UWB location and tracking for wireless embedded networks

Kegen Yu\textsuperscript{a,1}, Jean-philippe Montillet\textsuperscript{a,2}, Alberto Rabbachin\textsuperscript{a}, Paul Cheong\textsuperscript{a,3}, Ian Oppermann\textsuperscript{a,b,*}

\textsuperscript{a}Centre for Wireless Communications, University of Oulu, Finland
\textsuperscript{b}Nokia Networks, Espoo, Finland

Received 2 January 2005; accepted 4 July 2005
Available online 9 March 2006

Abstract

In this paper, we investigate the performance of different position estimation methods which make use of time-of-arrival (TOA) of ultra wideband (UWB) signals for low cost/low complexity UWB systems. We first propose a simple and robust two-stage, non-coherent TOA estimation approach. We then explore positioning algorithms utilizing both non-iterative and iterative techniques. A review of positioning in distributed networks is also performed and a positioning algorithm is proposed for node location in multi-hop distributed networks. Furthermore, we consider smoothing techniques to improve accuracy when tracking moving objects and we propose the use of sinc functions to smooth the estimate of the mobile position in order to achieve both good accuracy and low complexity. The system modelled and investigated corresponds to an actual test environment in a ski field where skiers are tracked.

\textcopyright{} 2006 Published by Elsevier B.V.

Keywords: Position location/tracking; Iterative/non-iterative estimation; Distributed networks; TOA/TDOA; Sinc smoothing; UWB

1. Introduction

There are many positioning techniques using radio signals. Signal strength, angle of arrival (AOA), time measurements (time-of-arrival (TOA), round trip time (RTT), and time difference of arrival (TDOA)) can all be exploited for positioning.

The most straightforward way to estimate position is to directly solve a set of simultaneous equations \cite{1} based on the TOA/TDOA measurements. Therefore, exact solutions can be obtained for 2D positioning with three fixed nodes using three TOA measurements (with known transmit time) or with four fixed nodes using four TDOA measurements. For 3D positioning, five fixed nodes are needed to obtain exact solutions using TDOA measurements. Here we assumed one fixed node is employed to clear the problem of multiple solutions.

For an over-determined system (with redundant fixed nodes), several different approaches have been proposed such as spherical interpolation (SI) \cite{2–5},
the two-stage maximum likelihood (ML) method [6], and the linear-correction least-square approach [7]. Also several iterative approaches have been investigated for positioning. Taylor series expansion can be used to iteratively produce a linearized least-square solution [8,9]. A different iterative method for positioning comes from nonlinear optimization theory. The gradient-based algorithms can be employed for position estimation [10,11]. One is the quasi-Newton algorithm [12] which has been used in the ultra wideband (UWB) precision assets location system [13]. The other is the Gauss–Newton type Levenberg–Marquardt method [14].

Some researchers started investigation and application of UWB signals more than three decades ago [15]. Early UWB applications include position location in Radar systems [16]. Existing and potential UWB applications result from the many desirable properties of UWB signals such as the fine delay resolution which enables high-resolution positioning and tracking. Very recently UWB technology has been successfully employed in the UWB precision assets location systems [13].

The goal of the paper is to investigate/develop practical and feasible algorithms using UWB technology for positioning and tracking ASIC (application-specific integrated circuit) UWB devices. The main contributions of the paper are as follows:

- propose and evaluate a two-stage TOA estimation scheme for locating low complexity UWB devices;
- propose and evaluate a positioning algorithm for localization in distributed sensor networks;
- employ sinc function for smoothing mobile tracks.

The remainder of the paper is organized as follows. Section 2 briefly describes the positioning system and Section 3 studies TOA estimation. Section 4 presents various positioning algorithms while Section 5 discusses about moving device tracking and smoothing.

2. Brief system description

In the system, there are a limited set of fixed nodes which are accurately synchronized by sharing the same local clock such as through cable connections. The fixed nodes are positioned at known coordinates in the area to be monitored. In the monitored area, there are a certain number of UWB devices carried by mobile users.

The data transmission is packet based using a TDMA (time division multiple access) scheme. Since each user transmits data in different and pre-assigned time slots, multiple access interference is greatly reduced. Due to the drift in the clock of the mobile nodes as well as the fixed nodes, synchronization between the fixed nodes and the mobile nodes is performed once every second. This is achieved by broadcasting beacons from the fixed nodes to the mobile nodes. The TOA of the beacon is used as the reference clock for the mobile nodes to transmit data according to the pre-assigned time slots.

At any given time, the TOA measurements from a specific group of fixed nodes are collected for position estimation. Since the mobile node is moving, the group of fixed nodes will change over time. In general, the fixed nodes with the strongest received signal powers are selected to provide the TOA estimates and their position coordinates are employed for the mobile node position estimation.

More information about the system structure can be found in [17] where a comprehensive system overview is provided for both physical layer and MAC (medium access control) layer. Differently, this paper focuses on the techniques and performance of TOA estimation, positioning algorithms, and tracking/smoothing.

3. Time-of-arrival estimation

To achieve accurate position estimation, we must first acquire accurate TOA measurements. There exist numerous TOA estimation algorithms in the literature. A comprehensive literature review on code acquisition and delay estimation for direct-sequence spread spectrum signals can be found in [18,19].

The extremely short (usually sub-nanosecond), very low-duty cycle UWB pulses with very low power spectral density, poses a challenge for synchronization in UWB systems. One method in the literature for UWB timing recovery employs the ML criteria [20–22]. A second method applies correlators in the traditional way, but makes use of techniques to obtain rapid timing acquisition [23–25]. However, the decrease of the acquisition time is not substantial. For low cost and low complexity applications, energy collection-based
timing acquisition [26] is a promising approach especially for indoor communications where dense multipath exists. However, the accuracy may not be satisfactory for some applications.

In this work, we employ a two-stage approach for fast timing acquisition to obtain the time-of-arrival of the desired signal. The scenario considered is based only on a LOS (line-of-sight) propagation. Since TDMA scheme is employed, no spreading is considered and non-coherent detection is employed for the sake of low complexity. Therefore, a symbol may consist of either one pulse or a sequence of pulses. As mentioned in Section 2, synchronization is performed once every second between the fixed and mobile nodes. For low complexity devices, the clock accuracy/stability may be around 10 parts per million (ppm). Then, in each second, the uncertainty of TOA could be around $T_u = 10 \mu s$ in addition to the propagation delay.

As shown in Fig. 1, after bandpass filtering and low noise amplification, the received signal is squared\(^4\) and then passed through a bank of integrators.

