_{1000}	800	1.4	0.1 %	0.2	0.21

The timings are from a MATLAB 7.1 implementation running on a laptop with 1.4GHz Intel Celeron. While the computation time of the grid method is several orders larger than that of EKF and particle filters, it is still under one second per fix and thus admissible for a real-time application, especially if implemented efficiently.

The RMSE (root mean square error) figure is the error of the mean estimate against the true track from which the measurement were simulated. The inconsistency column gives the percentage of time instants the error was "too large" in comparison to the estimated covariance, i.e. $\|\Sigma^{-\frac{1}{2}}(\hat{x}-x)\| > \sqrt{\frac{n}{0.05}}$ where $\hat{x}$ and Σ are the estimated mean and covariance, $n=2$ is the dimension of the state, and 0.05 is the risk level [11]. The smaller PM and PCG filters show 0% inconsistency because their large element size causes them to over-estimate the covariance.

The last two columns of the results table relate to a reference solution computed with a particle filter with one million particles, which we consider to be acceptably close to the optimal filtering solution. The reference error column gives a radius within which 95% of the mean estimates are from the corresponding reference mean. Finally, the pdf quality column gives the mean Kullback-Leibler distance (see Table I) between the reference density function and the approximated density function on the final step of each run. The Kullback-Leibler distances were computed numerically over a rectangular grid of 50 000 elements spanning over the support of the reference density. The probability density functions of the particle filters were evaluated by substituting each particle with a small Gaussian distribution. Figure V shows examples of the probability densities obtained by different filters.

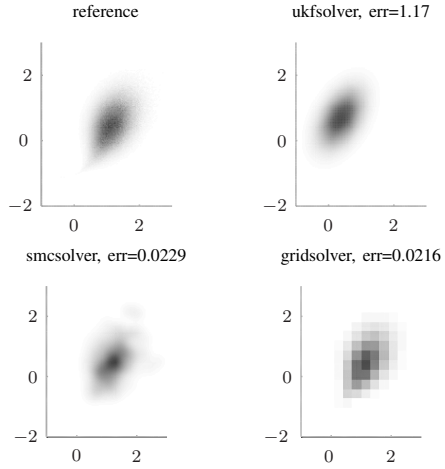


Fig. 1. Examples of density functions and their Kullback-Leibler distances from the reference density at time step $k=30$.

VI. CONCLUDING REMARKS

A piecewise nonlinear filter was given that generalizes the notion of the point-mass filter. A few specific implementation possibilities for the filter were described. The set of efficient implementations is expected to be quite small, because the more sophisticated the approximation is, the more work is required to compute the predictions and likelihood updates accurately.

Numerical testing shows that the efficiency and accuracy of a vanilla particle filter is hard to match even in a low dimension problem. The implemented point-mass and piecewise constant grid filters require several orders more computation time than a basic particle filter reaching almost the same accuracy.

The problem with particle filters is that we do not know how many particles are enough to keep the quality of the approximated posterior density good. Piecewise filters, on the other hand, can feasibly cover the whole significant region and maintain accuracy governed by the element size. The accuracy and efficiency of the piecewise algorithms can be further improved with a more advanced grid management strategy.

ACKNOWLEDGMENT

The author would like to thank Simo Ali-Löytty and Robert Piché for their comments and suggestions. The EKF used in testing were implemented by Simo Ali-Löytty, and the UKF by Tuomo Mäki-Marttunen.

REFERENCES

[1] R. S. Bucy and K. D. Senne, "Digital synthesis of non-linear filters," *Automatica*, vol. 7, no. 3, pp. 287–298, 1971.
[2] M. Šimandl, J. Královec, and T. Söderström, "Anticipative grid design in point-mass approach to nonlinear state estimation," *IEEE Transactions on Automatic Control*, vol. 47, no. 4, pp. 699–702, April 2002.
[3] Z. Cai, F. L. Gland, and H. Zhang, "An adaptive local grid refinement method for nonlinear filtering," INRIA Rennes, Research raport 2679, 1995.
[4] M. Šimandl, J. Královec, and T. Söderström, "Advanced point-mass method for nonlinear state estimation," *Automatica*, vol. 42, pp. 1133–1145, 2006.
[5] S. C. Kramer and H. W. Sorenson, "Recursive Bayesian estimation using piece-wise constant approximations," *Automatica*, vol. 24, no. 6, pp. 789–801, 1988.
[6] N. Sirola and S. Ali-Löytty, "Local positioning with parallelepiped moving grid," in *Proceedings of 3rd Workshop on Positioning, Navigation and Communication 2006 (WPNC'06)*, Hannover, March 16th 2006, pp. 179–188. [Online]. Available: http://www.wpnc.net/years/wpnc06/index06.html
[7] S. Challa and Y. Bar-Shalom, "Nonlinear filter design using Fokker-Planck-Kolmogorov probability density evolutions," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 309–315, 2000.
[8] R. Kulhavý, "Recursive nonlinear estimation: geometry of a space of posterior densities," *Automatica*, vol. 28, no. 2, pp. 313–323, 1992.
[9] A. Genz, "Numerical computation of multivariate normal probabilities," *Journal of Computational and Graphical Statistics*, vol. 1, no. 1, pp. 141–149, 1992.
[10] H. Pesonen, "Numerical integration in Bayesian positioning," M.Sc. Thesis, Tampere University of Technology, June 2006. [Online]. Available: http://math.tut.fi/posgroup/pesonen_mscth.pdf
[11] S. Ali-Löytty, N. Sirola, and R. Piché, "Consistency of three Kalman filter extensions in hybrid navigation," *European Journal of Navigation*, vol. 4, no. 1, 2006.

This thesis was typeset in LaTeX2e using the `memoir` document style. The body text is set with 12pt Adobe Utopia. All the figures were prepared with MATLAB and/or Xfig.

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O. Box 527
FIN-33101 Tampere, Finland