### 3.1. First stage processing

At the first stage, the uncertain region is divided into $K$ sectors and the length of each sector equals $T_{\text{int}} = T_u/K$. Each integrator integrates the squared signal of one sector\(^5\) as shown in Fig. 2. In the case of one pulse per symbol, $T_{\text{int}}$ may be chosen to be up to the multipath spread of the channel. In the case of multiple pulses per symbol, $T_{\text{int}}$ may be chosen to be the length of the pulse sequence plus a value less than the multipath spread. In practice, the multipath spread can only be estimated/predicted.

Based on the energy measurements, a decision is made according to a chosen criterion. In the hypothesis testing (decision making), three basic criteria may be considered. The first is the threshold crossing (TC) criterion. With TC, the search is performed serially and is stopped once a measurement value crosses the threshold. The corresponding sector is then chosen and its starting time provides the coarse TOA information. If necessary, a verification process may be pursued. In the event that no measurement crosses the threshold, new measurements are taken and the search resumes. The TC algorithm requires the setting of a threshold.

The other approach is the maximum selection (MAX) criterion. With this criterion, measurements at all sectors are first compared. Then, the maximal measurement is produced and the relevant sector is selected. In the event that no appropriate thresholds can be readily obtained, the MAX criterion could be desirable. Another criterion is the hybrid of MAX and TC. In this hybrid criterion, the maximal measurement is first obtained. Then, the maximum is examined against the threshold. If the threshold is crossed, the related integration sector is selected. If the maximum does not cross the threshold, the search resumes. With a probability dependent on the SNR, the coarse TOA estimate will satisfy

$$\tau_0 - \frac{1}{2}T_{\text{int}} < \hat{\tau}_0 < \tau_0 + \frac{1}{2}T_{\text{int}}, \quad \text{(1)}$$

where $\tau_0$ and $\hat{\tau}_0$ are the actual and the coarse TOA estimate, respectively.

### 3.2. Second stage processing

At the second stage, fine search is constrained to this reduced uncertain region given by Eq. (1). The same bank of integrators can be employed to obtain fine TOA estimate. The difference between the start time points of two adjacent integrators now can be as small as the clock period. Fig. 3 shows two examples of the time sequence and outputs of the integrators at the second stage, which represent the best (left) and the worst (right) case TOA estimation.

For simplicity the integration goes over an interval of $t_2 - \tau_0 + T_w$ for each integrator where $\tau_0$ is the actual TOA and $T_w$ is the pulse width. In practice, the integration interval at the second stage can be chosen based on the predicted/estimated channel parameters. If

---

\(^4\) The squared operation is necessary if the received pulses can be both positive and negative. In the event that all the received signal pulses have the same sign (either positive or negative) or energy of pulses of one sign is much greater than that of the other sign, the original signal may be directly integrated.

\(^5\) The integrators may be reused multiple times when $K$ is greater than the number of integrators. Another option is that only one single integrator is used to perform integration over the whole uncertain region provided that no saturation happens. Then, the output is read at $iT_{\text{int}}$ where $i \geq 0$. 
appropriate thresholds can be found, the TC criterion or the hybrid criterion can be employed. Otherwise, the MAX criterion should be considered. The process may continue over a sequence of symbols to produce multiple TOA estimates which can be further processed to obtain more accurate estimates.

To examine the accuracy of the proposed two-stage TOA estimation approach, we consider an outdoor environment with dimensions 400 m $\times$ 100 m $\times$ 100 m. A four path channel model [27] is employed to approximate an outdoor snow covered environment which provided motivation for the study. The first path signal has constant amplitude and the other three paths have Nakagami fading amplitudes [28] with a fading figure $m = 1.5$. The fading amplitudes can be either positive or negative with the same probability [29]. The delay of the second path is assumed to be 2 ns and the fourth path delay is 12 ns. Each bit consists of only one pulse [13] and pulse width is equal to 0.5 ns.

For indoor UWB communications, rich multipath exists and the multipath spread can be much larger.
It is assumed that the first stage search is successful and the TOA is constrained to the region given by Eq. (1). The power ratio (Rician factor) of the direct path to the fading paths is equal to one (i.e. 0 dB) and the SNR equals 8 dB and a total of 50,000 symbols are examined to produce 50,000 TOA estimates. Fig. 4 shows the amplitude distribution of the simulated TOA estimation errors while Fig. 5 shows the root mean squared error of the TOA estimation with respect to SNR for two different Rician factors. The MAX criterion is employed.

4. Positioning algorithms

In this section, we study various position estimation algorithms of either non-iterative or iterative. Details of several typical algorithms are presented. Also a positioning algorithm is proposed for node location in distributed sensor networks.

4.1. Non-iterative Methods

There exist various non-iterative position estimation algorithms. They include the direct method [1,30], the SI method [4], and other least-squares related techniques [6,7]. The non-iterative algorithms are simple and easy to implement compared to the iterative algorithms to be discussed in Section 4.2. For quick reference, the direct method using TDOA for 3D position estimation can be derived as follows.

In the Cartesian system, the range (distance) between fixed node \(i\) and the mobile node is given by

\[
\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} = c(t_i - t_0),
\]

\(i = 1, 2, 3, 4,\)    

where \(N\) is the number of the fixed nodes/base stations, \((x, y, z)\) are the unknown position coordinates of the mobile node, and \((x_i, y_i, z_i)\) are the known coordinates of fixed node \(i\). \(c\) is the speed of light, \(t_i\) is the signal TOA at fixed node \(i\) to be estimated, and \(t_0\) is the unknown transmit time at the mobile node. In the development of the

![Fig. 4. Amplitude distribution of TOA estimation errors when time instants of the integrators are as shown in Fig. 3.](image1)

![Fig. 5. RMS of TOA estimation errors. ‘syn’ denotes results when time instants of the integrators are as shown in the left part of Fig. 3 and other results are obtained when time instants of the integrators are as shown in the right part of Fig. 3.](image2)
expressions, we ignore the difference between the actual and the measured TOAs for simplicity. Squaring both sides of (2) gives

\[(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2 = c^2(t_i - t_0)^2, \quad i = 1, 2, 3, 4.\]

(3)

Subtracting (3) for \(i = 1\) from (3) for \(i = 2, 3, 4\) produces

\[ct_0 = \frac{c}{2}(t_1 + t_i) + \frac{1}{2c(t_1 - t_i)} \times (\beta_{1i} - 2x_{1i}x - 2y_{1i}y - 2z_{1i}z), \quad i = 2, 3, 4,\]

(4)

where

\[x_{1i} = x_i - x_1, \quad y_{1i} = y_i - y_1, \quad z_{1i} = z_i - z_1,\]

\[\beta_{1i} = x_1^2 + y_1^2 + z_1^2 - (x_i^2 + y_i^2 + z_i^2)\]

Define the TDOA between fixed nodes \(i\) and \(j\) as

\[\Delta t_{ij} = t_i - t_j.\]

Eliminating \(t_0\) in (4) yields

\[a_1x + b_1y + c_1z = g_1,\]

(5)

where

\[a_1 = \Delta t_{12}x_{31} - \Delta t_{13}x_{21}, \quad b_1 = \Delta t_{12}y_{31} - \Delta t_{13}y_{21}, \quad c_1 = \Delta t_{12}z_{31} - \Delta t_{13}z_{21}, \quad g_1 = \frac{1}{2}(c^2\Delta t_{12}\Delta t_{13}\Delta t_{32} + \Delta t_{12}\beta_{31} - \Delta t_{13}\beta_{21})\]

and

\[a_2x + b_2y + c_2z = g_2,\]

(6)

where

\[a_2 = \Delta t_{12}x_{41} - \Delta t_{14}x_{21}, \quad b_2 = \Delta t_{12}y_{41} - \Delta t_{14}y_{21}, \quad c_2 = \Delta t_{12}z_{41} - \Delta t_{14}z_{21}, \quad g_2 = \frac{1}{2}(c^2\Delta t_{12}\Delta t_{14}\Delta t_{42} + \Delta t_{12}\beta_{41} - \Delta t_{14}\beta_{21})\]

Combining (5) and (6) yields

\[x = A\hat{z} + B,\]

(7)

where

\[A = \frac{b_1c_2 - b_2c_1}{a_1b_2 - a_2b_1}, \quad B = \frac{b_2g_1 - b_1g_2}{a_1b_2 - a_2b_1}\]

and

\[y = C\hat{z} + D,\]

(8)

where

\[C = \frac{a_2c_1 - a_1c_2}{a_1b_2 - a_2b_1}, \quad D = \frac{a_1g_2 - a_2g_1}{a_1b_2 - a_2b_1}\]

Then, substitution of (7) and (8) into (4) for \(i = 2\) produces

\[c(t_1 - t_0) = Ez + F,\]

(9)

where

\[E = \frac{1}{c\Delta t_{12}}(x_{21}A + y_{21}C + z_{21}), \quad F = \frac{c\Delta t_{12}}{2} + \frac{1}{2c\Delta t_{12}}(2(x_{21}B + y_{21}D) - \beta_{21}).\]

Substituting (7), (8) and (9) back into (2) for \(i = 1\) followed by squaring yields

\[G\hat{z}^2 + Hz + I = 0,\]

(10)

where

\[G = A^2 + C^2 + 1 - E^2, \quad H = 2[A(B - x_1) + C(D - y_1) - z_1 - EF], \quad I = (B - x_1)^2 + (D - y_1)^2 + z_1^2 - F^2.\]

The two solutions to (10) are

\[\hat{z} = -\frac{H}{2G} \pm \sqrt{\left(\frac{H}{2G}\right)^2 - \frac{I}{G}}.\]

(11)

The two estimated \(z\) values (if both are reasonable) are then substituted into (7) and (8) to produce the estimate of the coordinates \(x\) and \(y\), respectively. However, there is only one desirable solution. We remove the one with either no physical meaning or which is beyond the monitored area. If both solutions are reasonable and they are very close, we may choose the average as the position estimate. Otherwise, an ambiguity occurs. In this case, one more fixed node may be required to clear this ambiguity. The cases where no acceptable results are produced include two complex solutions, or when both solutions are beyond the monitored area.

When the transmit time \(t_0\) is available, only three fixed nodes are required to determine the position variables and one extra fixed node might be needed to clear the problem of two solutions in some cases. It is trivial to derive the solution of the position coordinates in this case. For quick reference, we give the formulae as follows.

Define

\[f_{1i} = \frac{1}{2}c^2((t_1 - t_0)^2 - (t_i - t_0)^2) + \beta_{1i}), \quad i = 2, 3.\]
Also define

\[ A_1 = \frac{x_{21}z_{31} - x_{31}z_{21}}{x_{31}y_{21} - x_{21}y_{31}}, \quad B_1 = \frac{x_{31}f_{12} - x_{21}f_{13}}{x_{31}y_{21} - x_{21}y_{31}}, \]

\[ C_1 = \frac{y_{31}z_{21} - y_{21}z_{31}}{x_{31}y_{21} - x_{21}y_{31}}, \quad D_1 = \frac{y_{21}f_{13} - y_{31}f_{12}}{x_{31}y_{21} - x_{21}y_{31}}. \]

Then we have

\[ \frac{z}{E_1} = \frac{F_1}{E_1} \left( \frac{F_1}{E_1} \right)^2 - \frac{G_1}{E_1}, \]

where

\[ E_1 = A_1^2 + C_1^2 + 1, \]

\[ F_1 = A_1(y_1 - B_1) + C_1(x_1 - D_1) + z_1, \]

\[ G_1 = (x_1 - D_1)^2 + (y_1 - B_1)^2 + z_1^2 - c^2(t_1 - t_0)^2, \]

and

\[ \hat{x} = C_1\hat{z} + D_1, \quad \hat{y} = A_1\hat{z} + B_1. \]  

(13)

Note that different formulae may be derived for the direct position calculation, however, the presented method may be desirable since it does not involve any matrix operation.

Let us now study the SI method. First map the spatial origin to one of the fixed nodes, say fixed node 1, as shown in Fig. 6. Define

\[ R_i = \sqrt{x_i^2 + y_i^2 + z_i^2}, \quad p_i = [x_i, y_i, z_i]^T, \]

\[ R = \sqrt{x^2 + y^2 + z^2}, \quad d_{ij} = d_i - d_j. \]

Then, from the Pythagorean theorem, we have

\[ (R + d_{i,1})^2 = R_i^2 - 2p_i^T p + R^2, \quad i = 2, 3, \ldots, N. \]  

(14)

Fig. 6. Illustration for spherical interpolation approach.

In the presence of measurement errors, (14) becomes

\[ e_i = R_i^2 - d_{i,1}^2 - 2Rd_{i,1} - 2p_i^T p, \]

where \( e_i \) is the equation error. Eq. (15) can be written in compact form as

\[ \varepsilon = \delta - 2Rd - 2Ap, \]  

(16)

where \( \varepsilon \) is the equation error vector and

\[ [A]_{i,1} = x_{i+1}, \quad [A]_{i,2} = y_{i+1}, \]

\[ [A]_{i,3} = z_{i+1}, \quad i = 1, 2, \ldots, N - 1, \]

\[ \delta = [R_3^2 - d_{2,1}^2, R_3^2 - d_{3,1}^2, \ldots, R_N^2 - d_{N,1}^2]^T, \]

\[ d = [d_{2,1}, d_{3,1}, \ldots, d_{N,1}]^T. \]

The standard LS solution for \( p \), given \( R \), is

\[ p = \frac{1}{2}(A^T A)^{-1} A^T (\delta - 2Rd). \]

(17)

The key idea of the SI approach is to substitute (17) into (16) and minimize the equation error again but with respect to \( R \). The source location estimate is then obtained as

\[ \hat{p} = \frac{1}{2}(A^T W A)^{-1} A^T W \left(1 - \frac{dd^T B V B}{d^T B V d}\right). \]

(18)

where \( W \) and \( V \) are weighting matrices and

\[ B = I - A(A^T W A)^{-1} A^T W. \]

Let us examine the performance of the non-iterative methods. Performance evaluation is performed in terms of the RMSE of the coordinate estimation and the failure rate. The failure rate includes the cases where there is no solution or the solution is unreasonable. With the iterative methods, the failure rate includes situations where the algorithm does not converge to a solution, the maximum number of function evaluations/iterations is exceeded, or the results are beyond the area examined. With the non-iterative methods, ‘failures’ include cases when: the solutions are beyond the monitored area; both solutions are complex-valued; the two solutions are reasonable but not close to each other; or inversion of singular matrices is involved.

In the simulation results, the monitored area examined has dimensions of \( 90(l) \times 90(w) \times 10(h) \) m. The positions of the fixed nodes and the mobile node of interest are randomly generated. At each test point, 1000 runs are conducted with new random positions of the fixed nodes and the mobile node at each run. The performance is then averaged.
The TOA estimation errors are produced using the synchronization technique described in Section 3. Throughout the rest of the paper, the results corresponding to the time instants of the integrators as shown in the first part of Fig. 3 are employed.

Fig. 7 shows the accuracy and the failure rate of the direct method and the SI algorithm [4]. The TOA-based direct approach (results denoted by ‘DM3’) is rather sensitive to the accuracy of transmit time. ‘DM3(inacc t0)’ represent results with transmit time error of 4 ns while DM4(TDOA) denotes results for the TDOA-based direct method. When transmit time error is in the order of tens of ns, the TOA-based method does not work well at all. Therefore, only when nearly error-free transmit time information (compared to time reference at fixed nodes) is available, (12) to (13) are desirable.

Also shown are the results for the SI method with 5, 6 and 8 fixed nodes (denoted by ‘SI5’, ‘SI6’, and ‘SI8’). The technique does not work well in the case of four fixed nodes as indicated in [4], so the corresponding results are not presented. At relatively high SNR, the SI method performs well when at least five fixed nodes are employed.

4.2. Iterative methods

In the iterative methods, the estimation is performed iteratively and the iteration will not stop until some pre-defined criterion is satisfied. In this section we investigate Taylor series method and optimization techniques for position estimation.

4.2.1. Taylor series method

In Taylor series method, a set of nonlinear equations is linearized by expanding it in a Taylor series around a point (initially an estimate of the actual position) and keeping only terms below second order. The set of linearized equations is solved to produce a new approximate position and the process continues until a pre-specified criterion is satisfied.

When TDOA measurements are employed, this method may be described as follows.

Subtracting (2) for \( i = 1 \) from (2) for \( i = 2, 3, \ldots, N \) produces

\[
\sqrt{(x - x_1)^2 + (y - y_1)^2 + (z - z_1)^2} - \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} = c(t_i - t_1), \quad i = 2, 3, \ldots, N. \tag{19}
\]

Define

\[
f_i(x, y, z) = \sqrt{(x - x_{i+1})^2 + (y - y_{i+1})^2 + (z - z_{i+1})^2} - \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2},
\]

and let \( \hat{t}_i \) be the TOA estimate at fixed node \( i \). Then,

\[
f_i(x, y, z) = \hat{t}_{i+1,1} + \epsilon_{i+1,1}, \quad i = 1, 2, \ldots, N - 1,
\]
where $\hat{d}_{i+1} = c(t_i - t_1)$ and $\epsilon_{i+1}$ is the corresponding range difference estimation error with covariance $R$. If $x_v$, $y_v$, and $z_v$ are guesses of the actual mobile position, then,

$$x = x_v + \delta_x, \quad y = y_v + \delta_y, \quad z = z_v + \delta_z,$$

where $\delta_x$, $\delta_y$, and $\delta_z$ are the position errors to be determined. Expanding $f_i$ in Taylor series and retaining the first two terms produce

$$f_{i,v} + a_{i,1}\delta_x + a_{i,2}\delta_y + a_{i,3}\delta_z \approx \hat{d}_{i+1} + \epsilon_{i+1},$$

$$i = 1, 2, \ldots, N - 1,$$  \hspace{1cm} (20)

where

$$f_{i,v} = f_i(x_v, y_v, z_v),$$

$$a_{i,1} = \frac{\partial f_i}{\partial x}
|_{x_v, y_v, z_v} = \frac{x_1 - x_v}{d_1} - \frac{x_{i+1} - x_v}{d_{i+1}},$$

$$d_i = \sqrt{(x_v - x_i)^2 + (y_v - y_i)^2 + (z_v - z_i)^2},$$

$$a_{i,2} = \frac{\partial f_i}{\partial y}
|_{x_v, y_v, z_v} = \frac{y_1 - y_v}{d_1} - \frac{y_{i+1} - y_v}{d_{i+1}},$$

$$a_{i,3} = \frac{\partial f_i}{\partial z}
|_{x_v, y_v, z_v} = \frac{z_1 - z_v}{d_1} - \frac{z_{i+1} - z_v}{d_{i+1}}.$$  \hspace{1cm} (21)

Eq. (20) can be rewritten as

$$A\delta = D + e,$$  \hspace{1cm} (22)

where

$$A = \begin{bmatrix}
a_{1,1} & a_{1,2} & a_{1,3} \\
a_{2,1} & a_{2,2} & a_{2,3} \\
\vdots & \vdots & \vdots \\
a_{N-1,1} & a_{N-1,2} & a_{N-1,3}
\end{bmatrix}, \quad \delta = \begin{bmatrix}
\delta_x \\
\delta_y \\
\delta_z
\end{bmatrix},$$

$$D = \begin{bmatrix}
\hat{d}_{2,1} - f_{1,v} \\
\hat{d}_{3,1} - f_{2,v} \\
\vdots \\
\hat{d}_{N,1} - f_{N-1,v}
\end{bmatrix}, \quad e = \begin{bmatrix}
\epsilon_{2,1} \\
\epsilon_{3,1} \\
\vdots \\
\epsilon_{N,1}
\end{bmatrix}.$$  \hspace{1cm} (22)

The weighted least-square estimator for (22) produces

$$\delta = [A^TR^{-1}A]^{-1}A^TR^{-1}D.$$  \hspace{1cm} (23)

Given an initial position guess $(x_v, y_v, z_v)$ and compute $\delta$ with (23). Then update the position estimate according to

$$x_v = x_v + \delta_x, \quad y_v = y_v + \delta_y, \quad z_v = z_v + \delta_z.$$  \hspace{1cm} (24)

Continually refine the position estimate until $\delta$ is sufficiently small.

4.3. Optimization-based methods

After defining the objective function, several optimization-based position estimation methods are studied.

4.3.1. Objective function

An objective function is normally required for optimization algorithms. Since the aim of positioning is to obtain an accurate position estimate, it is natural to define the objective function as the sum of the squared range errors:

$$F(x, y, z, t_0) = \frac{1}{2} \sum_{i=1}^{N} f_i^2(x, y, z, t_0),$$  \hspace{1cm} (24)

where

$$f_i(x, y, z, t_0) = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - c(t_i - t_0).$$

The optimization purpose is to minimize this objective function to produce the optimal position estimate. For notational simplicity, we define

$$p = [x, y, z, t_0]^T, \quad f(p) = [f_1(p) f_2(p) \ldots f_N(p)]^T.$$  \hspace{1cm} (25)

Then (24) becomes

$$F(p) = \frac{1}{2}||f(p)||^2.$$  \hspace{1cm} (26)

There exist many optimization methods to minimize the pre-defined objective function to achieve the optimal performance (i.e. the optimal position estimate in our case). We choose two specific methods, the Gauss–Newton type methods and the quasi-Newton methods, for investigation.

4.3.2. Gauss–Newton type methods

Expanding the objective function in the Taylor series at the current point $p_k$ and taking the first three terms, we have

$$F(p_k + s_k) \approx F(p_k) + g_k^T s_k + \frac{1}{2} s_k^T G(p_k) s_k,$$  \hspace{1cm} (25)

where $s_k$ is the directional vector (or increment vector) to be determined, $g_k$ is a vector of the first partial derivatives (also called gradient) of the objective function at $p_k$, and $G(p_k)$ is the Hessian of the objective function. Minimization of the right-hand side of (25) yields

$$G(p_k) s_k = -g_k.$$  \hspace{1cm} (26)
The minimization in which \( s_k \) is defined by Eq. (26) is termed Newton’s method [10,11]. To avoid the calculation of the second order information in the Hessian, a simplified expression can be approached from (26), resulting in

\[
J_k^T J_k s_k = -J_k^T f(p_k),
\]

where \( J_k \) is the Jacobian matrix of \( f(p) \) at \( p_k \). This is termed the Gauss–Newton method. When \( J_k \) is full rank, which is the usual case of an over-determined system, we have the linear least-squares solution

\[
s_k = -(J_k^T J_k)^{-1} J_k^T f(p_k).
\]

The Gauss–Newton method may not perform well when the second order information in the Hessian is non-trivial. A method to overcome this is the Levenberg–Marquardt method. The Levenberg–Marquardt search direction is defined as the solution of the equations

\[
(J_k^T J_k + \lambda_k I) s_k = -J_k^T f(p_k),
\]

where \( \lambda_k \) is a non-negative scalar which controls both the magnitude and direction of \( s_k \).

To start the iteration, the initial position coordinates and the initial transmit time are required. The initial estimated values of the position coordinates may be chosen to be the mean position of the fixed nodes or the area being monitored. The initial estimated transmit time may be chosen to be some time point earlier than the earliest receive time. This will depend on the dimension of the monitored area. Certainly, if a more accurate initial position and transmit time estimate is available, the performance could be improved. This may be achieved by the non-iterative algorithms discussed in Section 4.1 or by position prediction discussed in Section 5.

4.3.3. Quasi-Newton methods

This method is similar to Newton’s method. The Hessian matrix \( G(p_k) \) in (26) is now approximated by a symmetric positive definite matrix \( B_k \), which is updated during each iteration. At the \( k \)th iteration, set

\[
s_k = -B_k g_k.
\]

Using line search along \( s_k \) produces

\[
p_{k+1} = p_k + \alpha s_k,
\]

where \( \alpha \) is the step size. Then \( B_k \) is updated to yield \( B_{k+1} \). There exist different ways of updating \( B_k \). One well known updating formula is the DFP (Davidon–Fletcher–Powell) formula [12], updating \( B_k \) according to

\[
B_{k+1} = B_k + h_k h_k^T - B_k q_k q_k^T B_k,
\]

where

\[
h_k = p_{k+1} - p_k, \quad q_k = g_{k+1} - g_k.
\]

The initial matrix \( B_1 \) can be any positive definite matrix. It is usually set to be the identity matrix in the absence of any better estimate [11]. Another important formula is the BFGS formula [11]. In this

Fig. 8. RMSE of DFP (left figure) and TS (right figure) algorithms using 4, 5, 6, and 8 fixed nodes.
formulae, $B_k$ is updated based on

$$B_{k+1} = B_k + \left( 1 + \frac{h_k^T B_k h_k}{q_k^T q_k} \right) q_k q_k^T - \frac{q_k^T h_k B_k + B_k h_k q_k q_k^T}{q_k^T q_k}.$$ 

Clearly the BFGS formula requires significantly greater computational effort. All the algorithms can be found in the Matlab optimization toolbox.

4.4. Performance evaluation

In this section we examine the performance of the iterative algorithms through simulation. The simulation setup is the same as in Section 4.1.

Figs. 8 and 9 show the RMSE of the position estimation and the failure rate, respectively, for the DFP and TS algorithms. Four different numbers of fixed nodes (i.e. 4, 5, 6 and 8) are examined. Clearly more fixed nodes achieve better performance.

The Gauss–Newton method did not perform well at all and the BFGS approach produces very similar performance to the DFP approach. Since the BFGS approach is more complicated, the DFP approach is preferable among the quasi-Newton methods. It is also shown that the Levenberg–Marquardt algorithm produced very similar results to the DFP formulae. Therefore, only results from the DFP approach and Taylor series method are presented.

Accuracy and failure rate are crucial measures in evaluating a positioning technique. Complexity of a position estimation algorithm is also an important issue in the design of a positioning system especially when low cost and low complexity are involved. Table 1 shows the normalized execution time for the four different algorithms, i.e. DFP algorithm, Taylor series method, SI method, and direct method. The number of iterations of TS is set to be 20. The Matlab optimization toolbox is employed to run DFP algorithm and the maximum number of iterations is set to be 20. The iterative algorithms are more complex than the non-iterative ones and the DFP algorithm requires much more computational time than any of the other three algorithms.

4.5. Positioning in distributed architectures

In this section, we consider localization in ad hoc and distributed networks. An ad hoc network is a collection of wireless nodes that self-configure themselves to form a network without the aid of any infrastructure. This sort of networks are characterized by large size, need for distributed coordination and ubiquitous connectivity, power constraints and the ability to be ad hoc deployable.
In the following subsections, an overview of related localization approaches is first presented. Then a positioning algorithm is proposed and finally some simulation results are provided.

4.5.1. Overview

In the recent years, there have been numerous algorithms (either centralized or distributed) to localize sensor nodes in wireless sensor networks. In [31,32], a centralized scheme is proposed which collects the entire topology in a server to minimize the errors using convex optimization. In [33–35], instead of directly solving the set of constraints of the whole wireless network, multi-dimensional scaling (MDS) is exploited. This technique uses local connectivity or distance measure to generate relative maps that represent the relative position of nodes. The main problem of the mentioned algorithms is the need to have some powerful node or server to perform the large computation. In [36], distributed algorithms are divided in two sub-families: range-based and range-free algorithms. In range-free localization [37,38], beacon nodes broadcast their positions to their neighbors that keep an account of all received beacons. Then, nodes calculate their positions based on the received beacon locations, the hop-count from the corresponding beacon and the average distance per hop. In [39], the distance per hop is averaged by taking into account the local density of nodes. In range-based algorithms, the distance between two neighboring sensors is first estimated, for example, by using TOA measurements. In [40] and more recently in [41], a distributed mechanism is proposed for GPS-free positioning in mobile ad hoc networks. A slightly modified version of the GPS-free algorithm is proposed in [42]. In [43], the distance between a sensor and a beacon is directly calculated using basic triangle rules and simple geometry. Collinearity is exploited in [44] and factor graphs are employed in [45].

4.5.2. Proposed algorithm

It is assumed that each node (sensor and beacon) has the ability of measuring both TOA and AOA with its 1-hop neighbors. In the first phase, the beacons broadcast their coordinates. Then, all sensors establish the shortest path with the beacons. As a result, a sensor should have the coordinates of at least three beacons and the path to reach them. After the shortest path is developed, all sensors and beacons belonging to the path calculate the TOA and AOA with the neighboring nodes in the path. Then, using the TOA and AOA measurements, a sensor is able to estimate the Euclidean distance to each of at least three beacons.

Fig. 10 gives an illustration of the algorithm. In the figure, sensor $S_1$ established a path to each of the three beacons ($B_1$, $B_2$ and $B_3$). Then, it stores the relevant information in its data base in the form

$S_1 \rightarrow B_1 : \Phi_1 = \{(S_1, S_3, B_1), (\hat{\theta}_{13}, \hat{\theta}_{3B_1}), (\hat{d}_{13}, \hat{d}_{3B_1})\},$

$S_1 \rightarrow B_2 : \Phi_2 = \{(S_1, S_2, B_2), (\hat{\theta}_{12}, \hat{d}_{1B_2})\},$

$S_1 \rightarrow B_3 : \Phi_3 = \{(S_1, S_2, B_3), (\hat{\theta}_{12}, \hat{\theta}_{B_2}), (\hat{d}_{12}, \hat{d}_{2B_2})\}.$

The distance from $S_1$ to the three beacons can be determined by

$$\hat{d}_{1B_1} = \sqrt{(\hat{d}_{13} \cos \hat{\theta}_{13} + \hat{d}_{3B_1} \cos \hat{\theta}_{3B_1})^2 + (\hat{d}_{13} \sin \hat{\theta}_{13} + \hat{d}_{3B_1} \sin \hat{\theta}_{3B_1})^2},$$

$$\hat{d}_{1B_2} = \hat{d}_{1B_1},$$

$$\hat{d}_{1B_3} = \sqrt{(\hat{d}_{12} \cos \hat{\theta}_{21} + \hat{d}_{2B_1} \cos \hat{\theta}_{B_1})^2 + (\hat{d}_{12} \sin \hat{\theta}_{21} + \hat{d}_{2B_1} \sin \hat{\theta}_{B_1})^2}.$$
After obtaining the distance to each of the beacons, the position coordinates of the node can be estimated using the algorithms studied in Section 4. When considering the optimization-based approaches, the cost function is defined as

\[
\varepsilon(x, y) = \sum_{k=1}^{N_B} \left( \hat{d}_k - \sqrt{(x_k - x)^2 + (y_k - y)^2} \right)^2,
\]

where \( \hat{d}_k \) is the estimated distance between the desired sensor and the \( k \)th beacon, and \( N_B \) is the number of beacons available in the network. \((x, y)\) and \((x_k, y_k)\) are the unknown coordinates of the sensor of interest and the known coordinates of the \( k \)th beacon, respectively. Although we focused on 2D sensor location, it is straightforward to extend the algorithm to 3D positioning.

4.5.3. Simulation results

The monitored area has dimensions of 100(w) \times 100(l) m. Hundred nodes are randomly positioned in the area together with a number (5, 15, and 20) of beacon nodes depending on the simulation scenarios. The transmission radius of any node in this network is equal to 30 m and it is kept constant. The position estimation algorithms considered are the DFP algorithm and the direct method (DM). The performance evaluation is performed by assuming that the TOA and AOA measurement errors are white Gaussian random variables of mean zero and variance \( \sigma_{\text{TOA}}^2 \) and \( \sigma_{\text{AOA}}^2 \), respectively. At each test point \((\sigma)\), 30 simulations are conducted with new random topology of the network and the performance is then averaged. In the case of the DFP algorithm, 50 iterations are used to update the estimated coordinates.

Fig. 11 shows the root mean square error of the coordinates estimations using either the DM or DFP algorithm. In general, the RMS errors of both algorithms are under 2 m and the DFP algorithm achieves higher accuracy. We also notice the quite flat curve (except for the case of DM with \( \sigma_{\text{TOA}} = 0 \)) as the AOA measurements error increases. It means that the algorithms, to localize the nodes, are not sensitive to the AOA measurement error.

Fig. 12 shows the RMS error with respect to \( \sigma_{\text{TOA}} \) at a given \( \sigma_{\text{AOA}} \). In this case, the slope of the curves becomes sharper. This phenomena tells the fact that the algorithms are more sensitive to TOA measurement error, comparatively.

5. Tracking moving objects

When considering mobile devices with velocity up to 100 km/h, tracking should be included in the positioning algorithms. The system should be able to update the position estimation at a reasonable rate to follow the moving devices. At each time instant, a number of TOA measurements are collected from a specific set of fixed nodes. Some set members can be different as the device moves. Usually the fixed nodes closest to the moving device are employed to provide the time measurements since in general shorter distance means higher signal power so better performance can be obtained.
Tracking performance can be improved by smoothing the individual position results. Kalman filtering has been widely used in modern control systems, tracking and navigation of all sorts of vehicles [46]. Some references about using Kalman filtering to smooth/filter position/velocity estimate of moving objects can be found in [47–52]. Another filtering approach used for smoothing position estimates is the linear least-squares approach [53]. The related formula of the two smoothing methods is provided in Appendices A and B, respectively.

When the track is rather irregular/nonlinear and/or the velocity of the moving nodes is time-varying, the LS smoothing approach may not perform well. Kalman filtering requires that the variances of the system noise and observation noise are known in a priori. However, in practice, the position estimation noise variances are usually unknown and implementation of Kalman filtering at low-complexity nodes may be not feasible due to its relatively high computational requirement. For those considerations, we propose to exploit the sinc function to smooth the tracks and improve the accuracy of the position estimates. Sinc function has been exploited to interpolate pilot symbol aided channel estimates [54]. Fig. 13 shows the block diagram of the proposed location and tracking system.

The principle of sinc smoothing is rather simple. The smoothing function is given by

$$h(j) = \frac{\sin(2\pi f_M T)}{2\pi f_M T}, \quad -\infty < j < \infty,$$

where $f_M$ is the maximum frequency of the track (function of time) and $T$ is the position updating period. In practice, truncation is used and the length of the window is equal to $K$ so that

$$-\frac{K - 1}{2} \leq j \leq \frac{K - 1}{2}.$$
When $K$ is not large such as about 10–20, some windowing techniques such as Hanning window or Hamming window may be required to smooth the truncation. Let $\{\tilde{x}_i\}$ be the sequence of the estimated coordinate (any of the three coordinates) and $\{\hat{x}_i\}$ be the corresponding results after smoothing. Then,

$$\hat{x}_i = \sum_{j=-p}^{p} h(j)\tilde{x}_{i+j}, \quad i > p.$$ 

It has been shown in [54] that sinc smoothing is optimal in the event that the power spectrum of the signal is bandlimited and has a flat spectrum up to $f_M$. Usually $f_M$ is unknown; however, low frequency components would be dominant so that it can be chosen empirically.

6. Simulation results

In this section, we examine the performance of the proposed position location and tracking system (as shown in Fig. 13). We use one of the realistic field structures, a snow covered slope of dimensions about 400 m x 100 m x 100 m. The fixed nodes will be deployed along both sides of the slope and mounted on poles of varying height. Fig. 14 shows the imagined track for examination. The skier moves from A to B (120 m) at a speed of 8 m per second (m/s). The skier moves from B to C (160 m) at a speed of 10 m/s and finally from C to D (120 m) at a speed of 8 m/s.

First we examine the performance of the different position estimation algorithms under the more realistic circumstance. Two hundred different combinations of fixed node positions are tested and then the results are averaged. Tables 2 and 3 compare the averaged results of the four algorithms at SNR of 16 dB.

For the parameters examined, the SI method provides the best tradeoff between performance and complexity when there are at least five fixed nodes. To achieve sub-meter accuracy, at least six fixed nodes are needed with SNR up to 16 dB.

Let us consider position smoothing by making use of the three smoothing techniques discussed in Section 7. Table 4 shows the averaged RMSEs before and after smoothing. The estimated tracks (before smoothing) are produced by using the SI algorithm with five fixed nodes under three sets of fixed node configurations. Since the one-Kalman-filter scheme and the three-Kalman-filter scheme produce the same results, only results from one of them are listed. Clearly, the sinc smoothing even achieves the best results on average.

Fig. 15 shows the original, estimated and smoothed tracks using the sinc smoothing. The

![Fig. 14. Track for examination.](image-url)
smoothing window length equals 11 and the five estimated values at both the beginning and the end of the tracks are not processed/smoothed. The initial position estimates are produced using the SI method under one specific combination of the fixed nodes and the RMSE before smoothing is 2.70 m. The effectiveness of smoothing is clearly demonstrated.

7. Conclusions

In this paper we investigated several position estimation approaches employing UWB technology for outdoor recreational activities. Focus was placed on two iterative algorithms (i.e. DFP and Taylor series) and two non-iterative methods (i.e. spherical interpolation and direct method). Performance comparisons of the four methods were performed under different scenarios. Using the proposed TOA estimation technique and certain number of fixed nodes, accurate position estimates can be obtained even under a realistic field structure. A sinc smoothing technique was employed to achieve the goal of both low complexity and high accuracy. A new multi-hop location technique was also proposed for distributed sensor networks.

Acknowledgment

This work is partly funded by the EU 6th framework project PULSERS.

 Appendix A. Kalman filtering for position smoothing

Define state vector at time $t_k$

$$x_k = [x_k y_k z_k v_{x,k} v_{y,k} v_{z,k}]^T,$$

smoothing window length equals 11 and the five estimated values at both the beginning and the end of the tracks are not processed/smoothed. The initial position estimates are produced using the SI method under one specific combination of the fixed nodes and the RMSE before smoothing is 2.70 m. The effectiveness of smoothing is clearly demonstrated.

7. Conclusions

In this paper we investigated several position estimation approaches employing UWB technology for outdoor recreational activities. Focus was placed on two iterative algorithms (i.e. DFP and Taylor series) and two non-iterative methods (i.e. spherical interpolation and direct method). Performance comparisons of the four methods were performed under different scenarios. Using the proposed TOA estimation technique and certain number of fixed nodes, accurate position estimates can be obtained even under a realistic field structure. A sinc smoothing technique was employed to achieve the goal of both low complexity and high accuracy. A new multi-hop location technique was also proposed for distributed sensor networks.

Acknowledgment

This work is partly funded by the EU 6th framework project PULSERS.

Appendix A. Kalman filtering for position smoothing

Define state vector at time $t_k$

$$x_k = [x_k y_k z_k v_{x,k} v_{y,k} v_{z,k}]^T,$$

where $v_{x,k}$, $v_{y,k}$ and $v_{z,k}$ are the velocity components along the $x$, $y$, and $z$ coordinates, respectively. Also define the observation vector

$$p_k = [x_k y_k z_k]^T.$$

Then the system dynamic model (process equation) is

$$x_k = \Phi_{k-1} x_{k-1} + B_{k-1} w_{k-1},$$

where $\Phi_{k-1}$ (state transition matrix) and $B_{k-1}$ are, respectively, given by

$$\Phi_{k-1} = \begin{bmatrix}
0 & 0 & 0 & \Delta t & 0 & 0 \\
0 & 1 & 0 & 0 & \Delta t & 0 \\
0 & 0 & 1 & 0 & 0 & \Delta t \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0.5\Delta t^2 & 0 & 0 & 0 \\
0 & 0.5\Delta t^2 & 0 & 0 \\
\Delta t & 0 & 0 & 0 \\
0 & \Delta t & 0 & 0 \\
0 & 0 & \Delta t & 0 & 0
\end{bmatrix},$$

and $w_{k-1}$ is the acceleration noise assumed to be white Gaussian random vector of mean zero and covariance $Q_{k-1}$. The measurement model (equation) is

$$p_k = H_k x_k + n_k,$$

where $H_k$ is the observation matrix given by

$$H_k = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0
\end{bmatrix},$$

and $n_k$ is the observation noise vector also assumed to be Gaussian with mean zero and covariance $R_k$.

Implementation of Kalman filtering can be summarized as follows. The initial estimate $\hat{x}_0$ and its error covariance $C_0$ are first given. Let $(-)$ and $(+)$ represent the a priori (before update) and the a posteriori (after update) value, respectively. The state estimate extrapolation is given by

$$\hat{x}_k(-) = \Phi_{k-1} \hat{x}_{k-1}(+),$$

and $w_{k-1}$ is the acceleration noise assumed to be white Gaussian random vector of mean zero and covariance $Q_{k-1}$. The measurement model (equation) is

$$p_k = H_k x_k + n_k,$$
and the error covariance extrapolation is performed by
\[ C_k(-) = \Phi_{k-1} C_{k-1}(+) \Phi_{k-1}^T + B_{k-1} Q_{k-1} B_{k-1}^T. \]
(35)
Then the Kalman gain matrix can be computed according to
\[ K_k = C_k(-) H_k^T (H_k C_k(-) H_k^T + R_k)^{-1}. \]
(36)
The state estimate is updated by
\[ \hat{x}_k(+) = \hat{x}_k(-) + K_k (p_k - H_k \hat{x}_k(-)), \]
and the error covariance is updated by
\[ C_k(+) = [I - K_k H_k] C_k(-). \]
(38)
Therefore, Kalman filter is updated recursively from (34) to (38).

In the case of an individual filter used for each position coordinate, the process equation and the observation equation for the x coordinate become
\[
\begin{bmatrix}
    x_k \\
    v_{x,k}
\end{bmatrix}
= \begin{bmatrix}
    1 & \Delta t \\
    0 & 1
\end{bmatrix}
\begin{bmatrix}
    x_{k-1} \\
    v_{x,k-1}
\end{bmatrix}
+ \begin{bmatrix}
    0.5 \Delta t^2 \\
    \Delta t
\end{bmatrix} w_k,
\]
(39)
\[ p_k = \begin{bmatrix}
    1 & 0 \\
    H_k
\end{bmatrix}
\begin{bmatrix}
    x_k \\
    v_{x,k}
\end{bmatrix}
+ n_k.
\]
(40)
In this case the system noise, the observation noise, and the observation are all scalar.

**Appendix B. Least-squares approach for position smoothing**

Assume that the target is in a linear motion with a constant velocity along each coordinate. This assumption would be reasonable when a short distance is considered, although the whole track may not be linear and the velocity can be varying over a long track. Then the actual target position at time \( t_k \) is given by
\[ p_k = p_0 + v t_k, \quad k \geq 1, \]
(41)
where \( p_0 \) is the position at time \( t_0 \) which is normalized to zero. The estimated target position at time \( t_k \) is
\[ \hat{p}_k = p_k + e_k, \]
(42)
where \( e_k \) is the estimation error vector. Let us make use of a sequence of \( K \) position estimates, \( \hat{p}_k, 1 \leq k \leq K \), for position smoothing. The LS estimator is found by minimizing
\[ \sum_{k=1}^{K} ||\hat{p}_k - (p_0 + v t_k)||^2. \]
This minimization may decompose into
\[
\begin{align*}
\min_{p_{x,0}, \ldots, v_x} & \sum_{k=1}^{K} [\hat{p}_{x,k} - (p_{x,0} + v_x t_k)]^2 \\
+ & \min_{p_{y,0}, \ldots, v_y} \sum_{k=1}^{K} [\hat{p}_{y,k} - (p_{y,0} + v_y t_k)]^2 \\
+ & \min_{p_{z,0}, \ldots, v_z} \sum_{k=1}^{K} [\hat{p}_{z,k} - (p_{z,0} + v_z t_k)]^2.
\end{align*}
\]
(43)
Let us pay attention to the first term in (43), which can be written as
\[ \min_{p_{x,0}, v_x} (p_x - M \theta_x)^T (p_x - M \theta_x), \]
(44)
where
\[ p_x = [p_{x,1} p_{x,2} \ldots p_{x,K}]^T, \quad \theta_x = [p_{x,0} v_x]^T, \]
\[ M = \begin{bmatrix}
    1 & 1 & \cdots & 1 \\
    t_1 & t_2 & \cdots & t_K
\end{bmatrix}^T. \]
The minimization in (44) yields \([55]\)
\[ \hat{\theta}_x = (M^T M)^{-1} M^T p_x. \]
(45)
After some mathematical manipulations, (45) becomes
\[ \hat{\theta}_x = \frac{1}{K \sum_{k=1}^{K} t_k^2 - (\sum_{k=1}^{K} t_k)^2} \times \begin{bmatrix}
    \sum_{k=1}^{K} \hat{t}_k \sum_{k=1}^{K} p_{x,k} - \sum_{k=1}^{K} t_k \sum_{k=1}^{K} t_k p_{x,k} \\
    K \sum_{k=1}^{K} t_k p_{x,k} - \sum_{k=1}^{K} t_k \sum_{k=1}^{K} p_{x,k}
\end{bmatrix}. \]
(46)
That is
\[ \hat{p}_{x,0} = \frac{\sum_{k=1}^{K} t_k^2 \sum_{k=1}^{K} p_{x,k} - \sum_{k=1}^{K} t_k \sum_{k=1}^{K} t_k p_{x,k}}{K \sum_{k=1}^{K} t_k^2 - (\sum_{k=1}^{K} t_k)^2}, \]
(47)
\[ \hat{v}_x = \frac{K \sum_{k=1}^{K} t_k p_{x,k} - \sum_{k=1}^{K} t_k \sum_{k=1}^{K} p_{x,k}}{K \sum_{k=1}^{K} t_k^2 - (\sum_{k=1}^{K} t_k)^2}. \]
(48)
Then we have the smoothed x-coordinate position estimate at time \( t_k \) as
\[ \hat{p}_{x,k} = \hat{p}_{x,0} + \hat{v}_x t_k. \]
(49)
In the same way we can find $\hat{p}_{y,0}$ and $\hat{p}_{z,0}$ in the form of (47), $\hat{v}_y$ and $\hat{v}_z$ in the form of (48), and $\hat{p}_{y,k}$ and $\hat{p}_{z,k}$ in the form of (49).

References


